

## SINUMERIK 840D sl/840D/840Di sl

### Cycles

#### Programming Manual

#### Preface

---

#### General

---

**1**

#### Drilling cycles and drilling patterns

---

**2**

#### Milling cycles

---

**3**

#### Turning cycles

---

**4**

#### Error messages and Error handling

---

**5**

#### List of abbreviations

---

**A**

#### References

---

**B**

#### List of parameters

---

**C**

Valid for

Controls

SINUMERIK 840D sl/840DE sl

SINUMERIK 840D powerline/840DE powerline

SINUMERIK 840Di sl/840DiE sl

Software

Version

NCU system software for 840D sl/840DE sl 1.5

HMI Advanced 7.5

each with cycles 7.5

**01/2008**

6FC5398-3BP20-1BA0

## Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

### **DANGER**

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

### **WARNING**

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

### **CAUTION**

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

### **CAUTION**

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

### **NOTICE**

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

## Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

## Prescribed Usage

Note the following:

### **WARNING**

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

## Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

## Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

# Preface

## Preface

### Structure of the documentation

The SINUMERIK documentation is organized in 3 parts:

- General documentation
- User documentation
- Manufacturer/service documentation

You can find a publications overview that is updated monthly indicating the currently available languages on the Internet at:

<http://www.siemens.com/motioncontrol>

Follow the menu items "Support" → "Technical Documentation" → "Publications Overview".

The Internet version of the DOConCD, the DOConWEB, is available at:

<http://www.automation.siemens.com/doconweb>

Information about training courses and FAQs (Frequently Asked Questions) can be found at the following website:

<http://www.siemens.com/motioncontrol>, under "Support"

### Target group

This documentation is intended for machine tool programmers.

### Benefits

This Programming manual enables the target group to develop, write, and test programs and to correct errors.

### Standard scope

The functionality of the standard scope (cycles SW 7.3) is described in this Programming manual. Additions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

It may be possible to execute other functions not described in this documentation in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

Further, for the sake of simplicity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation or maintenance.

## Technical Support

If you have questions, contact the following hotline:

Europe and Africa time zone

A&D Technical Support

Tel.: +49 (0) 180 / 5050 - 222

Fax: +49 (0) 180 / 5050 - 223

Internet: <http://www.siemens.com/automation/support-request>

E-mail: <mailto:adsupport@siemens.com>

Asia and Australia time zone

A&D Technical Support

Tel.: +86 1064 719 990

Fax: +86 1064 747 474

Internet: <http://www.siemens.com/automation/support-request>

E-mail: <mailto:adsupport@siemens.com>

America time zone

A&D Technical Support

Tel.: +1 423 262 2522

Fax: +1 423 262 2289

Internet: <http://www.siemens.com/automation/support-request>

E-mail: <mailto:adsupport@siemens.com>

---

### Note

Country telephone numbers for technical support are provided at the following Internet address:

<http://www.siemens.com/automation/service&support>

---

## Questions about the Manual

If you have any queries (suggestions, corrections) in relation to this documentation, please fax or e-mail us:

Fax: +49 (0) 9131 / 98 - 63315

E-mail: <mailto:motioncontrol.docu@siemens.com>



Fax form: See the reply form at the end of this publication

### **SINUMERIK Internet address**

<http://www.siemens.com/sinumerik>

### **Validity**

This Programming Guide is valid for Cycles Version 7.3.

### **Structure of descriptions**

All cycles and programming options have been described according to the same internal structure, as far as this is meaningful and practicable. The various levels of information have been organized such that you can selectively access the information you need for the task in hand.

### **Supplementary devices**

The applications of SIEMENS controls can be expanded for specific purposes through the addition of special add-on devices, equipment and expansions supplied by SIEMENS.



# Table of contents

	<b>Preface .....</b>	<b>3</b>
<b>1</b>	<b>General.....</b>	<b>11</b>
1.1	Overview of cycles .....	11
1.1.1	Drilling cycles, hole pattern cycles, milling cycles, and turning cycles .....	11
1.1.2	Cycle auxiliary subroutines .....	13
1.2	Programming cycles .....	14
1.2.1	Call and return conditions .....	14
1.2.2	Messages during execution of a cycle.....	15
1.2.3	Cycle call and parameter list.....	15
1.2.4	Simulation of cycles .....	19
1.3	Cycle support in the program editor.....	20
1.3.1	Menus, cycle selection.....	20
1.3.2	Functions of the input screen forms.....	21
1.4	Cycle support for user cycles.....	30
1.4.1	Overview of necessary files .....	30
1.4.2	Getting started with cycle support.....	30
1.4.3	Cycle support configuration .....	31
1.4.4	Bitmap size and screen resolution.....	32
1.4.5	Bitmap storage in the data management of HMI Advanced .....	32
1.4.6	Bitmap handling for HMI Embedded.....	33
1.5	Cycle startup .....	34
1.5.1	Machine data.....	34
1.5.2	Definition files for cycles GUD7.DEF and SMAC.DEF .....	35
1.5.3	New delivery forms for cycles in HMI Advanced.....	36
1.5.4	Upgrading the cycles with SW 6.4 and higher in HMI Advanced with SW 6.3 and higher.....	37
1.6	Additional functions for cycles.....	38
<b>2</b>	<b>Drilling cycles and drilling patterns.....</b>	<b>41</b>
2.1	Drilling cycles .....	41
2.1.1	General information .....	41
2.1.2	Requirements.....	42
2.1.3	Drilling, centering - CYCLE81.....	44
2.1.4	Drilling, counterboring - CYCLE82.....	47
2.1.5	Deep-hole drilling - CYCLE83.....	50
2.1.6	Rigid tapping - CYCLE84.....	58
2.1.7	Tapping with compensating chuck - CYCLE840 .....	65
2.1.8	Boring 1 - CYCLE85 .....	72
2.1.9	Boring 2 - CYCLE86 .....	75
2.1.10	Boring 3 - CYCLE87 .....	79
2.1.11	Boring 4 - CYCLE88 .....	82
2.1.12	Boring 5 - CYCLE89 .....	85
2.2	Modal call of drilling cycles .....	88

