SIEMENS

SINUMERIK 802D
SINUMERIK 802D base line

Programming and Operating
Turning

Valid for

<table>
<thead>
<tr>
<th>Control system</th>
<th>Software version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINUMERIK 802D</td>
<td>2</td>
</tr>
<tr>
<td>SINUMERIK 802D base line</td>
<td>1</td>
</tr>
</tbody>
</table>

08/05 Edition
Safety information

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

Danger

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

Warning

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

Caution

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

Caution

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

Attention

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

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

Qualified personnel

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

Proper use

Please note the following:

Warning

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

Trademarks

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

Disclaimer of liability

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

SINUMERIK Documentation

The SINUMERIK Documentation is organized in 3 levels:

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

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

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

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

Addressees of the documentation

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

Standard scope

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

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

Hotline

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

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

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

http://www.siemens.com/motioncontrol
## Contents

1 **Introduction** .......................................................... 1-11  
   1.1 Screen layout .......................................................... 1-11  
   1.2 Operating areas ...................................................... 1-14  
   1.3 Accessibility options .............................................. 1-15  
      1.3.1 Calculator ..................................................... 1-15  
      1.3.2 Editing Chinese characters .............................. 1-20  
      1.3.3 Hotkeys ..................................................... 1-21  
   1.4 The help system .................................................... 1-22  
   1.5 Coordinate systems .............................................. 1-24

2 **Setting Up** .............................................................. 2-27  
   3.1 Entering the tools and the tool offsets operating area# 3-30  
      3.1.1 Use this softkey to create a new tool. ................... 3-32  
      3.1.2 Determining the tool offsets (manually) ................... 3-33  
      3.1.3 Determining tool compensations using a probe .......... 3-36  
      3.1.4 Determining the tool offsets using optical measuring instruments 3-37  
      3.1.5 Probe settings  .......................................... 3-38  
   3.2 Tool monitoring ...................................................... 3-40  
   3.3 Entering/modifying a work offset ................................ 3-42  
      3.3.1 Determining the work offset ............................... 3-43  
   3.4 Programming setting data - "Parameter" operating area .... 3-44  
   3.5 R parameters - "Offset/Parameter" operating area .......... 3-47

4 **Manually Controlled Mode** .......................................... 4-49  
   4.1 Mode Jog - "Position" operating area .......................... 4-50  
      4.1.1 Assigning handwheels ..................................... 4-53  
   4.2 MDA mode (Manual input) - "Machine" operating area ....... 4-54  
      4.2.1 Face turning .............................................. 4-57

5 **Automatic Mode** ....................................................... 5-61  
   5.1 Selecting / starting a part program - "Machine" operating area 5-66  
   5.2 Block search - "Machine" operating area ...................... 5-67  
   5.3 Stopping / canceling a part program ........................... 5-68  
   5.4 Reapproach after cancellation .................................. 5-69  
   5.5 Repositioning after interruption ............................... 5-69  
   5.6 Program execution from external source (RS232 interface) 5-70

6 **Part Programming** .................................................... 6-71  
   6.1 Entering a new program - "Program" operating area .......... 6-74  
   6.2 Editing part programs - "Program" operating area ........... 6-75  
   6.3 Blueprint programming ......................................... 6-77  
   6.4 Simulation ...................................................... 6-95  
   6.5 Data transfer via the RS232 interface ......................... 6-96
7 System .................................................................................................................. 7-99
  7.1 PLC diagnosis represented as a ladder diagram ............................................. 7-120
  7.1.1 Screen layout ............................................................................................... 7-120
  7.1.2 Operating options ....................................................................................... 7-121
  8 Programming ...................................................................................................... 8-131
  8.1 Fundamentals of NC programming ................................................................ 8-131
  8.1.1 Program names ............................................................................................. 8-131
  8.1.2 Program structure ....................................................................................... 8-131
  8.1.3 Word structure and address ........................................................................ 8-132
  8.1.4 Block structure ............................................................................................ 8-133
  8.1.5 Character set ............................................................................................... 8-134
  8.1.6 Overview of the instructions ....................................................................... 8-136
  8.2 Positional data .................................................................................................. 8-148
  8.2.1 Absolute / incremental dimensioning: G90, G91, AC, IC ......................... 8-148
  8.2.2 Dimensions in metric units and inches: G71, G70, G710, G700 ............... 8-149
  8.2.3 Radius / diameter dimensional notation: DIAMOF, DIAMON ................. 8-150
  8.2.4 Programmable work offset: TRANS, ATRANS ........................................ 8-151
  8.2.5 Programmable scaling factor: SCALE, ASCALE ...................................... 8-152
  8.2.6 Workpiece clamping – settle work offset: G54 to G59, G500, G53, G153 ... 8-154
  8.2.7 Programmable working area limitation: G25, G26, WALIMON, WALIMOF .... 8-155
  8.3 Axis movements ............................................................................................... 8-157
  8.3.1 Linear interpolation with rapid traverse: G0 .............................................. 8-157
  8.3.2 Linear interpolation with feedrate: G1 ....................................................... 8-158
  8.3.3 Circular Interpolation: G2, G3 ..................................................................... 8-159
  8.3.4 Circular interpolation via intermediate point: CIP ...................................... 8-162
  8.3.5 Circle with tangential transition: CT .......................................................... 8-162
  8.3.6 Thread cutting with constant lead: G33 ...................................................... 8-163
  8.3.7 Thread cutting with variable lead: G34, G35 .............................................. 8-166
  8.3.8 Thread Interpolation: G331, G332 .............................................................. 8-167
  8.3.9 Fixed point approach: G75 .......................................................................... 8-168
  8.3.10 Reference point approach: G74 ............................................................... 8-168
  8.3.11 Measuring with touch–trigger probe: MEAS, MEAW .............................. 8-169
  8.3.12 Feedrate F .................................................................................................. 8-170
  8.3.13 Exact stop / continuous–path control mode: G9, G60, G64 ................. 8-171
  8.3.14 Acceleration pattern: BRISK, SOFT ....................................................... 8-171
  8.3.15 Percentage acceleration override: ACC .................................................. 8-174
  8.3.16 Traversing with feedforward control: FFVON, FFVOF ......................... 8-175
  8.3.17 3rd and 4th axes ....................................................................................... 8-176
  8.3.18 Dwell Time: G4 ......................................................................................... 8-176
  8.3.19 Travel to fixed stop ................................................................................... 8-177
  8.4 Spindle movements .......................................................................................... 8-181
  8.4.1 Spindle speed S, directions of rotation ..................................................... 8-181
  8.4.2 Spindle speed limitation: G25, G26 ............................................................ 8-181
  8.4.3 Spindle positioning: SPOS ......................................................................... 8-182
  8.4.4 Gear stages .................................................................................................. 8-183
  8.4.5 2nd spindle ................................................................................................ 8-183
  8.5 Special turning functions .................................................................................. 8-185
  8.5.1 Constant cutting rate: G96, G97 ................................................................. 8-185
  8.5.2 Rounding, chamfer ..................................................................................... 8-187
  8.5.3 Blueprint programming .............................................................................. 8-188
  8.6 Tool and tool offset .......................................................................................... 8-191
  8.6.1 General notes ............................................................................................... 8-191
  8.6.2 Tool T .......................................................................................................... 8-191
  8.6.3 Tool offset number D .................................................................................. 8-192
9 Cycles ................................................................. 9-235

9.1 Overview of cycles ........................................... 9-235
9.2 Programming cycles ........................................ 9-236
9.3 Graphical cycle support in the program editor .... 9-238
9.4 Drilling cycles ................................................ 9-240
9.4.1 General .................................................... 9-240
9.4.2 Prerequisites ............................................. 9-241
9.4.3 Drilling, centering – CYCLE81 ....................... 9-242
9.4.4 Drilling, counterboring – CYCLE82 ............... 9-245
9.4.5 Deep hole drilling – CYCLE83 ...................... 9-247
9.4.6 Rigid tapping – CYCLE84 ............................ 9-251
9.4.7 Tapping with compensating chuck – CYCLE840 9-254
9.4.8 Reaming1 (boring 1) – CYCLE85 ................... 9-258
9.4.9 Boring (boring 2) – CYCLE86 ......................... 9-261
9.4.10 Boring with Stop 1 (boring 3) – CYCLE87 ......... 9-264
9.4.11 Drilling with stop 2 (boring 4) – CYCLE88 ....... 9-267
9.4.12 Reaming 2 (boring 5) – CYCLE89 ................. 9-269
9.4.13 Row of holes – HOLES1 ............................ 9-271
9.4.14 Circle of holes – HOLES2 ........................... 9-275
9.5 Turning cycles ................................................. 9-278
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.5.1</td>
<td>Prerequisites</td>
<td>9-278</td>
</tr>
<tr>
<td>9.5.2</td>
<td>Grooving – CYCLE93</td>
<td>9-280</td>
</tr>
<tr>
<td>9.5.3</td>
<td>Undercut (forms E and F to DIN) – CYCLE94</td>
<td>9-288</td>
</tr>
<tr>
<td>9.5.4</td>
<td>Machining with undercut – CYCLE95</td>
<td>9-292</td>
</tr>
<tr>
<td>9.5.5</td>
<td>Thread undercut – CYCLE96</td>
<td>9-305</td>
</tr>
<tr>
<td>9.5.6</td>
<td>Thread cutting – CYCLE97</td>
<td>9-309</td>
</tr>
<tr>
<td>9.5.7</td>
<td>Chaining of threads – CYCLE98</td>
<td>9-315</td>
</tr>
<tr>
<td>9.6</td>
<td>Error Messages and Error Handling</td>
<td>9-322</td>
</tr>
<tr>
<td>9.6.1</td>
<td>General notes</td>
<td>9-322</td>
</tr>
<tr>
<td>9.6.2</td>
<td>Error handling in the cycles</td>
<td>9-322</td>
</tr>
<tr>
<td>9.6.3</td>
<td>Overview of cycle alarms</td>
<td>9-322</td>
</tr>
<tr>
<td>9.6.4</td>
<td>Messages in the cycles</td>
<td>9-324</td>
</tr>
</tbody>
</table>
SINUMERIK 802D Key Definition

- "Recall" key
- ETC key
- "Acknowledge alarm" key
- without function
- Info key
- Shift key
- Controlkey
- Altkey
- SPACE
- Backspace
- Clear key
- Insert key
- Tabulator
- ENTER / Input key
- "Position" operating area key
- "Program" operating area key
- "Parameter" operating area
- "Program Manager" operating area
- "Alarm/System" operating area
- not assigned
- PageUp / PageDown keys
- Cursor keys
- Selection key / toggle key
- Alphanumeric keys
  Double assignment on the Shift level
- Numeric keys
  Double assignment on the Shift level
External Machine Control Panel

- **RESET**
- **NC STOP**
- **NC START**
- **EMERGENCY STOP**
- **Spindle override**
- **User–defined key with LED**
- **User–defined key without LED**
- **INCREMENT**
- **JOG**
- **REFERENCE POINT**
- **AUTOMATIC**
- **SINGLE BLOCK**
- **MANUAL DATA**
- **SPINDLE START LEFT**
- **SPINDLE STOP**
- **SPINDLE START RIGHT**
- **RAPID TRAVERSE OVERRIDE**
- **X axis**
- **Z axis**
- **Feedrate override**
Note
The present Manual always uses "802D bl" as an acronym for SINUMERIK 802D base line.

1.1 Screen layout

The screen is divided into the following main areas:
- Status area
- Application area
- Tip and softkey area
Status area

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

<table>
<thead>
<tr>
<th>Screen Control</th>
<th>Display</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Active operating area, active mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Position</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JOG; 1 INC, 10 INC, 100 INC, 1000 INC, VAR INC (evaluation by increments in the JOG mode)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MDA AUTOMATIC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Offset</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Program</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Program Manager</td>
<td></td>
</tr>
<tr>
<td></td>
<td>System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alarm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Marked as an &quot;external language&quot; using G291</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Alarm and message line</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In addition, the following is displayed:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1. Alarm number with alarm text, or</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2. Message text</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Program status</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RESET</td>
<td>Program canceled / default status</td>
</tr>
<tr>
<td></td>
<td>RUN</td>
<td>Program running</td>
</tr>
<tr>
<td></td>
<td>STOP</td>
<td>Program stopped</td>
</tr>
<tr>
<td>4</td>
<td>Program controls in the AUTOMATIC mode</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>NC messages</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Selected part program (main program)</td>
<td></td>
</tr>
</tbody>
</table>
### Tip and softkey area

![Tip and softkey area diagram](image)

#### Fig. 1-3 Tip and softkey area

<table>
<thead>
<tr>
<th>Screen Control</th>
<th>Display</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="image">Recall symbol</a></td>
<td>Pressing the Recall key lets you return to the next higher menu level.</td>
</tr>
<tr>
<td>2</td>
<td>Tip line</td>
<td>Displays tips for the operator</td>
</tr>
<tr>
<td>3</td>
<td>MMC status information</td>
<td>ETC is possible (Pressing this key displays the horizontal softkey bar providing further functions.)</td>
</tr>
<tr>
<td>4</td>
<td>Softkey bar</td>
<td>Mixed notation active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data transfer running</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Connection to the PLC programming tool active</td>
</tr>
</tbody>
</table>

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

**Standard softkeys**

- **Back**
  - Use this softkey to quit the screenform.

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

- **Accept**
  - Selecting this softkey will complete your input and start the calculation.
Selecting this softkey will complete your input and accept the values you have entered.

This function is used to switch the screenform from diameter programming to radius programming.

### 1.2 Operating areas

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

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

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

### Protection levels

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

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

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

1.3.1 Calculator

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

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

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

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

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

Note

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

Fig. 1-4 Calculator

Characters permitted for input

+ , − , *, /  Basic arithmetic operations

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

O  Cosine function
   The X value (in degrees) in front of the input cursor is replaced by the cos(X) value.
Q  Square function  
The X value in front of the input cursor is replaced by the $X^2$ value.

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

()  Bracket function $(X+Y)*Z$

### Calculation examples

<table>
<thead>
<tr>
<th>Task</th>
<th>Input -&gt; Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$100 + (67*3)$</td>
<td>$100+67*3$      -&gt; 301</td>
</tr>
<tr>
<td>$\sin(45^\circ)$</td>
<td>$45 \ S$      -&gt; 0.707107</td>
</tr>
<tr>
<td>$\cos(45^\circ)$</td>
<td>$45 \ C$      -&gt; 0.707107</td>
</tr>
<tr>
<td>$4^2$</td>
<td>$4 \ Q$     -&gt; 16</td>
</tr>
<tr>
<td>$\sqrt{4}$</td>
<td>$4 \ R$     -&gt; 2</td>
</tr>
<tr>
<td>$(34+3*2)*10$</td>
<td>$(34+3*2)*10$ -&gt; 400</td>
</tr>
</tbody>
</table>

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

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

### Softkeys

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

Fig. 1-5

Enter the circle center, the angle of the tangent and the circle radius.
Use the G2 / G3 softkey to define the direction of rotation of the circle.

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

Example: Calculate the intersection point between the circle sector and the straight line in plane G18.

Given:
Radius: 10
Circle center: Z 147 X 103
Connection angle of the straight line: \(-45^\circ\)

Result: $Z = 154.071$
$X = 110.071$

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

Enter the following coordinates or angles:
- the coordinates of the given point (PP)
- the slope angle of the straight line (A1)
- the distance of the new point with reference to PP
- the slope angle of the connecting straight line (A2) with reference to A1
Use this softkey to calculate the Cartesian coordinates which are subsequently copied into two input fields following one after another. The abscissa value is copied into the input field from which the calculator function has been called, and the value of the ordinate is copied into the next following input field.

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

This function converts the given polar coordinates into Cartesian coordinates.

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

Use this softkey to calculate the Cartesian coordinates which are subsequently copied into two input fields following one after another. The abscissa value is copied into the input field from which the calculator function has been called, and the value of the ordinate is copied into the next following input field.

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

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

The following values of the straight line are known:

- **Straight line 1:**
  - Starting point and slope angle

- **Straight line 2:**
  - Length and one end point in the Cartesian coordinate system
This function is used to select the given coordinate of the end point. The ordinate value or the abscissa value is given.

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

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

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

**Example**

![Fig. 1-9](image)

*Add the drawing above by the value of the center circle in order to be able to calculate the intersection point between the circle sector of the straight line. The missing coordinate of the center point is calculated using the calculator function $\mathbf{2}$, since the radius in the tangential transition stands vertically on the straight line.*

![Fig. 1-10](image)
Calculating M1 in section 1:

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

Use the softkey to select the appropriate direction of rotation. Use the softkey to define the given end point.

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

Result: \( X = 60 \)
\( Z = -44.601 \)

1.3.2 Editing Chinese characters

This function is only available in the Chinese language version.

The control system provides a function for editing Chinese characters in the program editor and in the PLC alarm text editor. After activation, type the phonetic alphabet of the searched character in the input field. The editor will then offer various characters for this sound, from which you can choose the desired one by entering either of the digits 1 to 9.

Alt S Is used to turn on / turn off the editor
1.3.3 **Hotkeys**

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

- CTRL C: Copy
- CTRL B: Select
- CTRL X: Cut
- CTRL V: Paste
- Alt L: Is used to switch to the mixed notation or info key
- Alt H: Help system
1.4 The help system

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

In addition, the help offers the following topics:

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

This function opens the selected topic.

This function lets you return to the previous screenform.

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.

If you select a cross reference, in addition, the **Back to topic** softkey is displayed.
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.
1.5 Coordinate systems

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

![Coordinate system diagram](image1.png)

**Fig. 1-15** Determination of the axis directions another to one; coordinate system for programming when turning

**Machine coordinate system (MCS)**

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

![Machine coordinate system](image2.png)

**Fig. 1-16** Machine coordinates/machine axes using the example of a turning machine

The origin of the coordinate system is the **machine zero**. 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 be in the negative range.
Workpiece coordinate system (WCS)

The coordinate system described above (see Fig. 1-15) is also used to describe the geometry of a workpiece in the workpiece program. The workpiece zero can be freely selected by the programmer in the Z axis. In the X axis, it lies in the turning center.

![Fig. 1-17 Workpiece coordinate system](image)

Relative coordinate system

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

Clamping the workpiece

For machining, the workpiece is clamped on the machine. The workpiece must be aligned such that the axes of the workpiece coordinate system run in parallel with those of the machine. Any resulting offset of the machine zero with reference to the workpiece zero is determined along the Z axis and entered in a data area intended for the settable work offset. In the NC program, this offset is activated, e.g. using a programmed G54 (see also Section 8.2.6).

![Fig. 1-18 Workpiece on the machine](image)
1.5 Coordinate systems

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").
Turning On and Reference Point Approach

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 start screen](image)

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

The "Reference point approach" window (Fig. 2-1) displays whether the axes are referenced (approached to their reference points).

- Axis must be referenced
- Axis has reached its reference point

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

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

---

**Note**

“Reference point approach” is only possible in the Jog mode.
Setting Up

Preliminary remarks

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

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

Menu tree

![Menu tree](image)

Fig. 3-1 Menu tree "Parameters" operating area

Note

The softkeys marked in Fig. 3-1 with an asterisk ("※") are missing in the 802D bl.
3.1 Entering the tools and the tool offsets operating area

Functionality

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

See also Section 8.6 "Tool and tool compensation"

Operating sequences

Use this softkey to open the "Tool offset data" window which contains a list of the tools created. Use the cursor keys and the Page Up / Page Down keys to navigate in this list.

![Tool List](Image)

Enter the offsets by positioning the

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

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

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

Softkeys

Use this softkey to determine the tool compensation data.

Determining the tool offset data manually (see Section 3.1.2)
Determining the tool offset data semi–automatically (see Section 3.1.3)

Use this softkey to calibrate the sensing probe.

**Note**

With the 802D bl, the **Tool measur** softkey opens the “Measure tool” window directly.

Use this softkey to delete the tool.

Use this function to display all parameters of a tool. For the meanings of the parameters, please refer to the Section “Programming”.

![Input screen for special tools](image)

**Fig. 3-3** Input screen for special tools

Selecting this softkey activates the compensation values of the cutting edge.

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

Use this softkey to select the next higher edge number.

Use this softkey to select the next lower edge number.

Use this softkey to create a new edge.

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

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

Use this function to search for a tool by its number.
3.1 Entering the tools and the tool offsets operating area

Use this softkey to create tool offset data for a new tool. Max. 48 tools can be created with the 802D, and 18 tools with the 802D bl. For the 802D bl, no milling tools are offered.

3.1.1 Use this softkey to create a new tool.

Operating sequence

This function offers another two softkey functions to select the tool type. After selecting the tool type, type the desired tool number (max. 3 digits) in the input field.

For milling and drilling tools, the machining directions must be selected.

Select **OK** to confirm your input. A data record loaded with zero will be included in the tool list.
### 3.1.2 Determining the tool offsets (manually)

**Functionality**

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

**Prerequisite**

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

**Procedure**

Enter the reference point in the appropriate field Ø or Z0.

**Please observe:** The assignment of length 1 or 2 to the axis depends on the tool type (turning tool, drill).

With a turning tool, the reference point for the X axis is a diameter dimension.

Using the actual position of the point F (machine coordinate) and the reference point, the control system can calculate the offset value assigned to length 1 or length 2 for the axis.

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

---

**Fig. 3-6 Determination of the length offsets using the example of a cutting tool**

- F – toolholder reference point
- M – machine zero
- W – workpiece zero

The offset value in the X axis is a diameter value.
3.1 Entering the tools and the tool offsets operating area

Fig. 3-7 Determination of the length offset using the example of a drill: Length 1 / Z axis

Note
The diagram 3-7 only applies if the variables are the machine data MD 42950 TOOL_LENGTH_TYPE and MD 42940 TOOL_LENGTH_CONST ≠ 0; otherwise, length tool 2 will apply for the milling and drilling tools (see also Manufacturer Documentation "SINUMERIK 802D Start-up").

Operating sequence

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

Fig. 3-8 Selecting manual or semiautomatic measuring

Use this softkey to open the Measure tool window.
• Either type the workpiece diameter in the "Ø" field or the workpiece length in the "Z0" field. The machine coordinates and the values from the work offsets will apply. When using a spacer, it is also possible to enter the thickness of the spacer for taking into account.

• After selecting the Set length 1 or Set length 2 softkey, the control system will determine the searched length 1 or length 2 of the preselected axis. The offset value determined will be stored.

Selecting this softkey will save the X position. Thereafter, you can traverse in the X direction. Thus, it is possible to determine, for example, the workpiece diameter. The stored value of the axis position will then be used for calculating the length offset.

The activation of the softkey is dependent on the display machine data 373 MEAS_SAVE_POS_LENGTH2 (see also Manufacturer Documentation "SINUMERIK 802D Start-up")
3.1 Entering the tools and the tool offsets operating area

3.1.3 Determining tool compensations using a probe

**Note**
This function is only offered by the 802D.

**Operating sequence**

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

![Image](image_url)

**Fig. 3-10** The "Measure tool" window

In this screenform, you can enter tool and cutting edge numbers. In addition, the tool tip position is displayed after symbol.

After the screenform has been opened, the input fields are filled with the data of the tool currently working.

The tool can be either
- the currently active tool of the NC (loaded via a part program) or
- a tool loaded by the PLC.

If the tool was loaded by the PLC, the tool number in the input screenform can be different than that in the T,F,S window.

If you change the tool number, no automatic tool change will be performed using this function. The entered tool, however, are assigned measurement results.

**Measuring process**

Approach the probe using either the traversing keys or the handwheel.

After the "Probe triggered" has appeared, release the traversing key and wait until the measuring process is completed. During the automatic measurement, a dial appears symbolizing the active measuring process.
Note
To create the measuring program, the parameters "Safety clearance" from the Settings screenform and the feedrate from the Probe data screenform are used (see Section 3.1.5).
If several axes are moved simultaneously, no offset data can be calculated.

3.1.4 Determining the tool offsets using optical measuring instruments

Note
This function is only offered by the 802D.

Fig. 3-11 Measuring using an optical measuring instrument (for the T and D input fields, please refer to "Measuring using a probe")

Measuring process
For measuring, traverse the tool until its tip appears in the crosshair. With a milling tool, use the highest point of the cutting edge to determine the tool length.
Subsequently, select the Set length softkeys to calculate the offset values.
3.1.5   Probe settings

**Note**
This function is only offered by the 802D.

The screenform below is used to store the coordinates of the probe and to set the axis feedrate for the automatic measuring process.

All position values refer to the machine coordinate system.

![Screenform for storing probe coordinates](image)

**Fig. 3-12** The "Probe data" interactive screenform

**Table 3-1**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute position P1</td>
<td>Absolute position of the probe in the Z− direction</td>
</tr>
<tr>
<td>Absolute position P2</td>
<td>Absolute position of the probe in the X+ direction</td>
</tr>
<tr>
<td>Absolute position P3</td>
<td>Absolute position of the probe in the Z+ direction</td>
</tr>
<tr>
<td>Absolute position P4</td>
<td>Absolute position of the probe in the X− direction</td>
</tr>
<tr>
<td>Feedrate</td>
<td>Feedrate with which the tool approaches the probe</td>
</tr>
</tbody>
</table>

**Calibrating the probe**

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

Approach the four points of the probe.

For calibration, use a tool of the type 500 with tool tip position 3 or 4.
The offset parameters required to determine the four probe positions can be written to the data records of two cutting edges.

![Fig. 3-13 Calibrating the probe](image)

After the screenform has appeared, an animation signaling the step to be executed is displayed next to the current positions of the probe. This point must be approached with the appropriate axis.

After the "Probe triggered" has appeared, release the traversing key and wait until the measuring process is completed. During the automatic measurement, a dial appears symbolizing the active measuring process.

The positions delivered by the measuring program serve to calculate the real probe position. The measuring function can be quit without approaching all positions. The points already sensed are stored.

**Note**

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

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

Use the **Next Step** function to skip a point if this is not needed for measuring.
3.2 Tool monitoring

Note
This function is only offered by the 802D.

Each monitoring type is represented in 4 columns.
- Setpoint
- Prewarning limit
- Residual value
- active

Use the checkbox element of the 4th column to activate/deactivate the monitoring type.

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

⚠️ Prewarning limit reached
❌ Tool disabled
✔️ Tool is monitored

Use this softkey to reset the monitoring values of the selected tool.
Use this softkey to change the enable of the selected tool.
3.3 Entering/modifying a work offset

Functionality

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

Operating sequences

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

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

![The "Work offset" window](image)

Position the cursor bar on the input field to be changed

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

The compensation values of the cutting edge come into effect immediately.
3.3.1 Determining the work offset

Prerequisite

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

![Diagram of work offset determination](image)

Fig. 3-17 Determining the work offset for the Z axis

Procedure

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

Then scratch the workpiece with the tool tip. Enter the position for the workpiece in the workpiece coordinate system in the "Set position to:" field.

![Screenshots of work offset measurement](image)

Fig. 3-18 The Determine work offset in X” screenform The “Determine work offset in Z” screenform

Selecting this softkey will calculate the offset and display the result in the "Offset" field.
3.4 Programming setting data - “Parameter” operating area

Functionality

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

Operating sequences

Select Setting data using the Offset Parameter and Setting data keys.

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

![Setting data start screen](image)

**JOG feedrate**

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

**Spindle**

Spindle speed

**Minimum / maximum**

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

**Programmed (limitation)**

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

**Dry run feed (DRY)**

The feedrate which can be entered here will be used instead of the programmed feedrate in the AUTOMATIC mode if the "Dry run feed" function is selected.
Start angle for thread cutting (SF)

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

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

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

Softkeys

The working area limitation is active with geometry and additional axes. If you want to use a working area limitation, its values can be entered in this dialog box. Selecting the **Set Active** softkey will activate / deactivate the values for the axis highlighted by the cursor.

Fig. 3-20

Timers Counters

Fig. 3-21
Meaning:

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

In the AUTOMATIC mode, the runtimes of all programs between NC START and end of program / RESET are summed up. The timer is zeroed with each power-up of the control system.

- **Cycle time**: Tool action time

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

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

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

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**
- **channel setting data**

![Fig. 3-22](image-url)
3.5  R parameters – ”Offset/Parameter” operating area

Functionality

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

Fig. 3-23  The "R parameters" window

Operating sequence

Use the Parameters and the R parameters softkeys

Position the cursor bar on the input field to be changed

and enter the value(s).

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

Find R parameters
This sheet has been left empty for your notes
Manually Controlled Mode

The manually controlled mode is possible in the Jog and MDA modes.
The softkeys marked with an asterisk ("\*") are not available with the 802D bl.

<table>
<thead>
<tr>
<th>Set base</th>
<th>Measure workpiece</th>
<th>Tool measure</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>x=0</td>
<td></td>
<td>Measure manual</td>
<td>Data probe</td>
</tr>
<tr>
<td>z=0</td>
<td>Work offset</td>
<td>Measure auto</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set rel</td>
<td></td>
<td></td>
<td>Switch mm/inch</td>
</tr>
<tr>
<td>Delete base W0</td>
<td></td>
<td>Calibrate probe</td>
<td></td>
</tr>
<tr>
<td>All to zero</td>
<td>Set work offset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Back &lt;&lt;</td>
<td>Back &lt;&lt;</td>
<td>Back &lt;&lt;</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4-1 The "Jog" menu tree

| Set basis | | | Face | Settings |
|-----------| | | | |
| x=0       | | | Peripher. surface | Data probe |
| z=0       | | | |          |
|          | | | |          |
|          | | | |          |
| Set rel  | | | |          |
| Delete base W0 | | | | Switch mm/inch |
| All to zero | | | |          |
| Back <<  | | | |          |

Fig. 4-2 The MDA menu tree
4.1 Mode Jog - “Position” operating area

Operating sequences

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

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

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

If necessary set the velocity using the override switch.

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

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

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

Fig. 4-3 The "Jog" start screen
Parameters

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCS X Z</td>
<td>Displays the axes existing in the machine coordinate system (MCS) or in the workpiece coordinate system (WCS)</td>
</tr>
<tr>
<td>+X -Z</td>
<td>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.</td>
</tr>
<tr>
<td>Position mm</td>
<td>These fields display the current position of the axes in the MCS or WCS.</td>
</tr>
<tr>
<td>REPOS offset</td>
<td>If the axes are traversed in the &quot;Program interrupted&quot; condition in the Jog mode, the distance traversed by each axis is displayed referred to the interruption point.</td>
</tr>
<tr>
<td>G function</td>
<td>Displays important G functions</td>
</tr>
<tr>
<td>Spindle S r.p.m.</td>
<td>Displays the actual value and the setpoint of the spindle speed</td>
</tr>
<tr>
<td>Feed F mm/min</td>
<td>Displays the path feedrate actual value and setpoint</td>
</tr>
<tr>
<td>Tool</td>
<td>Displays the currently active tool with the current edge number</td>
</tr>
</tbody>
</table>

Note

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

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

- The master spindle (large display) is displayed:
  - in the idle condition;
  - when starting the spindle;
  - if both spindles are active.
- The workspindle (small display) is displayed:
  - when starting the workspindle.

The power bar applies to the spindle currently active.

Softkeys

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

- **Direct input of the desired axis position**
  In the input window, position the input cursor on the desired axis; thereafter, enter the new position. Then, press **Input** or move the cursor to confirm your input.

- **Setting of all axes to zero**
  The softkey function **All to zero** overwrites the current position of the appropriate axis with zero.

- **Setting of individual axes to zero**
  Use the \( X=0 \) or \( Z=0 \) softkey to overwrite the current position with zero.

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

**Note**

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

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

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

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.

**Retract plane**: The **Face** function retracts the tool to the specified position (Z position) after the function has been executed.
**Safety distance:** Safety clearance to the workpiece surface
This value defines the minimum distance between the workpiece surface and the workpiece. It is used by the functions "Face" and "Automatic tool gauging".

**JOG feedrate:** Feedrate value in the JOG mode

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

The screen form below is used to store the coordinates of the probe and to set the axis feedrate for the automatic measuring process (see Section 3.1.5). This only applies to the 802D.

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

### 4.1.1 Assigning handwheels

#### Operating sequence

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

After the window has been opened, all axis identifiers are displayed in the "Axis" column, which simultaneously appear in the softkey bar.

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

The symbol is display in the window.

![Handwheel menu screen](image)

Use the **MCS** softkey to select the axes from the machine or workpiece coordinate system for handwheel assignment. The current setting is displayed in the window.
4.2 MDA mode (Manual input) - "Machine" operating area

Functionality

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

Caution

The Manual mode is subject to the same safety interlocks as the fully automatic mode. Furthermore, the same prerequisites are required as in the fully automatic mode.

Operating sequences

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

Enter one or several blocks using the keyboard.

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

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCS X Z</td>
<td>Displays the existing axes in the MCS or WCS</td>
</tr>
<tr>
<td>+X -Z</td>
<td>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.</td>
</tr>
<tr>
<td>Position mm</td>
<td>These fields display the current position of the axes in the MCS or WCS.</td>
</tr>
<tr>
<td>Distance to go</td>
<td>This field displays the distance to go of the axes in the MCS or WCS.</td>
</tr>
<tr>
<td>G function</td>
<td>Displays important G functions</td>
</tr>
<tr>
<td>Spindle S r.p.m.</td>
<td>Displays the actual value and the setpoint of the spindle speed</td>
</tr>
<tr>
<td>Feed F</td>
<td>Displays the path feedrate actual value and setpoint in mm/min or mm/rev.</td>
</tr>
<tr>
<td>Tool</td>
<td>Displays the currently active tool with the current edge number (T..., D...).</td>
</tr>
<tr>
<td>Editing window</td>
<td>In the “Stop” or “Reset” program state, an editing window serves to input a part program block.</td>
</tr>
</tbody>
</table>

Note

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

The control system displays the spindle data according to the following aspects:
- The master spindle is displayed:
  - in the idle condition;
  - when starting the spindle;
  - if both spindles are active.
- The workspindle is displayed:
  - when starting the workspindle.

The power bar applies to the spindle currently active.
4.2 MDA mode (Manual input) - "Machine" operating area

Softkeys

- **Set base**
  - Use this softkey to set the base work offset (see Section 4.1).

- **Face**
  - Face milling (see also Section 4.2.1)

- **Settings**
  - see Section 4.1

- **G function**
  - The G function window displays G functions whereby each G function is assigned to a group and has a fixed position in the window. Use the **PageDown** and **PageUp** keys to display further G functions. Selecting the softkey repeatedly will close the window.

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

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

- **Delete MDI prog.**
  - Use this function to delete blocks from the program window.

- **Save MDI prog.**
  - Enter a name in the input field with which you wish the MDA program to be saved in the program directory. Alternatively, you can select an existing program from the list. To switch between the input field and the program list, use the TAB key.

![Fig. 4-7](image.png)

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

Functionality

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

Operating sequence

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

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

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

Important

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td>Input of the tool to be used. The tool is loaded prior to machining. To this end, the function calls a working cycle performing all steps required. This cycle is provided by the machine manufacturer.</td>
</tr>
<tr>
<td>Feed F</td>
<td>Input of the path feedrate, in mm/min or mm/rev.</td>
</tr>
</tbody>
</table>
4.2 MDA mode (Manual input) - "Machine" operating area

Table 4-3 Description of the parameters in the **Face turning** working window, cont’d

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spindle S r.p.m.</td>
<td>Input of the spindle speed</td>
</tr>
<tr>
<td>Machining</td>
<td>Definition of the surface quality</td>
</tr>
<tr>
<td></td>
<td>You can select between roughing and finishing.</td>
</tr>
<tr>
<td>Diameter</td>
<td>Input of the blank diameter of the part</td>
</tr>
<tr>
<td>Z0</td>
<td>Input of the Z position</td>
</tr>
<tr>
<td>Z1</td>
<td>Cutting dimension, incremental</td>
</tr>
<tr>
<td>DZ</td>
<td>Input of the cutting length in the Z direction</td>
</tr>
<tr>
<td></td>
<td>This dimension is always specified in increments and is referred to the workpiece edge.</td>
</tr>
<tr>
<td>UZ</td>
<td>Stock allowance in the Z direction</td>
</tr>
<tr>
<td>UX</td>
<td>Stock allowance in the X direction</td>
</tr>
</tbody>
</table>

Table 4-4 Description of the parameters in the **Longitudinal turning** working window

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool</td>
<td>Input of the tool to be used</td>
</tr>
<tr>
<td></td>
<td>The tool is loaded prior to machining. To this end, the function calls a working cycle performing all steps required. This cycle is provided by the machine manufacturer.</td>
</tr>
<tr>
<td>Feed F</td>
<td>Input of the path feedrate, in mm/min or mm/rev.</td>
</tr>
<tr>
<td>Spindle S r.p.m.</td>
<td>Input of the spindle speed</td>
</tr>
<tr>
<td>Mach.</td>
<td>Definition of the surface quality</td>
</tr>
<tr>
<td></td>
<td>You can select between roughing and finishing.</td>
</tr>
</tbody>
</table>
4.2 MDA mode (Manual input) - "Machine" operating area

Table 4-4 Description of the parameters in the **Longitudinal turning** working window, cont’d

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0</td>
<td>Blank diameter</td>
</tr>
<tr>
<td>X1</td>
<td>Cutting length, incremental, in the X direction</td>
</tr>
<tr>
<td>Z0</td>
<td>Position, input of the position of the workpiece edge in the Z direction</td>
</tr>
<tr>
<td>Z1</td>
<td>Cutting length, incremental, in the Z direction</td>
</tr>
<tr>
<td>DZ</td>
<td>Max. infeed</td>
</tr>
<tr>
<td>UZ</td>
<td>Input field for the stock allowance when roughing</td>
</tr>
<tr>
<td>UX</td>
<td>Stock allowance</td>
</tr>
</tbody>
</table>

Use this softkey to accept the current tool tip position into the Z0 or X0 input field.
Manually Controlled Mode

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

This sheet has been left empty for your notes
Automatic Mode

Prerequisites

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.

Note

The Real-time simulat. softkey is only offered by the 802D bl with color display option. Spindle power and load displays are missing with the 802D bl.
### Parameters

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCS X Z</td>
<td>Displays the existing axes in the MCS or WCS</td>
</tr>
<tr>
<td>+ X - Z</td>
<td>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.</td>
</tr>
<tr>
<td>Position mm</td>
<td>These fields display the current position of the axes in the MCS or WCS.</td>
</tr>
<tr>
<td>Distance to go</td>
<td>These fields display the current position of the axes in the MCS or WCS.</td>
</tr>
<tr>
<td>G function</td>
<td>Displays important G functions</td>
</tr>
<tr>
<td>Spindle S r.p.m.</td>
<td>Displays the actual value and the setpoint of the spindle speed</td>
</tr>
<tr>
<td>Feed F mm/min or mm/rev.</td>
<td>Displays the path feedrate actual value and setpoint</td>
</tr>
<tr>
<td>Tool</td>
<td>Displays the currently active tool with the current edge number (T..., D...).</td>
</tr>
<tr>
<td>Current block</td>
<td>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 &quot;Program progress&quot; window. To switch back to the seven–block display, use the &quot;Program sequence&quot; softkey.</td>
</tr>
</tbody>
</table>
Note

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

The control system displays the spindle data according to the following aspects:
- The master spindle is displayed:
  - in the idle condition;
  - when starting the spindle;
  - if both spindles are active.
- The workspindle is displayed:
  - when starting the workspindle.

The power bar applies to the spindle currently active.