2.3	Drilling pattern cycles .....	91
2.3.1	Requirements .....	91
2.3.2	Row of holes - HOLES1 .....	92
2.3.3	Row of holes - HOLES2 .....	95
2.3.4	Dot matrix - CYCLE801 .....	98
<b>3</b>	<b>Milling cycles .....</b>	<b>101</b>
3.1	General information .....	101
3.2	Requirements .....	101
3.3	Thread milling - CYCLE90 .....	104
3.4	Long holes located on a circle - LONGHOLE .....	111
3.5	Slots on a circle - SLOT1 .....	117
3.6	Circumferential slot - SLOT2 .....	125
3.7	Milling rectangular pockets - POCKET1 .....	132
3.8	Milling circular pockets - POCKET2 .....	137
3.9	Milling a rectangular pocket - POCKET3 .....	142
3.10	Milling a circular pocket - POCKET4 .....	150
3.11	Face milling - CYCLE71 .....	156
3.12	Path milling - CYCLE72 .....	163
3.13	Rectangular spigot milling - CYCLE76 .....	173
3.14	Circular spigot milling - CYCLE77 .....	179
3.15	Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75 .....	184
3.15.1	General information .....	184
3.15.2	Transfer pocket edge contour - CYCLE74 .....	185
3.15.3	Transfer island contour - CYCLE75 .....	186
3.15.4	Pocket milling with islands - CYCLE73 .....	189
3.15.4.1	General information .....	189
3.15.4.2	Examples .....	192
3.15.4.3	Description of technology in Example 2 .....	202
3.16	Swiveling – CYCLE800 .....	211
3.16.1	General information .....	211
3.16.2	Programming via screen form .....	213
3.16.2.1	General information .....	213
3.16.2.2	Parameters of input screen form .....	213
3.16.2.3	Operator and programmer instructions .....	220
3.16.2.4	Examples of input screen form .....	222
3.16.3	Programming using parameters .....	225
3.16.4	Setting of tools - CYCLE800 .....	228
3.16.5	Alignment of tools - CYCLE800 .....	230
3.16.6	Setting up workpieces with swiveled machining planes .....	233
3.16.6.1	General information .....	233
3.16.6.2	Parameters of input screen form .....	234
3.16.6.3	Data transfer of swivel data in "Swiveling in JOG" .....	238
3.16.7	Startup - CYCLE800 .....	239
3.16.7.1	General information .....	239
3.16.7.2	Startup of kinematic chain .....	244
3.16.7.3	Starting up kinematic rotary axes .....	252

3.16.7.4	Startup of fine kinematics.....	254
3.16.7.5	Startup examples for machine kinematics .....	255
3.16.8	Manufacturer Cycle TOOLCARR.SPF - CYCLE800 .....	265
3.17	High speed settings - CYCLE832 .....	270
3.17.1	General information .....	270
3.17.2	Programming via input screen form.....	273
3.17.2.1	General information .....	273
3.17.2.2	Parameters of input screen form.....	273
3.17.3	Programming via parameters.....	277
3.17.4	Customizing technology.....	278
3.17.4.1	General information .....	278
3.17.4.2	Customizing of machine setter/programmer.....	278
3.17.4.3	Customization by the machine manufacturer .....	279
3.17.4.4	Customizing additional program parameters CYC_832T .....	280
3.17.5	Ports.....	283
3.18	Engraving cycle CYCLE60.....	285
3.19	Trochoidal milling / plunge cutting - CYCLE899 .....	298
3.19.1	General information .....	298
3.19.2	Function .....	298
3.19.2.1	Vortex milling.....	298
3.19.2.2	Plunge cutting .....	300
3.19.3	Programming via a screen form.....	302
3.19.3.1	General information .....	302
3.19.3.2	Parameters of the screen form .....	303
3.19.4	Programming using parameters.....	305
3.19.5	Programming example.....	309
<b>4</b>	<b>Turning cycles .....</b>	<b>313</b>
4.1	General information .....	313
4.2	Conditions .....	313
4.3	Grooving cycle - CYCLE93.....	317
4.4	Undercut cycle - CYCLE94.....	328
4.5	Stock removal cycle - CYCLE95.....	333
4.6	Thread undercut - CYCLE96 .....	347
4.7	Thread cutting - CYCLE97.....	351
4.8	Thread chaining - CYCLE98.....	359
4.9	Thread recutting.....	366
4.10	Extended stock removal cycle - CYCLE950.....	367
<b>5</b>	<b>Error messages and Error handling .....</b>	<b>387</b>
5.1	General information .....	387
5.2	Error handling in the cycles.....	387
5.3	Messages in the cycles.....	388
<b>A</b>	<b>List of abbreviations.....</b>	<b>389</b>
<b>B</b>	<b>References .....</b>	<b>395</b>
<b>C</b>	<b>List of parameters .....</b>	<b>397</b>

Glossary ..... 401

Index..... 425

## General

The first section provides you with an overview of the available cycles. The following sections describe the general conditions that apply to all cycles regarding

- Programming the cycles and
- Operator guidance for calling the cycles.

### 1.1 Overview of cycles

Cycles are generally applicable technology subroutines, with which you can implement specific machining processes such as tapping a thread or milling a pocket. These cycles are adapted to individual tasks by parameter assignment.

The system provides you with the technologies

- Drilling
- Milling
- Turning

in various standard cycles.

#### 1.1.1 Drilling cycles, hole pattern cycles, milling cycles, and turning cycles

You can run the following cycles with the SINUMERIK 810D, 840D and 840Di controls:

##### Drilling cycles

CYCLE81	Drilling, centering
CYCLE82	Drilling, counterboring
CYCLE83	Deep-hole drilling
CYCLE84	Rigid tapping
CYCLE840	Tapping with compensating chuck
CYCLE85	Boring 1
CYCLE86	Boring 2
CYCLE87	Boring 3
CYCLE88	Boring 4
CYCLE89	Boring 5

**Hole pattern cycles**

HOLES1	Machining a row of holes
HOLES2	Machining a circle of holes
CYCLE801	Dot matrix

**Milling cycles**

CYCLE90	Thread milling
LONGHOLE	Milling pattern of elongated holes on a circle
SLOT1	Groove milling pattern on a circle
SLOT2	Circumferential groove milling pattern
POCKET1	Rectangular pocket milling (with face cutter)
POCKET2	Circular pocket milling (with face cutter)
POCKET3	Rectangular pocket milling (with any milling tool)
POCKET4	Circular pocket milling (with any milling tool)
CYCLE71	Face milling
CYCLE72	Contour milling
CYCLE76	Rectangular spigot milling
CYCLE77	Circular spigot milling
CYCLE73	Pocket milling with islands
CYCLE74	Transfer of pocket edge contour
CYCLE75	Transfer of island contour
CYCLE800	Swiveling
CYCLE832	High speed settings
CYCLE60	Engraving cycle
CYCLE899	Trochoidal milling / plunge cutting

**Turning cycles**

CYCLE93	Recess
CYCLE94	Undercut (DIN form E and F)
CYCLE95	Stock removal with relief cutting
CYCLE96	Thread undercut (DIN forms A, B, C, and D)
CYCLE97	Thread cutting
CYCLE98	Chaining of threads
CYCLE950	Extended stock removal



### 1.1.2 Cycle auxiliary subroutines

Included in the cycle package is the auxiliary subroutine

- PITCH

This auxiliary subroutine must always be loaded in the control.

## 1.2 Programming cycles

A standard cycle is defined as a subroutine with name and parameter list. The conditions described in the "SINUMERIK Programming Guide Part 1: Fundamentals" are applicable for calling a cycle.

---

### Note

The cycles are supplied on diskette/CD or with the relevant software version for HMI Advanced. They are loaded to the part program memory of the control via the V.24 interface or from the diskette drive (see operator's guide).

---

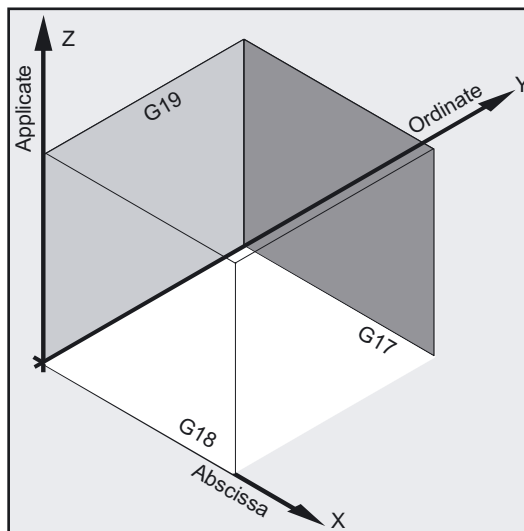
### 1.2.1 Call and return conditions

The G functions active before the cycle is called and the programmable frame remain active beyond the cycle.

You define the machining plane (G17, G18, G19) before the cycle call. A cycle operates in the current plane with the

- Abscissa (1st geometry axis)
- Ordinate (2nd geometry axis)
- Applicate (3rd geometry axis of the plane in space)

In the drilling cycles, the drilling is executed in the axis that is vertical on the plane (3rd geometry plane). In milling, the depth infeed is carried out in this axis.



**Plane and axis assignment:**

Command	Plane	Vertical infeed axis
G17	X/Y	Z
G18	Z/X	Y
G19	Y/Z	X

### 1.2.2 Messages during execution of a cycle

During various cycles, messages that refer to the state of machining are displayed on the screen of the control system during program execution. These messages do not interrupt the program execution and remain until the next message appears or the cycle is completed.

The message texts and their meaning are listed together with the cycle to which they refer.

---

**Note**

You can find a summary of all messages in Appendix A of this programming guide.

---

### Block display during execution of a cycle

The cycle call is displayed in the current block display for the duration of the cycle.

### 1.2.3 Cycle call and parameter list

The standard cycles use user-defined variables. The defining parameters for the cycles can be transferred via the parameter list when the cycle is called.

---

**Note**

Cycle calls always require a block for themselves.

---

### Fundamentals of standard cycle parameter assignment

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,

- variables or
- constants

can be transferred.

If variables are transferred in the parameter list, they must first be defined in the calling program and assigned values. Cycles can be called

- with an incomplete parameter list or
- by leaving out parameters.

If you want to exclude the last transfer parameters that have to be written in a call, you can prematurely terminate the parameter list with ")". If any parameters are to be omitted within the list, a comma "... ,..." must be written as a placeholder.

---

**Note**

No plausibility checks are made of parameter values with a discrete or limited value range unless an error response has been specifically described for a cycle.

During a cycle call, if the parameter list contains more entries than there are parameters defined in the cycle, the general NC alarm 12340 "Too many parameters" appears, and the cycle is not executed.

---

**Note****Transfer parameters and calculation resolution of the NCU**

The value ranges defined in the Programming Guide Fundamentals apply to the transfer parameters of standard and measuring cycles. The value range for angle values is defined as follows (see ROT / AROT in the Programming Guide Fundamentals):

- Rotation around 1st geometry axis: -180 degrees to +180 degrees
- Rotation around 2nd geometry axis: -90 degrees to +90 degrees
- Rotation around 3rd geometry axis: -180 degrees to +180 degrees

When angle values are passed to a standard or measuring cycle, please note that they must be rounded down to zero if they are lower than the calculation resolution of the NCU. The calculation resolution for angle positions on the NCU is specified in machine data 10210 \$MN\_INT\_INCR\_PER\_DEG.