Softkeys

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

If "Program test" is selected, the output of setpoints to axes and spindles is disabled. The setpoint display will "simulate" the traversing motion.

If you select this softkey, all traversing motions will be performed with the feedrate setpoint specified via the "Dry run feed" setting data. In other words: Instead of the programmed motion commands, the dry run feedrate will act.

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

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

If this function is enabled, the part program blocks are executed separately as follows: Each block is decoded separately, and a stop is performed at each block; an exception are only the thread blocks without dry run feedrate. In such blocks, a stop is only performed at the end of the current thread block. "Single Block fine" can only be selected in the RESET status.

The feedrate override switch will also act on the rapid traverse override.

Use this softkey to quit the screenform.

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

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

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

The cursor is positioned on the main program block of the interruption point. The search tar-
get is set in the subroutine levels automatically.

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

Using broken-line graphics, the programmed tool path can be traced.
(see also Section 6.4)

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

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

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

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

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

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

Use this softkey to switch from the seven-block to the three-block display.
Switches the axis value display between the machine, workpiece and relative coordinate systems.

Use this softkey to transmit an external program to the control system via the RS232 interface; to execute this program, press **NC START**.
5.1 Selecting / starting a part program - "Machine" operating area

Functionality

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

Operating sequence

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

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

Position the cursor bar on the desired program.

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

Use this softkey to specify settings for program execution (as necessary).

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.

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

Fig. 5-5 Block search

Block search to the block start

Block search to the end of the block

Block search without calculation

The interruption point is loaded.

Use this softkey to perform the block search by entering a term you are looking for.
5.3 Stopping / canceling a part program

Search result

The required block is displayed in the Current blockwindow.

Operating sequence

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

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

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

Operating sequence

Select the AUTOMATIC mode.

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

The interruption point is loaded.

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

Press NC START to continue the program execution.

5.5 Repositioning after interruption

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

Operating sequence

Select the AUTOMATIC mode.

Press NC START to continue the program execution.

Caution

When reapproaching the interruption point, all axes will traverse at the same time. Make sure that the traversing area is not obstructed.
5.6 Program execution from external source (RS232 interface)

Functionality

Use this softkey to transmit an external program to the control system via the RS232 interface; to execute this program, press **NC START**. While the contents of the buffer memory are being processed, the blocks are reloaded automatically.

For example, a PC with the PCIN tool installed for data transfer can be used as the external device.

**Important**

Before connecting / removing the cable between the external device and the control system, always first make sure that the device is disconnected from the mains.

Operating sequence

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

Select the softkey.

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

The program is transmitted into the buffer memory and selected and displayed in the Program Selection automatically.
Before starting program execution by pressing NC START, first make sure that the buffer is filled completely.

The program execution starts with **NC-START**; subsequently, the program is reloaded successively.

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

**Note**

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

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

Operating sequence

Press the **Program Manager** key to open the program directory.

![Program Manager](image)

**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 letters of the program name. The control system will automatically position the cursor on a program with matching characters.
Softkeys

**Programs**

Use this softkey to display all files contained in the part program directory.

**Execute**

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

**New**

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

**Copy**

Use the **Copy** softkey to copy the selected program into another program with a new name.

**Open**

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

**Delete**

Use this softkey to delete either only the program highlighted by the cursor or all part programs; first, however, a warning confirmation is displayed.

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

**Rename**

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

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

**Read out**

Use this softkey to saved files via the RS232 interface.

**Read in**

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

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

**Cycles**

Use the **User cycles** softkey to display the "Standard cycles" directory. This softkey will only appear unhidden if you have the relevant access right.

**Delete**

Use this softkey to delete the cycle highlighted by the cursor; first, a confirmation warning will appear.

**User cycles**

Use the **User cycles** softkey to display the "User cycles" directory. With the appropriate access right, the softkeys **New**, **Copy**, **Open**, **Delete**, **Rename**, **Read out** and **Read in** are displayed.
Save data
This function is used to save the contents of the volatile memory into a non-volatile memory area.
Prerequisite: There is no program currently executed.
Do not carry out any operator actions while the data backup is running!
6.1 Entering a new program - "Program" operating area

Operating sequences

Select the **Program** operating area displaying an overview of the programs already created in the NC.

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

The files in the "User cycles" directory is also assigned the extension ".SPF".

Enter the name for the new program.

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

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 can only be edited if it is currently not being executed. Any modifications to the part program are stored immediately.

Menu tree

The softkeys marked with an asterisk ("※") are only offered by the 802D bl with the color display option.
### Operating sequence

Use the "Program manager" to select the program you want to edit and select **Open** to open the program.

### Softkeys

- **Edit**
  - Use this softkey to edit a file.
- **Execute**
  - Use this softkey to execute the selected file.
- **Mark block**
  - Use this softkey to select a text segment up to the current cursor position (alternatively: <ctrl>B).
- **Copy block**
  - Use this softkey to copy a selected block to the clipboard (alternatively: <ctrl>C).
- **Insert block**
  - Use this softkey to paste a text from the clipboard at the current cursor position (alternatively: <ctrl>V).
- **Delete block**
  - Use this softkey to delete a selected text (alternatively: <ctrl>X).
- **Find**
  - Use the **Find** and the **Find Next** softkeys 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.
  - If the string you are searching for is not found in the program file, an error message is issued.
  - Use **Back** to quit the dialog box without starting the search process.
- **Renumber**
  - Use this softkey to replace the block numbers from the current cursor position up to the program end.
- **Contour**
  - For programming the contour ("blueprint programming"), see Section 6.3
- **Drill**
  - see Manual "Cycles"
- **Milling**
  - see Manual "Cycles" (with the options "Transmit" and "Tracyl")
- **Turning**
  - see Manual "Cycles"
  - 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.
- **Simulation**
  - The simulation is described in Section 6.4.
6.3 Blueprint programming

Functionality

The control system offers various contour screenforms for the fast and reliable creation of part programs. Use these screenforms to enter the required parameters.

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

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

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

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

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

![Setting the starting point](image)

Fig. 6-7 Setting the starting point

Use this interactive screenform to define whether the following contour sections are to be programmed using radius or diameter programming or whether the transformation axes are to be used for TRANSMIT or TRACYL.

Note

The TRANSMIT and TRACYL softkeys are not offered by the 802D bl. The interactive screenform should therefore only be used to define whether the following contour sections are to be programmed using radius or diameter programming.

The **Approach start point** softkey function will generate an NC block approaching the entered coordinates.
Programming aid for the programming of straight line sections

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

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

If the end point is determined via polar coordinates, you will need the length of the vector between the pole and the end point, as well as the angle of the vector referred to the pole.

The prerequisite is that a pole was set beforehand. This pole will be applicable until a new pole is set.

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

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

If necessary you can enter additional functions in the fields. The commands can be separated from each other by spaces, commas or semicolons.
This interactive screenform is provided for all contour elements.

Pressing the OK softkey will accept all commands into the part program. Select Abort to quit the interactive screenform without saving the values.

This function is intended to calculate the point of intersection between two straight lines. Specify the coordinates of the end point of the second straight line and the angles of the straight lines.

Table 6-1  Input in the interactive screenform

<table>
<thead>
<tr>
<th>End point of straight line 2</th>
<th>E</th>
<th>Enter the end point of the straight line.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle of straight line 1</td>
<td>A1</td>
<td>The angle is specified in the counterclockwise direction from 0 to 360 degrees.</td>
</tr>
<tr>
<td>Angle of straight line 2</td>
<td>A2</td>
<td>The angle is specified in the counterclockwise direction from 0 to 360 degrees.</td>
</tr>
<tr>
<td>Feedrate</td>
<td>F</td>
<td>Feedrate</td>
</tr>
</tbody>
</table>
Use this interactive screenform to create a circular block using the coordinates end point and center point.

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

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

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

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

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

Table 6-2  Input in the interactive screenform

<table>
<thead>
<tr>
<th>Description</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>End point of the circle</td>
<td>E</td>
<td>Enter the end point of the circle.</td>
</tr>
<tr>
<td>Angle of straight line</td>
<td>A</td>
<td>The angle is specified in the counterclockwise direction from 0 to 360 degrees.</td>
</tr>
<tr>
<td>Radius of the circle</td>
<td>R</td>
<td>Input field for the circle radius</td>
</tr>
<tr>
<td>Feedrate</td>
<td>F</td>
<td>Input field for the interpolation feedrate</td>
</tr>
<tr>
<td>Center point of the circle</td>
<td>M</td>
<td>If there is no tangential transition between the straight line and the circle, the circle center point must be known. The specification is performed depending on the type of calculation (absolute, incremental or polar coordinates) selected in the previous block.</td>
</tr>
</tbody>
</table>

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

You can choose between tangential or any transition.

The screenform generates a straight line and a circle block from the data you have entered. If several points of intersection exist, the desired point of intersection must be selected from a dialog box. If one coordinate was not entered, the program tries to calculate it from the existing specifications. If there are several possibilities, a dialog box is provided to choose from.

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

Fig. 6-14  Tangential transition
Table 6-3  Input in the interactive screenform

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Enter the end point of the straight line in absolute, incremental or polar coordinates.</th>
</tr>
</thead>
<tbody>
<tr>
<td>End point of straight line</td>
<td>E</td>
<td>Center point</td>
<td>Enter the center point of the circle in absolute, incremental or polar coordinates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Radius of the circle</td>
<td>Input field for the circle radius</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Angle of straight line 1</td>
<td>The angle is specified in the counterclockwise direction from 0 to 360 degrees and with reference to the point of intersection.</td>
</tr>
</tbody>
</table>
|                      |   | Feedrate              | Input field for the interpolation feedrate                                              

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

You can choose between tangential or any transition.

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

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

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

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

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

Fig. 6-15
Table 6-4  Input in the interactive screenform

<table>
<thead>
<tr>
<th>End point</th>
<th>E</th>
<th>1. and 2nd geometry axes of the plane</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>If no coordinates are entered, this function provides the point of intersection between the circle sector you have inserted and sector 2.</td>
</tr>
<tr>
<td>Center point of the circle 1</td>
<td>M1</td>
<td>1st and 2nd geometry axes of the plane (absolute coordinates)</td>
</tr>
<tr>
<td>Radius of circle 1</td>
<td>R1</td>
<td>Input field for radius 1</td>
</tr>
<tr>
<td>Center point of circle 2</td>
<td>M2</td>
<td>1st and 2nd geometry axes of the plane (absolute coordinates)</td>
</tr>
<tr>
<td>Radius of circle 1</td>
<td>R2</td>
<td>Input field for radius 2</td>
</tr>
<tr>
<td>Feedrate</td>
<td>F</td>
<td>Input field for the interpolation feedrate</td>
</tr>
</tbody>
</table>

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

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

```
sector 1   sector 2
G2         G3, G3
G3         G2, G3
G2         G2
G3         G3
```

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

Example DIAMON

Fig. 6-16
Part Programming

6.3 Blueprint programming

Given:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>50 mm</td>
</tr>
<tr>
<td>R2</td>
<td>100 mm</td>
</tr>
<tr>
<td>R3</td>
<td>40 mm</td>
</tr>
<tr>
<td>M1</td>
<td>Z = -159 X 138</td>
</tr>
<tr>
<td>M2</td>
<td>Z = -316 X 84</td>
</tr>
<tr>
<td>M3</td>
<td>Z = -413 X 292</td>
</tr>
</tbody>
</table>

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

After you have confirmed the starting point, use the screenform to calculate the contour section \( C_1 - L_1 - C_2 \).

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

The center point coordinates must be entered as absolute coordinates, i.e. the X coordinate with reference to zero.

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

![Image](image1.png)

**Fig. 6-19** Result of step 1

Since the end point has been left open, the point of intersection of the straight line \( L_1 \) with the circle sector \( C_2 \) will be used as the starting point for the next contour definition.

Now, call the interactive screenform for calculating the contour section \( C_2 - C_3 \) again. The end point of the contour section are the coordinates \( Z=-413.0 \) and \( X=212 \).

![Image](image2.png)

**Fig. 6-20** Calling the screenform

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

![Image](image3.png)

**Fig. 6-21** Result of step 2
6.3 Blueprint programming

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

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

You can choose between tangential or any transition.

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

**Selecting the point of intersection**

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

Fig. 6-23 Selecting the point of intersection

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

Fig. 6-24

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

Fig. 6-25
Pressing OK accepts the point of intersection of the displayed contour into the part program.

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

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

A help screen is provided to display the selected settings.

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

![Screenform for calculating the contour section circle-circle-circle](image)

**Fig. 6-26 Screenform for calculating the contour section circle-circle-circle**

<table>
<thead>
<tr>
<th>Table 6-6 Input in the interactive screenform</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>End point</strong></td>
</tr>
<tr>
<td><strong>Center point of the circle 1</strong></td>
</tr>
<tr>
<td><strong>Radius of circle 1</strong></td>
</tr>
<tr>
<td><strong>Center point of circle 2</strong></td>
</tr>
<tr>
<td><strong>Radius of circle 1</strong></td>
</tr>
<tr>
<td><strong>Radius of circle 3</strong></td>
</tr>
<tr>
<td><strong>Feedrate</strong></td>
</tr>
</tbody>
</table>

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

Use this softkey to define the direction of rotation of the two circles. You can choose between **G2/G3**.
6.3 Blueprint programming

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

Example DIAMON – G23

Given:

- R1 39 mm
- R2 69 mm
- R3 39 mm
- R4 49 mm
- R5 39 mm

M1 Z −111 X 196
M2 Z −233 X 260
M3 Z −390 X 162

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

After you have confirmed the starting point, use the screenform to calculate the contour section. The end point is left open, since the coordinates are not known.
Use softkey 1 to set the direction of rotation of the two circles (G2 – G2 – G3) and to fill out the parameter list.

Fig. 6-28 Setting the starting point

Fig. 6-29 Input of step 1

Fig. 6-30 Result of step 1

The function provides the point of intersection between circle sector 2 and circle sector 3 as the end point.

In the second step, use the screenform to calculate the contour section. For calculation, select the direction of rotation G2 – G3 – G2. Starting point is the end point of the first calculation.
The function provides the point of intersection between circle sector 4 and circle sector 5 as the end point.

To calculate the tangential transition between \( \odot \) and \( \square \) use the screenform "Circle – straight line".
The function inserts a circle sector (with tangential transitions) between two straight lines. The circle sector is described by the center point and the radius. Specify the coordinates of the end point of the second straight line and, optionally, the angle A2. The first straight line is described by the starting point and the angle A1.

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

<table>
<thead>
<tr>
<th>Point</th>
<th>Given coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting point</td>
<td>• Both coordinates in a Cartesian coordinate system</td>
</tr>
<tr>
<td></td>
<td>• Starting point as a polar coordinate</td>
</tr>
<tr>
<td>Circle sector</td>
<td>• Both coordinates in the Cartesian coordinate system and the radius</td>
</tr>
<tr>
<td></td>
<td>• Center point as a polar coordinate</td>
</tr>
<tr>
<td>End point</td>
<td>• Both coordinates in a Cartesian coordinate system</td>
</tr>
<tr>
<td></td>
<td>• End point as a polar coordinate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Point</th>
<th>Given coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting point</td>
<td>• Both coordinates in a Cartesian coordinate system</td>
</tr>
<tr>
<td></td>
<td>• Starting point as a polar coordinate</td>
</tr>
<tr>
<td>Circle sector</td>
<td>• One coordinate in the Cartesian coordinate system and the radius</td>
</tr>
<tr>
<td></td>
<td>• Angle A1 or A2</td>
</tr>
<tr>
<td>End point</td>
<td>• Both coordinates in a Cartesian coordinate system</td>
</tr>
<tr>
<td></td>
<td>• End point as a polar coordinate</td>
</tr>
</tbody>
</table>

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

Table 6-7 Input in the interactive screenform

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

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

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

**G2/G3**
6.4 Simulation

Note
This function is only offered by the 802D bl with the color display option.

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

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

The start screen will appear.

![Simulation start screen](image)

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

Softkeys
If you select this softkey, the recorded tool path is scaled automatically.

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

Select this softkey to display the whole workpiece.

Use this softkey to enlarge the displayed section.
6.5 Data transfer via the RS232 interface

Functionality

The RS232 interface of the control system can be used to output data (e.g. part programs) to an external data backup device or to read in data from there. The RS232 interface and your data backup device must be matched with each other. (see also Chapter 7)

File types

- Part programs
  - Part programs
  - Subroutines
- Cycles
  - Standard cycles

Operating sequence

You have selected the Program Manager area. A list of all programs already created is displayed.

Use this softkey to saved files via the RS232 interface.
6.5 Data transfer via the RS232 interface

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

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

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

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:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>Transmission completed successfully</td>
</tr>
<tr>
<td>ERR EOF</td>
<td>End-of-text character received, but archive file incomplete</td>
</tr>
<tr>
<td>Time Out</td>
<td>The time monitoring is reporting an interruption of the data transfer</td>
</tr>
<tr>
<td>User Abort</td>
<td>Data transfer aborted by the Stop softkey</td>
</tr>
<tr>
<td>Error Com</td>
<td>Error at the COM 1 port</td>
</tr>
<tr>
<td>NC / PLC Error</td>
<td>Error message from the NC</td>
</tr>
<tr>
<td>Error Data</td>
<td>Data error</td>
</tr>
<tr>
<td>Error File Name</td>
<td>The file name does not correspond to the name convention of the NC.</td>
</tr>
</tbody>
</table>
Functionality

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

Fig. 7-1 The "System" start screen

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

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

The softkeys marked with an asterisk ("\(^{*}\)) are not available with the 802D bl.
Softkey

Setting the password

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

- System password
- Manufacturer password
- User password

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

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

After you have selected the OK softkey, the password is set.

Use ABORT to return without any action to the System main screen.

Changing the password

After you have selected the OK softkey, the password is set.

Use ABORT to return without any action to the System main screen.
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 main screen.

Resetting the access right

Switching the language

Use this softkey to switch between foreground and background language.

Saving data

This function will save the contents of the volatile memory into a nonvolatile memory area. 

Prerequisite: There is no program currently executed.

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

Start-up

Use this softkey to select the power-up mode of the NC.

Select the desired mode using the cursor.

- Normal power-up
  The system will be restarted.

- Power-up with default data
  Cold restart with the default values (will restore the default data as on delivery)

- Power-up with saved data
  Cold restart with the data saved last (see "Data backup")

The PLC can be started in the following modes:

- Restart    Cold restart
- Overall reset    Overall reset

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

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

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

Machine data

Any changes in the machine data have a substantial influence on the machine.
Caution
Faulty parameterization may result in destruction of the machine.

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

**General machine data**

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

**Axis–specific machine data**

Open the *Axis–specific machine data* window. The softkey bar will be added by the softkeys **Axis +** and **Axis -**.
The data of axis 1 are displayed.

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

**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.

Use this softkey to continue searching for the next match.

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

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

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

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

**Deselect all** softkey: Selecting this softkey deselects all data groups.
Fig. 7-8  Display filter

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

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

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

Note for the reader
For a description of the machine data, please refer to the Manufacturer’s Documentation:  
"SINUMERIK 802D, Start–up"  
"SINUMERIK 802D, Description of Functions".

Selecting this softkey displays the Service axes window.

This window displays information on the axis operation.

The Axis+ or Axis- softkeys are additionally displayed. These can be used to display the values for the next or previous axis.

This window displays information in respect of the digital drive.

This window displays information in respect of the PROFIBUS settings.
To optimize the drives, an oscilloscope function is provided for graphical representation
- of the velocity setpoint
  The velocity setpoint corresponds to the \( \pm 10 \text{V} \) interface.
- of the contour violation
- of the following error
- of the actual position value
- of the position setpoint
- of exact stop coarse / fine

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

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

![Fig. 7-9 The Servo trace start screen](image)

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

The diagram shown above can be moved within the visible screen area using the cursor keys.
Use this menu to parameterize the measuring channel.

- **Selecting the axis:** To select the axis, use the "Axis" toggle field.
- **Signal type:** Following error
  - Servo difference
  - Contour violation
  - Actual position value
  - Velocity actual value
  - Velocity setpoint
  - Compensation value
  - Parameter set
  - Position setpoint at controller input
  - Velocity setpoint at controller input
  - Acceleration setpoint at controller input
  - Velocity feedforward control value
  - Exact stop fine signal
  - Exact stop coarse signal

- **Status:**
  - On: The tracing is carried out in this channel
  - Off: The channel is inactive.

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

- **Determining the measuring time:** The measuring time in ms is entered directly in the "Measuring time" input field (max. 6,133 ms).
• **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

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

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

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.

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

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

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

*Fig. 7-12*
The markers are moved using the cursor keys at a step size of one increment. Larger step sizes can be set using the input fields. The value specifies how many grid units the marker must be moved per cursor movement. If a marker reaches the margin of the diagram, the grid automatically appears in the horizontal or vertical direction.

Use this softkey to save or load trace data.

![Fig. 7-13](image)

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. Subsequently, the file can be exported via the RS232 interface, and the data can be processed in MS Excel.

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

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

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

<table>
<thead>
<tr>
<th>Designation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Softkey</td>
<td>SK1 to SK7 Hardkey assignment 1 to 7</td>
</tr>
<tr>
<td>DLL name</td>
<td>Name of the program to be executed</td>
</tr>
<tr>
<td>Class name</td>
<td>Identifier for receiving messages</td>
</tr>
<tr>
<td>Start method</td>
<td>Number of the function executed after starting the program</td>
</tr>
</tbody>
</table>
| 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 7-1  Meanings of the entries under [DLL arrangement], cont’d

<table>
<thead>
<tr>
<th>Designation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password level</td>
<td>The execution of the program depends on the password level.</td>
</tr>
<tr>
<td>Class SK</td>
<td>Reserved</td>
</tr>
<tr>
<td>SK file</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

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

Defining the start program

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

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

Fig. 7-16

Fig. 7-17  Modifying the start-up DLL
This softkey offers further functions for diagnostics and start-up of the PLC.

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

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

The RS232 interface is initialized with activation of the connection.

---

**Fig. 7-18** Setting the baud rate

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

**Fig. 7-19** Settings with the modem active

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

- Parity:  "None" with 10–bit format
  "Even" with 11–bit format
- Stop bits: 1 (set by default; active with initialization of the control system)
- Data bits: 8 (set by default; active with initialization of the control system)
Use this softkey to activate the connection between the control system and the PC/PG. It is waited for the call of the Programming Tool. No modifications to the settings are possible in this state.

The softkey designations will change to Connect off.

You can cancel the transmission from the control system at any time by pressing Connect off. Now it is possible again to make changes in the settings.

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

Press RECALL to quit the menu.

In this area, the modem settings are made.

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

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

Fig. 7-20 Settings for an analog modem

When specifying several AT strings, AT must merely be written once; the remaining commands can simply be attached, e.g. AT&FS0=1E1X0&W. For the individual commands and their parameters, please refer to the manuals of the appropriate manufacturers. The default values of the control system are therefore only a real minimum and should be verified very exactly in any case before they are used for the first time. To be on the safe side, it is recommended to connect the devices first to a PC/PG and then to test and optimize the establishment of the connection.
Use this function to display and change the current states of the memory areas listed in Table 7-2.

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

<table>
<thead>
<tr>
<th>Table 7-2 Memory areas</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inputs</strong></td>
</tr>
<tr>
<td><strong>Outputs</strong></td>
</tr>
<tr>
<td><strong>Flags</strong></td>
</tr>
<tr>
<td><strong>Timers</strong></td>
</tr>
<tr>
<td><strong>Counter</strong></td>
</tr>
<tr>
<td><strong>Data</strong></td>
</tr>
<tr>
<td><strong>Format</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The binary representation is not possible with double words. Counters and timers are represented decimally.
The operand address displays the value incremented by 1.

The operand address displays the value decremented by 1.

Use this softkey to delete all operands.

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

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

![PLC status list start screen](image)

Use this softkey to assign the active column a new area. To this end, the interactive screen-form 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 screen-form, the control system will save your settings.

Use the cursor keys and the PageUp / PageDownkeys to navigate in and between the columns.
Use this softkey to change the value of the variable. Select the **Accept** softkey to confirm your changes.

PLC diagnosis shown as a ladder diagram (LAD) (see Section 7.1)
This function is not offered by the 802D bl.

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.

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

This function is not offered by the 802D bl.
.. writes the selected file name to the clipboard

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

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

**Structure of the reference list** (file PLCPROG.LST)

It is divided into 3 areas:

<table>
<thead>
<tr>
<th>Number</th>
<th>Area</th>
<th>Protection level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ... 100</td>
<td>User area</td>
<td>User</td>
</tr>
<tr>
<td>101 ... 200</td>
<td>Machine manufacturer</td>
<td>Machine manufacturer</td>
</tr>
<tr>
<td>201 ... 255</td>
<td>Siemens</td>
<td>Siemens</td>
</tr>
</tbody>
</table>

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

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

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.

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.

The window is divided into two columns. The left column is used to select the data group, and the right−hand column is used to select individual data for transfer. If the cursor is positioned in the left−hand column, the whole data group is output when **Read out** is selected. If it is positioned in the right−hand column, only the selected file is transferred. You can use the TAB key to switch between the two columns.
In the **NC Card** selection area, the set interface parameters are ineffective. When reading in data from **NC Card**, first the desired area must be selected.

If when reading in one of the areas

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

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

---

**Note**

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

---

**Note**

These functionalities are not applicable to the 802D bl. 

- Part programs NC -> NC_CARD
- Part programs NC_CARD -> NC.

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

To read in data from an external device, use the "Read in" function. For reading in, it is not necessary to select the data group, since the target is determined by the data flow.
Use this function to display and change the interface parameters. The type of data to be transferred can be selected using the **Text Format** and **Binary Format** softkey functions.

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.

### Interface parameters

Table 7-3 Interface parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td><strong>RTS/CTS</strong>&lt;br&gt;The signal RTS (Request to Send) controls the Send mode of the data transfer device. Active: Data are to be sent.&lt;br&gt;Passive: The Send mode is only quit after all data have been transmitted.&lt;br&gt;The CTS signal indicates the readiness to transmit data as the acknowledgment signal for RTS.</td>
</tr>
<tr>
<td>Baud rate</td>
<td>... used to set the interface transmission rate.&lt;br&gt;300 Baud&lt;br&gt;600 Baud&lt;br&gt;1200 Baud&lt;br&gt;2400 Baud&lt;br&gt;4800 Baud&lt;br&gt;9600 Baud&lt;br&gt;19200 Baud&lt;br&gt;38400 Baud&lt;br&gt;57600 Baud&lt;br&gt;115200 Baud</td>
</tr>
<tr>
<td>Stop bits</td>
<td>Number of stop bits with asynchronous transmission&lt;br&gt;Input:&lt;br&gt;1 stop bit (default setting)&lt;br&gt;2 stop bits</td>
</tr>
</tbody>
</table>
### Table 7-3 Interface parameters, cont’d

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parity</strong></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Input:</td>
</tr>
<tr>
<td></td>
<td>No parity (default setting)</td>
</tr>
<tr>
<td></td>
<td>Parity even</td>
</tr>
<tr>
<td></td>
<td>Parity odd</td>
</tr>
<tr>
<td><strong>Data bits</strong></td>
<td>Number of data bits with asynchronous transmission</td>
</tr>
<tr>
<td></td>
<td>Input:</td>
</tr>
<tr>
<td></td>
<td>7 data bits</td>
</tr>
<tr>
<td></td>
<td>8 data bits (default setting)</td>
</tr>
<tr>
<td><strong>Overwriting with confirmation</strong></td>
<td>Y: When reading in, it is checked whether the file already exists in the NC.</td>
</tr>
<tr>
<td></td>
<td>N: The files are overwritten without confirmation warning.</td>
</tr>
</tbody>
</table>
7.1 PLC diagnosis represented as a ladder diagram

**Note**

This function is not offered by the 802D bl.

**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**

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

The project stored in the permanent memory is opened.

**7.1.1 Screen layout**

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

<table>
<thead>
<tr>
<th>Screen Control</th>
<th>Display</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Application area</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Supported PLC program language</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Name of the active program block</td>
<td>Representation: Symbolic name (absolute name)</td>
</tr>
<tr>
<td>4</td>
<td>Program status</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Display of the active keys</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Focus</td>
<td>performs the tasks of the cursor</td>
</tr>
<tr>
<td>7</td>
<td>Tip line</td>
<td>contains notes for searching</td>
</tr>
</tbody>
</table>

#### Operating options

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

**Key combinations**

The cursor keys move the focus over the PLC user program. When reaching the window borders, it is scrolled automatically.
### 7.1 PLC diagnosis represented as a ladder diagram

#### Table 7-4 Key combinations

<table>
<thead>
<tr>
<th>Key combination</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>to the first line of the row</td>
</tr>
<tr>
<td></td>
<td>to the last line of the row</td>
</tr>
<tr>
<td></td>
<td>up a screen</td>
</tr>
<tr>
<td></td>
<td>down a screen</td>
</tr>
<tr>
<td></td>
<td>one field to the left</td>
</tr>
<tr>
<td></td>
<td>one field to the right</td>
</tr>
<tr>
<td></td>
<td>up a field</td>
</tr>
<tr>
<td></td>
<td>down a field</td>
</tr>
<tr>
<td></td>
<td>to the first field of the first network</td>
</tr>
<tr>
<td></td>
<td>to the last field of the first network</td>
</tr>
<tr>
<td></td>
<td>opens the next program block in the same window</td>
</tr>
<tr>
<td></td>
<td>opens the previous program block in the same window</td>
</tr>
<tr>
<td></td>
<td>The function of the Select key depends on the position of the input focus.</td>
</tr>
<tr>
<td></td>
<td>- Table line: Displays the complete text line</td>
</tr>
<tr>
<td></td>
<td>- Network title: Displays the network comment</td>
</tr>
<tr>
<td></td>
<td>- Command: Displays the complete operands</td>
</tr>
<tr>
<td></td>
<td>If the input focus is positioned on a command, all operands including the comments are displayed.</td>
</tr>
</tbody>
</table>
7.1 PLC diagnosis represented as a ladder diagram

Softkeys

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

![Fig. 7-30 PLCinfo](image)

Use this softkey to refresh the data in the window.

![Fig. 7-31 PLC status display](image)

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

Use the **PLC status list** function to display and modify PLC signals.
7.1 PLC diagnosis represented as a ladder diagram

Fig. 7-32 Status list

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

Fig. 7-33 Window 1

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

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

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

Pressing this softkey displays the table of local variables of the selected program block. There are two types of program blocks.

- **OB1**  
  only temporary local variable

- **SBRxx**  
  temporary local variable

A table of variables exists for each program block.
System

7.1 PLC diagnosis represented as a ladder diagram

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

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

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

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

**Fig. 7-37** "Program status" ON – symbolic representation
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.

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%

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

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

Selection in the upper toggle field:

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

Further search criteria:

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

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

Depending on the display settings, you can search for symbolic or absolute operands.
Press the **OK** softkey to start the search. The found search element is highlighted by the focus. If nothing is found, an appropriate error message will appear in the notes line.

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

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

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

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

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.

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.

... 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)

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.
8.1 Fundamentals of NC programming

8.1.1 Program names

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

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

Example: SHAFT527

8.1.2 Program structure

Structure and contents

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

Each block represents a machining step.

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

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

Table 8-1 NC program structure

<table>
<thead>
<tr>
<th>Block</th>
<th>Word</th>
<th>Word</th>
<th>Word</th>
<th>...</th>
<th>; Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block</td>
<td>N10</td>
<td>G0</td>
<td>X20</td>
<td>...</td>
<td>; 1st block</td>
</tr>
<tr>
<td>Block</td>
<td>N20</td>
<td>G2</td>
<td>Z37</td>
<td>...</td>
<td>; 2nd block</td>
</tr>
<tr>
<td>Block</td>
<td>N30</td>
<td>G91</td>
<td>...</td>
<td>...</td>
<td>; ...</td>
</tr>
<tr>
<td>Block</td>
<td>N40</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>..</td>
</tr>
<tr>
<td>Block</td>
<td>N50</td>
<td>M2</td>
<td></td>
<td></td>
<td>; End of program</td>
</tr>
</tbody>
</table>
8.1 Fundamentals of NC programming

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
- **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.

<table>
<thead>
<tr>
<th>Word</th>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X-20.1</td>
<td>Travel or end position for the X axis: −20.1mm</td>
<td></td>
</tr>
<tr>
<td>F300</td>
<td>Feedrate: 300 mm/min</td>
<td></td>
</tr>
</tbody>
</table>

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 parameters (R parameters)
- H  H function
- I, J, K  Interpolation parameters / intermediate point
- M  Miscellaneous function M; only pertains to the spindle
- S  Spindle speed (spindle 1 or 2),

the address is extended by 1 through 4 digits to achieve a larger 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  M2=5 S2=400
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 "LF" (LineFeed). This character is automatically generated when pressing the line feed key or the Input key.

<table>
<thead>
<tr>
<th>/N...</th>
<th>Word1</th>
<th>Word2</th>
<th>...</th>
<th>Wordn</th>
<th>:Comment</th>
<th>LF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space</td>
<td>Space</td>
<td>Space</td>
<td>Space</td>
<td></td>
<td>End-of-block character</td>
<td></td>
</tr>
<tr>
<td>Block number – stands in front of the instructions; only if necessary; instead of &quot;N&quot;, a colon &quot; &quot; is used in main blocks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skip block; only if necessary; stands in the beginning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total number of characters in a block: 200 characters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8-2 Block structure diagram

Word order

If a block contains several instructions, 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). It is also possible to skip a whole program section by skipping several blocks using the "/".

If block skip is active during the program execution, all blocks marked with "/" are skipped. All instructions contained in the blocks concerned will not be considered. The program is continued with the next block without marking.
Comment, remark

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

Messages

Messages are programmed in a separate block. A message is displayed in a special field and remains active until a block with a new message is executed or until the end of the program is reached. Max. 65 characters of a text message can be displayed. A message without message text will delete any previous message.

MSG ("THIS IS THE MESSAGE TEXT")

Programming example

```
N10 ; Company G&S, order no. 12A71
N20 ; Pump part 17, drawing no.: 123 677
N30 ; Program created by H. Adam, Dept. TV 4
N40 MSG("BLANK, ROUGHING")
N50 G54 F4.7  S220 D2 M3 ; Main block
N60 G0 G90 X100 Z200
N70 G1 Z185.6
N80 X112
N90 X118 Z180 ; Block can be skipped
N100 X118 Z120
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


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]
8.1 Fundamentals of NC programming

[ ] Right square bracket
< Less than
> Greater than
: Main block, completion of label
= Assignment; part of equality
/ Division; block skip
* Multiplication
+ Addition; positive sign
− Subtraction; negative sign
, Comma, delimiter
; Start of a comment
% Reserved; do not use
& reserved; do not use
' Reserved; do not use
$ System–internal variable identifier
? Reserved; do not use
!

Non-printable special characters

LF End–of–block character
Blank Delimiter between the words; blank
Tabulator Reserved; do not use
## 8.1.6 Overview of the instructions

Valid as of software version 2.0.

<table>
<thead>
<tr>
<th>Address</th>
<th>Meaning</th>
<th>Value assignment</th>
<th>Information</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Tool offset number</td>
<td>0 ... 9, only integer, no sign</td>
<td>Contains offset data for a certain tool T...; D0-&gt; offset values= 0, max. 9 D numbers per tool</td>
<td>D...</td>
</tr>
<tr>
<td>F</td>
<td>Feedrate</td>
<td>0.001 ... 99 999.999</td>
<td>Path velocity of a tool/workpiece; unit: mm/min or mm/revolution depending on G94 or G95</td>
<td>F...</td>
</tr>
<tr>
<td>F</td>
<td>Dwell time (block with G4)</td>
<td>0.001 ... 99 999.999</td>
<td>Dwell time in seconds</td>
<td>G4 F... ; separate block</td>
</tr>
<tr>
<td>F</td>
<td>Thread lead change (block with G34, G35)</td>
<td>0.001 ... 99 999.999</td>
<td>in ( \text{mm}/\text{rev}^2 )</td>
<td>see with G34, G35</td>
</tr>
<tr>
<td>G</td>
<td>G function (preparatory function)</td>
<td>Only integer, specified values</td>
<td>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 canceled by another function of the same group) or only effective for the block in which it is programmed (non-modal).</td>
<td>G... or symbolic name, e.g.: CIP</td>
</tr>
<tr>
<td>G0</td>
<td>Linear interpolation at rapid traverse rate</td>
<td>1: Motion commands</td>
<td>(type of interpolation)</td>
<td>G0 X... Z...</td>
</tr>
<tr>
<td>G1 *</td>
<td>Linear interpolation at feedrate</td>
<td></td>
<td></td>
<td>G1 X... Z... F...</td>
</tr>
<tr>
<td>G2</td>
<td>Circular interpolation CW</td>
<td></td>
<td></td>
<td>G2 X... Z... I... K... F... ; center and end points</td>
</tr>
<tr>
<td></td>
<td>Circular interpolation CCW</td>
<td></td>
<td></td>
<td>G2 AR=... X... Z... F... ; aperture angle and end point</td>
</tr>
<tr>
<td>G3</td>
<td>Circular interpolation via intermediate point</td>
<td></td>
<td></td>
<td>G3 .... ; otherwise, as with G2</td>
</tr>
<tr>
<td>CIP</td>
<td>Circular interpolation via intermediate point</td>
<td></td>
<td></td>
<td>CIP X... Z... I1=... K1=... F... ; I1, K1 is intermediate point</td>
</tr>
<tr>
<td>CT</td>
<td>Circular interpolation; tangential transition</td>
<td></td>
<td></td>
<td>N10 ....</td>
</tr>
<tr>
<td></td>
<td>Circular interpolation; tangential transition</td>
<td></td>
<td></td>
<td>N20 CT Z... X... F... ; circle; tangential transition to the previous path segment N10</td>
</tr>
<tr>
<td>G33</td>
<td>Thread cutting with constant lead</td>
<td>modally effective</td>
<td></td>
<td>G33 Z... K... SF=... ; Cylindrical thread</td>
</tr>
<tr>
<td></td>
<td>Thread cutting with constant lead</td>
<td></td>
<td></td>
<td>G33 X... I... SF=... ; Transversal thread</td>
</tr>
<tr>
<td></td>
<td>Thread cutting with constant lead</td>
<td></td>
<td></td>
<td>G33 Z... X... K... SF=... ; Tapered thread; travel along the Z axis</td>
</tr>
<tr>
<td></td>
<td>Thread cutting with constant lead</td>
<td></td>
<td></td>
<td>G33 Z... X... I... SF=... ; Tapered thread; travel along the X axis larger than along the Z axis</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Example</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------------------------</td>
<td>----------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G34</td>
<td>Thread cutting, increasing lead</td>
<td>G33 Z... K... SF=... ; Cylindrical thread; constant lead</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>G34 Z... K... F17.123 ; increasing lead with 17.123 mm/rev.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G35</td>
<td>Thread cutting, decreasing lead</td>
<td>G33 Z... K... SF=... ; Cylindrical thread</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>G35 Z... K... F7.321 ; decreasing lead with 7.321 mm/rev.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G331</td>
<td>Thread interpolation</td>
<td>N10 SPOS=... ; Position–controlled spindle</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>N20 G331 Z... K... S... ; Rigid tapping (without compensation chuck, e.g. along the Z axis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>; RH or LH thread is defined by the sign of the lead (e.g. K+): + : as with M3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>− : as with M4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G332</td>
<td>Thread interpolation – retraction</td>
<td>G332 Z... K... ; rigid tapping (without compensation chuck, e.g. along the Z axis, retraction motion)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>; Sign of the lead as with G331</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G4</td>
<td>Dwell time</td>
<td>G4 F... ; separate block, F: Time in seconds</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>or G4 S..... ; separate block, S: in spindle revolutions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G74</td>
<td>Reference point approach</td>
<td>G74 X1=0 Z1=0 ; separate block (machine axis identifier!)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G75</td>
<td>Fixed–point approach</td>
<td>G75 X1=0 Z1=0 ; separate block (machine axis identifier!)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANS</td>
<td>Programmable offset</td>
<td>TRANS X... Z... ; separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCALE</td>
<td>Programmable scaling factor</td>
<td>SCALE X... Z... ; scaling factor in the direction of the specified axis, separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROT</td>
<td>Programmable rotation</td>
<td>ROT RPL=... ; rotation in the current plane G17 ... G19, separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIRROR</td>
<td>Programmable mirroring</td>
<td>MIRROR X0 ; coordinate axis whose direction is changed; separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATRANS</td>
<td>Additive programmable offset</td>
<td>ATRANS X... Z... ; separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASCALE</td>
<td>Additive programmable scaling factor</td>
<td>ASCALE X... Z... ; scaling factor in the direction of the specified axis, separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AROT</td>
<td>Additive programmable rotation</td>
<td>AROT RPL=... ; Add. rotation in the current plane G17 ... G19, separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMIRROR</td>
<td>additive programmable mirroring</td>
<td>AMIRROR X0</td>
<td>; coordinate axis whose direction is changed; separate block</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------</td>
<td>------------</td>
<td>-------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>G25</td>
<td>Lower spindle speed limitation or lower working area limitation</td>
<td>G25 S... ; separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G25 X... Z... ; separate block</td>
<td>G26 S... ; separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G26</td>
<td>Upper spindle speed limitation or upper working area limitation</td>
<td>G26 X... Z... ; separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G17</td>
<td>X/Y plane (when center–drilling, TRANSMIT milling required)</td>
<td>6: Plane selection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G18 *</td>
<td>Z/X plane (standard turning)</td>
<td>7: Tool radius compensation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G19</td>
<td>Y/Z plane (required for TRACYL milling)</td>
<td>modally effective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G40 *</td>
<td>Tool radius compensation OFF</td>
<td>8: Settable work offset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G41</td>
<td>Tool radius compensation left of the contour</td>
<td>modally effective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G42</td>
<td>Tool radius compensation right of the contour</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G500 *</td>
<td>Settable work offset OFF</td>
<td>9: Skipping of the settable work offset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G54</td>
<td>1st settable work offset</td>
<td>non–modal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G55</td>
<td>2nd settable work offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G56</td>
<td>3rd settable work offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G57</td>
<td>4th settable work offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G58</td>
<td>5th settable work offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G59</td>
<td>6th settable work offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G53</td>
<td>Non–modal skipping of the settable work offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G153</td>
<td>Non–modal skipping of the settable work offset including base frame</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G60 *</td>
<td>Exact stop</td>
<td>10: Approach behavior</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G64</td>
<td>Continuous–path control mode</td>
<td>modally effective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G9</td>
<td>Non–modal exact stop</td>
<td>11: Non–modal exact stop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G601 *</td>
<td>Exact stop window, fine, with G60, G9</td>
<td>non–modal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G602</td>
<td>Exact stop window, coarse, with G60, G9</td>
<td>12: Exact stop window</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>modally effective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function Code</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G70</td>
<td>Inch dimension input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G71 *</td>
<td>Metric dimension data input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G700</td>
<td>Inch dimension data input; also for feedrate F</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G710</td>
<td>Metric dimension data input; also for feedrate F</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G90 *</td>
<td>Absolute dimension data input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G91</td>
<td>Incremental dimension data input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G94</td>
<td>Feed F in mm/min</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G95 *</td>
<td>Feedrate F in mm/spindle revolutions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G96</td>
<td>Constant cutting rate ON (F in mm/rev., S in m/min)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G97</td>
<td>Constant cutting rate OFF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G450 *</td>
<td>Transition circle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G451</td>
<td>Point of intersection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRISK *</td>
<td>Jerking path acceleration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOFT</td>
<td>Jerk-limited path acceleration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FFWOF *</td>
<td>Feedforward control OFF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FFWON</td>
<td>Feedforward control ON</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WALIMON *</td>
<td>Working area limitation ON</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WALIMOF</td>
<td>Working area limitation OFF</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAMOF</td>
<td>Radius dimension input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAMON *</td>
<td>Diameter dimensioning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G290 *</td>
<td>SIEMENS mode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G291</td>
<td>External mode (not with 802D–bl)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

13: Inch / metr. dimension input
14: Absolute / incremental dimension
15: Feedrate / spindle
18: Behavior at corners when working with tool radius compensation
21: Acceleration profile
24: Feedforward control
28: Working area limitation
29: Dimension input
47: External NC languages

The functions marked with an asterisk (*) act when starting the program (in the default condition of the control system, unless otherwise programmed and if the machine manufacturer has preserved the default settings for the turning technology).
<table>
<thead>
<tr>
<th>Address</th>
<th>Meaning</th>
<th>Value assignment</th>
<th>Information</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>H function</td>
<td>± 0.00000001 ... 99 999 9999 (8 decimals) or with specification of an exponent: 10^-300 ... 10^300</td>
<td>Value transfer to the PLC; meaning defined by the machine manufacturer</td>
<td>H0=... H9999=... e. g.: H7=23.456</td>
</tr>
<tr>
<td>I</td>
<td>Interpolation parameters</td>
<td>±0.001 ... 99 999.999 Thread: 0.001 ... 2000.000</td>
<td>Belongs to the X axis; meaning dependent on G2,G3 -&gt; circle center or G33, G34, G35</td>
<td>See G2, G3 and G33, G34, G35</td>
</tr>
<tr>
<td>K</td>
<td>Interpolation parameters</td>
<td>±0.001 ... 99 999.999 Thread: 0.001 ... 2000.000</td>
<td>Belongs to the Z axis; otherwise, as with I</td>
<td>See G2, G3 and G33, G34, G35</td>
</tr>
<tr>
<td>I1=</td>
<td>Intermediate point for circular interpolation</td>
<td>±0.001 ... 99 999.999</td>
<td>Belongs to the X axis; specification for circular interpolation with CIP</td>
<td>See CIP</td>
</tr>
<tr>
<td>K1=</td>
<td>Intermediate point for circular interpolation</td>
<td>±0.001 ... 99 999.999</td>
<td>Belongs to the Z axis; specification for circular interpolation with CIP</td>
<td>See CIP</td>
</tr>
<tr>
<td>L</td>
<td>Subroutine; name and call</td>
<td>7 decimals; integer only, no sign</td>
<td>It is also possible to use L1 ...L99999999. Instead of a free name; thus, the subroutine will be called in a separate block. Please observe: L0001 is not always equal to L1. The name &quot;LL6&quot; is reserved for the tool change subroutine.</td>
<td>L... ; separate block</td>
</tr>
<tr>
<td>M</td>
<td>Miscellaneous function</td>
<td>0 ... 99 integer only, no sign</td>
<td>For example, for initiating switching actions, such as &quot;Coolant ON&quot;; max. 5 M functions per block</td>
<td>M...</td>
</tr>
<tr>
<td>M0</td>
<td>Programmed stop</td>
<td></td>
<td>The machining is stopped at the end of a block containing M0; to continue, press NC START.</td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>Optional stop</td>
<td></td>
<td>As with M0, but the stop is only performed if a special signal (Program control: &quot;M01&quot;) is present.</td>
<td></td>
</tr>
<tr>
<td>M2</td>
<td>End of program</td>
<td></td>
<td>Can be found in the last block of the processing sequence</td>
<td></td>
</tr>
<tr>
<td>M30</td>
<td></td>
<td></td>
<td>Reserved; do not use</td>
<td></td>
</tr>
<tr>
<td>M17</td>
<td></td>
<td></td>
<td>Reserved; do not use</td>
<td></td>
</tr>
<tr>
<td>M3</td>
<td>CW rotation of spindle (for master spindle)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M4</td>
<td>CCW rotation of spindle (for master spindle)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M5</td>
<td>Spindle stop (for master spindle)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mn=3</td>
<td>CW rotation of spindle (for spindle n)</td>
<td>n = 1 or = 2</td>
<td></td>
<td>M2=3 ; CW rotation stop for spindle 2</td>
</tr>
<tr>
<td>Mn=4</td>
<td>CCW rotation of spindle (for spindle n)</td>
<td>n = 1 or = 2</td>
<td></td>
<td>M2=4 ; CCW rotation stop for spindle 2</td>
</tr>
<tr>
<td>Mn=5</td>
<td>Spindle stop (for spindle n)</td>
<td>n = 1 or = 2</td>
<td></td>
<td>M2=5 ; Spindle stop for spindle 2</td>
</tr>
<tr>
<td>Address</td>
<td>Meaning</td>
<td>Value assignment</td>
<td>Information</td>
<td>Programming</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>M6</td>
<td>Tool change</td>
<td></td>
<td>Only if activated with M6 via the machine control panel; otherwise, change directly using the T command</td>
<td></td>
</tr>
<tr>
<td>M40</td>
<td>Automatic gear stage switching</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mn=40</td>
<td>Automatic gear stage switching</td>
<td>n = 1 or = 2</td>
<td>M1=40; gear stage selected automatically; for spindle 1</td>
<td></td>
</tr>
<tr>
<td>M41 to M45</td>
<td>Gear stage 1 to gear stage 5 (for master spindle)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mn=41 to Mn=45</td>
<td>Gear stage 1 to gear stage 5 (for spindle n)</td>
<td>n = 1 or = 2</td>
<td>M2=41; 1st gear stage for spindle 2</td>
<td></td>
</tr>
<tr>
<td>M70, M19</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M...</td>
<td>Remaining M functions</td>
<td></td>
<td>Functionality is not defined by the control system and can therefore be used freely by the machine manufacturer</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>Block number of an auxiliary block</td>
<td>0 ... 9999 9999</td>
<td>Can be used to identify blocks with a number; is written in the beginning of a block</td>
<td>N20</td>
</tr>
<tr>
<td>:</td>
<td>Block number of a main block</td>
<td>0 ... 9999 9999</td>
<td>Special block identification, used instead of N...; such a block should contain all instructions for a complete subsequent machining step.</td>
<td>:20</td>
</tr>
<tr>
<td>P</td>
<td>Number of subroutine passes</td>
<td>1 ... 9999</td>
<td>Is used if the subroutine is run several times and is contained in the same block as the call</td>
<td>L781 P...</td>
</tr>
<tr>
<td>R0 to R299</td>
<td>Arithmetic parameters</td>
<td>± 0.0000001 ...</td>
<td>R1=7.9431 R2=4 With specification of an exponent: R1=−1.9876EX9 ; R1=−1 987 600 000</td>
<td></td>
</tr>
<tr>
<td>SIN()</td>
<td>Sine</td>
<td>Degrees</td>
<td>R1=SIN(17.35)</td>
<td></td>
</tr>
<tr>
<td>COS()</td>
<td>Cosine</td>
<td>Degrees</td>
<td>R2=COS(R3)</td>
<td></td>
</tr>
<tr>
<td>TAN()</td>
<td>Tangent</td>
<td>Degrees</td>
<td>R4=TAN(R5)</td>
<td></td>
</tr>
<tr>
<td>ASIN()</td>
<td>Arc sine</td>
<td>Degrees</td>
<td>R10=ASIN(0.35) ; R10: 20.487 degrees</td>
<td></td>
</tr>
<tr>
<td>ACOS()</td>
<td>Arc cosine</td>
<td>R20=ACOS(R2)     ; R20: ... degrees</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>Meaning</td>
<td>Value assignment</td>
<td>Information</td>
<td>Programming</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------</td>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>ATAN2( , )</td>
<td>Arc tangent2</td>
<td></td>
<td>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</td>
<td>R40=ATAN2(30.5,80.1) ; R40: 20.8455 degrees</td>
</tr>
<tr>
<td>SQRT( )</td>
<td>Square root</td>
<td></td>
<td></td>
<td>R6=SQRT(R7)</td>
</tr>
<tr>
<td>POT( )</td>
<td>Square</td>
<td></td>
<td></td>
<td>R12=POT(R13)</td>
</tr>
<tr>
<td>ABS( )</td>
<td>Amount</td>
<td></td>
<td></td>
<td>R8=ABS(R9)</td>
</tr>
<tr>
<td>TRUNC( )</td>
<td>Integer portion</td>
<td></td>
<td></td>
<td>R10=TRUNC(R2)</td>
</tr>
<tr>
<td>LN( )</td>
<td>Natural logarithm</td>
<td></td>
<td></td>
<td>R12=LN(R9)</td>
</tr>
<tr>
<td>EXP( )</td>
<td>Exponential function</td>
<td></td>
<td></td>
<td>R13=EXP(R1)</td>
</tr>
<tr>
<td>RET</td>
<td>End of subroutine</td>
<td></td>
<td>Used instead of M2 – to maintain the continuous–path control mode</td>
<td>RET ; separate block</td>
</tr>
<tr>
<td>S...</td>
<td>Spindle speed (master spindle)</td>
<td>0.001 ... 99 999.999</td>
<td>Unit of measurement of the spindle r.p.m.</td>
<td>S...</td>
</tr>
<tr>
<td>S1=...</td>
<td>Spindle speed for spindle 1</td>
<td>0.001 ... 99 999.999</td>
<td>Unit of measurement of the spindle r.p.m.</td>
<td>S1=725 ; speed 725 r.p.m. for spindle 1</td>
</tr>
<tr>
<td>S2=...</td>
<td>Spindle speed for spindle 2</td>
<td>0.001 ... 99 999.999</td>
<td>Unit of measurement of the spindle r.p.m.</td>
<td>S2=730 ; speed 730 r.p.m. for spindle 2</td>
</tr>
<tr>
<td>S</td>
<td>Cutting rate with G96active</td>
<td>0.001 ... 99 999.999</td>
<td>Cutting rate unit m/min with G96; for master spindle only</td>
<td>G96 S...</td>
</tr>
<tr>
<td>S</td>
<td>Dwell time in block with G4</td>
<td>0.001 ... 99 999.999</td>
<td>Dwell time in spindle revolutions</td>
<td>G4 S... ; separate block</td>
</tr>
<tr>
<td>T</td>
<td>Tool number</td>
<td>1 ... 32 000 integer only, no sign</td>
<td>The tool change can be performed either directly using the T command or only with M6. This can be set in the machine data.</td>
<td>T...</td>
</tr>
<tr>
<td>X</td>
<td>axis</td>
<td>±0.001 ... 99 999.999</td>
<td>G command</td>
<td>X...</td>
</tr>
<tr>
<td>Y</td>
<td>axis (not with 802D bl)</td>
<td>±0.001 ... 99 999.999</td>
<td>Positional data, e.g. with TRACYL, TRANSMIT</td>
<td>Y...</td>
</tr>
<tr>
<td>Z</td>
<td>axis</td>
<td>±0.001 ... 99 999.999</td>
<td>G command</td>
<td>Z...</td>
</tr>
<tr>
<td>AC</td>
<td>Absolute coordinate</td>
<td></td>
<td>The dimension can be specified for the end or center point of a certain axis, irrespective of G91.</td>
<td>N10 G91 X10 Z=AC(20) ; X – incremental dimension, Z – absolute dimension</td>
</tr>
<tr>
<td>ACC[axis]</td>
<td>Percentage path acceleration override</td>
<td>1 ... 200, integer</td>
<td>Acceleration override for an axis or spindle; specified as a percentage</td>
<td>N10 ACC[X]=80 ; N20 ACC[S]=50 ; for the X axis: 80% ; for the spindle: 50%</td>
</tr>
</tbody>
</table>
| ACP        | Absolute coordinate; approach position in the positive direction (for rotary axis, spindle) |                  | It is also possible to specify the dimensions for the end point of a rotary axis with ACP(...) irrespective of G90/G91; also applies to spindle positioning | N10 A=ACP(45.3) ; A: Approach absolute position of the A axis in the positive direction N20 SPOS=ACP(33.1) ; SPOS: Position spindle
<table>
<thead>
<tr>
<th>Address</th>
<th>Meaning</th>
<th>Value assignment</th>
<th>Information</th>
<th>Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACN</td>
<td>Absolute coordinate; approach position in the negative direction (for rotary axis, spindle)</td>
<td>–</td>
<td>It is also possible to specify the dimensions for the end point of a rotary axis with ACN(…), irrespective of G90/G91; also applies to spindle positioning</td>
<td>N10 A=ACN(45.3) ;Approach absolute position of the A axis in the negative direction</td>
</tr>
<tr>
<td>ANG</td>
<td>Angle for the specification of a straight line for the contour definition</td>
<td>±0.00001 ... 359.99999</td>
<td>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</td>
<td>N10 G1 X... Z... ANG=... or contour over several blocks: N10 G1 X... Z... N11 ANG=...</td>
</tr>
<tr>
<td>AR</td>
<td>Aperture angle for circular interpolation</td>
<td>0.00001 ... 359.99999</td>
<td>Specified in degrees; one possibility of defining the circle when using G2/G3</td>
<td>See G2, G3</td>
</tr>
<tr>
<td>CALL</td>
<td>Indirect cycle call</td>
<td>–</td>
<td>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</td>
<td>N10 CALL VARNAME ; variable name</td>
</tr>
<tr>
<td>CHF</td>
<td>Chamfer; general use</td>
<td>0.001 ... 99 999.999</td>
<td>Inserts a chamfer of the specified chamfer length between two contour blocks</td>
<td>N10 X... Z... CHF=... N11 X... Z...</td>
</tr>
<tr>
<td>CHR</td>
<td>Chamfer; in the contour definition</td>
<td>0.001 ... 99 999.999</td>
<td>Inserts a chamfer of the specified leg length between two contour blocks</td>
<td>N10 X... Z... CHR=... N11 X... Z...</td>
</tr>
<tr>
<td>CR</td>
<td>Radius for circular interpolation</td>
<td>0.010 ... 99 999.999</td>
<td>Negative sign for selecting the circle: greater than semicircle</td>
<td>One possibility of defining a circle when using G2/G3</td>
</tr>
<tr>
<td>CYCLE...</td>
<td>Machining cycle</td>
<td>Only specified values</td>
<td>The call of the machining cycles requires a separate block; the appropriate transfer parameters must be loaded with values. Special cycle calls are also possible with an additional MCALL or CALL.</td>
<td></td>
</tr>
<tr>
<td>CYCLE82</td>
<td>Drilling, counterboring</td>
<td>N5 RTP=110 RFP=100 ..., N10 CYCLE82(RTP, RFP, ...)</td>
<td>;assign values ;separate block</td>
<td></td>
</tr>
<tr>
<td>CYCLE83</td>
<td>Deep-hole drilling</td>
<td>N10 CYCLE83(110, 100, ...) ;or transfer values directly ;separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCLE84</td>
<td>Rigid tapping</td>
<td>N10 CYCLE84(...) ;separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCLE840</td>
<td>Tapping with compensation chuck</td>
<td>N10 CYCLE840(...) ;separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCLE85</td>
<td>Reaming</td>
<td>N10 CYCLE85(...) ;separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCLE86</td>
<td>Boring</td>
<td>N10 CYCLE86(...) ;separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCLE88</td>
<td>Boring with stop</td>
<td>N10 CYCLE88(...) ;separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYCLE93</td>
<td>Grooving</td>
<td>N10 CYCLE93(...) ;separate block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>Meaning</td>
<td>Value assignment</td>
<td>Information</td>
<td>Programming</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>------------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CYCLE94</td>
<td>Undercut DIN76 (forms E and F), finishing</td>
<td></td>
<td></td>
<td>N10 CYCLE94(...) ;separate block</td>
</tr>
<tr>
<td>CYCLE95</td>
<td>Stock removal with relief cutting</td>
<td></td>
<td></td>
<td>N10 CYCLE95(...) ;separate block</td>
</tr>
<tr>
<td>CYCLE97</td>
<td>Thread cutting</td>
<td></td>
<td></td>
<td>N10 CYCLE97(...) ;separate block</td>
</tr>
<tr>
<td>DC</td>
<td>Absolute coordinate; approach position directly (for rotary axis, spindle)</td>
<td></td>
<td>It is also possible to specify the dimensions for the end point of a rotary axis with DC(...) irrespective of G90/G91; also applies to spindle positioning</td>
<td>N10 A=DC(45.3) ;Approach absolute position of the A axis directly N20 SPOS=DC(33.1) ;Position spindle</td>
</tr>
<tr>
<td>DEF</td>
<td>Definition instruction</td>
<td></td>
<td>Defining a local user variable of the type BOOL, CHAR, INT, REAL, directly at the beginning of the program</td>
<td>DEF INT VAR1=24, VAR2 : 2 variables of the type INT ; the name is defined by the user</td>
</tr>
<tr>
<td>FXS [axis]</td>
<td>Travel to fixed stop</td>
<td>=1: Selection =0: Deselection</td>
<td>Axis: Use the machine identifier</td>
<td>N20 G1 X10 Z25 FXS[Z1]=1 FXST[Z1]=12.3 FXSW[Z1]=2 ...</td>
</tr>
<tr>
<td>FXST [axis]</td>
<td>Clamping torque, travel to fixed stop</td>
<td>&gt; 0.0 ... 100.0</td>
<td>in %, max. 100% from the max. torque of the drive, axis: Use the machine identifier</td>
<td>N30 FXST[Z1]=12.3</td>
</tr>
<tr>
<td>FXSW [axis]</td>
<td>Monitoring window, travel to fixed stop</td>
<td>&gt; 0.0</td>
<td>Unit of measurement mm or degrees, axis-specific, axis: Use the machine identifier</td>
<td>N40 FXSW[Z1]=2.4</td>
</tr>
<tr>
<td>GOTOB</td>
<td>GoBack instruction</td>
<td></td>
<td>A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the program start.</td>
<td>N10 LABEL1: ... N100 GOTOB LABEL1</td>
</tr>
<tr>
<td>GOTOF</td>
<td>GoForward instruction</td>
<td></td>
<td>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.</td>
<td>N10 GOTOF LABEL2 ... N130 LABEL2: ...</td>
</tr>
<tr>
<td>IC</td>
<td>Coordinate specified using incremental dimensions</td>
<td></td>
<td>The dimension can be specified for the end or center point of a certain axis irrespective of G90.</td>
<td>N10 G90 X10 Z=IC(20) ;Z – incremental dimension, X – absolute dimension</td>
</tr>
<tr>
<td>IF</td>
<td>Jump condition</td>
<td></td>
<td>If the jump condition is fulfilled, the GoTo operation to the block with the label is performed; otherwise, the next instruction/block will follow. In one block, several IF instructions are possible. <strong>Comparison operands:</strong> = = equal to, &lt;&gt; not equal &gt; greater than, &lt; less than &gt;= greater than or equal to &lt;= less than or equal to</td>
<td>N10 IF R1&gt;5 GOTOF LABEL3 ... N80 LABEL3: ...</td>
</tr>
<tr>
<td>LIMS</td>
<td>Upper limit speed of the spindle with G96, G97</td>
<td>0.001 ... 99 999.999</td>
<td>Limits the spindle speed with the G96 function enabled – constant cutting rate and G97</td>
<td>See G96</td>
</tr>
<tr>
<td>MEAS</td>
<td>Measuring with deletion of the distance to go</td>
<td>+1, −1</td>
<td>++1: Measuring input 1, rising edge −1: Measuring input 1, falling edge</td>
<td>N10 MEAS=+1 G1 X... Z... F...</td>
</tr>
<tr>
<td>MEAW</td>
<td>Measuring without deletion of the distance to go</td>
<td>+1, −1</td>
<td>++1: Measuring input 1, rising edge −1: Measuring input 1, falling edge</td>
<td>N10 MEAW=+1 G1 X... Z... F...</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Type</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>SA_DB[n]</td>
<td>Data byte</td>
<td>Data word</td>
<td>Real data</td>
<td></td>
</tr>
<tr>
<td>$A_DBR[n]$</td>
<td>Reading and writing PLC variables</td>
<td>N10 $A_DB[n]=16.3$ : write real variables with offset position 5 (position, type and meaning are agreed between NC and PLC)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SA_MONIFACT</td>
<td>Factor for tool life monitoring</td>
<td>&gt; 0.0</td>
<td>Initialization value: 1.0</td>
<td></td>
</tr>
<tr>
<td>$A_MONIFACT$</td>
<td>N10 $A_MONIFACT=5.0$ : tool life elapsed 5 times faster</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_MONIFACT$</td>
<td>N10 $A_MONIFACT=5.0$ : tool life elapsed 5 times faster</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_MONIFACT$</td>
<td>N10 $A_MONIFACT=5.0$ : tool life elapsed 5 times faster</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AA_FXS[X]$</td>
<td>Status, travel to fixed stop</td>
<td>–</td>
<td>Values: 0 ... 5 (axis: machine axis identifier)</td>
<td></td>
</tr>
<tr>
<td>$A_MONIFACT$</td>
<td>N10 IF $AA_FXS[X]==1$ GOTOF ....</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AA_MM[X]$</td>
<td>Measurement result for an axis in the machine coordinate system</td>
<td>–</td>
<td>Axis: Identifier of an axis (X, Z) traversing when measuring</td>
<td></td>
</tr>
<tr>
<td>$A_MONIFACT$</td>
<td>N10 R1=$AA_MM[X]$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AA_MW[X]$</td>
<td>Measurement result for an axis in the workpiece coordinate system</td>
<td>–</td>
<td>Axis: Identifier of an axis (X, Z) traversing when measuring</td>
<td></td>
</tr>
<tr>
<td>$A_MONIFACT$</td>
<td>N10 R2=$AA_MW[X]$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AC_MEA[X]$</td>
<td>Measurement task status</td>
<td>–</td>
<td>Default condition: 0: Default condition, probe did not switch 1: Probe switched</td>
<td></td>
</tr>
<tr>
<td>$A_MONIFACT$</td>
<td>N10 IF $AC_MEA[X]==1$ GOTOF .... ; continue program if the probe has switched ...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AN_SETUP_TIME$</td>
<td>Timer for runtime</td>
<td>0.0 ... 10^300 min (read-only value)</td>
<td>System variable: Time since the control system has last booted</td>
<td></td>
</tr>
<tr>
<td>$AN_POWERON_TIME$</td>
<td>Time since the control system has last booted normally</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AN_OPERATING_TIME$</td>
<td>Total runtime of all NC programs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AC_CYCLE_TIME$</td>
<td>Runtime of the NC program (only of the selected program)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AC_CUTTING_TIME$</td>
<td>Tool action time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AC_CYCLES_TIME$</td>
<td>N10 IF $AC_CYCLE_TIME==50.5$ ....</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AC_TOTAL_PARTS$</td>
<td>Workpiece counter</td>
<td>0 ... 999 999 999, integer</td>
<td>System variable: Total actual count; Set number of workpiece</td>
<td></td>
</tr>
<tr>
<td>$AC_REQUIRED_PARTS$</td>
<td>Current actual count</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AC.ACTUAL_PARTS$</td>
<td>Count of workpieces – specified by the user</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AC_SPECIAL_PARTS$</td>
<td>N10 IF $AC_ACTUAL_PARTS==15$ ....</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AC_MSNUM$</td>
<td>Number of the active master spindle</td>
<td>read-only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AC_MSNUM$</td>
<td>Number of programmed master spindle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$AC_NUM_SPINDLES$</td>
<td>Number of configured spindles</td>
<td>read-only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_S[n]$</td>
<td>Actual speed of spindle n</td>
<td>Spindle number n =1 or =2, read-only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_S[n]$</td>
<td>Last programmed speed of spindle n</td>
<td>Spindle number n =1 or =2, read-only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAC_</td>
<td>Current direction of rotation of spindle n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDIR[n]</td>
<td>Spindle number n =1 or =2, read-only</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP_</td>
<td>Direction of rotation of spindle n, which was last programmed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDIR[n]</td>
<td>Spindle number n =1 or =2, read-only</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP_TOOLNO</td>
<td>Number of the active tool T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>read-only</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP_TOOL</td>
<td>Active D number of the active tool</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>read-only</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_{TOOLNO}$</td>
<td>Number of the active tool T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>read-only</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_{TOOL}$</td>
<td>Active D number of the active tool</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>read-only</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TC_MOP$</td>
<td>Tool life warning limit (not with 802D bl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1[t,d]</td>
<td>0.0 ... in minutes, writing or reading values for tool t, D number d</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TC_MOP$</td>
<td>Residual life warning limit (not with 802D bl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2[t,d]</td>
<td>0.0 ... in minutes, writing or reading values for tool t, D number d</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TC_MOP$</td>
<td>Count warning limit (not with 802D bl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3[t,d]</td>
<td>0 ... 999 999 999, integer writing or reading values for tool t, D number d</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TC_MOP$</td>
<td>Residual count (not with 802D bl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4[t,d]</td>
<td>0 ... 999 999 999, integer writing or reading values for tool t, D number d</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TC_MOP$</td>
<td>Required tool life (not with 802D bl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11[t,d]</td>
<td>0.0 ... in minutes, writing or reading values for tool t, D number d</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TC_MOP$</td>
<td>Required count (not with 802D bl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13[t,d]</td>
<td>0 ... 999 999 999, integer writing or reading values for tool t, D number d</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TC_TP8$</td>
<td>Tool status (not with 802D bl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1[t]</td>
<td>default status – coding by bits for tool t, (bit 0 to bit 4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$TC_TP9$</td>
<td>Type of tool monitoring (not with 802D bl)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1[t]</td>
<td>Monitoring type for tool t, writing or reading</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSG()</td>
<td>Message</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>max. 65 characters</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Message text in inverted commas</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OFFN</td>
<td>Groove width with TRACYL, otherwise specification of stock allowance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Only effective with the tool radius compensation G41, G42 active</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RND</td>
<td>Rounding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>inserts a rounding with the specified radius value tangentially between two</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>contour blocks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPL</td>
<td>Angle of rotation with ROT, AROT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Specification in degrees; angle for a programmable rotation in the current</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>plane G17 to G19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SET(, , , , )</td>
<td>Set values for the variable fields</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REP()</td>
<td>SET: Various values, from the specified element up to: according to the</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>number of values</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>REP: the same value, from the specified element up to the end of the field</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

N10 IF $$P_{TOOLNO}$$==12 GOTOF ....

N10 IF $$P_{TOOL}$$==1 GOTOF ....

N10 IF $$TC_MOP1[13,1]<15.8$$ GOTOF ....

N10 IF $$TC_MOP2[13,1]<15.8$$ GOTOF ....

N10 IF $$TC_MOP3[13,1]<15$$ GOTOF ....

N10 IF $$TC_MOP4[13,1]<8$$ GOTOF ....

N10 $$TC_MOP11[13,1]=247.5$$

N10 $$TC_MOP13[13,1]=715$$

N10 IF $$TC_TP8[1]==1$$ GOTOF ....

N10 $$TC_TP9[1]==2$$ ; Select count monitoring

MSG("MESSAGE TEXT"); separate block
N150 MSG() ; cancels the previous message

N10 OFFN=12.4

N10 X... Z.... RND=...

N10 R10=SET(1.1,2.3,4.4) ; R10=1.1, R11=2.3, R4=4.4
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETMS(n)</td>
<td>Define spindle as master spindle</td>
<td>n = 1 or n = 2</td>
<td>n: Number of the spindle, if only SETMS is set, the default master spindle comes into effect</td>
</tr>
<tr>
<td>SF</td>
<td>Thread starting point when using G33</td>
<td>0.001 ... 359.999</td>
<td>Specified in degrees; the thread starting point with G33 will be offset by the specified value</td>
</tr>
<tr>
<td>SPI(n)</td>
<td>Converts the spindle number n into the axis identifier</td>
<td>n = 1 or n = 2</td>
<td>Axis identifier: e.g. &quot;SP1&quot; or &quot;C&quot;</td>
</tr>
<tr>
<td>SPOS</td>
<td>Spindle position</td>
<td>0.0000 ... 359.9999</td>
<td>Specified in degrees; the spindle stops at the specified position (to achieve this, the spindle must provide the appropriate technical prerequisites: position control)</td>
</tr>
<tr>
<td>STOPRE</td>
<td>Preprocessing stop</td>
<td>–</td>
<td>Special function; the next block is only decoded if the block before STOPRE is completed.</td>
</tr>
<tr>
<td>TRACYL(d)</td>
<td>Milling of the peripheral surface (not with 802D bl)</td>
<td>d: 1.000 ... 99 999.999</td>
<td>Kinematic transformation; only available if the relevant option exists; to be configured</td>
</tr>
<tr>
<td>TRANSMIT</td>
<td>Milling of the end face (not with 802D bl)</td>
<td>–</td>
<td>Kinematic transformation; only available if the relevant option exists; to be configured</td>
</tr>
<tr>
<td>TRAFOOF</td>
<td>Deactivate TRANSMIT, TRACYL (not with 802D bl)</td>
<td>–</td>
<td>Disables all kinematic transformations</td>
</tr>
</tbody>
</table>
8.2 Positional data

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

Functionality

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

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

Programming

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

Fig. 8-3 Different dimensioning types in the drawing

Absolute dimensioning G90

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

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

Incremental dimensioning G91

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

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

Specification with \texttt{=AC(...), =IC(...)}

After the end point coordinate, write an equality sign. The value must be specified in parentheses (round brackets).

Absolute dimensions are also possible for circle center points using \texttt{=AC(...)}. Otherwise, the reference point for the circle center is the circle starting point.

Programming example

\begin{verbatim}
N10 G90 X20 Z90 ;Absolute dimensioning
N20 X75 Z=IC(−32) ;X dimensioning still absolute, Z incremental dimensioning
... 
N180 G91 X40 Z20 ;Switching to incremental dimensioning
N190 X−12 Z=AC(17) ;X still incremental dimensioning, Z absolute
\end{verbatim}

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

Functionality

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

Programming

\begin{verbatim}
G70 ;Inch dimensional notation
G71 ;Metric dimensional notation
G700 ;Inch dimensional notation, also for feed F
G710 ;Metric dimensional notation, also for feed F
\end{verbatim}

Programming example

\begin{verbatim}
N10 G70 X10 Z30 ;Inch dimensional notation
N20 X40 Z50 ;G70 remains active
...
N80 G71 X19 Z17.3 ;Metric dimensional notation from here
...
\end{verbatim}

Information

Depending on the \textbf{default setting} you choose, the control interprets all geometric values as either metric or inch dimensions. Tool offsets and settable zero offsets including their displays are also to be understood as geometrical values; this also applies to the feedrate F in mm/min or inch/min.

The default setting can be changed via machine data.

All examples listed in this Manual are based on a \textbf{metric default setting}. 

---

**SINUMERIK 802D, 802D bl Operation and Programming Turning (BP−D), 08/05 Edition**

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

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

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

### 8.2.3 Radius / diameter dimensional notation: DIAMOF, DIAMON

#### Functionality

For the machining of parts on turning machines, it is typical to program the positional data for the **X axis** (transverse axis) as diameter dimensioning. When necessary, it is possible to switch to radius dimensioning in the program.

DIAMOF or DIAMON interprets the end point specification for the X axis as a radius or diameter specification. Correspondingly, the actual value is displayed with the workpiece coordinate system.

#### Programming

```
DIAMOF ;Radius dimensioning
DIAMON ;Diameter dimensioning
```

![Diagram](image-url)

**Fig. 8-4 Diameter and radius dimensioning for the transverse axis**
Programming example

N10 DIAMON X44 Z30 ;For X axis diameter
N20 X48 Z25 ;DIAMON remains active
N30 Z10
...
N110 DIAMOF X22 Z30 ;Switching to radius dimensioning for the X axis from here
N120 X24 Z25
N130 Z10
...

Note

A programmable offset with TRANS X... or ATRANS X..., is always evaluated as radius dimensioning. Description of this function: see the next section.

8.2.4 Programmable work offset: TRANS, ATRANS

Functionality

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

Note:

In the X axis, the workpiece zero should lie in the turning center due to the functions ‘diameter programming’ with DIAMON and ‘constant cutting rate’ with G96. Therefore: Use no offset or only a small offset (e.g. as allowance) in the X axis.

Fig. 8-5 Effect of the programmable offset
8.2 Positional data

Programming

TRANS Z... ; Programmable offset, clears old instructions for offset, rotation, scale factor, mirroring
ATRANS Z... ; Programmable offset, cumulative with existing instructions
TRANS ; without values: clears old instructions for offset, rotation, scale factor, mirroring

The instruction with TRANS/ATRANS always requires a separate block.

Programming example

N10 ...
N20 TRANS Z5 ; Programmable offset, 5 mm in the Z axis
N30 L10 ; Subroutine call, contains the geometry to be offset
...
N70 TRANS ; Offset cleared

Subroutine call – see Section 8.11 "Subroutine technique "

8.2.5 Programmable scaling factor: SCALE, ASCALE

Functionality

With SCALE, ASCALE, a scaling factor used to zoom in or zoom out the relevant axis can be used for all axes.

The currently set coordinate system is used as the reference for the scale change.

Programming

SCALE X... Z... ; Programmable scaling factor; clears old instructions for offset, rotation, scaling factor, mirroring
ASCALE X... Z... ; Programmable scaling factor; additively to existing instructions
SCALE ; No values: clears old instructions for offset, rotation, scaling factor, mirroring

The instruction with SCALE, ASCALE always requires a separate block.
Notes

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

![Diagram of programmable scaling factor]

Fig. 8-6 Example of a programmable scaling factor

Programming example

```
N20 L10 ; Programmable contour original
N30 SCALE X2 Z2 ; Contour enlarged twice in X and Z
N40 L10
...
```

Subroutine call – see Section 8.11 "Subroutine technique"

Information

In addition to the programmable offset and the scale factor, the following functions exist: programmable rotation ROT, AROT and programmable mirroring MIRROR, AMIRROR.

These functions are primarily used in milling. On turning machines, this is possible with TRANSMIT or TRACYL (see Section 8.14 "Milling on turning machines").

Examples of rotation and mirroring: see Section 8.1.6 "Overview of statements"
For detailed information, see:

References: "Operation and Programming – Milling" SINUMERIK 802D
8.2.6 Workpiece clamping – settable work offset: G54 to G59, G500, G53, G153

Functionality

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

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

Programming

G54 ;1st settable work offset
G55 ;2nd settable work offset
G56 ;3rd settable work offset
G57 ;4th settable work offset
G58 ;5th settable work offset
G59 ;6th settable work offset
G500 ;Settable work offset OFF – modal
G53 ;Settable work offset OFF – non-modal;
      also suppresses the programmable offset
G153 ;As with G53, but additionally suppresses base frame

Fig. 8-7 Settable work offset

Programming example

N10 G54 ... ;Call of 1st settable work offset
N20 X... Z... ;Workpiece machining
...
N90 G500 G0 X... ;Settable work offset OFF
8.2.7 Programmable working area limitation:  
G25, G26, WALIMON, WALIMOF

Functionality

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

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

Programming

G25 X... Z... ; Lower working area limitation
G26 X... Z... ; Upper working area limitation
WALIMON ; Working area limitation ON
WALIMOF ; Working area limitation OFF

Notes

- For G25, G26, the channel axis identifier consisting of machine data 20080: AXCONF_CHANAX_NAME_TAB is to be used. With SW 2.0 and higher, kinematic transformations are possible for the SINUMERIK 802D. In this case, different axis identifiers are configured for MD 20080 and for the geometry axis identifiers MD 20060: AXCONF_GEOAX_NAME_TAB.
- G25, G26 is also used in connection with the address S for the spindle speed limitation (see also Section "Spindle speed limitation").
A working area limitation can only be activated if the reference point for the relevant axes has been approached.

Programming example

N10 G25 X0 Z40 ; Values for the lower working area limitation
N20 G26 X80 Z160 ; Values for the upper working area limitation
N30 T1
N40 G0 X70 Z150
N50 WALIMON ; Working area limitation ON
... ; Only within the working area
N90 WALIMOF ; Working area limitation OFF
8.3 Axis movements

8.3.1 Linear interpolation with rapid traverse: G0

Functionality

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

For each axis, the maximum speed (rapid traverse) is defined in machine data. If only one axis traverses, it uses its rapid traverse. If two axes are traversed simultaneously, the path velocity (resulting velocity) is selected to achieve the greatest possible path velocity in consideration of both axes.

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, ...).

![Fig. 8-9 Linear interpolation with rapid traverse from point P1 to P](image)

Programming example