Example of the parameter \_OVR[21] of measuring cycle CYCLE998 (measure angle):

```
_OVR[21]=-0.000345 ;calculation resolution MD $MN_INT_INCR_PER_DEG=1000
```

```
IF ((ABS(_OVR[21] * $MN_INT_INCR_PER_DEG)) < 1)
```

```
_OVR[21]=0
```

```
ENDIF
```

Explanation:

If the value of parameter \_OVR[21] is less than the programmed calculation resolution, it is rounded down to zero.

**Cycle call**

The various methods for writing a cycle call are shown in the following example, CYCLE100, which requires the following input parameters.

**Example**

FORM	char	;Definition of the form to be machined, values: E and F
MID	real	;Infeed depth (to be entered without sign)
FFR	real	;Feed
VARI	integer	;Machining type, values: 0, 1, or 2
FAL	real	;Finishing allowance

This cycle is called via the Cycle100 (FORM, MID, FFR, VARI, FAL) command.

**1. Parameter list with constant values**

Rather than input individual parameters, you can directly enter the concrete values to be used in the cycle.

**Example**

```
| CYCLE100 ("E", 5, 0.1, 1, 0) ;Cycle call
```

**2. Parameter list with variables as transfer parameters**

You can transfer the parameters as R variables that you define before calling the cycle and to which you must assign variables.

**Example**

```

DEF CHAR FORM="E"                ;Definition of a parameter, value assignment
DEF REAL MID=5, FFR, FAL         ;Definition of parameters with and without
DEF INT VARI=1                   ;value assignments
N10 FFR=0.1 FAL=0                ;Value assignments
N20 CYCLE100 (FORM, MID, FFR, VARI, ;Cycle call
FAL)

```

**3. Use of predefined variables as transfer parameters**

To assign parameters to cycles, you can also use variables such as R parameters.

**Example**

```

DEF CHAR FORM="E"                ;Definition of a parameter, value assignment
N10 R1=5 R2=0.1 R3=1 R4=0        ;Value assignments
N20 CYCLE100 (FORM, R1, R2, R3, R4) ;Cycle call

```

As R parameters are predefined as type real, it is important to ensure that the type of the target parameter in the cycle is compatible with the type real.

---

**Note**

More detailed information about data types and type conversion and compatibility is given in the Programming Guide.

In the event of type incompatibilities, the system displays alarm 12330 "Type of parameter ... incorrect".

---

**4. Incomplete parameter list and omission of parameters**

If an assigned parameter for a cycle call is not required or if it should have a value of zero, it can be omitted from the parameter list. In its place, only the comma "...," is to be written in order to ensure the correct assignment of subsequent parameters or if the parameter list is to be closed early with ")".

**Example**

```

CYCLE100 ("F", 3, 0.3, , 1)      ;Cycle call, 4th parameter omitted
                                ;(i.e., the value zero)
CYCLE100 ("F", 3, 0.3)          ;Cycle call, a value of zero is assigned
                                ;to the last two parameters
                                ;(e.g., they have been omitted)

```

### 5. Expressions in the parameter list

Expressions, the results of which are assigned to the corresponding parameter in the cycle, are also permitted in the parameter list.

#### Example

```
DEF REAL MID=7, FFR=200           ;Definition of the parameters, value
                                  assignments
CYCLE100 ("E", MID*0.5, FFR+100,1) ;Cycle call infeed depth 3.5, Feed rate 300
```

## 1.2.4 Simulation of cycles

Programs with cycle calls can be tested first in simulation.

### Function

In HMI Embedded configurations, the program runs normally in the NC and the traversing motion is traced on screen during simulation.

In HMI Advanced configurations, the program is simulated on the HMI only. This makes it possible to run cycles without tool data or without prior selection of a tool offset in the MM.

The finished contour is then traversed in the case of cycles that have to include tool offset data in the calculation of their traversing motion (e.g., milling pockets and grooves, turning with recess) and a message indicates that simulation without tool is active.

This function can be used, for example, to check the position of the pocket.

### 1.3 Cycle support in the program editor

The program editor provides cycle support for Siemens and user cycles.

#### Function

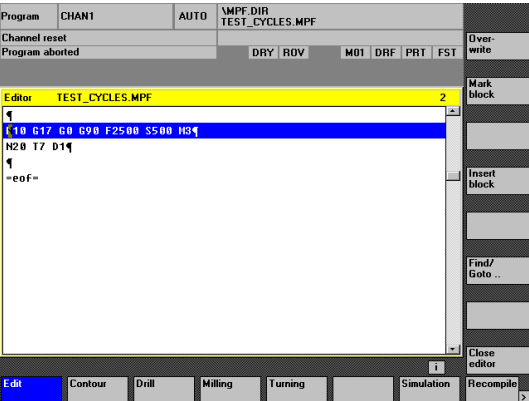
The cycle support offers the following functions:

- Cycle selection via soft keys
- Input screen forms for parameter assignment with help displays
- Online help for each parameter (HMI Advanced only)
- Support of contour input

From the individual screen forms, a program code that can be reset is generated.

#### 1.3.1 Menus, cycle selection

Technology-oriented cycle selection is carried out using soft keys:



Contour

Geometry input via the geometry processor or contour definition screen forms

Bore

Input screen forms for drilling cycles and drilling patterns

Milling

Input screen forms for milling cycles

Turning

Input screen forms for turning cycles

After confirming the screen form input by clicking OK, the technology selection bar remains visible. Similar cycles are supplied from shared screen forms.



Within the screen forms, it is then possible to switch between cycles via soft key, e.g., for tapping or undercut.