```
N10 G0 X100 Z65
```

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

Information

Another group of G functions exists for movement to the position (see Section 8.3.13 "Exact stop / continuous–path control mode: G60, G64"). For G60 exact stop, a window with various precision values can be selected with another G group. For exact stop, an alternative instruction with non-modal effectiveness exists: G9.

You should consider these options for adaptation to your positioning tasks.
8.3.2  Linear interpolation with feedrate: G1

Functionality

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

![Fig. 8-10 Linear interpolation with G1](image)

Programming example

N05 G54 G0 G90 X40 Z200 S500 M3 ; Tool moves with rapid traverse rate, Spindle speed = 500 rpm, clockwise rotation
N10 G1 Z120 F0.15 ; Linear interpolation with feed 0.15 mm/revolution
N15 X45 Z105
N20 Z80
N25 G0 X100 ; Retraction with rapid traverse rate
N30 M2 ; End of program

Note: Another option for linear programming is available with the angle specification ANG= (see Section "Blueprint programming").
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**: Clockwise
- **G3**: Counterclockwise

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

- **G2/G3 and center point parameter (+end point)**: E.g. G2 X... Z... I... K...
- **G2/G3 and radius parameter (+end point)**: E.g. G2 X... Z... CR= ...
- **G2/G3 and aperture angle parameter (+center point)**: E.g. G2 AR=... I... K...
- **G2/G3 and aperture angle parameter (+end point)**: E.g. G2 AR=... X... Z...

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

**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.

Programming example for center point and end point specification:

![Diagram](image)

Fig. 8-13 Example for center point and end point specification

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

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

Programming example for end point and radius specification:

![Diagram](image)

Fig. 8-14 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 semicircle is selected.

**Programming example for end point and aperture angle:**

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

**Programming example for center point and aperture angle:**

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

Note: Center point values refer to the circle starting point!
8.3.4 Circular interpolation via intermediate point: CIP

Functionality

The direction of the circle results here from the position of the intermediate point (between starting and end points). 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.

![Diagram of a circle with intermediate point](image)

Fig. 8-17 Circle with end point and intermediate point specification using the example of G90

Programming example

N5 G90 Z30 X40 ; Circle starting point for N10
N10 CIP Z50 X40 K1=40 I1=45 ; End point and intermediate point

8.3.5 Circle with tangential transition: CT

With CT and the programmed end point in the current plane (G18: Z/X plane), a circle is produced which tangentially connects to the previous path segment (circle or straight line). 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.

![Diagram of a circle with tangential transition](image)

Fig. 8-18 Circle with tangential transition to the previous path section

Programming:

N10 G1 Z20 F3 ; Straight line
N20 CT X... Z... ; Circle with tangential connection
8.3.6  Thread cutting with constant lead: G33

Functionality

The function G33 can be used to machine threads with constant lead of the following type:

- Threads on cylindrical structures
- Threads on tapered structures
- Outside / inside thread
- Single- and multiple-start thread
- Multi-block thread (series of threads)

This requires a spindle with position measuring system.

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

![External / internal thread with cylindrical thread as an example](image)

**Right-hand or left-hand threads**

Right-hand or left-hand threads are set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) – see Section 8.4 "Spindle movement"). To do this, the rotation value must be programmed under address S or a rotation speed set.
Remark: Run-in and run-out paths must be taken into account for the thread lengths.

![Diagram of axis movements](Image)

**Fig. 8-20** Programmable values for threads with G33

<table>
<thead>
<tr>
<th>Programming:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cylinder thread</strong></td>
</tr>
<tr>
<td>G33  Z... K...</td>
</tr>
<tr>
<td><strong>Tapered thread</strong></td>
</tr>
<tr>
<td>G33  Z... X... K...</td>
</tr>
<tr>
<td>(Lead K because the larger distance is in the Z axis)</td>
</tr>
<tr>
<td>G33  Z... X... I...</td>
</tr>
<tr>
<td>(Lead I because the larger distance is in the X axis)</td>
</tr>
<tr>
<td><strong>Face thread</strong></td>
</tr>
<tr>
<td>G33  X... I...</td>
</tr>
</tbody>
</table>

**Fig. 8-21** Pitch assignment using the example of the Z / X axis

**Tapered thread**

For tapered threads (2 axis values required), the required lead address I or K of the axis with the greater distance (greater thread length) must be used. A second lead is not defined.
Starting point offset SF=

A starting point offset is required for the spindle if multiple threads or threads in offset sections are to be machined. The starting point offset is programmed in the thread block with G33 under the address SF (absolute position).

If no starting point offset is programmed, the value from the setting data will be active.

**Please note:** A programmed value for SF= is always also to be entered in the setting data.

### Programming example

**Cylindrical thread**, double-thread, starting point offset 180 degrees, thread length (including run-in and run-out) 100 mm, thread lead 4 mm/rev.

Right-hand thread, cylinder already premanufactured:

N10 G54 G0 G90  X50 Z0 S500 M3 ;Approach starting point, spindle direction clockwise

N20 G33 Z−100 K4 SF=0 ;Lead: 4 mm/rev.

N30 G0 X54

N40 Z0

N50 X50

N60 G33 Z−100 K4 SF=180 ;2nd thread, offset 180 degrees

N70 G0 X54 ...

### Multi-block thread

If multiple thread blocks are programmed consecutively (multi-block thread), it only makes sense to define a starting point offset in the 1st thread block. The value is only used here.

Multi-block threads are automatically connected by G64 continuous-path control mode (see Section 8.3.13 "Exact stop / continuous-path control mode: G60, G64").

![Multi-turn thread (thread chaining)](image)

### Axis velocity

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

Information

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

8.3.7 Thread cutting with variable lead: G34, G35

Functionality

G34, G35 can be used to manufacture threads with variable lead in one block:
- G34 ; thread with increasing lead
- G35 ; thread with decreasing lead
Both functions otherwise have the same functionality as G33 and have the same prerequisites.

G34 or G35 are effective until they are canceled by another statement of this G group (G0, G1, G2,G3, G33, ...).

Pitch:
- I or K ; Starting thread lead in mm/rev., associated with X or Z axis

Lead change:
In the block with G34 or G35, the address F contains the meaning of the lead change:
The lead (mm per revolution) changes per revolution.
- F ; Pitch change in mm/rev. ².

Note: Outside of G34, G35, the address F also indicates the feed or the dwell time for G4. The values programmed there remain saved.

Determining F

If you already know the initial and final pitch of a thread, you can calculate the thread pitch change F to be programmed according to the following equation:

\[ F = \frac{|K_a^2 - K_e^2|}{2 \times L_G} \text{ [mm/rev. ²]} \]

The parameters have the following meaning:
- K_e Thread lead of the axis target coordinate [mm/rev.]
- K_a Thread start lead (programmed under I, K) [mm/rev.]
- L_G thread length in [mm]

Programming

G34 Z... K... F... ; Cylindrical thread with increasing pitch
G35 X... I... F... ; Transversal thread with degressive pitch
G35 Z... X... K... F... ; Tapered thread with degressive pitch
Programming example

; Cylindrical thread; followed by degressive pitch
N10 M3 S40 ; Activate spindle
N20 G0 G54 G90 G64 Z10 X60 ; Approach starting point
N30 G33 Z-100 K5 SF=15 ; Thread; constant lead 5mm/rev.,
    ; Starting point at 15 degrees
N40 G35 Z-150 K5 F0.16 ; Initial thread 5 mm/rev.,
    ; Thread reduction 0.16 mm/rev. 2,
    ; Thread length 50 mm,
    ; Desired thread at the end of the block 3 mm/rev.
N50 G0 X80
N60 Z120
N100 M2

8.3.8 Thread Interpolation: G331, G332

Functionality

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

Drilling is done using G331, retraction is done using G332.
The drilling depth is specified via the axis, e.g. Z, and the thread lead via the appropriate interpolation parameter (here: K).
For G332, the same lead is programmed as for G331. Reversal of the spindle's direction of rotation occurs automatically.
The spindle speed is programmed with S, and without M3/M4.
Before tapping the thread with G331/G332, the spindle must be switched to the position controlled mode with SPOS=... (see also Section 8.4.3 "Spindle positioning").

Right–hand or left–hand thread

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

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

Axis velocity

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

**Programming example**

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

N5 G54 G0 G90 X10 Z5 ;Approach starting point
N10 SPOS=0 ;Spindle in position control mode
N20 G331 Z−25 K0.8 S600 ;Thread tapping, K positive = clockwise rotation of the spindle, end point −25 mm
N40 G332 Z5 K0.8 ;Retraction
N50 G0 X...

**8.3.9 Fixed point approach: G75**

**Functionality**

By using G75, a fixed point on the machine, e.g. tool change point, can be approached. The position is stored permanently in the machine data for all axes. No offset is effective. The speed of each axis is its rapid traverse.

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

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

**Programming example**

N10 G75 X1=0 Z1=0

Remark: The programmed position values for X1, Z1 (here = 0) are ignored, but must still be written.

**8.3.10 Reference point approach: G74**

**Functionality**

The reference point can be approached in the NC program with G74. The direction and speed of each axis are stored in machine data.

G74 requires a separate block and is non-modal. The machine axis identifier must be programmed!

In the block after G74, the previous G command of the group "interpolation type" (G0, G1, G2, ...) is active again.
Programming example

N10 G74 X1=0 Z1=0

Remark: The programmed position values for X1, Z1 (here = 0) are ignored, but must still be written.

8.3.11 Measuring with touch–trigger probe: MEAS, MEAW

Functionality

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

Programming

MEAS=1 G1 X... Z... F... ;Measuring with the rising edge of the probe; deletion of the distance to go
MEAS=−1 G1 X... Z... F... ;Measuring with the falling edge of the probe; deletion of the distance to go
MEAW=1 G1 X... Z... F... ;Measuring with the rising edge of the probe; without deletion of the distance to go
MEAW=−1 G1 X... Z... F... ;Measuring with the falling edge of the probe; without deletion of the distance to go

Caution

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

Measurement task status

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

Measurement result

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

Axis stands for X or Z.
## 8.3 Axis movements

### Programming example

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

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

### 8.3.12 Feedrate F

#### Functionality

The feed F is the **path velocity** and represents the value of the geometrical sum of the velocity components of all involved axes.

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

Remark:

This unit of measure applies to metric dimensions. According to Section 8.2.2 "Metric and inch dimensioning", settings with inch dimensioning are also possible.

#### Programming example

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

Remark: Write a new F word if you change G94 – G95!
Information

The G group with G94, G95 also contains the functions G96, G97 for the constant cutting rate. These functions also have an influence on the S word (see Section 8.5.1 "Constant cutting rate").

8.3.13 Exact stop / continuous-path control mode: G9, G60, G64

Functionality

To set the movement behavior at the block limits and to continue with the next block, G functions are provided for optimum adaptation to different requirements. For example, you would like to quickly position with the axes or you would like to machine path contours over multiple blocks.

Programming

G60 ;Exact stop – modally effective
G64 ;Continuous-path-control mode
G9 ;Exact stop – non-modal
G601 ;Exact stop fine window
G602 ;Exact stop coarse window

Exact stop G60, G9

If the exact stop function (G60 or G9) is active, the velocity for reaching the exact end position at the end of a block is decelerated to zero.

Another modal G group can be used here to set when the traversing movement of this block is considered ended and the next block is started.

- G601 Exact stop window fine
  Block advance takes place when all axes have reached the "Exact stop window fine" (value in the machine data).
- G602 Exact stop window coarse
  Block advance takes place when all axes have reached the "Exact stop window coarse" (value in the machine data).

The selection of the exact stop window has a significant influence on the total time if many positioning operations are executed. Fine adjustments require more time.
8.3 Axis movements

Programming example

N5 G602 ;Exact stop window coarse
N10 G0 G60 Z... ;Exact stop modal
N20 X... Z... ;G60 is still effective
...
N50 G1 G601 ... ;Exact stop window fine
N80 G64 Z... ;Switching to continuous–path control mode
...
N100 G0 G9 Z... ;Exact stop is only effective for this block
N111 ... ;Continuous–path control mode again

Remark: The command G9 only creates exact stop for the block in which it is programmed; G60, however, is active until it is canceled by G64.

Continuous–path control mode G64

The objective of the continuous–path control mode is to avoid deceleration at the block boundaries and to switch to the next block with the most constant path velocity possible (during tangential transitions). The function works with look–ahead velocity control via several blocks.

For non–tangential transitions (corners), the velocity can reduced rapidly enough so that the axes are subject to a relatively high velocity change over a short time. This may lead to a significant jerk (acceleration change). The magnitude of the jerk can be limited by activating the SOFT function.
Programming example

N10 G64 G1 Z... F... ;Continuous–path control mode
N20 X.. ;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 automatically determines the velocity control for multiple NC blocks in advance. This enables acceleration and deceleration across multiple blocks with almost tangential transitions. For paths that consist of short sections in the NC blocks, higher velocities can be achieved than without look ahead.

![Feedrate Graph](image)

Fig. 8-24 Comparison of the G60 and G64 velocity behavior with short paths in the blocks

8.3.14 Acceleration pattern: BRISK, SOFT

**BRISK**

The axes of the machine change their speeds using the maximum allowable acceleration value until reaching the final speed. BRISK allows time–optimized working. The set velocity is reached in a short time. However, jumps are present in the acceleration pattern.
SOFT

The axes of the machine accelerate with nonlinear, constant curves 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.

![Diagram showing velocity and time for BRISK and SOFT]

Fig. 8-25 Principle course of the path speed when using BRISK/SOFT

Programming

BRISK ; Stepped path acceleration
SOFT ; Path acceleration with jerk limitation

Programming example

N10 SOFT G1 X30 Z84 F6.5 ; Path acceleration with jerk limitation
...      
N90 BRISK X87 Z104 ; Continue with stepped path acceleration
...

8.3.15 Percentage acceleration override: ACC

Functionality

In certain program sections, it can be necessary to change the axis or spindle acceleration set via machine data via the program. This programmable acceleration is a percentage acceleration correction.

For each axis (e.g., X) or spindle (S), a percentage value 0% and v 200% can be programmed. The axis interpolation is then carried out with this proportional acceleration. The reference value (100%) is the valid machine data value for the acceleration (depending on whether it is the axis or spindle; for the spindle it depends further on the gear step and whether it is positioning mode or speed mode).

Programming

ACC[axis name] = percentage value ; for the axis
ACC[S] = percentage value ; for the spindle
Programming example

N10 ACC[X]=80 ; 80% acceleration for the X axis
N20 ACC[S]=50 ; 50% acceleration for the spindle
...
N100 ACC[X]=100 ; Disable X axis override

Activation

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

The override is disabled by assigning the value ACC[...] = 100; this can also be achieved using RESET and end of program.

The programmed override value is also active with dry run feedrate.

Caution

A value greater than 100% may only be programmed if this load is permissible for the machine mechanics and the drives have the corresponding reserves. Failure to adhere to the limits can lead to damage to the mechanical parts and/or error messages.

8.3.16 Traversing with feedforward control: FFWON, FFWOF

Functionality

The feedforward control reduces the following error approximately to zero.
Traversing with feedforward control permits higher path accuracy and thus improved machining results.

Programming

FFWON ; Feedforward control ON
FFWOFS ; Feedforward control OFF

Programming example

N10 FFWON ; Feedforward control ON
N20 G1 X... Z... F9
...
N80 FFWOF ; Feedforward control OFF
8.3 Axis movements

8.3.17 3rd and 4th axes

Functionality

**Prerequisite:** Expanded control system configuration for 4 axes

Depending on the machine design, a 3rd and even a 4th axis can be required. These axes can be implemented as linear or rotary axes. Accordingly, the identifier for these axes can be configured, e.g.: U, C, or A etc. For rotary axes, the traversing range can be configured between 0 ... < 360 degrees (modulo–behavior).

A 3rd or 4th axis can be linear traversed with the remaining axes at the same time with a corresponding machine design. If the axis is traversed in a block with G1 or G2/G3 with the remaining axes (X, Z), it is not assigned a component of the feedrate F; its velocity will depend on the path time of the X and Z axes. Its movement starts and ends with the remaining path axes. The velocity, however, must not be greater than the defined limiting value.

If programmed solely in the block, with G1, the axis will traverse with the active feedrate F. If it is a rotary axis, the unit of measurement for F is correspondingly degrees/min (with G94) or degrees/revolution of the spindle (with G95).

For this axis, it is also possible to specify (G54 ... G57) and program (TRANS, ATRANS) offsets.

**Programming example**

Assumed the 4th axis is a rotary axis and has the axis identifier A:

```
N5 G94 ; F in mm/min or degrees/min
N10 G0 X10  Z30 A45 ; Traverse X–Z path with rapid traverse, A simultaneously
N20 G1 X12  Z33 A60 F400 ; X–Z path at 400 mm/min, A – simultaneously
N30 G1 A90 F3000 ; A axis traverses alone to 90 deg. position
                   at a velocity of 3,000 degrees/min
```

**Special instructions for rotary axes: DC, ACP, ACN**

- e.g. for rotary axis A:
  - A=DC(...) ; Absolute dimensions, approach position directly (along shortest path)
  - A=ACP(...) Absolute dimensions, approach position in positive direction
  - A=ACN(...) Absolute dimensions, approach position in negative direction

**Example:**

```
N10 A=ACP(55.7) ; Absolute position 55.7 degrees; approach position in positive direction
```

8.3.18 Dwell Time: G4

Functionality

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

The words with F... or S... are only used for this block for the specified time. Any previously programmed feedrate F and a spindle speed S remain valid.
**Programming**

G4 F... ;Dwell time in seconds  
G4 S... ;Dwell time in spindle rev.'s

**Programming example**

```
N5 G1 F3.8 Z-50 S300 M3 ;Feedrate F, spindle speed S  
N10 G4 F2.5 ;Dwell time 2.5 s  
N20 Z70  
N30 G4 S30 ;Dwell for 30 spindle revolutions, with S  
=300 rpm and a speed override of 100 %, this corresponds  
to: t=0.1 min  
N40 X... ;Feedrate and spindle speed continue to be effective
```

**Remark**

G4 S.. is only possible if a controlled spindle is available (if the speed preset is also pro-
grammed via S...).

8.3.19 **Travel to fixed stop**

**Functionality**

This function is an option and available as of SW 2.0.  
The travel to fixed stop (FXS = Fixed Stop) function can be used to establish defined forces  
for clamping workpieces, such as those required for sleeves and grippers. This function can  
also be used for approaching mechanical reference points. With sufficiently reduced torque,  
it is also possible to perform simple measurement operations without connecting a probe.

**Programming**

```
FXS[axis]=1 ; Select "Travel to fixed stop"  
FXS[axis]=0 ; Deselect "Travel to fixed stop"  
FXST[axis]=... ; Clamping torque, specified in % of the max. torque of the drive  
FXSW[axis]=... ; Width of the window for fixed-stop monitoring in mm/degrees
```

**Remark:** The **machine axis identifier**, e.g: X1, should be used as the axis identifier. X1.  
The channel axis identifier (e.g.: X) is only permitted, if e.g. no coordinate rotation is active  
and this axis is directly assigned to a machine axis.  

The commands are modal. The traversing path and the selection of the function FXS[axis]=1  
must be programmed **in a separate block**.
8.3 Axis movements

Programming example – selection

N10 G1 G94 ...  
N100 X250 Z100 F100 FXS[Z1]=1 FXST[Z1]=12.3 FXSW[Z1]=2  
; For the Z1 machine axis  
; FXS function selected,  
; Clamping torque 12.3%,  
; Window width 2 mm

Notes

- When selected, the fixed stop must be located between the start and end positions.
- Torque (FXST[ ]= ) and window width (FXSW[ ]= ) may be specified optionally. If they are not written, the value of the existing setting data are used. Programmed values are imported into the setting data. At the start, the setting data are loaded with values from machine data. FXST[ ]=... or FXSW[ ]=... can be changed at any time in the program. The changes are active before traversing movements in the block.

Further programming examples

N10 G1 G94 ...  
N20 X250 Z100 F100 FXS[X1]=1  
; FXS selected for the X1 machine axis;  
; clamping torque and window width from the SDs

N20 X250 Z100 F100 FXS[X1]=1 FXST[X1]=12.3  
; FXS selected for the X1 machine axis,  
; clamping torque 12.3 %, window width from SD

N20 X250 Z100 F100 FXS[X1]=1 FXST[X1]=12.3 FXSW[X1]=2  
; FXS selected for the X1 machine axis;  
; clamping torque 12.3%,  
; window width 2 mm

N20 X250 Z100 F100 FXS[X1]=1 FXSW[X1]=2  
; FXS selected for the X1 machine axis;  
; clamping torque from the SD, window width 2 mm
8.3 Axis movements

Fixed stop reached

After the fixed stop has been reached:

- The distance to go is deleted and the position setpoint is followed up.
- The drive torque increases to the programmed limit value $\text{FXST}[] = \ldots$ or the value from SD and then remains constant.
- The monitoring of the fixed stop is active within the specified window width ($\text{FXSW}[] = \ldots$ or value from SD).

Deselecting the function

Deselecting the function results in a preprocessing stop. In the block with $\text{FXS}[X1] = 0$, traversing movements should stop.

Example:
N200 G1 G94 X200 Y400 F200 FXS[X1] = 0 ; The X1 axis is retracted from fixed stop to the X= 200 mm position.

Important

The traversing motion to the retraction position must lead away from the fixed stop; otherwise, the fixed stop or the machine may be damaged.

The block change takes place when the retraction position has been reached. If no retraction position is specified, the block change takes place immediately after the torque limit has been deactivated.

Further information

- "Measure and delete distance-to-go" (MEAS command) and "Travel to fixed stop" cannot be programmed in the same block.
- Contour monitoring is not performed while "Travel to fixed stop" is active.
- If the torque limit is reduced too far, the axis will not be able to follow the specified setpoint; the position controller then goes to the limit and the contour deviation increases. In this operating state, an increase in the torque limit may result in sudden, jerky movements. To make sure that the axis may still follow, make sure that the contour deviation is not greater than with unlimited torque.
- A rate of rise ramp for the new torque limit can be defined in MD to prevent any abrupt changes to the torque limit setting (e.g. insertion of a spindle sleeve or quill).
8.3 Axis movements

System variable for status: $AA_FXS[axis]

This system variable provides the "Travel to fixed stop" status for the axis specified:

Value =

0: Axis not at stop
1: The stop was approached successfully
   (axis is in fixed-stop monitoring window)
2: Approach to fixed stop has failed (axis is not at fixed stop)
3: Travel to fixed stop activated
4: Fixed stop was detected
5: Travel to fixed stop is deselected. The deselection is not yet completed.

The interrogation of the system variable in the part program triggers a preprocessing stop.

With the SINUMERIK 802D, only the static states before selection/deselection may be acquired.

Alarm suppression

The issuing of the following alarms can be suppressed with machine data:

- 20091 "Fixed stop not reached"
- 20094 "Fixed stop aborted"

References: "Description of Functions", Section "Travel to fixed stop"
8.4 Spindle movements

8.4.1 Spindle speed S, directions of rotation

Functionality

The spindle speed is programmed under the address S in RPM, if the machine has a controlled spindle.
The direction of rotation and the start or end of the movement are specified via M commands (also see Chapter 8.7 "Miscellaneous function M").

M3 Spindle CW
M4 Spindle CCW
M5 Spindle STOP

Remark: For integer S values, the decimal point can be omitted, e.g. S270

Information

If you write M3 or M4 in a block with axis movements, the M commands become active before the axis movements.

Default setting: The axis movements will only start once the spindle has accelerated to speed (M3, M4). M5 is also issued before the axis movement. However, it does not wait for the spindle to stop. The axis movements begin before the spindle stops.
The spindle is stopped using program end or RESET.
At program start, spindle speed zero (S0) is in effect.
Remark: Other settings can be configured via machine data.

Programming example

N10 G1 X70 Z20 F3 S270 M3 ; The spindle starts rotating CW to 270 rpm before traversing the X and Z axes
; ...
N80 S450 ... ; Speed change
...
N170 G0 Z180 M5 ; Z motion the block, spindle stops

8.4.2 Spindle speed limitation: G25, G26

Functionality

In the program, you can limit the limit values that would otherwise apply by writing G25 or G26 and the spindle address S with the speed limit value. This overwrites the values entered in the setting data at the same time.
G25 or G26 each requires a separate block. A previously programmed speed S is maintained.
8.4 Spindle movements

Programming

G25 S...
G26 S...

; Lower spindle speed limitation
; Upper spindle speed limitation

Information

The outermost limits of the spindle speed are set in machine data. By making inputs via the operator panel, setting data can be active for further limiting.

When working with the function G96 "Constant cutting rate", an additional upper limit can be programmed / entered.

Programming example

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

8.4.3 Spindle positioning: SPOS

Functionality

Prerequisite: The spindle must be technically designed for position control.

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

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

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

Exception: The spindle movement is completed first, when the measurement system is not yet synchronized. In this case, the direction is specified in machine data.

Other movement specifications for the spindle are possible with SPOS=ACP(...), SPOS=ACN(...), ... as for rotary axes (see Section "3rd and 4th axes").

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

Programming

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

```
N10  SPOS=14.3 ;Spindle position 14.3 degrees
...
N80 G0 X89  Z300 SPOS=25.6 ;Positioning of the spindle with axis motions This block is
ended when both movements are finished.
N81 X200 Z300 ;The N81 block will only start if the spindle position from
;N80 is reached.
```

8.4.4 Gear stages

**Function**

Up to 5 gear stages can be configured for a spindle for speed / torque adaptation. The selection of a gear stage takes place in the program via M commands (see Section 8.7 "Miscellaneous function M"):

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

8.4.5 2nd spindle

With the SINUMERIK 802D, with SW 2.0 and higher, a 2nd spindle is available. This is not applicable to the 802D bl.

**Function**

With SW 2.0 and higher, the kinematic transformation functions TRANSMIT and TRACYL are possible for the milling machining on turning machines. These functions require a 2nd spindle for the driven milling tool. When using these functions, the main spindle is operated as a rotary axis (see Section 8.14).

**Master spindle**

The master spindle results in various functions which are only possible with this spindle:

- G95 ; Rev. feedrate
- G96, G97 ; Constant cutting rate
- LIMS ; Upper speed limit with G96, G97
- G33, G34, G35, G331, G332 ; Thread cutting, thread interpolation
- M3, M4, M5, S... ; Simple specifications for the direction of rotation, stop and speed

The master spindle is defined via configuration (machine data). As a rule, the main spindle (spindle 1) is the master spindle. A different spindle can be defined as master spindle in the program:

- SETMS(n) ; Spindle n (= 1 or 2) is master spindle as of now.
Switching back can also be performed via:

- SETMS ; Configured master spindle is now master spindle again or
- SETMS(1) ; Spindle 1 is master spindle again as of now.

The definition of the master spindle changed in the program is only valid until program end/program abort. Thereafter, the configured master spindle is again active.

### Programming via spindle number

Some spindle functions can also be selected via the spindle number:

- \( S1=..., S2=... \) ; Spindle speed for spindle 1 or 2
- \( M1=3, M1=4, M1=5 \) ; Specifications for the direction of rotation; stop for spindle 1
- \( M2=3, M2=4, M2=5 \) ; Specifications for the direction of rotation; stop for spindle 2
- \( M1=40, ..., M1=45 \) ; Gear stages for spindle 1 (where provided)
- \( M2=40, ..., M2=45 \) ; Gear stages for spindle 2 (where provided)
- \( \text{SPOS}[\,n\,] \) ; Positioning of spindle \( n \)
- \( \text{SPI}(\,\,n\,\,) \) ; Converts the spindle number \( n \) to an axis identifier, e.g. "SP1" or "CC"
  ; "n" must be a valid spindle number (1 or 2)
  ; The functions of the spindle identifiers \( \text{SPI}(n) \) and \( S_n \) are identical.
- \( P\_S[\,n\,] \) ; Last programmed speed of spindle \( n \)
- \( AA\_S[\,n\,] \) ; Actual speed of spindle \( n \)
- \( P\_SDIR[\,n\,] \) ; Last programmed direction of rotation of spindle \( n \)
- \( AC\_SDIR[\,n\,] \) ; Current direction of rotation of spindle \( n \)

### 2 spindles installed

The following can be interrogated in the program via the system variable:

- \( P\_\text{NUM}\_\text{SPINDLES} \) ; Number of configured spindles (in the channel)
- \( P\_\text{MSNUM} \) ; Number of programmed master spindle
- \( AC\_\text{MSNUM} \) ; Number of the active master spindle
8.5 Special turning functions

8.5.1 Constant cutting rate: G96, G97

Functionality

**Prerequisite:** A controlled spindle must be present.

With activated G96 function, the spindle speed is adapted to the currently machined workpiece diameter (transverse axis) such that a programmed cutting rate S remains constant on the tool edge (spindle speed multiplied with the diameter = constant).

The S word is evaluated as the cutting rate as of the block with G96. G96 is modally effective until cancellation by another G function of the group (G94, G95, G97).

Programming

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

S ;Cutting rate, unit m/min
LIMS= ;Upper limit speed of the spindle, effective with G96, G97
F ;feed in the units mm/revolution -- as for G95
```

Remark:
If G94 was previously active instead of G95, an appropriate F value must be programmed!

![Diagram of constant cutting rate G96](image)

Fig. 8-27 Constant cutting rate G96

Rapid traverse

With rapid traverse G0, there is no change in speed.

**Exception:** If the contour is approached at rapid traverse and the next block contains an interpolation type G1 or G2, G3, CIP, CT (contour block), then the speed for the contour block is applied already in the approach block with G0.
Upper speed limit LIMS=

During machining from large to small diameters, the spindle speed can increase significantly. In this case, it is recommended the upper spindle speed limitation LIMS=... . LIMS is only effective with G96 and G97.

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

The upper limit speed either programmed via G26 or defined via machine data cannot be exceeded with LIMS=.

Deactivate constant cutting rate: G97

The function "Constant cutting rate" is deactivated with G97. If G97 is effective, a programmed S word will be interpreted again as the spindle speed specified in revolutions per minute.

If no new S word is programmed, the spindle will continue rotating at the speed last determined with the G96 function active.

Programming example

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

Information

The G96 function can also be deactivated with G94 or G95 (same G group). In this case, the last programmed spindle speed S is active for the remaining machining sequence if no new S word is programmed.

The programmable offset TRANS or ATRANS (see section of that name) should not be used on the transverse axis X or used only with low values. The workpiece zero point should be located at the turning center. Only then is the exact function of G96 guaranteed.
### 8.5.2 Rounding, chamfer

**Functionality**
In a contour corner, you can insert the elements chamfer or rounding. The respective instruction `CHF=...` or `RND=...` is written in the block, which leads to the corner.

**Programming**

```plaintext
CHF=... ; insert chamfer, value: Length Side length of the chamfer
RND=... ; insert rounding, value: Radius of the rounding
```

**Chamfer CHF=**

A linear contour element is inserted between linear and circle contours in any combination. The edge is broken.

![Chamfer Diagram](image)

**Fig. 8-28** Inserting a chamfer CHF between two straight lines (example)

**Programming example for a chamfer**

```plaintext
N10 G1 Z... CHF=5 ; Insert 5 mm chamfer
N20 X... Z...
```
8.5 Special turning functions

Rounding RND=

A circle contour element can be inserted with tangential link between the linear and circle contours in any combination.

<table>
<thead>
<tr>
<th>Straight line/straight line:</th>
<th>Straight line/circle:</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10 G1 ... RND=...</td>
<td>N50 G1 ... RND=...</td>
</tr>
<tr>
<td>N20 X... Z...</td>
<td>N60 G3 X... Z...</td>
</tr>
</tbody>
</table>

![Fig. 8-29 Inserting roundings as examples](image)

Programming example for a rounding

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

Information

The programmed value for chamfer and rounding is automatically reduced if the contour length of an involved block is insufficient.
No chamfer/rounding is inserted if subsequently more than one block is programmed which does not contain any traversing information.

8.5.3 Blueprint programming

Functionality

If direct end point values for the contour are not visible in a machining drawing, angle values can also be used for straight line determination. In a contour corner, you can insert the elements chamfer or rounding. The respective instruction CHR=... or RND=... is written in the block, which leads to the corner.
The contour definition programming can be used in blocks with G0 or G1.
Theoretically, any number of straight line blocks can be linked and a rounding or a chamfer can be inserted between them. Every straight line must be clearly identified by point values and/or angle values.

Programming

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

If only one end point coordinate of the plane is known for a straight line, or for contours across multiple blocks the cumulative end point, an angle parameter can be used for unique definition of the straight line path. The angle is always referred to the Z axis (normal case: G18 active). Positive angles are aligned counterclockwise.

<table>
<thead>
<tr>
<th>Contour</th>
<th>Programming</th>
</tr>
</thead>
</table>
| ![Diagram](angle_angle.png) | End point in N20 not completely known
| ![Diagram](angle_angle.png) | N10 G1 X1 Z1
| ![Diagram](angle_angle.png) | N20 X2 ANG=...
| ![Diagram](angle_angle.png) | or:
| ![Diagram](angle_angle.png) | N10 G1 X1 Z1
| ![Diagram](angle_angle.png) | N20 Z2 ANG=...
| ![Diagram](angle_angle.png) | The values are only symbolic.

Fig. 8-30  Angle value for determination of a straight line

Rounding RND=

A contour element is inserted into the corner of two linear blocks with tangential corner (see also Fig. 8-29).

Chamfer CHR=

Another linear contour element (chamfer) is inserted into the corner of two linear blocks. The programmed value is the side length of the chamfer.

<table>
<thead>
<tr>
<th>Contour</th>
<th>Programming</th>
</tr>
</thead>
</table>
| ![Diagram](chamfer_chamfer.png) | Insert a chamfer with side length e.g. 5 mm:
| ![Diagram](chamfer_chamfer.png) | N10 G1 Z...
| ![Diagram](chamfer_chamfer.png) | CHR=5
| ![Diagram](chamfer_chamfer.png) | N20 X... Z..

Fig. 8-31  Inserting a chamfer using CHR
Information

- If radius and chamfer are programmed in one block, only the radius is inserted regardless of the programming sequence.
- In fields other than blueprint programming, you will also find specification of a chamfer in the form CHF=... In such cases, the value after CHF= is the chamfer length, instead of CHR=...

<table>
<thead>
<tr>
<th>Contour</th>
<th>Programming</th>
</tr>
</thead>
</table>
| ![Contour Diagram 1](image1.png) | End point in N20 unknown:  
N10 G1 X1 Z1  
N20 ANG=...1 RND=...1  
N30 X3 Z3 ANG=...2  

The values are only symbolic. |
| ![Contour Diagram 2](image2.png) | End point in N20 unknown:  
insert chamfer:  
N10 G1 X1 Z1  
N20 ANG=...1 CHR=...1  
N30 X3 Z3 ANG=...2  

analogously:  
Insert chamfer:  
N10 G1 X1 Z1  
N20 ANG=...1 CHR=...1  
N30 X3 Z3 ANG=...2 |
| ![Contour Diagram 3](image3.png) | End point in N20 known:  
insert rounding:  
N10 G1 X1 Z1  
N20 X2 Z2 RND=...1  
N30 X3 Z3  

analogously:  
Insert rounding:  
N10 G1 X1 Z1  
N20 X2 Z2 RND=...1  
N30 X3 Z3 |
| ![Contour Diagram 4](image4.png) | End point in N20 unknown:  
insert roundings:  
N10 G1 X1 Z1  
N20 ANG=...1 RND=...1  
N30 X3 Z3 ANG=...2 RND=...2  
N40 X4 Z4  

analogously:  
Insert chamfer:  
N10 G1 X1 Z1  
N20 ANG=...1 CHR=...1  
N30 X3 Z3 ANG=...2 CHR=...2  
N40 X4 Z4 |

Fig. 8-32  Examples of multi–block contours
8.6 Tool and tool offset

8.6.1 General notes

Functionality

During program creation for the workpiece machining, you do not have to take tool lengths or cutting radius into consideration. You can program workpiece dimensions directly, e.g. as specified in the drawing.

You enter the tool data separately in a special data section. Simply call the required tool with its offset data in the program. The control executes the required path corrections based on this data to create the described workpiece.

![Diagram of machining a workpiece with different tool dimensions](image)

8.6.2 Tool T

Functionality

The tool selection takes place when the T word is programmed. Whether this is a tool change or only a preselection, is defined in the machine data:

- A tool change (tool function) takes place directly with the T word (e.g. typical for tool turrets on turning machines) or
- the change takes place after the preselection with the T word by an additional instruction M6 (see also Section 8.7 "Miscellaneous functions M").

Please note:

If a certain tool was activated, it remains stored as an active tool even beyond the end of the program and after turning off / turning on the control system. If you change a tool manually, input the change also in the control system so that the control system 'knows' the correct tool. For example, you can start a block with the new T word in the MDA mode.
8.6 Tool and tool offset

Programming

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

Note: With the 802D, max. 32 tools, and with the 802D bl, 18 tools can be stored in the control system at a time.

Programming example

Tool change without M6:
N10 T1 ;Tool 1
...
N70 T588 ;Tool 588

8.6.3 Tool offset number D

Functionality

It is possible to assign between 1 to 9 (12) data fields with different tool offset blocks (for multiple cutting edges) to a specific tool. If a special cutting tool is required, it can be programmed with D and the corresponding number.
If no D word is written, D1 is automatically in effect.
If D0 is programmed, the offsets for the tool are ineffective.

Programming

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

Note: Max. 64 data fields (802D bl 36) with tool offset blocks can be stored in the control system at a time.

Each tool has separate offset blocks – a maximum of 9.

Fig. 8-34 Examples for assigning tool offset numbers / tool

Information

Tool length offsets are active immediately if the tool is active; if no D numbers have been programmed, the values of D1 will be used.
The offset is applied with the first programmed traverse of the respective length compensation axis.
A tool radius compensation must also be activated by G41/G42.
Programming example

Tool change:

N10 T1 ; Tool 1 is activated with the relevant D1
N11 G0 X... Z... ; The length compensation is overlaid here
N50 T4 D2 ; Load tool 4, D2 of T4 is active
...
N70 G0 Z... D1 ; D1 for tool 4 active, only edge changed

Contents of an offset memory

- Geometric quantities: Length, radius
  These consist of several components (geometry, wear). The control computes the components to a certain dimension (e.g., overall length 1, total radius). The respective overall dimension becomes effective when the offset memory is activated.
  How these values are calculated in the axes is determined by the tool type and the commands G17, G18, G19 (see following illustrations).

- Tool type
  The tool type (drill, turning tool or cutter) determines which geometry data are required and how they will be computed.

- Cutting edge position
  For the tool type "turning tool", you must also enter the cutting edge position.

The following figures provide information on the required tool parameters for the respective tool type.

![Fig. 8-35 Length offset values for turning tools](image-url)
8.6 Tool and tool offset

Two offset blocks required, e.g.: D1 – cutting edge 1
D2 – cutting edge 2

<table>
<thead>
<tr>
<th>Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>G18: Length 1 in X Length 2 in Z</td>
</tr>
</tbody>
</table>

Fig. 8-36 Turning tool with two cutting edges – length offset

<table>
<thead>
<tr>
<th>Activation</th>
</tr>
</thead>
<tbody>
<tr>
<td>G18: Length 1 in X Length 2 in Z</td>
</tr>
</tbody>
</table>

Fig. 8-37 Offsets for turning tool with tool radius offset

Note: The values for length1 and length2 refer to point P for the edge positions 1..8; for position 9, however, to S (S= P)
Center hole

Switch to G17 for application of a center hole. This makes the length offset take effect for the drill in the Z axis. After drilling, switch back to standard offset for turning tools using G18.

Example:
N10 T... ; Drill
N20 G17 G1 F... Z... ; Length compensation is effective along the Z axis
N30 Z...
N40 G18 .... ; Drilling completed
8.6.4 Selecting the tool radius compensation: G41, G42

Functionality

A tool with a corresponding D number must be active. The tool radius offset (cutting edge radius offset) is activated by G41/G42. The controller automatically calculates the required equidistant tool paths for the programmed contour for the respective current tool radius. G18 must be active.

Programming

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

Remark: The selection can only be made for linear interpolation (G0, G1). Program both axes. If you only specify one axis, the second axis is automatically completed with the last programmed value.
Starting the compensation

The tool travels in a straight line directly to the contour and is positioned perpendicular to the path tangent at the starting point of the contour.
Select the starting point such that a collision-free travel is ensured.

![Diagram of starting contour: Straight line and Circle](image)

Starting contour: Straight line
Corrected tool path
G42
R – cutting edge radius
P1 – starting point of the contour

Starting contour: Circle
P0 – starting point
Circle radius
G42 Corrected tool path
P1 Tangent

Fig. 8-42 Start of the tool radius offset with the example G42, cutting edge position =3

Information

As a rule, the block with G41/G42 is followed by the block with the workpiece contour. However, the contour description may be interrupted by an intervening block that does not contain information for the contour path, e.g. only M command.

Programming example

N10 T... F...
N15 X... Z... ;P0 – starting point
N20 G1 G42 X... Z... ;Selection right of the contour, P1
N30 X... Z... ;Initial contour; circle or straight line

8.6.5 Corner behavior: G450, G451

Functionality

Using the functions G450 and G451, you can set the behavior for non-continuous transition from one contour element to another contour element (corner behavior) when G41/G42 is active.
Internal and external corners are detected by the control system itself. For inside corners, the intersection of the equidistant paths is always approached.
8.6 Tool and tool offset

**Programming**

- G450 ; Transition circle
- G451 ; Point of intersection

**Fig. 8-43** Corner behavior at an external corner

**Transition circle G450**

The tool center point travels around the workpiece external corner in an arc with the tool radius.

In view of the data, for example, as far as the feedrate value is concerned, the transition circle belongs to the next block containing traversing movements, for example, with reference to the feedrate value.

**Point of intersection G451**

For a G451 intersection of the equidistant paths, the point (intersection) that results from the center point paths of the tool (circle or straight line) is approached.
8.6.6 Tool radius compensation OFF: G40

Functionality

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

The tool ends the block before G40 in the normal position (compensation vector vertically to the tangent at the end point); irrespective of the approach angle.

If G40 is active, the reference point is the tool center point. The tool tip then travels to the programmed point upon deselection.

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

Programming

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

Remark: The compensation mode can only be deselected with linear interpolation (G0, G1). Program both axes. If you only specify one axis, the second axis is automatically completed with the last programmed value.

![Diagram](image-url)

Fig. 8-45 Ending the tool radius compensation with G40, with the example of G42, cutting edge position = 3

Programming example

...  
N100 X... Z... ; Last block at the contour; circle or straight line, P1  
N110 G40 G1 X... Z... ; Deactivate tool radius compensation, P2
8.6.7 Special cases of the tool radius compensation

Change of the compensation direction

The compensation direction G41 <-> G42 can be changed without writing G40. The last block with the old compensation direction ends with the normal position of the compensation vector at the end point. The new compensation direction is executed as a compensation start (normal position at starting point).

Repetition of G41, G41 or G42, G42

The same contour can be reprogrammed without writing G40. The last block in front of the new compensation call ends with the normal position of the compensation vector at the end point. The new compensation is carried out as a compensation start (behavior as described for change in compensation direction).

Changing the offset number D

The offset number D can be changed in compensation mode. A modified tool radius is active with effect from the block in which the new D number is programmed. Its complete modification is only achieved at the end of the block. In other words: The modification is traversed continuously over the entire block; this also applies to circular interpolation.

Cancellation of compensation by M2

If the offset mode is canceled with M2 (program end) without writing the command G40, the last block with coordinates ends in the normal offset vector setting. No compensating movement is executed. The program ends with this tool position.

Critical machining cases

When programming, pay special attention to cases where the contour travel is smaller than the tool radius; in case of two successive internal corners, this is smaller than the diameter. Such cases should be avoided.

Also check over multiple blocks that the contour contains no "bottlenecks".

When carrying out a test/dry run, use the largest tool radius you are offered.

Acute contour angles

If very sharp outside corners (\( \alpha \)) occur in the contour with active G451 intersection, the control system automatically switches to transition circle. This avoids long idle motions.
8.6.8 Example of tool radius compensation

Programming example

N1 ;Section of the contour
N2 T1 ;Tool 1 with offset D1
N10 DIAMON F... S... M... ;Radius dimensioning, technological values
N15 G54 G0 G90 X100 Z15
N20 X0 Z6
N30 G1 G42 G451 X0 Z0 ;Start compensation mode
N40 G91 X20 CHF=(5* 1.1223 ) ;Insert chamfer, 30 degrees
N50 Z−25
N60 X10 Z−30
N70 Z−8
N80 G3 X20 Z−20 CR=20
N90 G1 Z−20
N95 X5
N100 Z−25
N110 G40 G0 G90 X100 ;Quit compensation mode
N120 M2
8.6.9 Use of milling cutters

Function

The kinematic transformation functions TRANSMIT and TRACYL are associated with the use of milling cutters on turning machines (see Section 8.14). The tool offsets for milling cutters act differently than with for tools.

<table>
<thead>
<tr>
<th>Activation</th>
<th>Length 1 in Z</th>
<th>Length 2 in Y</th>
<th>Length 3 in X</th>
<th>Radius in X/Y</th>
<th>Radius in Z/X</th>
<th>Radius in Y/Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>G17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The radius is not taken into account for the drill type.

Fig. 8-47 Effect of the offsets on tool type milling cutter

Fig. 8-48 Effect of the tool length compensation – 3D (special case)
Milling cutter radius correction G41, G42

![Diagram](image)

**Starting the compensation**

The tool travels in a straight line directly to the contour and is positioned perpendicular to the path tangent at the starting point of the contour. Select the starting point such that a collision-free travel is ensured.

![Diagram](image)

**Information**

Otherwise, the cutter radius compensation acts as the radius compensation when working with a turning tool (see Sections 8.6.5 to 8.6.7). For detailed information, please refer to

**References:** "Operation and Programming – Milling" SINUMERIK 802D
8.6.10 Special handling of tool compensation

With the SINUMERIK 802D with SW 2.0 and higher, the following special cases are available for the tool compensation.

Influence of setting data

With the use of the following setting data, the operator / programmer can exert an influence on the calculation of the length offsets of the tool used:

- SD 42940: TOOL_LENGTH_CONST
  (allocation of the tool length components to the geometry axes)
- SD 42950: TOOL_LENGTH_TYPE
  (allocation of the tool length components independent of tool type)

Note: The modified setting data will become effective with the next cutting edge selection.

Examples

With SD 42950: TOOL_LENGTH_TYPE =2
the length offset for a loaded milling cutter is calculated as for a turning tool:

- G17: Length 1 in Y axis, length 2 in X axis
- G18: Length 1 in X axis, length 2 in Z axis
- G19: Length 1 in Z axis, length 2 in Y axis

With SD 42940: TOOL_LENGTH_CONST =18
the length assignment is performed in all planes G17 to G19 as for G18:

- Length 1 in X axis, length 2 in Z axis

Setting data in the program

In addition to setting of setting data via operator input, these can also be written in the program.

Example:
N10 $MC_TOOL_LENGTH_TYPE=2
N20 $MC_TOOL_LENGTH_CONST=18

Information

Detailed information on tool offset special actions can be found in

References: Description of Functions, Section "Tool compensation special cases"
8.7 Miscellaneous function (M)

Functionality

The special function M initiates switching operations, for example, such as "Coolant ON/OFF" and other operations on the machine.

Permanent functions have already been assigned to some of the M functions by the control manufacturer. The machine manufacturer can freely dispose of the other functions.

Note:
An overview of the M special functions used and reserved in the control can be found in Section 8.1.6 "Overview of the instructions".

Programming

M... ; a maximum of 5 M functions per block,

Activation

Activation in blocks with axis movements:
If the , M1, M2 functions are contained in a block with traversing movements of the axes, then these M functions become effective after the traversing movements.

The functions M3, M4, M5 are output to the internal PLC before the traversing movements. The axis movements only begin once the controlled spindle has ramped up for M3, M4. For M5, however, the spindle standstill is not waited for. The axis movements already begin before the spindle stops (default setting).

The remaining M functions are output to the PLC with the traversing movements.

If you would like to program an M function directly before or after an axis movement, then insert a separate block with this M function. Please note: This block interrupts G64 continuous path mode and generates exact stop!

Programming example

N10 S...
N20 X... M3 ;M function in a block with axis motion
Spindle ramps up prior to the X axis motion
N180 M78 M67 M10 M12 M37 ;Max. 5 M functions in the block

Note

In addition to M and H functions, T, D, and S functions can also be transferred to the PLC (programmable logic controller). In all, a maximum of 10 such function outputs are possible in a block.
8.8  H function

Information

With SW 2.0 and higher, two spindles are possible. This provides extended programming possibilities for the spindle when using M commands:
- M1=3, M1=4, M1=5, M1=40, ... ; M3, M4, M5, M40, ... for spindle 1
- M2=3, M2=4, M2=5, M2=40, ... ; M3, M4, M5, M40, ... for spindle 2

Functionality

With H functions, floating point data (REAL data type – as with arithmetic parameters, see Section "Arithmetic Parameters R") can be transferred from the program to the PLC.
The meaning of the values for a given H function is defined by the machine manufacturer.

Programming

H0=... to H9999=... ; max. 3 H functions per block

Programming example

N10  H1=1.987  H2=978.123  H3=4 ; 3 H functions in the block
N20  G0 X71.3 H99=−8978.234 ; with axis movements in the block
N30  H5 ; corresponds to: H0=5.0

Note

In addition to M and H functions, T, D, and S functions can also be transferred to the PLC (programmable logic controller). A total of 10 such function outputs are possible in one NC block.
8.9 Arithmetic parameters R, LUD and PLC variables

8.9.1 Arithmetic parameters R

Functionality

The arithmetic parameters are used if an NC program is not only to be valid for values assigned once, or if you must calculate values. The required values can be set or calculated by the control during program execution. The arithmetic parameter values can also be set by operator inputs. If values have been assigned to the arithmetic parameters, they can be assigned to other variable-setting NC addresses in the program.

Programming

R0=... to R299=...

Value assignment

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

\[ \pm (0.000 \, 0001 \ldots 9999 \, 9999) \]  
(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:

\[ R0=3.5678 \quad R1=-37.3 \quad R2=2 \quad R3=-7 \quad R4=-45678.123 \]

Use the **exponential notation** to assign an extended range of numbers:

\[ \pm (10^{-300} \ldots 10^{+300}) \].

The exponent value is written after the **EX** characters; max. number of characters: 10 (including the leading sign and decimal point)

Value range of EX: \(-300\) to \(+300\)

Example:

\[ R0=-0.1EX-5 \quad ; \text{Meaning: } R0 = -0.000 \, 001 \]
\[ R1=1.874EX8 \quad ; \text{Meaning: } R1 = 187 \, 400 \, 000 \]

Remark: Several assignments can be provided per block; this also applies to the assignment of arithmetic expressions.

Assignments to other addresses

The flexibility of an NC program lies in assigning these arithmetic parameters or expressions with arithmetic parameters to other NC addresses. Values, arithmetic expressions and arithmetic parameters can be assigned to all addresses; **Exception: addresses N, G, and L.**
When assigning, write the " = " sign after the address character. It is also possible to have an assignment with a minus sign.

A separate block is required for assignments to axis addresses (traversing instructions).

**Example:**

```
N10 G0 X=R2  //Assignment to the X axis
```

### Arithmetic operations/arithmetic functions

When operators/arithmetic functions are used, it is necessary to use conventional mathematical notation. Machining priorities are set by parentheses. Otherwise, multiplication and division take precedence over addition and subtraction.

Degrees are used for the trigonometrical functions.

Permitted arithmetic functions: see Section "Overview of the instructions"

#### Programming example: R parameters

```
N10 R1= R1+1  //The new R1 results from the old R1 plus 1
N20 R1=R2+R3  R4=R5−R6  R7=R8+ R9  R10=R11/R12
N30 R13=SIN(25.3)  //R13 yields the sine of 25.3 degrees
N40 R14=R1*R2+R3  //Multiplication before addition R14=(R1 *R2)+R3
N50 R14=R3+R2*R1  //Result as block N40
N60 R15=SQRT(R1*R1+R2*R2)  //Meaning: R15 = \sqrt{R1^2 + R2^2}
```

#### Programming example: Assignment to the axes

```
N10 G1 G91 X=R1 Z=R2 F3
N20 Z=R3
N30 X=−R4
N40 Z=−R5
```

### 8.9.2 Local User Data (LUD)

#### Functionality

The operator/programmer (user) can define his/her own variable in the program from various data types (LUD = Local User Data). These variables are only available in the program in which they were defined. The definition takes place immediately at the start of the program and can also be associated with a value assignment at the same time. Otherwise the starting value is zero.
The name of a variable can be defined by the programmer. The naming is subject to the following rules:

- maximum length 32 characters
- The first two characters must be letters. Only use letters, digits, or underscore.
- Do not use a name that is already being used in the controller (NC addresses, keywords, names of programs, subroutines, etc.)

**Programming**

```
DEF BOOL  varname1 ; Bool type, values: TRUE (=1), FALSE (=0)
DEF CHAR  varname2 ; Char type, 1 character in the ASCII code: "a", "b", ...
                    ; Code numerical value: 0 ... 255
DEF INT   varname3 ; Integer type, integer values, 32–bit value range:
                    ; −2 147 483 648 to +2 147 483 648 (decimal)
DEF REAL  varname4 ; Real type, natural number (such as arithmetic parameters R),
                    ; Range of values: ±(0.000 0001 ... 9999 9999)  
                    ; (8 decimal places, arithmetic sign and decimal point) or
                    ; exponential notation: ± (10⁻³⁰⁰ ... 10⁺³⁰⁰).
```

Each type requires a separate program line. It is, however, possible to define several variables of the same type in a line.

Example:

```
DEF INT PVAR1, PVAR2, PVAR3=12, PVAR4 ; 4 variables of the INT type
```

**Fields**

In addition to the individual variables, one or two–dimensional fields of variables of these data types can also be defined:

```
DEF INT PVAR5[n] ; One–dimensional field of the INT type, n: integer
DEF INT PVAR6[n,m] ; Two–dimensional field of the INT type, n, m: integer
```

Example:

```
DEF INT PVAR7[3] ; Field with 3 elements of the INT type
```

Access to the individual field elements is granted in the program via the field index; each individual field element can be handled as an individual variable. The field index ranges from 0 to "less number of elements".

Example:

```
N10 PVAR7[2]=24 ; The third field element (with index 2) is assigned the value 24.
```

Value assignment for field with SET instruction:

```
N20 PVAR5[2]=SET(1,2,3) ; Beginning with the 3rd field element, different values are assigned.
```

Value assignment for field with REP instruction:

```
N20 PVAR7[4]=REP(2) ; As of field element [4], all elements is assigned the same value, here 2.
```
Number of LUDs

With the SINUMERIK 802D, max. 200 LUDs may be defined. However, please note: The standard cycles of SIEMENS also use LUDs and they share this quantity with the user. Always keep a sufficient reserve if you are working with these cycles.

Note for display

There is no special display for LUDs. They would only be visible during execution of the program anyway.

For testing purposes, when creating the program, the LUDs may be assigned to the arithmetic parameters R and are thus visible via the arithmetic parameter display, but are converted into the REAL type.

Another possibility of displaying is offered in the STOP condition of the program via a message output:

```
MSG(" VAR1 value: "<<PVAR1<<" VAR2 value: ": "<<PVAR2) ; Values of PVAR1, PVAR2
```

8.9.3 Reading and writing PLC variables

Functionality

To allow rapid data exchange between NC and PLC, a special data area exists in the PLC user interface with a length of 512 bytes. In this area, PLC data are compatible in data type and position offset. In the NC program, these compatible PLC variables can be read or written.

Special system variables are provided here:

- `$A_DBB[n]` ; Data byte (8-bit value)
- `$A_DBW[n]` ; Data word (16-bit value)
- `$A_DBD[n]` ; Data double-word (32-bit value)
- `$A_DBR[n]` ; REAL data (32-bit value)

`n` stands here for the position offset (between the data area start and the variable start) in bytes

Example:

```
R1=$A_DBR[5] ; Reading of a REAL value, offset 5 (starts at byte 5 of this range)
```

Notes

- When reading variables, a preprocessing stop is generated (internal STOPRE).
- A maximum of 3 variables can be written simultaneously (in a block).
8.10 Program jumps

8.10.1 Jump destination for program jumps

Functionality

A label or a block number serve to mark blocks as jump destinations for program jumps. Program jumps can be used to branch to the program sequence.

Labels can be freely selected, but must contain a minimum of 2 and a maximum of 8 letters or numbers, and the first two characters must be letters or underscores.

Labels that are in the block that serves as the jump destination are ended by a colon. They are always at the start of a block. If a block number is also present, the label is located after the block number.

Labels must be unique within a program.

Programming example

N10 LABEL1: G1 X20 ; LABEL1 is the label, jump destination
...
TR789: G0 X10 Z20 ; TR789 is the label, jump destination
− no block number exists
N100 ... ; A block number can be a jump destination
...

8.10.2 Unconditional program jumps

Functionality

NC programs process their blocks in the sequence in which they were arranged when they were written.

The processing sequence can be changed by introducing program jumps.

The jump destination can be a block with a label or with a block number. This block must be located within the program.

The unconditional jump instruction requires a separate block.
8.10 Program jumps

Programming

GOTOF Label ; Jump forward (towards the last block of the program)
GOTOB Label ; Jump backward (towards the first block of the program)

Label ; Selected character string for the label (jump mark) or block number

8.10.3 Conditional program jumps

Functionality

Jump conditions are formulated after the IF instruction. If the jump condition (value not zero) is satisfied, the jump takes place.

The jump destination can be a block with a label or with a block number. This block must be located within the program.

Conditional jump instructions require a separate block. Several conditional jump instructions can be located in the same block.

By using conditional program jumps, you can also considerably shorten the program, if necessary.

Programming

IF condition GOTOF label ; Jump forward
IF condition GOTOB label ; Jump backward

GOTOF ; Jump direction forward (in the direction of the last block of the program)
GOTOB ; Jump direction backward (in the direction of the first block of the program)
Label ; Selected string for the label (jump label) or block number

Fig. 8-51 Unconditional jumps using an example
Programming example for comparing operators

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

Programming example

N10 IF R1 GOTOF LABEL1 ; If R1 is not zero, go to block with LABEL1
... N90 LABEL1: ...
N100 IF R1>1 GOTOF LABEL2 ; If R1 is greater than 1, go to the block with LABEL2
... N150 LABEL2: ...
... N800 LABEL3: ...
... N1000 IF R45==R7+1 GOTOB LABEL3 ; If R45 is equal to R7 plus 1, go to the block with LABEL3
... Several conditional jumps in the block:
N10 MA1: ...
... N20 IF R1==1 GOTOB MA1  IF R1==2 GOTOF MA2 ...
... N50 MA2: ...
Remark: The jump is executed for the first fulfilled condition.
8.10.4 Program example for jumps

Task

Approaching points on a circular section:
Given:
- Starting angle: 30° in R1
- Circle radius: 32 mm in R2
- Distance of the positions: 10° in R3
- Number of points: 11 in R4
- Position of the circle center point in Z: 50 mm in R5
- Position of the circle center point in X: 20 mm in R6

![Diagram showing points on a circular section](image)

Fig. 8-52 Approaching points on a circular section

Programming example

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

Explanation

In block N10, the starting conditions are assigned to the corresponding arithmetic parameters. The calculation of the coordinates in X and Z and the processing takes place in N20.

In block N30, R1 is incremented by the clearance angle R3; R4 is decremented by 1.
If R4 > 0, N20 is executed again; otherwise, N50 with end of program.
8.11 Subroutine technique

8.11.1 General

Use

Basically, there is no difference between a main program and a subroutine.

Frequently recurring machining sequences are stored in subroutines, e.g. certain contour shapes. These subroutines are called at the appropriate locations in the main program and then executed.

One form of subroutine is the machining cycle. Machining cycles contain universally valid machining scenarios (e.g.: thread cutting, stock removal, etc.). By assigning values via included transfer parameters, you can adapt the subroutine to your specific application.

Structure

The structure of a subroutine is identical to that of a main program (see Section 8.1.2 "Program structure"). Like main programs, subroutines contain M2 program end in the last block of the program sequence. This means a return to the program level where the subroutine was called from.

End of program

The end instruction RET can also be used instead of the M2 program end in the subroutine. RET requires a separate block.

The RET instruction is used when G64 continuous-path mode is not to be interrupted by a return. With M2, G64 is interrupted and exact stop is initiated.

Fig. 8-53 Example of sequence when calling a subroutine twice
**Subroutine technique**

**Subroutine name**

The subprogram is given a unique name allowing it to be selected from among the others. When you create the program, the program name may be freely selected provided the following conventions are observed:

The same rules apply as for the names of main programs.

Example: **BUCHSE7**

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

Note: With address L, leading zeros are meaningful for differentiation.

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

These are three different subroutines.

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

**Subroutine call**

Subroutines are called in a program (main or subprogram) with their names.

To do this, a separate block is required.

**Example:**

```
N10 L785    ; Call of the subroutine L785
N20 SHAFT7  ; Call of the subroutine SHAFT7
```

**Program repetition P...**

If a subroutine is to be executed several times in succession, write the number of times it is to be executed in the block of the call after the subroutine name under the address **P**. A maximum of 9999 passes are possible (P1 ... P9999).

**Example:**

```
N10 L785 P3  ; Call of the subroutine L785, 3 passes
```

**Nesting depth**

Subroutines can also be called from a subroutine, not only from a main program. Totally, 8 program levels, including the main program level are available for such a nested call.
8.11 Subroutine technique

Information

Modal G functions can be changed in the subroutine, e.g. G90 -> G91. When returning to the calling program, ensure that all modal functions are set the way you need them to be. This also applies to the arithmetic parameters R. Please make sure that the values of your arithmetic parameters used in upper program levels are not inadvertently changed in lower program levels.

When working with SIEMENS cycles, up to 4 program levels are needed.

8.11.2 Calling machining cycles

Functionality

Cycle are technology subroutines that realize a certain machining process, such as drilling or thread cutting, generally. Adaptation to the concrete problem is done directly via supply parameters/values when calling the respective cycle.

Programming example

N10  CYCLE83(110, 90, ...) ; Call of cycle 83, transfer values directly, separate block
...
N40 RTP=100  RFP= 95.5 ... ; Set transfer parameters for cycle 82
N50 CYCLE82(RTP, RFP, ...) ; Call of cycle 82, separate block
### 8.12 Timers and workpiece counters

#### 8.12.1 Runtime timer

**Functionality**

The timers are prepared as system variables ($A...$) that can be used for monitoring the technological processes in the program or only in the display. These timers can only be read. There are timers that are always active. Others can be deactivated via machine data.

**Timers – always active**

- Time since the last "Control power–up with default values" (in minutes):
  
  $\text{AN SETUP TIME}$ (read–only)
  
  "Power–up of the control system with default values" will reset it to zero automatically.

- Time since the last "Power–up of the control system" (in minutes):
  
  $\text{AN POWERON TIME}$ (read–only)
  
  "It is reset to zero automatically with each power–up of the control system.

**Timers that can be deactivated**

The following timers are activated via machine data (default setting).
The start is timer–specific. Each active run–time measurement is automatically interrupted in the stopped program state or for feedrate–override–zero.
The behavior of the activated timers for active dry run feedrate and program testing can be specified using machine data.

- Total runtime of NC programs in the AUTOMATIC mode (in seconds)
  
  $\text{AC OPERATING TIME}$
  
  In the AUTOMATIC mode, the runtimes of all programs between NC START and end of program / RESET are summed up. The timer is set to zero upon every booting of the control system.

- Runtime of the selected NC program (in seconds):
  
  $\text{AC CYCLE TIME}$
  
  The runtime between NC START and end of program / RESET is measured in the selected NC program. The timer is reset with the start of a new NC program.

- Tool action time (in seconds):
  
  $\text{AC CUTTING TIME}$
  
  The runtime of the path axes is measured in all NC programs between NC START and end of program / RESET (without active rapid traverse) with the tool active.
The measurement is also interrupted if a dwell time is active.
The timer is automatically set to zero upon every booting of the control system.
Programming example

N10 IF $AC_CUTTING_TIME>=R10 GOTO WZZEIT ; Tool action time limit value?
...
N80 WZZEIT:
N90 MSG("Tool action time: Limit value reached")
N100 M0

Display

The contents of the active system variables is displayed on the screen in the “OFFSET/PARAM” operating area -> “Setting data” softkey (2nd page):

- **Run time** = $AC_OPERATING_TIME
- **Cycle time** = $AC_CYCLE_TIME
- **Cutting time** = $AC_CUTTING_TIME
- **Setup time** = $AN_SETUP_TIME
- **Power on time** = $AN_POWERON_TIME

"Cycle time" is also displayed in the AUTOMATIC mode in the "Position" operating area in the information line.

8.12.2 Workpiece counter

Functionality

The "Workpiece counter” function provides counters for counting workpieces. These counters exist as system variables with write and read access from the program or via operator input (observe the protection level for writing!). Machine data can be used to control counter activation, counter reset timing and the counting algorithm.

Counters

- **Number of workpieces required (required number of workpieces):**
  
  $AC_REQUIRED_PARTS

  The number of workpieces at which the number of current workpieces $AC_ACTUAL_PARTS is set to zero can be defined in this counter.

  The generation of the display alarm 21800 "Workpiece setpoint reached" can be activated via machine data.

- **Number of workpieces produced in total (actual total):**
  
  $AC_TOTAL_PARTS

  The counter specifies the number of all workpieces manufactured from the time of starting.

  The counter is automatically set to zero upon every booting of the control system.

- **Number of current workpieces (actual total):**
  
  $AC_ACTUAL_PARTS

  This counter registers the number of all workpieces produced since the starting time. When the workpiece setpoint is reached ($AC_REQUIRED_PARTS, value greater than zero), the counter is automatically zeroed.
8.12 Timers and workpiece counters

- Number of workpieces specified by the user:
  $AC\_SPECIAL\_PARTS
  This counter allows user-defined workpiece counting. Alarm output can be defined for
  the case of identity with $AC\_REQUIRED\_PARTS (workpiece target). The user must re-
  set the counter himself.

**Programming example**

N10 IF $AC\_TOTAL\_PARTS==R15 GOTO SIST ; Count reached?
...
N80 SIST:
N90 MSG("Workpiece setpoint reached")
N100 M0

**Display**

The contents of the active system variables are displayed on the screen in the
"OFFSET/PARAM" operating area -> "Setting data" softkey (2nd page):
- **Part total** = $AC\_TOTAL\_PARTS
- **Part required** = $AC\_REQUIRED\_PARTS
- **Part count** = $AC\_ACTUAL\_PARTS
  $AC\_SPECIAL\_PARTS (not displayed)

"Part count" is also displayed in the AUTOMATIC mode in the "Position" operating area in
the information line.
8.13 Language commands for tool monitoring

8.13.1 Tool monitoring overview

With SINUMERIK 802D, this function is an option and available with software release 2.0 and higher.

Functionality

The tool monitoring is activated via machine data.

The following types of active cutting edge monitoring for the active tool are possible:

- Monitoring of the **tool life**
- Monitoring of the **workpiece count**

For a tool, the above-mentioned types of monitoring can be activated simultaneously.

The control / data input of tool monitoring is preferably done by operator input. In addition, functions are programmable.

Monitoring counter

Monitoring counters exist for each monitoring type. The monitoring counters count from a set value > 0 down to zero. When a counter has decremented to a value of <=0, the limit value is reached. A corresponding alarm message is issued.

System variable for type and condition of the monitoring

- $TC_TP8[t]$ – status of the tool with number t:
  - Bit 0 = 1: Tool is **active**
  - = 0: Tool not active
  - Bit 1 = 1: Tool is **enabled**
  - = 0: not enabled
  - Bit 2 = 1: Tool is **disabled**
  - = 0: not disabled
  - Bit 3: Reserved
  - Bit 4 = 1: **Prewarning limit reached**
  - = 0: Not reached

- $TC_TP9[t]$ – type of the monitoring function for the tool with number t:
  - = 0: No monitoring
  - = 1: (Tool monitored for the tool life
  - = 2: Count–monitored tool

These system variables can be read/written in the NC program.

System variables for tool monitoring data

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
<th>Data type</th>
<th>Default setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TC_MOP1[t,d]$</td>
<td>Prewarning limit for tool life in minutes</td>
<td>REAL</td>
<td>0.0</td>
</tr>
<tr>
<td>$TC_MOP2[t,d]$</td>
<td>Residual tool life in minutes</td>
<td>REAL</td>
<td>0.0</td>
</tr>
<tr>
<td>$TC_MOP3[t,d]$</td>
<td>Count prewarning limit</td>
<td>INT</td>
<td>0</td>
</tr>
<tr>
<td>$TC_MOP4[t,d]$</td>
<td>Remaining part quantity</td>
<td>INT</td>
<td>0</td>
</tr>
</tbody>
</table>
System variables for active tool

The following can be read in the NC program via system variables:

- $P_{\text{TOOLNO}}$ – number of the active tool T
- $P_{\text{TOOL}}$ – active D number of the active tool

8.13.2 Tool life monitoring

Tool life monitoring is done for the tool cutting edge that is currently in use (active cutting edge D of the active tool T).

As soon as the path axes traverse (G1, G2, G3, ... but not for G0), the residual tool life ($TC_{\text{MOP2}[t,d]}$) of this tool cutting edge is updated. If the residual tool life of a tool's cutting edge runs below the value of "Prewarning limit for tool life" ($TC_{\text{MOP1}[t,d]}$), it is reported via an interface signal " to the PLC.

If the residual tool life $\leq 0$, an alarm is issued and an additional interface signal is set. The tool changes to the “disabled” condition and cannot be programmed again until this condition changes. The operator must intervene: The operator must change the tool or ensure that he has an operational tool for machining.

$A_{\text{MONIFACT}}$ system variable

The $A_{\text{MONIFACT}}$ system variable (REAL data type) allows the monitoring clock to be run slower or faster. This factor can be set before using the tool, in order to take the different kinds of wear into consideration according to the workpiece material used, for example.

After booting of the control, reset / program end, the factor $A_{\text{MONIFACT}}$ has the value 1.0. Real time is in effect.

Examples for accounting:

- $A_{\text{MONIFACT}}=1 \quad 1 \text{ minute real time} = 1 \text{ minute tool life which is decremented}$
- $A_{\text{MONIFACT}}=0.1 \quad 1 \text{ minute real time} = 0.1 \text{ minute tool life which is decremented}$
- $A_{\text{MONIFACT}}=5 \quad 1 \text{ minute real time} = 5 \text{ minutes tool life which are decremented}$

Setpoint update with RESETMON

The RESETMON(state, t, d, mon) function sets the actual value to the setpoint:

- either for all cutting edges or only for a certain cutting edge of a certain tool
- either for all monitoring type or only for a certain monitoring type.
Transfer parameters:

INT  state  Status of executing the command:
   = 0  Command executed successfully
   = −1  The edge with the specified D number d does not exist.
   = −2  The tool with the specified T number t does not exist.
   = −3  The specified tool t does not have a defined monitoring function.
   = −4  The monitoring function is not activated, i.e. the command is not executed.

INT  t  Internal T number:
   = 0  For all tools
   <> 0  For this tool (t < 0: Absolute value formation |t|)

INT  d  optional: D number of the tool with the number t:
   > 0  for this D number
   without d /= 0 all cutting edges of tool t

INT  mon  optional: bit-encoded parameters for the monitoring type (values like $TC_TP9):
   = 1: Tool life
   = 2: Count
   without monitoring or = 0: All actual values of the monitoring functions active for tool t are set to the setpoints.

Notes:

− RESETMON( ) has no effect during active Program test.
− The variable for the state status feedback must be defined at the beginning of the program using a DEF instruction: DEF INT state
  You can also define a different name for the variable (instead of state, with a maximum of 15 characters, beginning with 2 letters). The variable is only available in the program in which it was defined.
  The same applies to the monitoring type variable mon. If an entry is required for this, it may also be passed directly as a number (1 or 2).

8.13.3 Workpiece count monitoring

The workpiece count of the active cutting edge of the active tool is monitored.
Monitoring of the unit quantity covers all tool cutting edges, which are used to manufacture a workpiece. If the count is changed by new parameters, the monitoring data are adapted to all of the tool cutting edges that became active since the last unit count.

Updating the workpiece count by operator input or SETPIECE( )

The workpiece count can be updated by an operator input (HMI) or in the NC program through the SETPIECE( ) language command.
By using the **SETPIECE** function, the programmer can update the count monitoring data of the tools involved in the machining process. All tools that became active since the last activation of **SETPIECE** are acquired with their D numbers. If a tool is active at the time when **SETPIECE()** is called, it is also counted.

If a block containing path axis motions is programmed after **SETPIECE()**, the appropriate tool is also taken into account in the next **SETPIECE** call.

**SETPIECE(x)** ;

- **x**: 1...32000 Number of workpieces produced since the last execution of the **SETPIECE** function.

  - The counter status for the remaining part quantity ($TC_MOP4[t,d]$) is reduced by this value.
  - **x**: 0 Deletion of all counters for the residual count ($TC_MOP4[t,d]$) for the tools/D number involved in machining since then.

  Alternatively, the deletion via operation is recommended (HMI).

**Programming example**

```plaintext
N10 G0 X100
N20 ...
N30 T1
N40 M6
N50 D1
N60 SETPIECE(2) ;$TC_MOP4[1,1] (T1,D2) is decremented by "2"
N70 T2
N80 M6
N90 SETPIECE(0) ;Deletion command for the tools stored
N91 D2
N100 SETPIECE(1) ;$TC_MOP4[2,2] (T2,D2) is decremented by "1"
N110 SETPIECE(0) ;Deletion command for the tools stored
N120 M30
```

**Notes:**

- The **SETPIECE()** command is not active in the block search.
- Direct writing of $TC_MOP4[t,d]$ is recommended only in simple cases. A subsequent block with the STOPRE command is required.

**Setpoint update**

The setpoint update, i.e. setting the remaining workpiece counters ($TC_MOP4[t,d]$) to the workpiece count setpoint ($TC_MOP13[t,d]$) is typically performed via operator input (HMI). It can, however, also be performed through the **RESETMON** (state, t, d, ,mon) function as already described for the service life monitoring.
Example:
DEF INT state ; Defining a variable for the status feedback in the beginning of the program
...
N100 RESETMON(state,12,1,2) ; Updating the setpoint of the workpiece counter for
T12, D1
...

Programming example

DEF INT state ; Define variable for the status feedback from RESETMON()
;
G0 X... ; Retraction
T7 ; Load new tool, possibly with M6
$TC_MOP3[SP_TOOLNO,SP_TOOL]=100 ; Prewarning limit 100 pcs.
$TC_MOP4[SP_TOOLNO,SP_TOOL]=700 ; Residual count
$TC_MOP13[SP_TOOLNO,SP_TOOL]=700 ; Setpoint of count
; Activation after setting:
$TC_TP9[SP_TOOLNO,SP_TOOL]=2 ; Activation of count monitoring, active tool
STOPRE
ANF:
BEARBEIT ; Subroutine for workpiece machining
SETPIECE(1) ; Update counter
M0 ; Next tool; press NC START to continue
IF ($TC_MOP4[SP_TOOLNO,SP_TOOL]>1) GOTOB ANF
MSG("Tool T7 worn – please change")
M0 ; after changing the tool, press NC START to continue
RESETMON(state,7,1,2) ; Workpiece counter setpoint update
IF (state<>0) GOTOF ALARM
GOTOB ANF
ALARM: ; display errors:
MSG("Error RESETMON: " <<state)
M0
M2
8.14 Milling on turning machines

Note
This is not applicable to the 802D bl.

8.14.1 Milling of the front face – TRANSMIT

With SINUMERIK 802D, this function is an option and available with software release 2.0 and higher.

Functionality

- The kinematic transformation function TRANSMIT allows milling / drilling on the front face of turned parts in the turning clamp.
- A Cartesian coordinate system is used to program these machining operations.
- The control system transforms the programmed traversing motions of the Cartesian coordinate system into motions of the real machine axes. The main spindle functions here as the machine rotary axis.
- TRANSMIT must be configured via special machine data elements. A tool center offset relative to the turning center is permitted and is also configured via these machine data elements.
- In addition to the tool length offset, it is also possible to work with the tool radius offset (G41, G42).
- The velocity control makes allowance for the limits defined for the rotations.

Fig. 8-55 Milling of the front face
Programming

TRANSMIT ; Activate TRANSMIT (separate block)
TRAFOOF ; Deactivate (separate block)

TRAFOOF deactivates any active transformation function.

Programming example

Fig. 8-56 Cartesian coordinate system X, Y, Z with origin at the turning center when programming
TRANSMIT

; Mill square, eccentric and turned
N10 T1 F400 G94 G54 ; Cutter tool, feedrate, feedrate type
N20 G0 X50 Z60 SPOS=0 ; Approach starting position
N25 SETMS(2) ; Master spindle is now milling spindle
N30 TRANSMIT ; Activate TRANSMIT function
N35 G55 G17 ; Work offset; activate X/Y plane
N40 ROT RPL=−45 ; Programmable rotation in the X/Y plane
N50 ATRANS X−2 Y3 ; Programmable rotation
N55 S600 M3 ; Activate milling spindle
N60 G1 X12 Y−10 G41 ; Activate tool radius compensation
N65 Z−5 ; Feed cutting tool
N70 X−10
N80 Y10
N90 X10
N100 Y−12
N110 G0 Z40 ; Retract cutting tool
N120 X15 Y−15 G40 ; Deactivate tool radius compensation
N130 TRANS ; Deactivate programmable offset and rotation
N140 M5 ; Deactivate milling spindle
N150 TRAFOOF ; Deactivate TRANSMIT
N160 SETMS ; Master spindle is now main spindle again
N170 G54 G18 G0 X50 Z60 SPOS=0 ; Approach starting position
N200 M2
Information

The turning center with X0/Y0 is designated as the pole. Workpiece machining operations close to the pole are not recommended since these may require sharp feedrate reductions to prevent overloading of the rotary axis. Do not select TRANSMIT if the tool is positioned exactly on the pole. Do not traverse with the tool tip center across the pole X0/Y0.