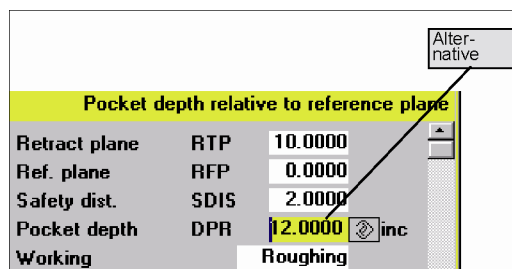
The editor cycle support also contains screen forms that insert a multi-line DIN code in the program instead of a cycle call, e.g., contour definition screen forms and the input of any drilling positions.

## 1.3.2 Functions of the input screen forms

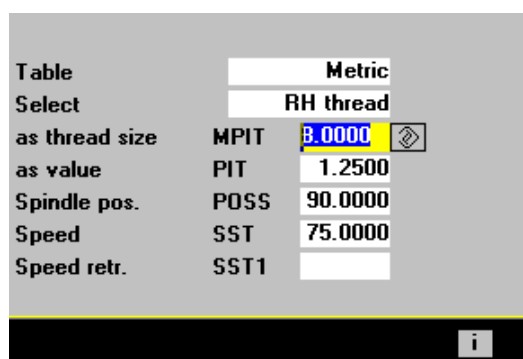
### Function

- Many cycles allow you to influence the processing type via the VARI parameter. It often contains several settings composing one code. In the cycle support screen forms, these individual settings can be divided among different input fields that can be switched using the toggle key.

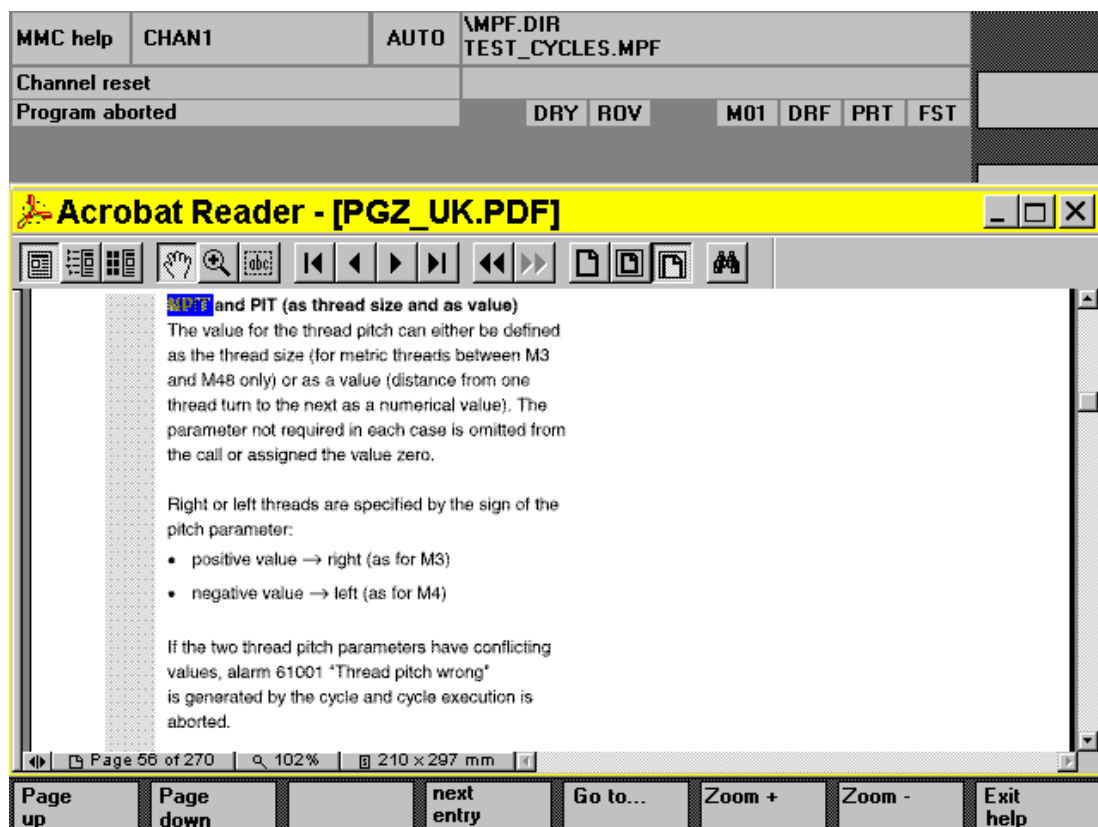
- The input screen forms are changed dynamically. Only the input fields required for the selected processing type are displayed. Unnecessary input fields are not shown.
- One input may, therefore, automatically assign several dependent parameters. This is the case for thread cutting, where metric thread tables are currently supported. During the thread cutting cycle CYCLE97, for example, with input 12 in the input box for the thread size (parameter MPIT), the thread pitch (parameter PIT) is automatically assigned a value of 1.75 and the thread depth (parameter TDEP) is automatically assigned a value of 1.074. This function is not active if the metric thread table has not been selected.
- If a screen form is displayed a second time, the most recently entered values are assigned to all fields. When cycles are called up several times in a row in the same program (e.g., pocket milling when roughing and finishing), few parameters then have to be changed.
- In screen forms for drilling and milling cycles, certain parameters may be input as absolute or incremental values. The abbreviation ABS for absolute and INC for incremental input is displayed behind the input field. This can be toggled with the "Alternative" soft key. This setting will remain the next time these screen forms are called.



- HMI Advanced allows you to view additional information on each cycle parameter in the online help. If the cursor is positioned on a parameter and the help icon appears in the lower right corner, the help function can be activated.



By pressing the info key, the parameter explanation is displayed from the Cycle Programming Guide.