References: Description of Functions, Section "Kinematic transformations"

8.14.2 Milling of the peripheral surface – TRACYL

With SINUMERIK 802D, this function is an option and available with software release 2.0 and higher.

Functionality

- The kinematic transformation function TRACYL is used for milling machining of the peripheral surface of cylindrical objects and allows the production of slots at any position.
- The path of the slot is programmed in the level peripheral surface, which was logically developed for a specific machining cylinder diameter.

![Cartesian coordinate system X, Y, Z when programming TRACYL](image)

- The control system transforms the programmed traversing motions in the Cartesian coordinate system X, Y, Z into motions of the real machine axes. The main spindle functions here as the machine rotary axis.
- TRACYL must be configured using special machine data. The rotary axis position at which the value Y=0 is also defined here.
• If the machine has a real machine Y axis (YM), an expanded TRACYL variant can also be configured. This allows slots with slot side offset to be produced: The slot side and base are perpendicular to each other – even if the milling tool's diameter is smaller than the slot width. This is otherwise only possible with exact fitting milling cutters.

![Fig. 8-58 Special machine kinematics with additional machine Y axis (YM)](image)

![Fig. 8-59 Various slots (cross section)](image)

**Programming**

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

TRAFOOF deactivates any active transformation function.
OFFN address

Distance from the slot side to the programmed path
The slot center line is generally programmed. OFFN defines the (half) slot width for activated cutter radius compensation (G41, G42).
Programming: OFFN=...; Distance in mm

Note:
Set OFFN=0 once the slot has been completed. OFFN is also used outside of TRACYL – for offset programming in combination with G41, G42.

Programming notes

In order to mill with TRACYL, the slot center line is programmed in the part program with the coordinates and the (half) slot width is programmed with OFFN.
OFFN does not go into effect until tool radius compensation is selected. Furthermore, it must be guaranteed that OFFN >= tool radius to avoid that the opposite slot side is damaged.

A part program for milling a slot generally consists of the following steps:
1. Select a tool
2. Select TRACYL
3. Select a suitable work offset
4. Positioning
5. Program OFFN
6. Select the tool radius compensation
7. Approach block (position TRC and approach slot side)
8. Program the slot path via slot center line
9. Deselect the tool radius compensation
10. Retraction block (retract TRC from slot side)
11. Positioning
12. Clear OFFN
13. TRAFOOF (deselect TRACYL)
14. Reselect the original work offset (see also following programming example)
Information

- Guiding slots:
  By using a tool diameter that corresponds exactly to the slot width, it is possible to produce an exact slot. The tool radius compensation (TRC) is not disabled in this case. With TRACYL, slots can also be produced, whose tool diameter is smaller than the slot width. For this, the tool radius compensation (G41, GG42) and OFFN are used. To avoid problems of accuracy, the tool diameter should only be slightly smaller than the slot width.

- When working with TRACYL with slot side correction, the axis used for the correction (YM) should stand on the turning center. Thus, the slot is created centered on the programmed slot center line.

- Selecting the tool radius compensation (TRC):
  The TRC is in effect for the programmed slot center line. The slot side results. G42 is input so that the tool traverses to the left of the slot side (to the right of the slot center line). Accordingly, G41 is to be written to the right of the slot side (to the left of the slot center line).
  As an alternative to exchanging G41<→G42, you can input the slot width with a minus sign in OFFN.

- Since, even without TRACYL, OFFN is included when TRC is active, OFFN should be reset to zero after TRAFOOF. OFFN acts differently with TRACYL than it does without TRACYL.

- It is not possible to change OFFN within the part program. This allows the actual slot center line to be offset from the center.

References: Description of Functions, Section "Kinematic transformations"

Programming example

Making a hook-shaped slot

Fig. 8-61 Producing a slot (example)
Programming the slot, values at the slot base

; Machining diameter of the cylinder at the slot base: 35.0 mm
; Desired total slot width: 24.8 mm, the cutter in use has a radius of: 10.123 mm
N10 T1 F400 G94 G54 ; Cutter tool, feedrate, feedrate type, work offset
N30 G0 X25 Z50 SPOS=200 ; Approach starting position
N35 SETMS(2) ; Master spindle is now the milling spindle
N40 TRACYL (35.0) ; Activate TRACYL, machining diameter 35.0 mm
N50 G55 G19 ; Work offset, plane selection: Y/Z plane
N60 S800 M3 ; Activate milling spindle
N70 G0 Y70 Z10 ; Y / Z start position
N80 G1 X17.5 ; Feed cutter to slot base
N70 OFFN=12.4 ; 12.4 slot side distance to slot center line
N90 G1 Y70 Z1 G42 ; Activate TRC, approach slot side
N100 Z−30 ; Slot section parallel to cylinder axis
N110 Y20 ; Slot section parallel to circumference
N120 G42 G1 Y20 Z−30 ; Restart TRC, approach other slot side, slot distance continues to be 12.4 mm to slot center line
N130 Y70 F600 ; Slot section parallel to the circumference
N140 Z1 ; Slot section parallel to cylinder axis
N150 Y70 Z10 G40 ; Deactivate TRC
N160 G0 X25 ; Retract cutter
N170 M5 OFFN=0 ; Deactivate milling spindle, clear slot side distance
N180 TRAFOOF ; Deactivate TRACYL
N190 SETMS ; Master spindle is now main spindle again
N200 G54 G18 G0 X25 Z50 SPOS=200 ; Approach starting position
N210 M2
### 8.15 G functions equivalent to the SINUMERIK 802S – Turning

<table>
<thead>
<tr>
<th>SINUMERIK 802S</th>
<th>SINUMERIK 802D</th>
</tr>
</thead>
<tbody>
<tr>
<td>G5</td>
<td>CIP</td>
</tr>
<tr>
<td>G158</td>
<td>TRANS</td>
</tr>
<tr>
<td>G22</td>
<td>DIAMOF</td>
</tr>
<tr>
<td>G23</td>
<td>DIAMON</td>
</tr>
</tbody>
</table>

All the other G functions are the same as with 802S and 802D provided that they exist there.
8.15  G functions equivalent to the SINUMERIK 802S – Turning

This sheet has been left empty for your notes
Cycles

9.1 Overview of cycles

Cycles are generally applicable technology subroutines that can be used to carry out a specific machining process, such as tapping. These cycles are adapted to individual tasks by parameter assignment.

The cycles described here are the same as supplied for the SINUMERIK 840D/810D.

Drilling cycles and turning cycles

The following standard cycles can be carried out using the SINUMERIK 802D control system:

- Drilling cycles
  - CYCLE81 Drilling, centering (not with 802D bl)
  - CYCLE82 Drilling, counterboring
  - CYCLE83 Deep hole drilling
  - CYCLE84 Rigid tapping
  - CYCLE840 Tapping with compensating chuck
  - CYCLE85 Reaming 1 (boring 1)
  - CYCLE86 Boring (boring 2) (not with 802D bl)
  - CYCLE87 Boring with stop (boring 3) (not with 802D bl)
  - CYCLE88 Drilling with stop 2 (boring 4)
  - CYCLE89 Reaming 2 (boring 5)
  - HOLES1 Row of holes
  - HOLES2 Circle of holes

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

- Turning cycles
  - CYCLE93 Recess
  - CYCLE94 Recess (forms E and F to DIN)
  - CYCLE95 Stock removal with relief cutting
  - CYCLE96 Thread undercut
  - CYCLE97 Thread cutting
CYCLE98    Thread chain (not with 802D bl)
The cycles are supplied with the tool box. They are loaded via the RS232 interface into the part program memory during the start-up of the control system.

Auxiliary cycle subroutines
The cycle package includes the following auxiliary subroutines:
- cyclest.spf
- steigung.spf and
- meldung.spf
These must always be loaded in the control.

9.2 Programming cycles

A standard cycle is defined as a subroutine with name and parameter list.

Call and return conditions
The G functions effective prior to the cycle call and the programmable offsets remain active beyond the cycle.
The machining plane G17 for drilling cycles or G18 for turning cycles is to be defined before calling the cycle.
With drilling cycles, the drilling operation is carried out in the axis standing vertically to the current plane.

Messages output during execution of a cycle
During some cycles, messages that refer to the state of machining are displayed on the screen of the control system during program execution.
These message do not interrupt the program execution and continue to be displayed on the screen until the next message appears.
The message texts and their meaning are listed together with the cycle to which they refer.
A summary of all relevant messages is to be found in Section 9.4.

Block display during execution of a cycle
The cycle call is displayed in the current block display for the duration of the cycle.

Cycle call and parameter list
The defining parameters for the cycles can be transferred via the parameter list when the cycle is called.
Note
Cycle calls must always be programmed in a separate block.

Basic instructions with regard to the assignment of standard cycle parameters

The Programming Guide describes the parameter list of every cycle with the
- order and the
- type.

It is imperative to observe the order of the defining parameters.

Each defining parameter of a cycle has a certain data type. The parameter being used must be specified when the cycle is called. In the parameter list, you can transfer
- R parameters (only numerical values)
- constants.

If R parameters are used in the parameter list, they must first be assigned values in the calling program. Cycles can be called
- with an incomplete parameter list
  or
- by leaving out parameters.

If transfer parameters are omitted at the end of the parameter list, the parameter list must be prematurely ended with ")". If any parameters are to be omitted within the list, a comma "..., ...," must be written as a placeholder.

No plausibility checks are made for parameter values with a limited range of values unless an error response has been specifically described for a cycle.

If when calling the cycle the parameter list contains more entries than parameters are defined in the cycle, the general NC alarm 12340 "Too many parameters" is displayed and the cycle is not executed.

Cycle call

The individual methods for writing a cycle are shown in the programming examples provided for the individual cycles.

Simulation of cycles

Programs with cycle calls can be tested first in simulation.

During simulation, the traversing movements of the cycle are visualized on the screen.
9.3 Graphical cycle support in the program editor

The program editor in the control system provides you with programming support to add cycle calls to the program and to enter parameters.

**Function**

The cycle support consists of three components:
1. Cycle selection
2. Input screenforms for parameter assignment
3. Help display per cycle.

**Overview of required files**

The following files constitute the basis for cycle support:
- sc.com
- cov.com

**Note**

These files are loaded during the start-up of the control system and must always remain loaded.

**Operating the cycle selection**

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

- Branching to selection bars for the individual cycles is possible in the horizontal softkey bar using the "Drilling" and "Turning" softkeys provided.
- The cycle selection is carried out using the vertical softkey bar until the appropriate input screenform with the help display appears on the screen.
- The values can be entered either directly (numerical values) or indirectly (R parameters, e.g. R27, or expressions consisting of R parameters, e.g. R27+10). If numerical values are entered, a check is carried out to see whether the value is within the admissible range.
- Some parameters that may have only a few values are selected using the toggle key.
- For drilling cycles, it is also possible to call a cycle modally using the vertical "Modal Call" softkey. The modal call is selected via "Deselect modal" from the drilling cycles list box.
- Press "OK" to confirm (or "Abort" in case of error).

**Recompiling**
Recompiling of program codes serves to make modifications to an existing program using the cycle support.

Position the cursor on the line to be modified and press the "Recompile" softkey. This will reopen the input screenform from which the program piece has been created, and you can modify the values.
9.4 Drilling cycles

9.4.1 General

Drilling cycles are motional sequences defined to DIN 66025 for drilling, boring, tapping etc. They are called in the form of a subroutine with a defined name and a parameter list. They all follow a different technological procedure and are therefore parameterized differently.

The drilling cycles can be modal, i.e. they are executed at the end of each block which contains motion command (see Section 8.1.6 or 9.3). Other cycles written by the user can also be called modally.

There are two types of parameters:

- Geometrical parameters and
- Machining parameters

The geometrical parameters are identical with all drilling cycles. They define the reference and retraction planes, the safety clearance and the absolute or relative final drilling depth. Geometrical parameters are assigned once during the first drilling cycle CYCLE82.

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

Call and return conditions

Drilling cycles are programmed independently of the actual axis names. The drilling position must be approached in the higher-level program before the cycle is called.

The required values for feedrate, spindle speed and direction of spindle rotation must be programmed in the part program if there are no defining parameters in the drilling cycle.

The G functions and the current data record active before the cycle was called remain active beyond the cycle.

Plane definition

In the case of drilling cycles, it is generally assumed that the current workpiece coordinate system in which the machining operation is to be performed is to be defined by selecting plane G17 and activating a programmable offset. The drilling axis is always the axis of this coordinate system which stands vertically to the current plane.

A tool length compensation must be selected before the cycle is called. Its effect is always perpendicular to the selected plane and remains active even after the end of the cycle.

In turning, the drilling axis is thus the Z axis. Drilling is performed to the end face of the workpiece.

![Diagram of drilling axis and tool length compensation](image)

Dwell time programming

The parameters for dwell times in the drilling cycles are always assigned to the F word and must therefore be assigned with values in seconds. Any deviations from this procedure must be expressly stated.
9.4.3 Drilling, centering – CYCLE81

Note
This standard cycle is not offered by the 802D bl.

Programming

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

Table 9-1 Parameters for CYCLE81

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
<td>real</td>
<td>Retraction plane (absolute)</td>
</tr>
<tr>
<td>RFP</td>
<td>real</td>
<td>Reference plane (absolute)</td>
</tr>
<tr>
<td>SDIS</td>
<td>real</td>
<td>Safety clearance (enter without sign)</td>
</tr>
<tr>
<td>DP</td>
<td>real</td>
<td>Final drilling depth (absolute)</td>
</tr>
<tr>
<td>DPR</td>
<td>real</td>
<td>Final drilling depth relative to the reference plane (enter without sign)</td>
</tr>
</tbody>
</table>

Function

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

Sequence

Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the final drilling depth at the feedrate programmed in the calling program (G1)
- Retraction to the retraction plane with G0

Explanation of the parameters

RFP and RTP (reference plane and retraction plane)

Normally, reference plane (RFP) and return plane (RTP) have different values. In the cycle, it is assumed that the retraction plane is ahead of the reference plane. This means that the distance from the retraction plane to the final drilling depth is larger than the distance from the reference plane to the final drilling depth.
SDIS (safety clearance)

The safety clearance (SDIS) acts with reference to the reference plane. This is brought forward by the safety clearance.
The direction in which the safety clearance acts is determined by the cycle automatically.

DP and DPR (final drilling depth)

The final drilling depth can be specified either absolute (DP) or relative (DPR) to the reference plane.

With relative specification, the cycle will calculate the resulting depth automatically using the positions of reference and retraction planes.

Note

If a value is entered both for DP and for DPR, the final drilling depth is derived from DPR. If this differs from the absolute depth programmed via DP, the message “Depth: Corresponding to value for relative depth” is output in the message line.

If the values for reference and retraction planes are identical, a relative depth specification is not permitted. The error message 61101 “Reference plane defined incorrectly” is output and the cycle is not executed. This error message is also output if the retraction plane is located after the reference plane, i.e. its distance to the final drilling depth is smaller.

Programming example: Drilling_centering

Using this program, you can produce 3 drill holes using the CYCLE81 drilling cycle, whereby this is called using different parameters. The drilling axis is always the Z axis.
### Cycles

#### 9.4 Drilling cycles

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10</td>
<td>G0 G17 G90 F200 S300 M3</td>
<td>Specification of the technological values</td>
</tr>
<tr>
<td>N20</td>
<td>D3 T3 Z110</td>
<td>Approaching the retraction plane</td>
</tr>
<tr>
<td>N30</td>
<td>X40 Y120</td>
<td>Approach of the first drilling position</td>
</tr>
<tr>
<td>N40</td>
<td>CYCLE81(110, 100, 2, 35)</td>
<td>Cycle call with absolute final drilling depth, safety clearance and incomplete parameter list</td>
</tr>
<tr>
<td>N50</td>
<td>Y30</td>
<td>Approach of next drill position</td>
</tr>
<tr>
<td>N60</td>
<td>CYCLE81(110, 102, , 35)</td>
<td>Cycle call without safety clearance</td>
</tr>
<tr>
<td>N70</td>
<td>G0 G90 F180 S300 M03</td>
<td>Specification of the technological values</td>
</tr>
<tr>
<td>N80</td>
<td>X90</td>
<td>Approach next position</td>
</tr>
<tr>
<td>N90</td>
<td>CYCLE81(110, 100, 2, , 65)</td>
<td>Cycle call with relative final drilling depth and safety clearance</td>
</tr>
<tr>
<td>N100</td>
<td>M2</td>
<td>End of program</td>
</tr>
</tbody>
</table>

Fig. 9-4

---

**Notes:**

- The table above summarizes the code lines and their corresponding actions in the context of drilling cycles.
- The diagram on the right side illustrates the coordinate system and the approach and cycle positions. The letters X, Y, Z, A, and B are used to denote specific points and directions.
- The cycle calls for various positions and parameters, ensuring safe and precise drilling operations.

---

This section from the manual provides detailed guidance on how to implement drilling cycles, emphasizing safety and precision in manufacturing processes.
9.4.4 Drilling, counterboring – CYCLE82

Programming

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

Parameters

Table 9-2 Parameters for CYCLE82

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
<td>real</td>
<td>Retraction plane (absolute)</td>
</tr>
<tr>
<td>RFP</td>
<td>real</td>
<td>Reference plane (absolute)</td>
</tr>
<tr>
<td>SDIS</td>
<td>real</td>
<td>Safety clearance (enter without sign)</td>
</tr>
<tr>
<td>DP</td>
<td>real</td>
<td>Final drilling depth (absolute)</td>
</tr>
<tr>
<td>DPR</td>
<td>real</td>
<td>Final drilling depth relative to the reference plane (enter without sign)</td>
</tr>
<tr>
<td>DTB</td>
<td>real</td>
<td>Dwell time at final drilling depth (chip breaking)</td>
</tr>
</tbody>
</table>

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. A dwell time can be allowed to elapse when the final drilling depth has been reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the final drilling depth with the feedrate (G1) programmed prior to the cycle call
- Dwell time at final drilling depth
- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81
DTB (dwell time)

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

**Note**

If a value is entered both for DP and for DPR, the final drilling depth is derived from DPR. If this differs from the absolute depth programmed via DP, the message "Depth: Corresponding to value for relative depth" is output in the message line.

If the values for reference and retraction planes are identical, a relative depth specification is not permitted. The error message 61101 "Reference plane defined incorrectly" is output and the cycle is not executed. This error message is also output if the retraction plane is located after the reference plane, i.e. its distance to the final drilling depth is smaller.

**Programming example: Boring_counterboring**

The program machines a single hole of a depth of 20 mm at position X0 with cycle CYCLE82.

The dwell time programmed is 3 s, the safety clearance in the drilling axis Z is 2,4 mm.

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10 G0 G90 G54 F2 S300 M3</td>
<td>Specification of the technological values</td>
</tr>
<tr>
<td>N20 D1 T6 Z50</td>
<td>Approaching the retraction plane</td>
</tr>
<tr>
<td>N30 G17 X0</td>
<td>Approaching the drill position</td>
</tr>
<tr>
<td>N40 CYCLE82(3, 1.1, 2.4, -20, , 3)</td>
<td>Cycle call with absolute final drilling depth and safety clearance</td>
</tr>
<tr>
<td>N50 M2</td>
<td>End of program</td>
</tr>
</tbody>
</table>
9.4.5 Deep hole drilling – CYCLE83

Programming

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

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
<td>real</td>
<td>Retraction plane (absolute)</td>
</tr>
<tr>
<td>RFP</td>
<td>real</td>
<td>Reference plane (absolute)</td>
</tr>
<tr>
<td>SDIS</td>
<td>real</td>
<td>Safety clearance (enter without sign)</td>
</tr>
<tr>
<td>DP</td>
<td>real</td>
<td>Final drilling depth (absolute)</td>
</tr>
<tr>
<td>DPR</td>
<td>real</td>
<td>Final drilling depth relative to the reference plane (enter without sign)</td>
</tr>
<tr>
<td>FDEP</td>
<td>real</td>
<td>First drilling depth (absolute)</td>
</tr>
<tr>
<td>FDPR</td>
<td>real</td>
<td>First drilling depth relative to the reference plane (enter without sign)</td>
</tr>
<tr>
<td>DAM</td>
<td>real</td>
<td>Amount of degression (enter without sign)</td>
</tr>
<tr>
<td>DTB</td>
<td>real</td>
<td>Dwell time at final drilling depth (chip breaking)</td>
</tr>
<tr>
<td>DTS</td>
<td>real</td>
<td>Dwell time at starting point and for swarf removal</td>
</tr>
<tr>
<td>FRF</td>
<td>real</td>
<td>Feedrate factor for the first drilling depth (enter without sign) Range of values: 0.001 ... 1</td>
</tr>
<tr>
<td>VARI</td>
<td>int</td>
<td>Machining type: Chip breaking=0 Swarf removal=1</td>
</tr>
</tbody>
</table>

Function

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

Deep hole drilling is performed with a depth infeed of a maximum definable depth executed several times, increasing gradually until the final drilling depth is reached.

The drill can either be retracted to the reference plane + safety clearance after every infeed depth for swarf removal or retracted in each case by 1 mm for chip breaking.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.
The cycle creates the following sequence of motions:

Deep hole drilling with swarf removal (VARI=1):

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the first drilling depth with G1, the feedrate for which is derived from the feedrate defined with the program call which is subject to parameter FRF (feedrate factor)
- Dwell time at final drilling depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance for swarf removal by using G0
- Dwell time at the starting point (parameter DTS)
- Approach of the drilling depth last reached, reduced by anticipation distance by using G0
- Traversing to the next drilling depth with G1 (sequence of motions is continued until the final drilling depth is reached)
- Retraction to the retraction plane with G0

Deep hole drilling with chip breaking (VARI=0):

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the first drilling depth with G1, the feedrate for which is derived from the feedrate defined with the program call which is subject to parameter FRF (feedrate factor)
- Dwell time at final drilling depth (parameter DTB)
- Retraction by 1 mm from the current drilling depth with G1 and the feedrate programmed in the calling program (for chip breaking)
- Traversing to the next drilling depth with G1 and the programmed feedrate (sequence of motions is continued until the final drilling depth is reached)
- Retraction to the retraction plane with G0
Explanation of the parameters

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

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

The intermediate drilling depth are calculated in the cycle on the basis of final drilling depth, first drilling depth and amount of degression as follows:

- In the first step, the depth parameterized with the first drilling depth is traversed as long as it does not exceed the total drilling depth.
- From the second drilling depth on, the drilling stroke is obtained by subtracting the amount of degression from the stroke of the last drilling depth, provided that the latter is greater than the programmed amount of degression.
- The next drilling strokes correspond to the amount of degression, as long as the remaining depth is greater than twice the amount of degression.
- The last two drilling strokes are divided and traversed equally and are therefore always greater than half of the amount of degression.
- If the value for the first drilling depth is incompatible with the total depth, the error message 61107 "First drilling depth defined incorrectly" is output and the cycle is not executed.

The parameter FDPR has the same effect in the cycle as the parameter DPR. If the values for the reference and retraction planes are identical, the first drilling depth can be defined as a relative value.

If the first drilling depth is programmed larger than the final drilling depth, the final drilling depth is never exceeded. The cycle will reduce the first drilling depth automatically as far as the final drilling depth is reached when drilling only once, and will therefore drill only once.
DTB (dwell time)

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

DTS (dwell time)

The dwell time at the starting point is only performed if VARI=1 (swarf removal).

FRF (feedrate factor)

With this parameter, you can enter a reduction factor for the active feedrate which only applies to the approach to the first drilling depth in the cycle.

VARI (machining type)

If parameter VARI=0 is set, the drill retracts 1 mm after reaching each drilling depth for chip breaking. If VARI=1 (for swarf removal), the drill traverses in each case to the reference plane brought forward by the safety clearance.

Note

The anticipation distance is calculated internally in the cycle as follows:

- If the drilling depth is 30 mm, the value of the anticipation distance is always 0.6 mm.
- For larger drilling depths, the formula drilling depth /50 is used (maximum value 7 mm).

Programming example – deep hole drilling

This program executes the cycle CYCLE83 at the position X0. The first drill hole is drilled with a dwell time zero and machining type chip breaking. The final drilling depth and the first drilling depth are entered as absolute values. The drilling axis is the Z axis.

<table>
<thead>
<tr>
<th>N10 G0 G54 G90 F5 S500 M4</th>
<th>Specification of the technological values</th>
</tr>
</thead>
<tbody>
<tr>
<td>N20 D1 T6 Z50</td>
<td>Approaching the retraction plane</td>
</tr>
<tr>
<td>N30 G17 X0</td>
<td>Approaching the drill position</td>
</tr>
<tr>
<td>N40 CYCLE83(3.3, 0, 0, -80, 0, -10, 0, 0, 0, 0, 1, 0)</td>
<td>Call of cycle; depth parameters with absolute values</td>
</tr>
<tr>
<td>N50 M2</td>
<td>End of program</td>
</tr>
</tbody>
</table>
9.4.6 Rigid tapping – CYCLE84

Programming

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

Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
<td>real</td>
<td>Retraction plane (absolute)</td>
</tr>
<tr>
<td>RFP</td>
<td>real</td>
<td>Reference plane (absolute)</td>
</tr>
<tr>
<td>SDIS</td>
<td>real</td>
<td>Safety clearance (enter without sign)</td>
</tr>
<tr>
<td>DP</td>
<td>real</td>
<td>Final drilling depth (absolute)</td>
</tr>
<tr>
<td>DPR</td>
<td>real</td>
<td>Final drilling depth relative to the reference plane (enter without sign)</td>
</tr>
<tr>
<td>DTB</td>
<td>real</td>
<td>Dwell time at thread depth (chip breaking)</td>
</tr>
<tr>
<td>SDAC</td>
<td>int</td>
<td>Direction of rotation after end of cycle Values: 3, 4 or 5 (for M3, M4 or M5)</td>
</tr>
<tr>
<td>MPIT</td>
<td>real</td>
<td>Pitch as thread size (signed) Range of values 3 (for M3) ... 48 (for M48); the sign determines the direction of rotation in the thread</td>
</tr>
<tr>
<td>PIT</td>
<td>real</td>
<td>Pitch as a value (signed) Range of values: 0.001 ... 2000.000 mm; the sign determines the direction of rotation in the thread</td>
</tr>
<tr>
<td>POSS</td>
<td>real</td>
<td>Spindle position for oriented spindle stop in the cycle (in degrees)</td>
</tr>
<tr>
<td>SST</td>
<td>real</td>
<td>Speed for tapping</td>
</tr>
<tr>
<td>SST1</td>
<td>real</td>
<td>Speed for retraction</td>
</tr>
</tbody>
</table>

Function

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

CYCLE84 can be used to perform rigid tapping operations.

Note

CYCLE84 can be used if the spindle to be used for the boring operation is technically able to be operated in the position–controlled spindle operation.

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

Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:
- Approach of the reference plane brought forward by the safety clearance by using G0
- Oriented spindle stop (value in the parameter POSS) and switching the spindle to axis mode
- Tapping to final drilling depth and speed SST
- Dwell time at thread depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance, speed SST1 and direction reversal
- Retraction to the retraction plane with G0; spindle mode is reinitiated by reprogramming the spindle speed active before the cycle was called and the direction of rotation programmed under SDAC

Explanation of the parameters

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

![Diagram](image)

Fig. 9-8

DTB (dwell time)
The dwell time must be programmed in seconds. When tapping blind holes, it is recommended to omit the dwell time.

SDAC (direction of rotation after end of cycle)
The direction of rotation of the spindle at the end of cycle is programmed under SDAC. The direction reversal when tapping is carried out automatically internally in the cycle.
MPIT and PIT (as a thread size and as a value)

The value for the thread pitch can be defined either as the thread size (for metric threads between M3 and M48 only) or as a value (distance from one thread turn to the next as a numerical value). The parameter not required in each case is omitted in the call or is assigned the value zero.

RH or LH threads are defined by the sign of the pitch parameters:

- positive value → RH (as for M3)
- negative value → LH (as for M4)

If the two thread pitch parameters have conflicting values, alarm 61001 "Thread pitch wrong" is generated by the cycle and cycle execution is aborted.

POSS (spindle position)

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

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

SST (speed)

Parameter SST contains the spindle speed for the tapping block.

SST1 (retraction speed)

The speed for retraction from the tapped hole is programmed under SST1 with G332. If this parameter is assigned the value zero, retraction is carried out at the speed programmed under SST.

Note

The direction of rotation when tapping in the cycle is always reversed automatically.

Programming example: Rigid tapping

Rigid tapping is carried out at position X0; the drilling axis is the Z axis. No dwell time is programmed; the depth is programmed as a relative value. The parameters for the direction of rotation and for the pitch must be assigned values. A metric thread M5 is tapped.

<table>
<thead>
<tr>
<th>N10</th>
<th>G0</th>
<th>G90</th>
<th>G54</th>
<th>T6</th>
<th>D1</th>
<th>Specification of the technological values</th>
</tr>
</thead>
<tbody>
<tr>
<td>N20</td>
<td>G17</td>
<td>X0</td>
<td>Z40</td>
<td></td>
<td></td>
<td>Approaching the drill position</td>
</tr>
</tbody>
</table>
9.4 Drilling cycles

9.4.7 Tapping with compensating chuck – CYCLE840

Programming

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

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
<td>real</td>
<td>Retraction plane (absolute)</td>
</tr>
<tr>
<td>RFP</td>
<td>real</td>
<td>Reference plane (absolute)</td>
</tr>
<tr>
<td>SDIS</td>
<td>real</td>
<td>Safety clearance (enter without sign)</td>
</tr>
<tr>
<td>DP</td>
<td>real</td>
<td>Final drilling depth (absolute)</td>
</tr>
<tr>
<td>DPR</td>
<td>real</td>
<td>Final drilling depth relative to the reference plane (enter without sign)</td>
</tr>
<tr>
<td>DTB</td>
<td>real</td>
<td>Dwell time at thread depth (chip breaking)</td>
</tr>
<tr>
<td>SDR</td>
<td>int</td>
<td>Direction of rotation for retraction Values: 0 (automatic reversal of the direction of rotation) 3 or 4 (for M3 or M4)</td>
</tr>
<tr>
<td>SDAC</td>
<td>int</td>
<td>Direction of rotation after end of cycle Values: 3, 4 or 5 (for M3, M4 or M5)</td>
</tr>
<tr>
<td>ENC</td>
<td>int</td>
<td>Tapping with/without encoder Values: 0 = with encoder 1 = without encoder</td>
</tr>
<tr>
<td>MPIT</td>
<td>real</td>
<td>Thread pitch as the thread size (signed) Range of values 3 (for M3) ... 48 (for M48)</td>
</tr>
<tr>
<td>PIT</td>
<td>real</td>
<td>Pitch as a value (signed) Range of values: 0.001 ... 2000.000 mm</td>
</tr>
</tbody>
</table>

Function

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

Using this cycle, you can perform tapping with compensating chuck

- without encoder and
- with encoder.
Sequence of operations: Tapping with compensating chuck without encoder

Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:
- Approach of the reference plane brought forward by the safety clearance by using G0
- Tapping to the final drilling depth
- Dwell time at tapping depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance
- Retraction to the retraction plane with G0

Sequence of operations: Tapping with compensating chuck with encoder

Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:
- Approach of the reference plane brought forward by the safety clearance by using G0
- Tapping to the final drilling depth
- Dwell time at thread depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance
- Retraction to the retraction plane with G0
Explanation of the parameters

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

DTB (dwell time)

The dwell time must be programmed in seconds. It is only effective in tapping without encoder.

SDR (direction of rotation for retraction)

SDR=0 must be set if the spindle direction is to reverse automatically.

If the machine data are defined such that no encoder is set (in this case, machine data MD30200 NUM_ENCS is 0), the parameter must be assigned the value 3 or 4 for the direction of rotation; otherwise, alarm 61202 "No spindle direction programmed" is output and the cycle is aborted.

SDAC (direction of rotation)

Because the cycle can also be called modally (see Section 9.3), it requires a direction of rotation for tapping further threaded holes. This is programmed in parameter SDAC and corresponds to the direction of rotation programmed before the first call in the higher-level program. If SDR=0, the value assigned to SDAC has no meaning in the cycle and can be omitted in the parameterization.

ENC (tapping)

If tapping is to be performed without encoder although an encoder exists, parameter ENC must be assigned value 1.
If, however, no encoder is installed and the parameter is assigned the value 0, it is ignored in the cycle.

**MPIT and PIT (as a thread size and as a value)**

The parameter for the spindle pitch is only relevant if tapping is performed with encoder. The cycle calculates the feedrate from the spindle speed and the pitch.

The value for the thread pitch can be defined either as the thread size (for metric threads between M3 and M48 only) or as a value (distance from one thread turn to the next as a numerical value). The parameter not required in each case is omitted in the call or is assigned the value zero.

If the two thread pitch parameters have conflicting values, alarm 61001 “Thread pitch wrong” is generated by the cycle and cycle execution is aborted.

**Further notes**

Depending on the settings in machine data MD30200 NUM_ENCS, the cycle selects whether tapping is to be performed with or without encoder.

The direction of rotation for the spindle must be programmed with M3 or M4.

In thread blocks with G63, the values of the feedrate override switch and spindle speed override switch are frozen to 100%.

A longer compensating chuck is usually required for tapping without encoder.

**Programming example: Tapping without encoder**

Tapping is carried out without encoder at position X0; the drilling axis is the Z axis. The parameters SDR and SDAC for the direction of rotation must be assigned; parameter ENC is assigned the value 1, the value for the depth is the absolute value. Pitch parameter PIT can be omitted. A compensating chuck is used in machining.

```
N10 G90 G0 G54 D1 T6 S500 M3  
Specification of the technological values
N20 G17 X0 Z60  
Approaching the drill position
N30 G1 F200  
Determination of the path feed
N40 CYCLE840(3, 0, , −15, 0, 1, 4, 3, 1, , )  
Cycle call, dwell time 1 s, direction of rotation for retraction M4, direction of rotation after cycle M3, no safety clearance
Parameters MPIT and PIT are omitted
N50 M2  
End of program
```
Example: Tapping with encoder

This program is used for tapping with encoder at position X0. The drilling axis is the Z axis. The pitch parameter must be defined, automatic reversal of the direction of rotation is programmed. A compensating chuck is used in machining.

<table>
<thead>
<tr>
<th>N10 G90 G0 G54 D1 T6 S500 M3</th>
<th>Specification of the technological values</th>
</tr>
</thead>
<tbody>
<tr>
<td>N20 G17 X0 Z60</td>
<td>Approaching the drill position</td>
</tr>
<tr>
<td>N30 G1 F200</td>
<td>Determination of the path feed</td>
</tr>
<tr>
<td>N40 CYCLE840(3, 0, -15, 0, 0, ,0, 3.5, )</td>
<td>Cycle call without safety clearance</td>
</tr>
<tr>
<td>N50 M2</td>
<td>End of program</td>
</tr>
</tbody>
</table>

9.4.8 Reaming1 (boring 1) – CYCLE85

Programming

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

Parameters

| Table 9-6 Parameters of CYCLE85 |
|-------------------------------|--------------------------------|
| RTP  | real | Retraction plane (absolute) |
| RFP  | real | Reference plane (absolute)  |
| SDIS | real | Safety clearance (enter without sign) |
| DP   | real | Final drilling depth (absolute) |
| DPR  | real | Final drilling depth relative to the reference plane (enter without sign) |
| DTB  | real | Dwell time at final drilling depth (chip breaking) |
| FFR  | real | Feedrate                   |
| RFF  | real | Retraction feedrate        |

Function

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

The inward and outward movement is performed at the feedrate assigned to FFR and RFF respectively.
**Sequence**

**Position reached prior to cycle start:**

The drilling position is the position in the two axes of the selected plane.

![Diagram of drilling positions](image)

**Fig. 9-11**

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the final drilling depth with G1 and at the feedrate programmed under the parameter FFR
- Dwell time at final drilling depth
- Retraction to the reference plane brought forward by the safety clearance with G1 and the retraction feedrate defined under the parameter RFF
- Retraction to the retraction plane with G0

**Explanation of the parameters**

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81
DTB (dwell time)

The dwell time to the final drilling depth is programmed under DTB in seconds.

FFR (feedrate)

The feedrate value programmed under FFR is active in drilling.

RFF (retraction feedrate)

The feedrate value programmed under RFF is active when retracting from the hole to the reference plane + safety clearance.

Programming example: First boring pass

CYCLE85 is called at Z70 X0. The drilling axis is the Z axis. The value for the final drilling depth in the cycle call is programmed as a relative value; no dwell time is programmed. The workpiece upper edge is at Z0.

<table>
<thead>
<tr>
<th>N10 G90 G0 S300 M3</th>
</tr>
</thead>
<tbody>
<tr>
<td>N20 T3 G17 G54 Z70 X0</td>
</tr>
<tr>
<td>N30 CYCLE85(10, 2, 2, , 25, , 300, 450)</td>
</tr>
<tr>
<td>N40 M2</td>
</tr>
</tbody>
</table>
9.4.9 Boring (boring 2) – CYCLE86

Note
This standard cycle is not offered by the 802D bl.

Programming

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

Parameters

Table 9-7 Parameters for CYCLE86

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
<td>real</td>
<td>Retraction plane (absolute)</td>
</tr>
<tr>
<td>RFP</td>
<td>real</td>
<td>Reference plane (absolute)</td>
</tr>
<tr>
<td>SDIS</td>
<td>real</td>
<td>Safety clearance (enter without sign)</td>
</tr>
<tr>
<td>DP</td>
<td>real</td>
<td>Final drilling depth (absolute)</td>
</tr>
<tr>
<td>DPR</td>
<td>real</td>
<td>Final drilling depth relative to the reference plane (enter without sign)</td>
</tr>
<tr>
<td>DTB</td>
<td>real</td>
<td>Dwell time at final drilling depth (chip breaking)</td>
</tr>
<tr>
<td>SDIR</td>
<td>int</td>
<td>Direction of rotation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Values: 3 (for M3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 (for M4)</td>
</tr>
<tr>
<td>RPA</td>
<td>real</td>
<td>Retraction path in the 1st axis of the plane (incremental, enter with sign)</td>
</tr>
<tr>
<td>RPO</td>
<td>real</td>
<td>Retraction path in the 2nd axis of the plane (incremental, enter with sign)</td>
</tr>
<tr>
<td>RPAP</td>
<td>real</td>
<td>Retraction path in the boring axis (incremental, enter with sign)</td>
</tr>
<tr>
<td>POSS</td>
<td>real</td>
<td>Spindle position for oriented spindle stop in the cycle (in degrees)</td>
</tr>
</tbody>
</table>

Function

The cycle supports the boring of holes with a boring bar.

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

With boring 2, oriented spindle stop is activated once the drilling depth has been reached. Then, the programmed retraction positions are approached in rapid traverse and, from there, the retraction plane.

Sequence

Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.
The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time to final drilling depth
- Oriented spindle stop at the spindle position programmed under POSS
- Traverse retraction path in up to three axes with G0
- Retraction in the boring axis to the reference plane brought forward by the safety clearance by using G0
- Retraction to the retraction plane with G0 (initial drilling position in both axes of the plane)

Explanation of the parameters

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

DTB (dwell time)

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

SDIR (direction of rotation)

With this parameter, you determine the direction of rotation with which boring is performed in the cycle. If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is generated and the cycle is not executed.

RPA (retraction path in the 1st axis)

Use this parameter to define a retraction movement in the 1st axis (abscissa), which is executed after the final drilling depth has been reached and oriented spindle stop has been performed.
RPO (retraction path in the 2nd axis)

Use this parameter to define a retraction movement in the 2nd axis (ordinate), which is executed after the final drilling depth has been reached and oriented spindle stop has been performed.

RPAP (retraction path in the boring axis)

Use this parameter to define a retraction movement in the boring axis, which is executed after the final drilling axis has been reached and oriented spindle stop has been performed.

POSS (spindle position)

Use POSS to program the spindle position for the oriented spindle stop in degrees which is performed after the final drilling depth has been reached.

Note

It is possible to stop the active spindle with orientation. The angular value is programmed using a transfer parameter.

Cycle CYCLE86 can be used if the spindle to be used for the boring operation is technically able to go into position-controlled spindle operation.

Programming example: Second boring pass

CYCLE86 is called at position X70 Y50 in the ZX plane. The drilling axis is the Z axis. The final drilling depth is programmed as an absolute value; no safety clearance is specified. The dwell time at the final drilling depth is 2 s. The workpiece upper edge is at Z110. In the cycle, the spindle is to rotate with M3 and to stop at 45 degrees.

Fig. 9-14
### 9.4 Drilling cycles

<table>
<thead>
<tr>
<th>G10 G0 G17 G90 F200 S300 M3</th>
<th>Specification of the technological values</th>
</tr>
</thead>
<tbody>
<tr>
<td>N20 T11 D1 Z112</td>
<td>Approaching the retraction plane</td>
</tr>
<tr>
<td>N30 X70 Y50</td>
<td>Approaching the drill position</td>
</tr>
<tr>
<td>N40 CYCLE86(112, 110, , 77, 0, 2, 3, –1, –1, 1, 45)</td>
<td>Cycle call with absolute drilling depth</td>
</tr>
<tr>
<td>N50 M2</td>
<td>End of program</td>
</tr>
</tbody>
</table>

#### 9.4.10 Boring with Stop 1 (boring 3) – CYCLE87

**Note**

This standard cycle is not offered by the 802D bl.

**Programming**

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

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
<td>real</td>
<td>Retraction plane (absolute)</td>
</tr>
<tr>
<td>RFP</td>
<td>real</td>
<td>Reference plane (absolute)</td>
</tr>
<tr>
<td>SDIS</td>
<td>real</td>
<td>Safety clearance (enter without sign)</td>
</tr>
<tr>
<td>DP</td>
<td>real</td>
<td>Final drilling depth (absolute)</td>
</tr>
<tr>
<td>DPR</td>
<td>real</td>
<td>Final drilling depth relative to the reference plane (enter without sign)</td>
</tr>
<tr>
<td>SDIR</td>
<td>int</td>
<td>Direction of rotation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Values: 3 (for M3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 (for M4)</td>
</tr>
</tbody>
</table>

**Function**

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

During boring 3, a spindle stop without orientation M5 is generated after reaching the final drilling depth, followed by a programmed stop M0. Pressing the NC START key continues the retraction movement at rapid traverse until the retraction plane is reached.

**Sequence**

**Position reached prior to cycle start:**

The drilling position is the position in the two axes of the selected plane.
The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Spindle stop with M5
- Press NC START
- Retraction to the retraction plane with G0

Explanation of the parameters

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

![Diagram of drilling cycle](image)

**SDIR (direction of rotation)**

This parameter determines the direction of rotation with which the drilling operation is carried out in the cycle.

If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is generated and the cycle is aborted.

**Programming example: Third boring**

CYCLE87 is called at position X70 Y50 in the XY plane. The drilling axis is the Z axis. The final drilling depth is specified as an absolute value. The safety clearance is 2 mm.
9.4 Drilling cycles

---

**Fig. 9-16**

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEF REAL DP, SDIS</td>
<td>Definition of the parameters</td>
</tr>
<tr>
<td>N10 DP=77 SDIS=2</td>
<td>Value assignments</td>
</tr>
<tr>
<td>N20 G0 G17 G90 F200 S300</td>
<td>Specification of the technological values</td>
</tr>
<tr>
<td>N30 D3 T3 Z113</td>
<td>Approaching the retraction plane</td>
</tr>
<tr>
<td>N40 X70 Y50</td>
<td>Approaching the drill position</td>
</tr>
<tr>
<td>N50 CYCLE87 (113, 110, SDIS, DP, , 3)</td>
<td>Cycle call with programmed direction of rotation of spindle M3</td>
</tr>
<tr>
<td>N60 M2</td>
<td>End of program</td>
</tr>
</tbody>
</table>
9.4.11 Drilling with stop 2 (boring 4) – CYCLE88

Programming

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

Parameters

Table 9-9 Parameters for CYCLE88

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
<td>real</td>
<td>Retraction plane (absolute)</td>
</tr>
<tr>
<td>RFP</td>
<td>real</td>
<td>Reference plane (absolute)</td>
</tr>
<tr>
<td>SDIS</td>
<td>real</td>
<td>Safety clearance (enter without sign)</td>
</tr>
<tr>
<td>DP</td>
<td>real</td>
<td>Final drilling depth (absolute)</td>
</tr>
<tr>
<td>DPR</td>
<td>real</td>
<td>Final drilling depth relative to the reference plane (enter without sign)</td>
</tr>
<tr>
<td>DTB</td>
<td>real</td>
<td>Dwell time at final drilling depth (chip breaking)</td>
</tr>
<tr>
<td>SDIR</td>
<td>int</td>
<td>Direction of rotation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Values: 3 (for M3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 (for M4)</td>
</tr>
</tbody>
</table>

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. During boring pass 4, a dwell time, a spindle stop without orientation M5 and a programmed stop M0 are generated when the final drilling depth is reached. Pressing the NC START key continues the retraction movement at rapid traverse until the retraction plane is reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time at final drilling depth
- Spindle and program stop with M5 M0. After program stop, press the NC START key.
- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81
### DTB (dwell time)

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

### SDIR (direction of rotation)

The programmed direction of rotation is active for the distance to be traversed to the final drilling depth.

If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is generated and the cycle is aborted.

### Programming example: Fourth boring pass

CYCLE88 is called at X0. The drilling axis is the Z axis. The safety clearance is programmed with 3 mm; the final drilling depth is specified relative to the reference plane. M4 is active in the cycle.

<table>
<thead>
<tr>
<th>N10</th>
<th>T1 S300 M3</th>
<th>Specification of the technological values</th>
</tr>
</thead>
<tbody>
<tr>
<td>N20</td>
<td>G17 G54 G90 F1 S450</td>
<td></td>
</tr>
<tr>
<td>N30</td>
<td>G0 X0 Z10</td>
<td>Approach drilling position</td>
</tr>
<tr>
<td>N40</td>
<td>CYCLE88 (5, 2, 3, , 72, 3, 4)</td>
<td>Cycle call with programmed direction of rotation of spindle M4</td>
</tr>
<tr>
<td>N50</td>
<td>M2</td>
<td>End of program</td>
</tr>
</tbody>
</table>
9.4.12 Reaming 2 (boring 5) – CYCLE89

Programming

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

Parameters

Table 9-10 Parameter CYCLE89

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP</td>
<td>real</td>
<td>Retraction plane (absolute)</td>
</tr>
<tr>
<td>RFP</td>
<td>real</td>
<td>Reference plane (absolute)</td>
</tr>
<tr>
<td>SDIS</td>
<td>real</td>
<td>Safety clearance (enter without sign)</td>
</tr>
<tr>
<td>DP</td>
<td>real</td>
<td>Final drilling depth (absolute)</td>
</tr>
<tr>
<td>DPR</td>
<td>real</td>
<td>Final drilling depth relative to the reference plane (enter without sign)</td>
</tr>
<tr>
<td>DTB</td>
<td>real</td>
<td>Dwell time at final drilling depth (chip breaking)</td>
</tr>
</tbody>
</table>

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. If the final drilling depth is reached, a dwell time may be programmed.

Sequence

Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time to final drilling depth
- Retraction up to the reference plane brought forward by the safety clearance using G1 and the same feedrate value
- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81
DTB (dwell time)

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

Programming example: Fifth boring

At X80 Y90 in the XY plane, the drilling cycle CYCLE89 is called with a safety clearance of 5 mm and specification of the final drilling depth as an absolute value. The drilling axis is the Z axis.

<table>
<thead>
<tr>
<th>DEF REAL RFP, RTP, DP, DTB</th>
<th>Definition of the parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFP=102 RTP=107 DP=72 DTB=3</td>
<td>Value assignments</td>
</tr>
<tr>
<td>N10 G90 G17 F100 S450 M4</td>
<td>Specification of the technological values</td>
</tr>
<tr>
<td>N20 G0 X80 Y90 Z107</td>
<td>Approach drilling position</td>
</tr>
</tbody>
</table>
9.4 Drilling cycles

9.4.13 Row of holes – HOLES1

Programming

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

Parameters

Table 9-11 Parameters of HOLES1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPCA</td>
<td>real</td>
<td>1st axis of the plane (abscissa) of a reference point on the straight line (absolute)</td>
</tr>
<tr>
<td>SPCO</td>
<td>real</td>
<td>2nd axis of the plane (ordinate) of this reference point (absolute)</td>
</tr>
<tr>
<td>STA1</td>
<td>real</td>
<td>Angle to the 1st axis of the plane (abscissa) (-180 &lt; STA1 &lt; 180) degrees</td>
</tr>
<tr>
<td>FDIS</td>
<td>real</td>
<td>Distance from the first hole to the reference point (enter without sign)</td>
</tr>
<tr>
<td>DBH</td>
<td>real</td>
<td>Distance between the holes (enter without sign)</td>
</tr>
<tr>
<td>NUM</td>
<td>int</td>
<td>Number of holes</td>
</tr>
</tbody>
</table>

Function

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

Sequence

To avoid unnecessary travel, the cycle calculates whether the row of holes is machined starting from the first hole or the last hole from the actual position of the plane axes and the geometry of the row of holes. The drilling positions are then approached one after the other at rapid traverse.

Fig. 9-20
Explanation of the parameters

![Diagram of drilling cycle parameters](image)

**SPCA and SPCO (reference point of 1st axis of the plane and 2nd axis of the plane)**

One point along the straight line of the row of holes is defined as the reference point for determining the distances between the holes. The distance to the first hole FDIS is defined from this point.

**STA1 (angle)**

The straight line can be in any position in the plane. It is specified both by the point defined by SPCA and SPCO and by the angle contained by the straight line with the 1st axis of the plane of the workpiece coordinate system that is active when the cycle is called. The angle is entered under STA1 in degrees.

**FDIS and DBH (distance)**

The distance of the first hole and the reference point defined under SPCA and SPCO is specified with FDIS. The parameter DBH contains the distance between any two holes.

**NUM (number)**

The NUM parameter is used to define the number of holes.
Programming example: Row of holes

Use this program to machine a row of holes consisting of 5 threaded holes arranged parallel to the Z axis of the ZX plane and which have a distance of 20 mm one to another. The starting point of the row of holes is at Z20 and X30 whereby the first hole has a distance of 10 mm from this point. The geometry of the row of holes is described by the cycle HOLES1.

First, drilling is carried out using CYCLE82, and then tapping is performed using CYCLE84 (tapping without compensating chuck). The holes are 80 mm in depth (difference between reference plane and final drilling depth).

![Diagram of row of holes](image_url)

Fig. 9-22

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10</td>
<td>G90 F30 S500 M3 T10</td>
<td>Specification of the technological values for the machining step</td>
</tr>
<tr>
<td>N20</td>
<td>G17 G90 X20 Z105 Y30</td>
<td>Approach starting position</td>
</tr>
<tr>
<td>N30</td>
<td>MCALL CYCLE82(105, 102, 2, 22, 0, 1)</td>
<td>Modal call of the drilling cycle</td>
</tr>
<tr>
<td>N40</td>
<td>HOLES1(20, 30, 0, 10, 20, 5)</td>
<td>Call of row-of-holes cycle; the cycle starts with the first hole; only the drill positions are approached in this cycle</td>
</tr>
<tr>
<td>N50</td>
<td>MCALL</td>
<td>Deselect modal call</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>Tool change</td>
</tr>
<tr>
<td>N60</td>
<td>G90 G0 X30 Z110 Y105</td>
<td>Traverse to position next to the 5th hole</td>
</tr>
<tr>
<td>N70</td>
<td>MCALL CYCLE84(105, 102, 2, 22, 0, 3, 4.2, 300, )</td>
<td>Modal call of the tapping cycle</td>
</tr>
<tr>
<td>N80</td>
<td>HOLES1(20, 30, 0, 10, 20, 5)</td>
<td>Call of the row-of-holes cycle started with the 5th hole in the row</td>
</tr>
<tr>
<td>N90</td>
<td>MCALL</td>
<td>Deselect modal call</td>
</tr>
<tr>
<td>N100</td>
<td>M2</td>
<td>End of program</td>
</tr>
</tbody>
</table>
**Programming example: Grid of holes**

Use this program to machine a grid of holes consisting of 5 rows with 5 holes each, which are arranged in the XY plane, with a distance of 10 mm between them. The starting point of the grid is at X30 Y20. The example uses R parameters as transfer parameters for the cycle.

| R10=102 | Reference plane |
| R11=105 | Retraction plane |
| R12=2  | Safety clearance |
| R13=75 | Drilling depth |
| R14=30 | Reference point: Row of holes of the 1st axis of the plane |
| R15=20 | Reference point: Row of holes of the 2nd axis of the plane |
| R16=0  | Starting angle |
| R17=10 | Distance of the 1st hole to the ref. point |
| R18=10 | Distance between the holes |
| R19=5  | Number of holes per row |
| R20=5  | Number of rows |
| R21=0  | Count of rows |
| R22=10 | Distance between the rows |

**Terms:**
- **N10 G90 F300 S500 M3 T10 D1**: Specification of the technological values
- **N20 G17 G0 X=R14 Y=R15 Z105**: Approach starting position
- **N30 MCALL CYCLE82(R11, R10, R12, R13, 0, 1)**: Modal call of the drilling cycle
- **N40 LABEL1**: Call of the row-of-holes cycle
- **N41 HOLES1(R14, R15, R16, R17, R18, R19)**: Call of the row-of-holes cycle
- **N50 R15=R15+R22**: Calculate y value for the next line
- **N60 R21=R21+1**: Increment line counter
- **N70 IF R21<R20 GOTO LABEL1**: Return to LABEL1 if the condition is fulfilled
- **N80 MCALL**: Deselect modal call
- **N90 G90 G0 X30 Y20 Z105**: Approach starting position
- **N100 M2**: End of program

---

**Fig. 9-23**
9.4.14 Circle of holes – HOLES2

Programming

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

Parameters

Table 9-12 Parameters for HOLES2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPA</td>
<td>real</td>
<td>Center point of circle of holes (absolute), 1st axis of the plane</td>
</tr>
<tr>
<td>CPO</td>
<td>real</td>
<td>Center point of circle of holes (absolute), 2nd axis of the plane</td>
</tr>
<tr>
<td>RAD</td>
<td>real</td>
<td>Radius of circle of holes (enter without sign)</td>
</tr>
</tbody>
</table>
| STA1      | real | Starting angle  
|           |      | Range of values: $-180 < STA1 <= 180$ degrees |
| INDA      | real | Incrementing angle |
| NUM       | int  | Number of holes |

Function

Use this circle to machine a circle of holes. The machining plane must be defined before the cycle is called.

The type of hole is determined by the drilling hole cycle that has already been called modally.

Sequence

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

CPA, CPO and RAD (center point position and radius)

The position of the circle of holes in the machining plane is defined via center point (parameters CPA and CPO) and radius (parameter RAD). Only positive values are permitted for the radius.

STA1 and INDA (starting and incremental angle)

These parameters define the arrangement of the holes on the circle of holes.
Parameter STA1 specifies the angle of rotation between the positive direction of the 1st axis (abscissa) in the workpiece coordinate system active before the cycle was called and the first hole. Parameter INDA contains the angle of rotation from one hole to the next.

If parameter INDA is assigned the value zero, the indexing angle is calculated internally from the number of holes which are positioned equally in a circle.

**NUM (number)**

Parameter NUM defines the number of holes.

**Programming example: Circle of holes**

The program uses CYCLE82 to produce 4 holes having a depth of 30 mm. The final drilling depth is specified as a relative value to the reference plane. The circle is defined by the center point X70 Y60 and the radius 42 mm in the XY plane. The starting angle is 33 degrees. The safety clearance along the drilling axis Z is 2 mm.

![Diagram of circle of holes](image)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10</td>
<td>G90 F140 S170 M3 T10 D1</td>
</tr>
<tr>
<td>N20</td>
<td>G17 G0 X50 Y45 Z2</td>
</tr>
<tr>
<td>N30</td>
<td>MCALL CYCLE82(2, 0, 2, , 30, 0)</td>
</tr>
<tr>
<td>N40</td>
<td>HOLES2 (70, 60, 42, 33, 0, 4)</td>
</tr>
<tr>
<td>N50</td>
<td>MCALL</td>
</tr>
<tr>
<td>N60</td>
<td>M2</td>
</tr>
</tbody>
</table>
9.5 Turning cycles

9.5.1 Prerequisites

The turning cycles are part of the configuration file setup_T.cnf which is loaded into the user memory of the control system.

Call and return conditions

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

Plane definition

The machining plane must be defined prior to the cycle call. With turning, it is usually the G18 (ZX plane). The two axes of the current plane in turning will be called in the following longitudinal axis (first axis of this plane) and transverse axis (second axis of this plane).

In the turning cycles, with diameter programming active, the second axis is taken into account as the transverse axis in all cases (see Programming Guide).

Contour monitoring referred to the tool clearance angle

Certain turning cycles in which traversing motions with relief cutting are generated monitor the clearance angle of the active tool for a possible contour violation. This angle is entered in the tool compensation as a value (in the D offset under the parameter DP24). A value between 1 and 90 degrees (0=no monitoring) without sign must be specified for the angle.
When entering the tool clearance angle, note that this depends on the machining type 'longitudinal' or 'face'. If you want to use one tool for longitudinal and face machining, two tool compensations must be used in the case of different tool clearance angles.

The cycle will check whether or not the programmed contour can be machined using the selected tool.

If the machining is not possible using this tool, then

- the cycle will abort and an error message is output (in stock removal) or
- the contour is continued to be machined and a message is output (with undercut cycles).

In this case, the contour is determined by the cutting edge geometry.

If the tool clearance angle is specified with zero in the tool compensation, this monitoring will not be performed. For details on the reactions, please refer to the individual cycles.
9.5 Turning cycles

9.5.2 Grooving – CYCLE93

Programming

CYCLE93(SPD, SPL, WIDG, DIAG, STA1, ANG1, ANG2, RCO1, RCO2, RCI1, RCI2, FAL1, FAL2, IDEP, DTB, VARI)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPD</td>
<td>real</td>
<td>Starting point in the transverse axis</td>
</tr>
<tr>
<td>SPL</td>
<td>real</td>
<td>Starting point in the longitudinal axis</td>
</tr>
<tr>
<td>WIDG</td>
<td>real</td>
<td>Groove width (enter without sign)</td>
</tr>
<tr>
<td>DIAG</td>
<td>real</td>
<td>Groove depth (enter without sign)</td>
</tr>
<tr>
<td>STA1</td>
<td>real</td>
<td>Angle between contour and longitudinal axis</td>
</tr>
<tr>
<td>ANG1</td>
<td>real</td>
<td>Flank angle 1: on the groove side determined by the starting point (enter without sign)</td>
</tr>
<tr>
<td>ANG2</td>
<td>real</td>
<td>Flank angle 2: on the other side (enter without sign)</td>
</tr>
<tr>
<td>RCO1</td>
<td>real</td>
<td>Radius/chamfer 1, externally: on the side determined by the starting point</td>
</tr>
<tr>
<td>RCO2</td>
<td>real</td>
<td>Radius/chamfer 2, externally:</td>
</tr>
<tr>
<td>RCI1</td>
<td>real</td>
<td>Radius/chamfer 1, internally: on the starting point side</td>
</tr>
<tr>
<td>RCI2</td>
<td>real</td>
<td>Radius/chamfer 2, internally:</td>
</tr>
<tr>
<td>FAL1</td>
<td>real</td>
<td>Finishing allowance at the recess base</td>
</tr>
<tr>
<td>FAL2</td>
<td>real</td>
<td>Finishing allowance at the flanks</td>
</tr>
<tr>
<td>IDEP</td>
<td>real</td>
<td>Infeed depth (enter without sign)</td>
</tr>
<tr>
<td>DTB</td>
<td>real</td>
<td>Dwell time at recess base</td>
</tr>
<tr>
<td>VARI</td>
<td>int</td>
<td>Machining type</td>
</tr>
</tbody>
</table>

Function

The grooving cycle can be used to carry out symmetrical and asymmetrical recesses for longitudinal and face machining at any straight contour elements. External and internal grooves can be produced.

Sequence

The infeed in the depth (towards the groove base) and in the width (from groove to groove) are calculated in the cycle internally and distributed equally with the maximum possible value.
When grooving at oblique faces, the tool will traverse from one groove to the next on the shortest path, i.e. parallel to the cone at which the groove is machined. During this process, a safety clearance to the contour is calculated internally in the cycle.

1st step

Paraxial roughing in individual infeed steps up to the base
The tool will retract for chip breaking after each infeed.

Fig. 9-31

2nd step

The groove is machined vertically to the infeed direction in one or several steps whereby each step, in turn, is divided according to the infeed depth. From the second cut along the groove width onwards, the tool will retract by 1 mm each before retraction.

Fig. 9-32
3rd step

Machining of the flanks in one step if angles are programmed under ANG1 or ANG2. Infeed along the groove width is carried out in several steps if the flank width is larger.

Fig. 9-33

4th step

Stock removal of the finishing allowance parallel to the contour from the edge to the groove center. During this operation, the tool radius compensation is selected and deselected by the cycle automatically.

Fig. 9-34
Explanation of the parameters

SPD and SPL (starting point)

These coordinates can be used to define the starting point of a groove starting from which the form is calculated in the cycle. The cycle starting point approached at the beginning is determined by the cycle itself. In the case of an external groove, first the tool will traverse in the direction of the longitudinal axis, and in the case of an internal groove, first in the direction of the transverse axis.

Grooves at bent contour elements can be realized differently. Depending on the form and radius of the bend, either a paraxial straight line can be laid over the maximum of the bend or a tangential oblique line can be created in a point of the edge points of the groove.

Radii and chamfers at the groove edge make sense with bent contours only if the appropriate edge point is on the straight line specified for the cycle.

![Diagram of groove parameters](image)

**Fig. 9-35**

WIDG and DIAG (groove width and groove depth)

The parameters groove width (WIDG) and groove depth (DIAG) are used to define the form of the groove. In its calculation, the cycle always assumes the point programmed under SPD and SPL.

If the groove width is larger than that of the active tool, the width is removed in several steps. When doing so, the whole width is distributed by the cycle equally. The maximum infeed is 95% of the tool width after deduction of the cutting edge radii. This provides a cutting overlap.

If the programmed groove width is smaller than the real tool width, the error message 61602 "Tool width defined incorrectly" and machining is aborted. The alarm will also appear if a cutting edge width equal to zero is detected in the cycle.
STA1 (angle)

Use the parameter STA1 to program the angle of the oblique line at which the groove is to be machined. The angle can assume values between 0 and 180 degrees and always refers to the longitudinal axis.

ANG1 and ANG2 (flank angle)

Asymmetric grooves can be described by flank angles specified separately. The angles can assume values between 0 and 89.999 degrees.

RCO1, RCO2 and RCI1, RCI2 (radius/chamfer)

The form of the groove can be modified by entering radii/chamfers at the edge or on the root. **Note that radii are entered with positive sign and chamfers with negative sign.**

How the programmed chamfers are taken into account is specified in dependence of the tens digit of the VARI parameter.

- If VARI<10 (tens digit=0) Chamfers with CHF=...
- If VARI>10 Chamfers with CHR programming

(CHF / CHR see Section 8.1.6)

FAL1 and FAL2 (finishing allowance)

It is possible to program separate finishing allowances for groove base and flanks. During roughing, stock removal is carried out up to these finishing allowances. Then a cut is carried out parallel to the contour with the same tool along the final contour.
IDEP (infeed depth)

You can divide the paraxial grooving into several depth infeeds by programming an infeed depth. After each infeed, the tool is retracted by 1 mm for chip breaking.

The parameter IDEP must be programmed in all cases.

DTB (dwell time)

The dwell time at the groove base should be selected such that at least one spindle revolution is carried out. It is programmed in seconds.

VARI (machining type)

The machining type of the groove is defined with the units digit of the VARI parameter. It can assume the values indicated in the illustration.

The tens digit of parameter VARI determines how the chamfers are taken into account.

VARI 1...8: The chamfers will be taken into account as CHF.
VARI 11...18: The chamfers will be taken into account as CHR.
If the parameter has a different value, the cycle will abort with alarm 61002 "Machining type defined incorrectly".

The cycle carries out a contour monitoring such that a reasonable groove contour results. This is not the case if the radii/chamfers come into contact or intersect at the groove base or if you try to carry out a face grooving operation at a contour segment located parallel to the longitudinal axis. In such cases, the cycle will abort with alarm 61603 "Groove form defined incorrectly".

**Further notes**

Before calling the grooving cycle, a double-edged tool must be enabled. The offset values for the two cutting edges must be stored in two successive D numbers of the tool whereby the first of which must be activated prior to the first cycle call. The cycle itself defines for which machining step it will use which of the two tool compensation values and will also enable them automatically. After completion of the cycle, the tool compensation number programmed prior to the cycle call is active again. If no D number is programmed for a tool compensation when the cycle is called, the execution of the cycle is aborted with the alarm 61000 "No tool compensation active".

**Programming example: Grooving**

This program is used to produce a groove externally at an oblique line in the longitudinal direction.

The starting point is on the right-hand side at X35 Z60.

The cycle will use the tool compensations D1 and D2 of tool T5. The cutting tool must be defined accordingly.
Fig. 9-39

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10</td>
<td>G0 G90 Z65 X50 T5 D1 S400 M3</td>
<td>Starting point prior to cycle start</td>
</tr>
<tr>
<td>N20</td>
<td>G95 F0.2</td>
<td>Specification of the technological values</td>
</tr>
<tr>
<td>N30</td>
<td>CYCLE93(35, 60, 30, 25, 5, 10, 20, 0, 0, -2, -2, 1, 1, 10, 1, 5)</td>
<td>Cycle call</td>
</tr>
<tr>
<td>N40</td>
<td>G0 G90 X50 Z65</td>
<td>Next position</td>
</tr>
<tr>
<td>N50</td>
<td>M02</td>
<td>End of program</td>
</tr>
</tbody>
</table>
9.5.3 Undercut (forms E and F to DIN) – CYCLE94

Programming

CYCLE94(SPD, SPL, FORM)

Parameters

Table 9-14 Parameters of CYCLE94

<table>
<thead>
<tr>
<th>SPD</th>
<th>real</th>
<th>Starting point in the transversal axis (enter without sign)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPL</td>
<td>real</td>
<td>Starting point of the tool compensation in the longitudinal axis (enter without sign)</td>
</tr>
<tr>
<td>FORM</td>
<td>char</td>
<td>Definition of the form</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>E (for form E)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F (for form F)</td>
</tr>
</tbody>
</table>

Function

Using this cycle, you can program undercuts to DIN509 of the forms E and F for usual stress and with a finished part diameter >3 mm.

Sequence

Position reached prior to cycle start:

The starting position can be any position from which the undercut can be approached without collision.
The cycle creates the following sequence of motions:

- Approach of the starting point determined in the cycle by using G0
- Selection of the cutter radius compensation according to the active tool point direction and traveling along the undercut contour at the feedrate programmed prior to the cycle call
- Retraction to the starting point with G0 and deselection of the cutter radius compensation with G40

Explanation of the parameters

**SPD and SPL (starting point)**

Use the parameter SPD to specify the finished part diameter for the undercut. The SPL parameter defines the finished dimension in the longitudinal axis.

If a final diameter <3 mm results for the value programmed for SPD, the cycle aborts with alarm 61601 "Finished part diameter too small".

![Fig. 9-41](image)

**FORM (definition)**

Form E and form F are fixed in DIN509 and must be defined using this parameter.

If the parameter has a value other than E or F, the cycle aborts and creates alarm 61609 "Form defined incorrectly".
The tool point direction is determined by the cycle automatically from the active tool compensation. The cycle can operate with the tool point directions 1 ... 4.

If the cycle detects either of the tool point directions 5 ... 9, the alarm 61608 "Wrong tool point direction programmed" and the cycle is aborted.

The cycle determines its starting point automatically. This is by 2 mm away from the end diameter and by 10 mm away from the finishing dimension in the longitudinal axis. The position of this starting point referred to the programmed coordinate values is determined by the tool point direction of the active tool.

The clearance angle of the active tool is monitored in the cycle if an appropriate value is specified in the appropriate parameter of the tool compensation. If it turns out that the form of the undercut cannot be machined using the selected tool since its tool clearance angle is too small, the message "Changed form of undercut" is displayed on the control system. The machining, however, is continued.
Further notes

Before calling the cycle, a tool compensation must be activated. Otherwise, the cycle is aborted after alarm 61000 "No tool compensation active" has been output.

Programming example: Undercut_form_E

This program can be used to program an undercut of form E.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>N10</td>
<td>T1 D1 S300 M3 G95 F0.3</td>
</tr>
<tr>
<td>N20</td>
<td>G0 G90 Z100 X50</td>
</tr>
<tr>
<td>N30</td>
<td>CYCLE94(20, 60, &quot;E&quot;)</td>
</tr>
<tr>
<td>N40</td>
<td>G90 G0 Z100 X50</td>
</tr>
<tr>
<td>N50</td>
<td>M02</td>
</tr>
</tbody>
</table>
9.5.4 Machining with undercut – CYCLE95

Programming

CYCLE95 (NPP, MID, FALZ, FALX, FAL, FF1, FF2, FF3, VARI, DT, DAM, _VRT)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPP</td>
<td>string</td>
<td>Name of contour subroutine</td>
</tr>
<tr>
<td>MID</td>
<td>real</td>
<td>Infeed depth (enter without sign)</td>
</tr>
<tr>
<td>FALZ</td>
<td>real</td>
<td>Finishing allowance in the longitudinal axis (enter without sign)</td>
</tr>
<tr>
<td>FALX</td>
<td>real</td>
<td>Finishing allowance in the transverse axis (enter without sign)</td>
</tr>
<tr>
<td>FAL</td>
<td>real</td>
<td>Finishing allowance according to the contour (enter without sign)</td>
</tr>
<tr>
<td>FF1</td>
<td>real</td>
<td>Feedrate for roughing without undercut</td>
</tr>
<tr>
<td>FF2</td>
<td>real</td>
<td>Feedrate for insertion into relief cut elements</td>
</tr>
<tr>
<td>FF3</td>
<td>real</td>
<td>Feedrate for finishing</td>
</tr>
<tr>
<td>VARI</td>
<td>real</td>
<td>Machining type [Range of values: 1 ... 12]</td>
</tr>
<tr>
<td>DT</td>
<td>real</td>
<td>Dwell time for chip breaking when roughing</td>
</tr>
<tr>
<td>DAM</td>
<td>real</td>
<td>Path length after which each roughing step is interrupted for chip breaking</td>
</tr>
<tr>
<td>_VRT</td>
<td>real</td>
<td>Retraction travel from the contour when roughing, incremental (enter without sign)</td>
</tr>
</tbody>
</table>

Function

Using the rough turning cycle, you can produce a contour, which has been programmed in a subroutine, from a blank by paraxial stock removal. The contour may contain relief cut elements. It is possible to machine contours using longitudinal and face machining, both externally and internally. The technology can be freely selected (roughing, finishing, complete machining). When roughing the contour, paraxial cuts from the maximum programmed infeed depth are programmed and burrs are also removed parallel to the contour after an intersection point with the contour has been reached. Roughing is carried out up to the programmed finishing allowance.

Finishing is carried out in the same direction as roughing. **The tool radius compensation is selected and deselected by the cycle automatically.**
Sequence

Position reached prior to cycle start:

The starting position is any position from which the contour starting point can be approached without collision.

The cycle creates the following sequence of motions:

The cycle starting point is calculated internally and approached with G0 in both axes at the same time.

Roughing without relief cut elements:

- The paraxial infeed to the current depth is calculated internally and approached with G0.
- Approach of paraxial roughing intersection point with G1 and at feedrate FF1.
- Rounding parallel to the contour along the contour + finishing allowance with G1/G2/G3 and FF1.
- Retraction by the amount programmed under _VRT in each axis and retraction with G0.
- This sequence is repeated until the total depth of the machining step is reached.
- When roughing without relief cut elements, retraction to the cycle starting point is carried out axis by axis.
Roughing the relief cut elements:

- Approach of the starting point for the next relief cut axis by axis with G0. When doing so, an additional cycle–internal safety clearance is observed.
- Infeed along the contour + finishing allowance with G1/G2/G3 and FF2.
- Approach of paraxial roughing intersection point with G1 and at feedrate FF1.
- Rounding along the contour, retraction and return are carried out as with the first machining step.
- If there are further relief cut elements, this sequence is repeated for each relief cut.
**Finishing:**

- The cycle starting point is approached axis by axis with G0.
- The contour starting point is approached with G0 in both axes at the same time.
- Finishing along the contour with G1/G2/G3 and FF3
- Retraction to the starting point with both axes and G0

**Explanation of the parameters**

**NPP (name)**

This parameter is used to specify the contour name.

1. The contour can be defined as a subroutine:
   NPP=name of subroutine
   The name of the contour subroutine is subject to all name conventions described in the Programming Guide.
   Input:
   - The subroutine already exists --> enter name, continue
   - The subroutine does not yet exist --> enter name and press softkey "new file". A program (main program) with the entered name is created and the program will jump to the contour editor.
   Use the "Technol. mask" softkey to confirm your input and return to the cycle help screenform.

2. The contour can also be a section of the calling program:
   NPP=name of starting label: name of end label
   Input:
   - Contour is already described --> name of starting label: name of end label
   - Contour is not yet described --> enter name of starting label and press softkey "contour append". Starting and end label are automatically created from the name you have entered; then the program will jump to the contour editor.
   Use the "Technol. mask" softkey to confirm your input and return to the cycle help screenform.
Examples:

<table>
<thead>
<tr>
<th>NPP=KONTUR_1</th>
<th>The rough turning contour is the complete program Kontur_1.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPP=ANFANG:ENDE</td>
<td>The rough turning contour is defined as a section in the calling program, which starts from the block containing label ANFANG to the block containing label ENDE.</td>
</tr>
</tbody>
</table>

**MID (infeed depth)**

Parameter MID is used to define the maximum possible infeed depth for the roughing process.

The cycle will automatically calculate the current infeed depth used for roughing.

With contours containing relief cut elements, the roughing process is divided by the cycle into individual roughing sections. The cycle calculates a new current infeed depth for each roughing section. This infeed depth is always between the programmed infeed depth and the half of its value. The number of required roughing steps is determined on the basis of the total depth of a roughing section and of the programmed maximum infeed depth to which the total depth to be machined is distributed equally. This provides optimum cutting conditions. For roughing this contour, the machining steps shown in the illustration result.
Example of calculating the current infeed depth:

Machining step 1 has a total depth of 39 mm. Therefore, 8 roughing steps are still required with a maximum infeed depth of 5 mm. These are carried out with an infeed of 4.875 mm.

In machining step 2, 8 roughing steps, too, are carried out with an infeed of 4.5 mm each (total difference 36 mm).

In machining step 3, two roughing passes are carried out with a current infeed of 3.5 (total difference 7 mm).

**FAL, FALZ and FALX (finishing allowance)**

A finishing allowance for roughing can be specified either using the parameters FALZ and FALX if you want to specify different finishing allowances axis-specifically or via the parameter FAL for a finishing allowance that follows the contour. In this case, this value is taken into account in both axes as a finishing allowance.

No plausibility check is carried out for the programmed values. In other words: If all three parameters are assigned values, all these finishing allowances are taken into account by the cycle. It is, however, reasonable to decide either on the one or other form of definition of a finishing allowance.

Roughing is always carried out up to these finishing allowances. The resulting residual corner is also removed parallel to the contour after each paraxial roughing process immediately so that no additional residual corner cut is required after completion of roughing. If no finishing allowances are programmed, stock is removed when roughing up to the final contour.

**FF1, FF2 and FF3 (feedrate)**

For different machining steps, different feedrates can be specified, as shown in Fig. 9-50.
### VARI (machining type)

#### Table 9-16 Type of machining

<table>
<thead>
<tr>
<th>Value</th>
<th>Longitudinal/face</th>
<th>Ext./Int.</th>
<th>Roughing/finishing/complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>L</td>
<td>A</td>
<td>Roughing</td>
</tr>
<tr>
<td>2</td>
<td>P</td>
<td>A</td>
<td>Roughing</td>
</tr>
<tr>
<td>3</td>
<td>L</td>
<td>I</td>
<td>Roughing</td>
</tr>
<tr>
<td>4</td>
<td>P</td>
<td>I</td>
<td>Roughing</td>
</tr>
<tr>
<td>5</td>
<td>L</td>
<td>A</td>
<td>Finishing</td>
</tr>
<tr>
<td>6</td>
<td>P</td>
<td>A</td>
<td>Finishing</td>
</tr>
<tr>
<td>7</td>
<td>L</td>
<td>I</td>
<td>Finishing</td>
</tr>
<tr>
<td>8</td>
<td>P</td>
<td>I</td>
<td>Finishing</td>
</tr>
<tr>
<td>9</td>
<td>L</td>
<td>A</td>
<td>Complete machining</td>
</tr>
<tr>
<td>10</td>
<td>P</td>
<td>A</td>
<td>Complete machining</td>
</tr>
<tr>
<td>11</td>
<td>L</td>
<td>I</td>
<td>Complete machining</td>
</tr>
<tr>
<td>12</td>
<td>P</td>
<td>I</td>
<td>Complete machining</td>
</tr>
</tbody>
</table>

In longitudinal machining, the infeed is always carried out along the transversal axis, and in face machining – along the longitudinal axis.

External machining means that the infeed is carried out in the direction of the negative axis. With internal machining, the infeed is carried out in the direction of the positive axis.
The parameter VARI is subjected to a plausibility check. If its value is not in the range 1 ... 12 when the cycle is called, the cycle is aborted with alarm 61002 "Machining type defined incorrectly".