## Operating the help display

Page up	Paging backward in the documentation
Page down	Paging forward in the documentation
Next entry	Enables the user to jump to another piece of text included in the help display
Go to	Enables the user to jump to a selected piece of text
Zoom +	Zoom the text in the help window
Zoom -	Reduce the text in the help window
Exit help	Return to the cycle screen form

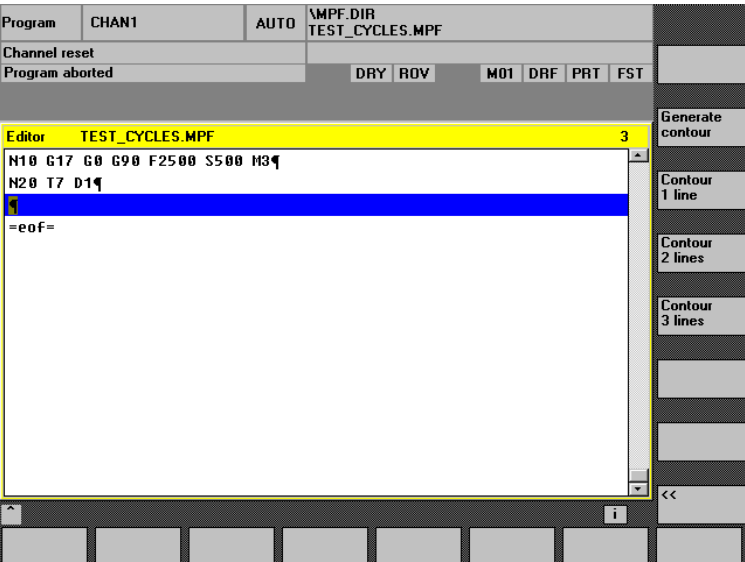
Contour input support

Generate  
contour

Free contour programming

Starts the free contour programming, which can be used to enter contiguous contour sections.

References: /BA/, Operator's Guide



Contour  
1 line

Contour  
2 line

Contour  
3 line

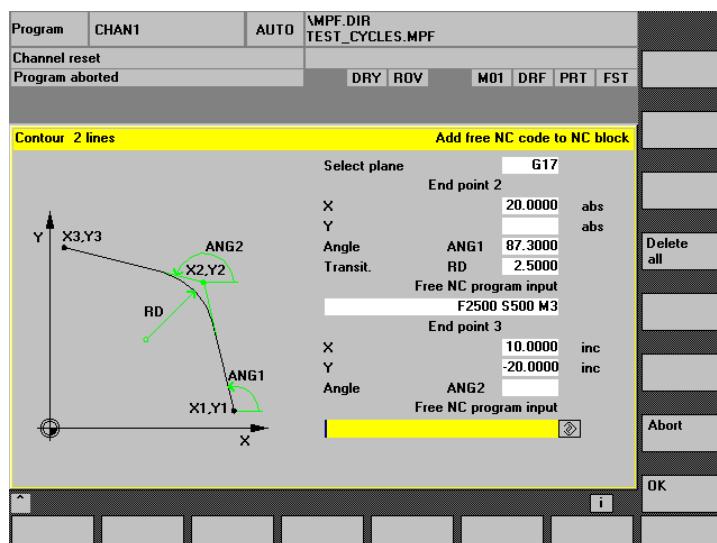
Contour definition programming

These soft keys support the contour definitions that are possible.

These consist of one or more straight lines with intermediary contour transition elements (radii, chamfers). Each contour element may be preassigned by means of end points or point and angle and supplemented by a free DIN code.

Example

The following DIN code is generated from the following input screen form for a contour definition with two straight lines:

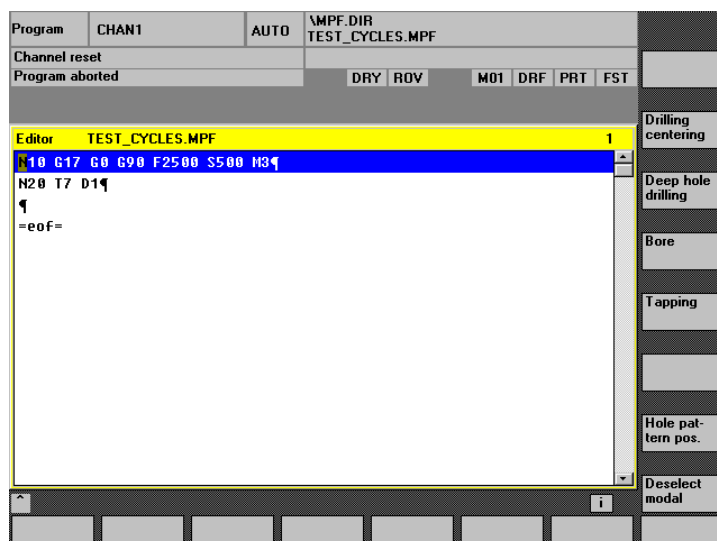


X=AC(20) ANG=87.3 RND=2.5 F2000 S500 M3

X=IC(10) Y=IC(-20); incremental end point

## Drilling support

The drilling support includes a selection of drilling cycles and drilling patterns.



The "Drilling pattern position" soft key branches into a submenu with a selection of several drilling patterns.

Drilling  
centering

Deep hole  
drilling

Bore

Tapping

Selection of drilling patterns

Hole  
pattern pos.

Deselect  
modal

---

**Note**

Cycles CYCLE81, CYCLE87 and CYCLE89 cannot be parameterized with this support. The function of CYCLE81 is covered by CYCLE82 ("Drilling, centering" soft key), as is the function of CYCLE89. The CYCLE87 function is covered by the function of CYCLE88 (soft keys "Drilling center." -> "Drilling with stop").

---

Drilling patterns may be repeated if, for example, drilling and tapping are to be executed in succession. To this end, a name for the drilling pattern, which is later entered in the "Repeat Position" screen form, is assigned in the drilling pattern.