DT and DAM (dwell time and path length)

These parameters can be used to achieve an interruption of the individual roughing steps after certain distances traversed in order to carry out chip breaking. These parameters are only relevant for roughing. The parameter DAM is used to define the maximum distance after which chip breaking is to be carried out. In DT, an appropriate dwell time (in seconds) can be programmed which is carried out at each of the cut interruption points. If no distance is specified for the cut interruption (DAM=0), uninterrupted roughing steps without dwell times are created.
_VRT (retraction travel)

Parameter _VRT can be used to program the amount by which the tool is retracted in both axes when roughing.

If _VRT=0 (parameter not programmed), the tool will retract by 1 mm.

Further notes:
Contour definition

The contour subroutine must contain at least 3 blocks with motions in the two axes of the machining plane.

If the contour program is shorter, the cycle is aborted after the alarms 10933 "Number of contour blocks contained in the contour program not sufficient" and 61606 "Error in contour preparation" have been output.

Relief cut elements can be connected directly one after the other. Blocks without motions in the plane can be written without restrictions.

In the cycle, all traversing blocks are prepared for the first two axes of the current plane since only these are involved in the cutting process. The contour program may contain any motions programmed for other axes; their distances to be traversed, however, will not come into effect during the whole cycle.

Only straight line and circle programming with G0, G1, G2 and G3 are permitted as the geometry in the contour. Furthermore, it is also possible to program the commands for rounding and chamfer. If any other motion commands are programmed in the contour, the cycle is aborted with the alarm 10930 "Illegal type of interpolation in the stock removal contour".

The first block with traversing motion in the current machining plane must contain a motion command G0, G1, G2 or G3; otherwise, the cycle is aborted with alarm 15800 "Illegal initial conditions for CONTPRON". This alarm will also appear with active G41/42. The starting condition of the contour is the first position in the machining plane which is programmed in the contour subroutine.

To machine the programmed contour, a cycle−internal memory is prepared which can ac-
commodate a certain maximum number of contour elements; how many, depends on the contour. If a contour contains too many contour elements, the cycle is canceled, and alarm 10934 "Overflow of contour table" is issued. In this case, the contour must be split over several contour sections, and the cycle for each section must be called separately.

If the maximum diameter is not at the programmed end or starting point of the contour, the cycle will automatically add an axis-parallel straight line to complete the contour maximum, and this part is removed as the undercut.

![Added straight line](Fig. 9-53)

If a tool radius compensation is programmed in the contour subroutine with G41/G42, the cycle is canceled, and alarm 10931 "Faulty cutting contour" is issued.

**Contour direction**

The direction in which the stock removal contour is programmed can be freely selected. In the cycle, the machining direction is defined automatically. In complete machining, the contour is finished in the same direction as machining was carried out when roughing.

When deciding on the machining direction, the first and the last programmed contour points are taken into account. Therefore, both coordinates must always be programmed in the first block of the contour subroutine.

**Contour monitoring**

The cycle provides contour monitoring with regard to the following:

- Clearance angle of the active tool
- Circle programming of arcs with an aperture angle > 180 degrees

With relief cut elements, the cycle checks whether the machining is possible using the active tool. If the cycle detects that this machining will result in a contour violation, it will be aborted after alarm 61604 "Active tool violates programmed contour" has been output.

If the tool clearance angle is specified with zero in the tool compensation, this monitoring will not be performed.
If too large arcs are found in the compensation, alarm 10931 "Incorrect machining contour" appears.

Protruding contours cannot be machined using CYCLE95; such contours are not monitored by the cycle and, therefore, no alarm message is issued.

![Example of a protruding contour element which cannot be machined with undercut.](image1)

**Starting point**

The cycle determines the starting point for machining operation automatically. The starting point is located in an axis in which the depth infeed is carried out, by the finishing allowance + retraction travel (parameter _VRT) from the contour away. In the other axis, it is by finishing allowance + _VRT ahead of the contour starting point.

When the starting point is approached, the cutter radius compensation is selected internally in the cycle.

The last point before the cycle is called must therefore be selected such that this approach is possible without collision and space enough is provided to carry out the appropriate compensatory motion.

![Total of finishing allowance in X+_VRT, STARTING POINT of the cycle, Total of finishing allowance in Z+_VRT](image2)
Cycle approach strategy

In roughing, the starting point determined by the cycle is always approached with both axes simultaneously, and in finishing, axis by axis. In finishing, the infeed axis traverses first.

Programming example 1: Stock removal cycle

The contour shown in the illustration to explain the defining parameters is to be machined longitudinally externally by complete machining. Axis-specific finishing allowances are specified. Cutting will not be interrupted when roughing. The maximum infeed is 5 mm.

The contour is stored in a separate program.

---

**N10 T1 D1 G0 G95 S500 M3 Z125 X81**

Approach position prior to the call

**N20 CYCLE95("KONTUR_1", 5, 1.2, 0.6, 0.2, 0.1, 0.2, 9, 0.5)**

Cycle call

**N30 G0 G90 X81**

Reapproach of starting position

**N40 Z125**

Traversing by axes

**N50 M2**

End of program

**%_N_KONTUR_1_SPF**

Start of contour subroutine

**N100 Z120 X37**

Traversing by axes

**N110 Z117 X40**

**N120 Z112 RND=5**

Rounding with radius 5

**N130 Z95 X65**

Traversing by axes

**N140 Z87**

**N150 Z77 X29**

**N160 Z62**

**N170 Z58 X44**

**N180 Z52**

**N190 Z41 X37**

**N200 Z35**

**N210 X76**

**N220 M17**

End of subroutine
Programming example 2: Stock removal cycle

The stock removal contour is defined in the calling program and is traversed directly after the cycle for finishing has been called.

```
N110 G18 DIAMOF G90 G96 F0.8
N120 S500 M3
N130 T1 D1
N140 G0 X70
N150 Z160
N160 CYCLE95("ANFANG:ENDE",2.5,0.8,0.8,0.8,0.75,0.6,1,0,0.8,0.75,0.6,1, Cycle call
Fig. 9-57
N170 G0 X70 Z160
N175 M02
ANFANG:
N180 G1 X10 Z100 F0.6
N190 Z90
N200 Z70 ANG=150
N210 Z50 ANG=135
N220 Z50 X50
ENDE:
N230 M02
```
9.5.5 Thread undercut – CYCLE96

Programming

CYCLE96 (DIATH, SPL, FORM)

Parameters

Table 9-17 Parameters of CYCLE94

<table>
<thead>
<tr>
<th></th>
<th>Nominal diameter of the thread</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIATH</td>
<td>real</td>
</tr>
<tr>
<td>SPL</td>
<td>real</td>
</tr>
<tr>
<td>FORM</td>
<td>char</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Starting point of the correction in the longitudinal axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIATH</td>
<td>real</td>
</tr>
<tr>
<td>SPL</td>
<td>real</td>
</tr>
<tr>
<td>FORM</td>
<td>char</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Definition of the form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>A (for form A)</td>
</tr>
<tr>
<td></td>
<td>B (for form B)</td>
</tr>
<tr>
<td></td>
<td>C (for form C)</td>
</tr>
<tr>
<td></td>
<td>D (for form D)</td>
</tr>
</tbody>
</table>

Function

Using this cycle, you can perform thread undercuts to DIN76 for parts with metrical ISO thread.

Sequence

Position reached prior to cycle start:

The starting position can be any position from which each thread undercut can be approached without collision.
The cycle creates the following sequence of motions:

- Approach of the starting point determined in the cycle by using G0
- Selection of tool radius compensation according to the active tool point direction. Traversing along the undercut contour using the feedrate programmed before the cycle was called
- Retraction to the starting point with G0 and deselection of the tool radius compensation with G40

Explanation of the parameters

**DIATH (nominal diameter)**

Use this cycle to perform thread undercuts for metric threads from M3 through M68.

If a final diameter <3 mm results according to the value programmed for DIATH, the cycle is terminated, generating the alarm 61601 "Finished part diameter too small"

If the parameter has a value other than specified in DIN76 Part 1, the cycle is also terminated, generated the alarm 61001 "Thread pitch defined incorrectly".

**SPL (starting point)**

The finished dimension in the longitudinal axis is defined using the parameter SPL.

![Fig. 9-59](image)

**FORM (definition)**

Thread undercuts of the forms A and B are defined for external threads, form A for standard run-outs of threads, and form B for short run-outs of threads.

Thread undercuts of the forms C and D are used for internal threads, form C for a standard run-out of the thread, and form D for a short run-out.

If the parameter has a value other than A ... D, the cycle aborts and creates alarm 61609 "Form defined incorrectly".
Internally in the cycle, the tool radius compensation is selected automatically.

The cycle uses only the tool point directions 1 ... 4. If the cycle recognizes one of the tool point directions 5 ... 9 or if the relevant undercut form cannot be machined using the selected tool point direction, the alarm 61608 "Wrong tool point direction programmed" appears, and the cycle is terminated.

The cycle will find the starting point determined by the tool point direction of the active tool and the thread diameter automatically. The position of this starting point referred to the programmed coordinate values is determined by the tool point direction of the active tool.

For the forms A and B, the undercut angle of the active tool is monitored in the cycle. If it is detected that the form of the undercut cannot be machined using the selected tool, the message "Changed form of undercut" is displayed on the control system; the machining, however, is continued.

![Diagram of FORMS A and B](image1)

![Diagram of FORMS C and D](image2)
Further notes

Before calling the cycle, a tool compensation must be activated. Otherwise, the cycle is terminated and the error message 61000 "No tool compensation active" is issued.

Programming example: Thread_undercut_form_A

This program can be used to program a thread undercut of form A.

Fig. 9-62

<table>
<thead>
<tr>
<th>N10  D3 T1 S300 M3 G95 F0.3</th>
<th>Specification of the technological values</th>
</tr>
</thead>
<tbody>
<tr>
<td>N20 G0 G90 Z100 X50</td>
<td>Selection of the starting position</td>
</tr>
<tr>
<td>N30 CYCLE96 (40, 60, &quot;A&quot;)</td>
<td>Cycle call</td>
</tr>
<tr>
<td>N40 G90 G0 X30 Z100</td>
<td>Approach next position</td>
</tr>
<tr>
<td>N50 M2</td>
<td>End of program</td>
</tr>
</tbody>
</table>
9.5.6 Thread cutting – CYCLE97

Programming

CYCLE97(PIT, MPIT, SPL, FPL, DM1, DM2, APP, ROP, TDEP, FAL, IANG, NSP, NRC, NID, VARI, NUMT)

Parameters

Table 9-18 Parameters of CYCLE97

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIT</td>
<td>real</td>
<td>Thread pitch as a value (enter without sign)</td>
</tr>
<tr>
<td>MPIT</td>
<td>real</td>
<td>Thread pitch as a thread size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range of values: 3 (for M3) .. 60 (for M60)</td>
</tr>
<tr>
<td>SPL</td>
<td>real</td>
<td>Thread starting point in the longitudinal axis</td>
</tr>
<tr>
<td>FPL</td>
<td>real</td>
<td>Thread end point in the longitudinal axis</td>
</tr>
<tr>
<td>DM1</td>
<td>real</td>
<td>Thread diameter at the starting point</td>
</tr>
<tr>
<td>DM2</td>
<td>real</td>
<td>Thread diameter at the end point</td>
</tr>
<tr>
<td>APP</td>
<td>real</td>
<td>Run-in path (enter without sign)</td>
</tr>
<tr>
<td>ROP</td>
<td>real</td>
<td>Run-out path (enter without sign)</td>
</tr>
<tr>
<td>TDEP</td>
<td>real</td>
<td>Thread depth (enter without sign)</td>
</tr>
<tr>
<td>FAL</td>
<td>real</td>
<td>Finishing allowance (enter without sign)</td>
</tr>
<tr>
<td>IANG</td>
<td>real</td>
<td>Infeed angle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range of values: &quot;+&quot; (for flank infeed at the flank)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;-&quot; (for alternating flank infeed)</td>
</tr>
<tr>
<td>NSP</td>
<td>real</td>
<td>Starting point offset for the first thread turn (enter without sign)</td>
</tr>
<tr>
<td>NRC</td>
<td>int</td>
<td>Number of roughing cuts (enter without sign)</td>
</tr>
<tr>
<td>NID</td>
<td>int</td>
<td>Number of idle passes (enter without sign)</td>
</tr>
<tr>
<td>VARI</td>
<td>int</td>
<td>Determination of machining type for the thread</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range of values: 1 ... 4</td>
</tr>
<tr>
<td>NUMT</td>
<td>int</td>
<td>Number of thread starts (enter without sign)</td>
</tr>
</tbody>
</table>

Function

Use the thread cutting cycle to produce cylindrical and tapered external and internal threads with constant pitch in longitudinal and face machining. The threads can be both single-start and multiple threads. With multiple threads, the individual thread starts are machined one after the other.

The infeed is performed automatically; you can choose between the variants constant infeed per cut or constant cutting cross-section.

Right-hand or left hand thread is determined by the direction of rotation of the spindle which must be programmed prior to the cycle start.

Both feed and spindle override are ineffective in the traversing blocks with thread.
Important
To be able to use this cycle, a speed-controlled spindle with position measuring system is required.

Sequence

Position reached prior to cycle start:
Starting position is any position from which the programmed thread starting point + run-in path can be approached without collision.

The cycle creates the following sequence of motions:

- Approach of the starting point determined in the cycle at the beginning of the run-in path for the first thread turn with G0
- Infeed for roughing according to the infeed type defined under VARI.
- Thread cutting is repeated according to the programmed number of roughing cuts.
- The finishing allowance is removed in the following step with G33.
- This step is repeated according to the number of idle passes.
- The whole sequence of motions is repeated for each further thread turn.
Explanation of the parameters

PIT and MPIT (value and thread size)

The thread pitch is an axis-parallel value and is specified without sign. To produce metric cylindrical threads, it is also possible to specify the thread start as a thread size via the parameter MPIT (M3 to M60). Only one of the two parameters should be used by option. If they contain contradicting values, the cycle generates the alarm 61001 “Invalid thread pitch” and is aborted.

DM1 and DM2 (diameter)

Use this parameter to define the thread diameter of starting and end point of the thread. In the case of internal threads, this is the tap-hole diameter.

Interrelation SPL, FPL, APP and ROP (starting, end point, run-in and run-out path)

The programmed starting point (SPL) or end point (FPL) constitutes the original starting point of the thread. The starting point used in the cycle, however, is the starting point brought forward by the run-in path APP, and, correspondingly, the end point is the programmed end point brought back by the run-out path ROP. In the transversal axis, the starting point defined by the cycle is always by 1 mm above the programmed thread diameter. This retraction plane is generated in the internally control system automatically.

Interrelation TDEP, FAL, NRC and NID (thread depth, finishing allowance, number of cuts)

The programmed finishing allowance acts paraxially and is subtracted from the specified thread depth TDEP; the remainder is divided into roughing cuts.

The cycle will calculate the individual infeed depth automatically, depending on the parameter VARI.
When the thread depth is divided into infeeds with constant cutting cross-section, the cutting force will remain constant over all roughing cuts. In this case, the infeed will be performed using different values for the infeed depth.

A second variant is the distribution of the whole thread depth to constant infeed depths. When doing so, the cutting cross-section becomes larger from cut to cut, but with smaller values for the thread depth, this technology can result in better cutting conditions.

The finishing allowance FAL is removed after roughing in one step. Then the idle passes programmed under parameter NID are executed.

**IANG (infeed angle)**

By using parameter IANG, the angle is defined under which the infeed is carried out in the thread. If you wish to infeed at a right angle to the cutting direction in the thread, the value of this parameter must be set to zero. If you wish to infeed along the flanks, the absolute value of this parameter may amount maximally to the half of the flank angle of the tool.

![Diagram](image)

The execution of the infeed is defined by the sign of this parameter. With a positive value, infeed is always carried out at the same flank, and with a negative value, at both flanks alternating. The infeed type with alternating flanks is only possible for cylindrical threads. If the value of IANG for tapered threads is nonetheless negative, the cycle will carry out a flank infeed along a flank.

**NSP (starting point offset) and NUMT (number)**

By using this parameter, you can program the angle value defining the point of the first cut of the thread turn at the circumference of the turned part. This is a starting point offset. The parameter can assume values between 0 and +359.9999 degrees. If no starting point offset is specified or the parameter is omitted from the parameter list, the first thread turn automatically starts at the zero-degree mark.
Use the NUMT parameter to define the number of thread turns with a multiple-turn thread. For a single-turn thread, the parameter must be assigned zero or can be dropped completely in the parameter list.

The thread turns are distributed equally over the circumference of the turned part; the first thread turn is determined by the parameter NSP.

To produce a multiple-turn thread with an asymmetrical arrangement of the thread turns on the circumference, the cycle for each thread turn must be called when programming the appropriate starting point offset.

**VARI (machining type)**

By using the VARI parameter, it is defined whether external or internal machining will be carried out and which technology will be used with regard to the infeed when roughing. The parameter VARI can assume values between 1 and 4 with the following meaning:
Table 9-19 Type of machining

<table>
<thead>
<tr>
<th>Value</th>
<th>Ext./int.</th>
<th>Const. infeed/const. cutting cross–section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>Constant infeed</td>
</tr>
<tr>
<td>2</td>
<td>I</td>
<td>Constant infeed</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>Constant cutting cross–section</td>
</tr>
<tr>
<td>4</td>
<td>I</td>
<td>Constant cutting cross–section</td>
</tr>
</tbody>
</table>

If a different value is programmed for the parameter VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

Further notes

**Differentiation between longitudinal and transversal thread**

The decision whether a longitudinal or transversal thread is to be machined is made by the cycle itself. This depends on the angle of the taper at which the threads are cut. If the angle at the taper is ≤45 degrees, the thread of the longitudinal axis is machined, otherwise – the transversal thread.

![Diagram showing longitudinal and transversal thread differentiation](image)

Fig. 9-68

**Programming example: Thread cutting**

By using this program, you can produce a metric external thread M42x2 with flank infeed. Infeed is carried out with constant cutting cross–section. 5 roughing cuts are carried out at a thread depth of 1.23 mm without finishing allowance. At completion of this operation, 2 idle passes will be carried out.
9.5.7 Chaining of threads – CYCLE98

Note
This standard cycle is not offered by the 802D bl.

Programming

CYCLE98 (PO1, DM1, PO2, DM2, PO3, DM3, PO4, DM4, APP, ROP, TDEP, FAL, IANG, NSP, NRC, NID, PP1, PP2, PP3, VARI, NUMT)

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO1</td>
<td>real</td>
<td>Thread starting point in the longitudinal axis</td>
</tr>
<tr>
<td>DM1</td>
<td>real</td>
<td>Thread diameter at the starting point</td>
</tr>
<tr>
<td>PO2</td>
<td>real</td>
<td>First intermediate point in the longitudinal axis</td>
</tr>
<tr>
<td>DM2</td>
<td>real</td>
<td>Diameter at the first intermediate point</td>
</tr>
</tbody>
</table>
**Table 9-20 Parameter CYCLE98, cont’d**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PO3</td>
<td>real</td>
<td>Second intermediate point</td>
</tr>
<tr>
<td>DM3</td>
<td>real</td>
<td>Diameter at the second intermediate point</td>
</tr>
<tr>
<td>PO4</td>
<td>real</td>
<td>Thread end point in the longitudinal axis</td>
</tr>
<tr>
<td>DM4</td>
<td>real</td>
<td>Diameter at the end point</td>
</tr>
<tr>
<td>APP</td>
<td>real</td>
<td>Run–in path (enter without sign)</td>
</tr>
<tr>
<td>ROP</td>
<td>real</td>
<td>Run–out path (enter without sign)</td>
</tr>
<tr>
<td>TDEP</td>
<td>real</td>
<td>Thread depth (enter without sign)</td>
</tr>
<tr>
<td>FAL</td>
<td>real</td>
<td>Finishing allowance (enter without sign)</td>
</tr>
<tr>
<td>IANG</td>
<td>real</td>
<td>Infeed angle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range of values: “+” (for flank infeed at the flank)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“–” (for alternating flank infeed)</td>
</tr>
<tr>
<td>NSP</td>
<td>real</td>
<td>Starting point offset for the first thread turn (enter without sign)</td>
</tr>
<tr>
<td>NRC</td>
<td>int</td>
<td>Number of roughing cuts (enter without sign)</td>
</tr>
<tr>
<td>NID</td>
<td>int</td>
<td>Number of idle passes (enter without sign)</td>
</tr>
<tr>
<td>PP1</td>
<td>real</td>
<td>Thread pitch 1 as a value (enter without sign)</td>
</tr>
<tr>
<td>PP2</td>
<td>real</td>
<td>Thread pitch 2 as a value (enter without sign)</td>
</tr>
<tr>
<td>PP3</td>
<td>real</td>
<td>Thread pitch 3 as a value (enter without sign)</td>
</tr>
<tr>
<td>VARI</td>
<td>int</td>
<td>Determination of machining type for the thread</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range of values: 1 ... 4</td>
</tr>
<tr>
<td>NUMT</td>
<td>int</td>
<td>Number of thread starts (enter without sign)</td>
</tr>
</tbody>
</table>

**Function**

This cycle can be used to produce several cylindrical or tapered threads in succession. The individual thread sections may possess different leads whereby the lead within one and the same thread section must be constant.

![Fig. 9-70](image_url)
Sequence

Position reached prior to cycle start:
Starting position is any position from which the programmed thread starting point + run-in path can be approached without collision.

The cycle creates the following sequence of motions:
- Approach of the starting point determined in the cycle at the beginning of the run-in path for the first thread turn with G0
- Infeed for roughing according to the infeed type defined under VARI.
- Thread cutting is repeated according to the programmed number of roughing cuts.
- The finishing allowance is removed in the following step with G33.
- This step is repeated according to the number of idle passes.
- The whole sequence of motions is repeated for each further thread turn.

Explanation of the parameters

Fig. 9-71

PO1 and DM1 (starting point and diameter)
These parameters are used to define the original starting point for the thread series. The starting point determined by the cycle itself and approached at the beginning using G0 is located by the run-in path before the programmed starting point (starting point A in the diagram on the previous page).

PO2, DM2 and PO3, DM3 (intermediate point and diameter)
These parameters are used to define two intermediate points in the thread.
PO4 and DM4 (end point and diameter)

The original end point of the thread is programmed using the parameters PO4 and DM4. In the case of internal threads, DM1 ... DM4 are the tap–hole diameter.

Interrelation between APP and ROP (run–in/run–out paths)

The starting point used in the cycle, however, is the starting point brought forward by the run–in path APP, and, correspondingly, the end point is the programmed end point brought back by the run–out path ROP.

In the transversal axis, the starting point defined by the cycle is always by 1 mm above the programmed thread diameter. This retraction plane is generated in the internally control system automatically.

Interrelation between TDEP, FAL, NRC and NID (thread depth, finishing allowance, number of roughing and idle passes)

The programmed finishing allowance acts paraxially and is subtracted from the specified thread depth TDEP; the remainder is divided into roughing cuts. The cycle will calculate the individual infeed depth automatically, depending on the parameter VARI. When the thread depth is divided into infeeds with constant cutting cross–section, the cutting force will remain constant over all roughing cuts. In this case, the infeed will be performed using different values for the infeed depth.

A second variant is the distribution of the whole thread depth to constant infeed depths. When doing so, the cutting cross–section becomes larger from cut to cut, but with smaller values for the thread depth, this technology can result in better cutting conditions.

The finishing allowance FAL is removed after roughing in one step. Then the idle passes programmed under parameter NID are executed.

IANG (infeed angle)

![Diagram showing infeed angles and infeed paths](image-url)

Fig. 9-72
By using parameter IANG, the angle is defined under which the infeed is carried out in the thread. If you wish to infeed at a right angle to the cutting direction in the thread, the value of this parameter must be set to zero. In other words, the parameter may also be omitted from the parameter list, since in this case, it is automatically loaded with zero by default. If you wish to infeed along the flanks, the absolute value of this parameter may amount maximally to the half of the flank angle of the tool.

The execution of the infeed is defined by the sign of this parameter. With a positive value, infeed is always carried out at the same flank, and with a negative value, at both flanks alternating. The infeed type with alternating flanks is only possible for cylindrical threads. If the value of IANG for tapered threads is nonetheless negative, the cycle will carry out a flank infeed along a flank.

**NSP (starting point offset)**

By using this parameter, you can program the angle value defining the point of the first cut of the thread turn at the circumference of the turned part. This is a starting point offset. The parameter can assume values between 0.0001 and +359.9999 degrees. If no starting point offset is specified or the parameter is omitted from the parameter list, the first thread turn automatically starts at the zero-degree mark.

**PP1, PP2 and PP3 (thread pitch)**

These parameters are used to define the value of the thread lead in the three sections of the thread series. The pitch value must be entered as a paraxial value without sign.

**VARI (machining type)**

By using the VARI parameter, it is defined whether external or internal machining will be carried out and which technology will be used with regard to the infeed when roughing. The parameter VARI can assume values between 1 and 4 with the following meaning:

- **Infeed with constant infeed depth**
- **Infeed with constant cutting cross-section**

Fig. 9-73
### Value Ext./int. Const. infeed/const. cutting cross-section

<table>
<thead>
<tr>
<th>Value</th>
<th>Ext./int.</th>
<th>Const. infeed/const. cutting cross-section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>external</td>
<td>Constant infeed</td>
</tr>
<tr>
<td>2</td>
<td>internal</td>
<td>Constant infeed</td>
</tr>
<tr>
<td>3</td>
<td>external</td>
<td>Constant cutting cross-section</td>
</tr>
<tr>
<td>4</td>
<td>internal</td>
<td>Constant cutting cross-section</td>
</tr>
</tbody>
</table>

If a different value is programmed for the parameter VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

**NUMT (number of thread starts)**

Use the NUMT parameter to define the number of thread turns with a multiple-turn thread. For a single-turn thread, the parameter must be assigned zero or can be dropped completely in the parameter list.

The thread turns are distributed equally over the circumference of the turned part; the first thread turn is determined by the parameter NSP.

To produce a multiple-turn thread with an asymmetrical arrangement of the thread turns on the circumference, the cycle for each thread turn must be called when programming the appropriate starting point offset.

![Fig. 9-74](image-url)

**Programming example: Thread chain**

By using this program, you can produce a thread chain starting with a cylindrical thread. The infeed is performed vertically to the thread; neither finishing allowance, nor starting point offset are programmed. 5 roughing cuts and one idle pass are performed. The specified machining type is longitudinal, external, with constant cut cross-section.
Cycles

9.5 Turning cycles

N10 G95 T5 D1 S1000 M4 Specification of the technological values

N20 G0 X40 Z10 Approach starting position

N30 CYCLE98 (0, 30, −30, 30, −60, 36, −80, 50, 10, 10, 0.92, , , , , , , 5, 1, 1.5, 2, 2, 3, 1) Cycle call

N40 G0 X55 Traversing by axes

N50 Z10

N60 X40

N70 M2 End of program
9.6 Error messages and error handling

9.6.1 General notes

If error conditions are detected in the cycles, an alarm is generated and the execution of the cycle is aborted.

Furthermore, the cycles display their messages in the message line of the control system. These message will not interrupt the program execution.

The errors with their reactions and the messages in the message line of the control system are described in conjunction with the individual cycles.

9.6.2 Error handling in the cycles

Alarms with numbers between 61000 and 62999 generated in the cycles. This range of numbers, in turn, is divided again with regard to alarm responses and cancel criteria.

The error text that is displayed together with the alarm number gives you more detailed information on the error cause.

Table 9-21

<table>
<thead>
<tr>
<th>Alarm Number</th>
<th>Clear Criterion</th>
<th>Alarm Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>61000 ... 61999</td>
<td>NC_RESET</td>
<td>Block preparation in the NC is aborted</td>
</tr>
<tr>
<td>62000 ... 62999</td>
<td>Clear key</td>
<td>The block preparation is interrupted; the cycle can be continued with NC START after the alarm has been cleared.</td>
</tr>
</tbody>
</table>

9.6.3 Overview of cycle alarms

The error numbers are classified as follows:

\[
\begin{array}{c c c c c c}
6 & _ & X & _ & _
\end{array}
\]

- X=0 General cycle alarms
- X=1 Alarms generated by the drilling, drilling pattern and milling cycles
- X=6 Alarms generated by the drilling cycles

The Table below includes a list of all errors occurring in the cycles with their location of occurrence and appropriate instructions for fault correction.
### Table 9-22

<table>
<thead>
<tr>
<th>Alarm Number</th>
<th>Alarm Text</th>
<th>Source</th>
<th>Explanation, Remedy</th>
</tr>
</thead>
<tbody>
<tr>
<td>61000</td>
<td>&quot;No tool compensation active&quot;</td>
<td>CYCLE93 to CYCLE96</td>
<td>D offset must be programmed prior to cycle call</td>
</tr>
<tr>
<td>61001</td>
<td>&quot;Illegal thread pitch&quot;</td>
<td>CYCLE84 CYCLE840 CYCLE96 CYCLE97</td>
<td>Check the parameters for the thread size or the specifications for the thread (are contradicting)</td>
</tr>
<tr>
<td>61002</td>
<td>&quot;Machining type defined incorrectly&quot;</td>
<td>CYCLE93 CYCLE95 CYCLE97</td>
<td>The value of parameters VARI for the machining type is specified incorrectly and must be changed</td>
</tr>
<tr>
<td>61101</td>
<td>&quot;Reference plane defined incorrectly&quot;</td>
<td>CYCLE81 to CYCLE89 CYCLE840</td>
<td>Either different values for reference and retraction plane must be selected in the case of relative specification of the depth or an absolute value must be specified for the depth.</td>
</tr>
<tr>
<td>61102</td>
<td>&quot;No spindle direction programmed&quot;</td>
<td>CYCLE88 CYCLE840</td>
<td>The parameter SDIR (or SDR in CYCLE840) must be programmed</td>
</tr>
<tr>
<td>61107</td>
<td>&quot;First drilling depth defined incorrectly&quot;</td>
<td>CYCLE83</td>
<td>First drilling depth is opposite to total drilling depth</td>
</tr>
<tr>
<td>61601</td>
<td>&quot;Finished part diameter too small&quot;</td>
<td>CYCLE94 CYCLE96</td>
<td>The finished part diameter programmed is too small.</td>
</tr>
<tr>
<td>61602</td>
<td>&quot;Too width defined incorrectly&quot;</td>
<td>CYCLE93</td>
<td>Cutting tool is larger than programmed groove width</td>
</tr>
</tbody>
</table>
| 61603        | "Groove form defined incorrectly"   | CYCLE93 | • Radii/chamfers on recess base do not match with groove width  
• Face groove at a contour element running parallel to the longitudinal axis is not possible. |
| 61604        | "Active tool violates programmed contour" | CYCLE95 | Contour violation in relief cut elements due to clear cutting angle of the tool used, i.e. use a different tool or check the contour subroutine |
| 61605        | "Contour programmed incorrectly"    | CYCLE95 | Illegal relief cut element detected                                                                                                           |
| 61606        | "Error in contour preparation"      | CYCLE95 | An error has been found in the contour preparation; this alarm always occurs in conjunction with an NCK alarm 10930 ... 10934, 15800 or 15810 |
| 61607        | "Starting point programmed incorrectly" | CYCLE95 | The starting point reached prior to the cycle call is not outside the rectangle described by the contour subroutine. |
| 61608        | "Invalid tool point direction programmed" | CYCLE94 CYCLE96 | A tool point direction 1...4 matching to the relief cut form must be programmed.                                                              |
| 61609        | "Form defined incorrectly"          | CYCLE94 CYCLE96 | Check the parameters for the relief cut form.                                                                                                  |
| 61611        | "No intersection point found"       | CYCLE95 | No point of intersection with the contour could be calculated. Check contour programming or change infeed depth.                                    |
9.6.4 Messages in the cycles

The cycles display their messages in the message line of the control system. These messages will not interrupt the program execution.

Messages provide information with regard to a certain behavior of the cycles and with regard to the progress of machining and are usually kept beyond a machining step or until the end of the cycle. The following messages are possible:

Table 9-23

<table>
<thead>
<tr>
<th>Message Text</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Depth: according to the value for the relative depth&quot;</td>
<td>CYCLE82...CYCLE88, CYCLE840</td>
</tr>
<tr>
<td>&quot;1st drilling depth: according to the value for the relative depth&quot;</td>
<td>CYCLE83</td>
</tr>
<tr>
<td>&quot;Thread turn &lt;no.&gt; – machining as a longitudinal thread&quot;</td>
<td>CYCLE97</td>
</tr>
<tr>
<td>&quot;Thread turn &lt;no.&gt; – machining as a transversal thread&quot;</td>
<td>CYCLE97</td>
</tr>
</tbody>
</table>

The number > in the message text always stands for the number of the figure currently machined.
Index

A
Absolute drilling depth, 9-243
Address, 8-132

B
Block search, 5-67
Block structure, 8-133
Boring, 9-240
Boring 1, 9-258
Boring 2, 9-261
Boring 3, 9-264
Boring 4, 9-267
Boring 5, 9-269

C
Call conditions, 9-236, 9-241
Centering, 9-242
Chaining of threads – CYCLE98, 9-315
Character set, 8-134
Circle of holes, 9-275
Configuring input screenforms, 9-238
Contour definition, 9-300
Contour monitoring, 9-278, 9-301
CONTPRON, 9-300
Cycle alarms, 9-322
Cycle call, 9-236
Cycle support in the program editor, 9-238
CYCLE81, 9-242
CYCLE82, 9-245
CYCLE83, 9-247
CYCLE84, 9-251
CYCLE840, 9-254
CYCLE85, 9-258
CYCLE86, 9-261
CYCLE87, 9-264
CYCLE88, 9-267
CYCLE89, 9-269
CYCLE93, 9-280
CYCLE94, 9-288
CYCLE95, 9-292
CYCLE96, 9-305
CYCLE97, 9-309
CYCLE98, 9-315

D
Data transfer, 6-96
Deep hole drilling, 9-247
Deep hole drilling with chip breaking, 9-248
Deep hole drilling with swarf removal, 9-248
Determining the tool offsets, 3-33
Drilling, 9-242
Drilling cycles, 9-235
Drilling, counterboring, 9-245

E
Entering the tools and the tool offsets, 3-30

F
Fundamentals of NC programming, 8-131

G
Geometrical parameters, 9-240
Grooving cycle – CYCLE93, 9-280

H
Handwheel, 4-53
HOLES1, 9-271
HOLES2, 9-275

I
Interface parameters, 7-118

J
Jog, 4-50
Jog mode, 4-50

L
Longitudinal thread, 9-314

M
Machine operating area, 4-50
Machine zero, 3-42
Machining cycle – CYCLE95, 9-292
Machining parameters, 9-240
Machining plane, 9-236
Manual input, 4-54
MDA mode, 4-54
Index

Messages, 9-324

N
Non-printable special characters, 8-135

O
Operating areas, 1-14
Operating the cycle selection, 9-238
Overview of cycle alarms, 9-322
Overview of cycle files, 9-238

P
Parameters, 3-30
Part program, selecting, starting, 5-66
Part program, stopping, canceling, 5-68
Plane definition, 9-236
Printable special characters, 8-134

R
R parameters, 3-47
Reapproach after cancellation, 5-69
Reference plane, 9-242
Relative drilling depth, 9-243
Repositioning after interrupting, 5-69
Retraction plane, 9-242
Return conditions, 9-236
Rigid tapping, 9-251
Row of holes, 9-271

S
Safety clearance, 9-243

Screen layout, 1-11
Setting data, 3-44
Simulation of cycles, 9-237
SPOS, 9-252, 9-253
Starting point, 9-302

T
Tapping with compensating chuck, 9-254
Tapping with compensating chuck with encoder, 9-255
Tapping with compensating chuck without encoder, 9-255
Thread cutting – CYCLE97, 9-309
Thread undercut – CYCLE96, 9-305
tool clearance angle, 9-278
Tool zero, 3-42
Transversal thread, 9-314
Turning cycles, 9-235

U
Undercut cycle – CYCLE94, 9-288

V
V24 interface, 6-96

W
Word structure, 8-132
Work offset, 3-42
<table>
<thead>
<tr>
<th>From:</th>
<th>Suggestions and/or corrections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.</td>
</tr>
<tr>
<td>Company/Dept.</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>Zip code: City:</td>
<td></td>
</tr>
<tr>
<td>Telephone: /</td>
<td></td>
</tr>
<tr>
<td>Telefax: /</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To:</th>
<th>Suggestions and/or corrections</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIEMENS AG</td>
<td>Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.</td>
</tr>
<tr>
<td>A&amp;D MC BMS</td>
<td></td>
</tr>
<tr>
<td>Postfach 3180</td>
<td></td>
</tr>
<tr>
<td>D–91050 Erlangen</td>
<td></td>
</tr>
<tr>
<td>(Tel. +49 (0) 180 5050 – 222 [hotline]</td>
<td></td>
</tr>
<tr>
<td>Fax +49 (0) 9131 98 – 63315 [documentation]</td>
<td></td>
</tr>
<tr>
<td>E-mail: <a href="mailto:motioncontrol@siemens.com">motioncontrol@siemens.com</a></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>From:</th>
<th>Suggestions and/or corrections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.</td>
</tr>
<tr>
<td>Company/Dept.</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>Zip code: City:</td>
<td></td>
</tr>
<tr>
<td>Telephone: /</td>
<td></td>
</tr>
<tr>
<td>Telefax: /</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To:</th>
<th>Suggestions and/or corrections</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIEMENS AG</td>
<td>Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.</td>
</tr>
<tr>
<td>A&amp;D MC BMS</td>
<td></td>
</tr>
<tr>
<td>Postfach 3180</td>
<td></td>
</tr>
<tr>
<td>D–91050 Erlangen</td>
<td></td>
</tr>
<tr>
<td>(Tel. +49 (0) 180 5050 – 222 [hotline]</td>
<td></td>
</tr>
<tr>
<td>Fax +49 (0) 9131 98 – 63315 [documentation]</td>
<td></td>
</tr>
<tr>
<td>E-mail: <a href="mailto:motioncontrol@siemens.com">motioncontrol@siemens.com</a></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>From:</th>
<th>Suggestions and/or corrections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.</td>
</tr>
<tr>
<td>Company/Dept.</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>Zip code: City:</td>
<td></td>
</tr>
<tr>
<td>Telephone: /</td>
<td></td>
</tr>
<tr>
<td>Telefax: /</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To:</th>
<th>Suggestions and/or corrections</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIEMENS AG</td>
<td>Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.</td>
</tr>
<tr>
<td>A&amp;D MC BMS</td>
<td></td>
</tr>
<tr>
<td>Postfach 3180</td>
<td></td>
</tr>
<tr>
<td>D–91050 Erlangen</td>
<td></td>
</tr>
<tr>
<td>(Tel. +49 (0) 180 5050 – 222 [hotline]</td>
<td></td>
</tr>
<tr>
<td>Fax +49 (0) 9131 98 – 63315 [documentation]</td>
<td></td>
</tr>
<tr>
<td>E-mail: <a href="mailto:motioncontrol@siemens.com">motioncontrol@siemens.com</a></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>From:</th>
<th>Suggestions and/or corrections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.</td>
</tr>
<tr>
<td>Company/Dept.</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>Zip code: City:</td>
<td></td>
</tr>
<tr>
<td>Telephone: /</td>
<td></td>
</tr>
<tr>
<td>Telefax: /</td>
<td></td>
</tr>
</tbody>
</table>
SINUMERIK 802D Document Structure

General Documentation: Catalog

User Documentation  Operation and Programming

User Documentation  Diagnostics Guide

Manufacturer/Service Documentation: Start-up

Manufacturer/Service Documentation: Description of Functions