### Example with cycle support generated

N100 G17 G0 G90 Z20 F2000 S500 M3	;Main block
N110 T7 M6	;Change drilling machine
N120 G0 G90 X50 Y50	;Initial drilling position
N130 MCALL CYCLE82(10,0,2,0,30,5)	;Modal drilling cycle call
N140 Circle of holes 1:	;Label – name of drilling pattern
N150 HOLES2(50,50,37,20,20,9)	;Call drilling pattern cycle
N160 ENDLABEL:	
N170 MCALL	;Deselect modal call
N180 T8 M6	;Change tap
N190 S400 M3	
N200 MCALL CYCLE84(10,0,2,0,30,,→ →3,5,0.8,180,300,500)	;Modal call of tapping cycle
N210 REPEAT Circle of holes 1	;Repeat drilling pattern
N220 MCALL	;Deselect modal call

Moreover, any drilling position may be entered as a repeatable drilling pattern by means of screen forms.

Program	CHAN1	AUTO	\MPF.DIR TEST_CYCLES.MPF	
Channel reset				
Program aborted		DRY	ROV	M01 DRF PRT FST

Arbitrary position		4th position		Alter-native
	Select plane	G17		Delete all          Abort OK
	Name of label	PATTERN_1		
	X0	40.0000	abs	
	Y0	40.0000	abs	
	X1	15.0000	inc	
	Y1	80.0000	abs	
	X2	154.0000	inc	
	Y2	60.0000	abs	
	X3		abs	
	Y3		abs	
X4		abs		
Y4		abs		

Up to 5 positions can be programmed in the plane; all values are optionally absolute or incremental (can be toggled with "Alternative" soft key). The "Delete all" soft key generates an empty screen form.

## Milling support

Milling support includes the following selection possibilities:

Program	CHAN1	AUTO	\MPF.DIR TEST_CYCLES.MPF	
Channel reset				
Program aborted		DRY	ROV	M01 DRF PRT FST

Editor	TEST_CYCLES.MPF	2	Face milling
<pre> N10 G17 G0 G90 F2500 S500 M3 [20 T7 D1 </pre>			Path milling
<pre> =eof= </pre>			Standard pockets
			Grooves
			Thread milling
			Spigot

The "Standard pockets", "Grooves", and "Spigot" soft keys each branch into submenus with a selection of several pocket, groove, or spigot cycles.

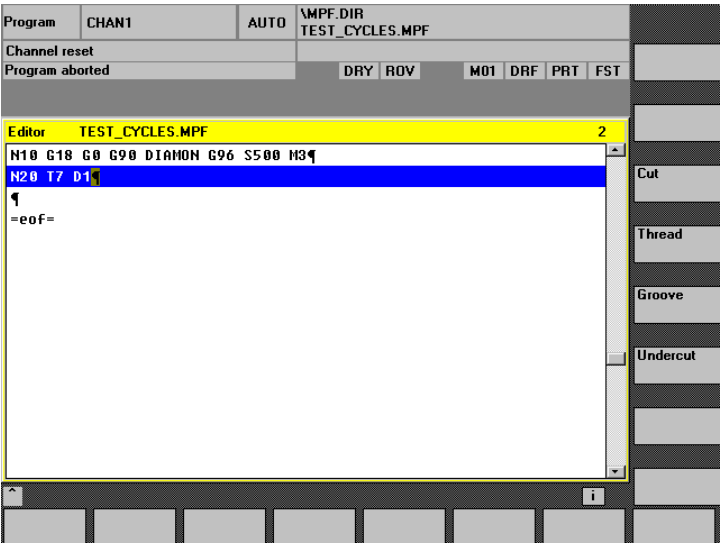
Face milling	Thread milling
Path milling	Swiveling cycles
Standard milling	
Grooves	
Spigot	
>>	<<

**Note**

Pocket milling cycles POCKET1 and POCKET2 cannot be parameterized with this support.

**Turning support**

Turning support includes the following selection possibilities:



Cut
Thread
Groove
Undercut

The undercut cycles for forms E and F (CYCLE94) as well as for the thread undercuts of forms A to D (CYCLE96) are grouped together under the "Undercut" soft key.

The "Thread" soft key contains a submenu for selection between single thread cutting or thread chaining.



## Recompiling

Retranslating program codes serves to change an existing program with the help of cycle support. The cursor is placed on the line to be changed, and the "Recompile" soft key is pressed.

This reopens the corresponding input screen form, which created the program piece, and values may be modified.

Entering modifications directly into the created DIN code may prevent recompilation. Therefore, consistent use of the cycle support is required and modifications are to be carried out with the help of recompilation.

## Default settings for cycle support

Cycles SW 6.4 and higher includes a field for setting data `_SC_MASK[10]` for cycle support.

These are integer type data and are created as NCK global data in the `GUD7_SC.DEF`.

Date	Value	Meaning
<code>_SC_MASK [0]</code>	-	CYCLE84: Default settings for the <code>_TECHNO</code> parameter (set by the machine tool manufacturer during commissioning)
<code>_SC_MASK [1]</code>	-	CYCLE840: Default settings for the <code>_TECHNO</code> parameter (set by the machine tool manufacturer during commissioning)
<code>_SC_MASK [2]</code>		CYCLE97:
	0	Suggested values when using a metric thread table according to DIN13-1 version 11.1999
	1	Suggested values when using a metric thread table according to DIN13-1 version prior to 1999 (for backward compatibility of existing programs)
<code>_SC_MASK [3 to 9]</code>		internal

## Support for programming user cycles

**References:** /IAM/, HMI Installation and Startup Guide

BE1 "Expanding the User Interface"

IM2 "Startup of HMI Embedded"

IM4 "Startup of HMI Advanced"

## 1.4 Cycle support for user cycles

### 1.4.1 Overview of necessary files

The following files constitute the basis for cycle support:

Assignment	File	Application	File type
Cycle selection	aeditor.com	Standard and user cycles	Text file
	common.com (HMI Embedded only)	Standard and user cycles	Text file
Input screen form for parameter assignment	*.com	Standard or user cycles	Text file
Help screens	*.bmp	Standard or user cycles	Bitmap
Online help (HMI Advanced only)	pgz_<language>.pdf and pgz_<language>.txt	Standard cycles only	pdf file

#### Note

Any names can be chosen for the cycle support configuration files (\*.com ).

### 1.4.2 Getting started with cycle support

#### Function

The horizontal soft key HS6 in the program editor is designated as the Entry soft key for user cycles. Its function must be configured in file aeditor.com. Assign a text to the soft key and configure a function in the press block for soft key operation.

#### Example

```
//S(Start)
...
HS5=($80270,,se1)
PRESS(HS5)
LS("Turning",,1)
END_PRESS
HS6=("Usercycle",,se1)      ;HS6 is configured with the "Usercycle" text
PRESS(HS6)
LS("SK_Cycles1","cycproj1") ;When the soft key is pressed, a soft key bar is
                             ;loaded from the file cycproj1.com
END_PRESS
```

A detailed description of the configuration is given in:

**References:** /IAM/, HMI Installation and Startup Guide: BE1 "Expanding the User Interface"

```
%_N_COMMON_COM
; $PATH=/_N_CUS_DIR
...
[MMC_DOS]
...
SC315=AEDITOR.COM
SC316=AEDITOR.COM
```

### 1.4.3 Cycle support configuration

#### Function

The soft key bars and input screen forms of cycle support can be configured in any file and stored as type \*.com in the HMI of the control.

A detailed description of the configuration is given in:

**References:** /IAM/, HMI Installation and Startup Guide: BE1 "Expanding the User Interface"

In HMI Advanced, the \*.com files are stored in data management in the directories:

- dh\cst.dir (for Siemens cycles)
- dh\cma.dir or
- dh\cus.dir

and the usual search sequence is followed: cus.dir, cma.dir, cst.dir. The files are not loaded into the NCU.

For HMI Embedded, the \*.com files can be loaded into the NCU (read in via "Services" using V.24). But as they occupy NC memory there, it is better to integrate them in the HMI. They must be packed and incorporated into the application software of the HMI version. The tool to pack the files is included with the standard cycle software under \hmi\_emb\tools.

#### Step sequence for creation

1. Copy file arj.exe from directory \hmi\_emb\tools to an empty directory on a PC.
2. Copy the separate \*.com configuration files to this directory.
3. Pack each individual com file with the command:
 

```
arj a <Target file name> <Source file name>
```

The target files must have the co\_target extension.

Example: Pack configured file cycproj1.com to:

```
arj a cycproj1.co_ cycproj1.com.
```
4. Copy the \*.co\_ files to the relevant directory of the HMI application software and create a version.

**References:**

/BEM/, HMI Embedded Operator's Guide

/IAM/, HMI Installation and Startup Guide: IM2 "Startup of HMI Embedded"

**1.4.4 Bitmap size and screen resolution**

Three different screen resolutions are available on the HMI.

For each of the resolutions, there is a maximum bitmap size for the cycle screen forms (see the following table), which is to be observed when creating your own bitmaps.

Screen resolution	Bitmap size
640 * 480	224 * 224 pixels
800 * 600	280 * 280 pixels
1024 * 768	352 * 352 pixels

Bitmaps are created and stored as 256-color bitmaps.

**1.4.5 Bitmap storage in the data management of HMI Advanced**

New paths in the data management are set up for the different screen resolutions, such that the bitmaps can be stored in parallel in different sizes.

**Standard cycles:**

- dh\cst.dir\hlp.dir\640.dir
- dh\cst.dir\hlp.dir\800.dir
- dh\cst.dir\hlp.dir\1024.dir

**Manufacturer cycles:**

- dh\cma.dir\hlp.dir\640.dir
- dh\cma.dir\hlp.dir\800.dir
- dh\cma.dir\hlp.dir\1024.dir

**User cycles:**

- dh\cus.dir\hlp.dir\640.dir
- dh\cus.dir\hlp.dir\800.dir
- dh\cus.dir\hlp.dir\1024.dir

Depending on the current resolution, the appropriate directory (for example, dh\...\hlp.dir\640.dir for 640 \* 480) is searched first, followed by dh\...\hlp.dir.

Otherwise, the search sequence cus.dir, cma.dir, cst.dir applies.

## 1.4.6 Bitmap handling for HMI Embedded

### Introduction

With HMI Embedded, the bitmaps are incorporated in the HMI software. They are grouped together into a package `cst.arj`. Bitmaps can always be integrated in `*.bmp` format. However, to save more space and display faster, use a `*.bin` binary format.

### Requirement

In order to generate this, you need the tools supplied in the `\hmi_emb\tools` directory of the standard cycles software:

- `arj.exe`, `bmp2bin.exe` and
- `sys_conv.col`
- `arj_idx.exe`

and the script files:

- `mcst_640.bat`,
- `mcst_800.bat` or
- `mcst1024.bat`.

File `cst.arj` contains all standard and user cycle bitmaps, so you will have to link together the standard cycle bitmaps and your own bitmaps.

### Step sequence for creation

1. Copy all the files from directory `\hmi_emb\tools` to an empty directory on a PC.
2. Create a subdirectory `\bmp_file` in this new directory.
3. Copy your own `*.bmp` bitmaps to this `\bmp_file` subdirectory.
4. Depending on the resolution, for which a `cst.arj` is being created, start `mcst_640.bat/mcst_800.bat` or `mcst1024.bat`.
5. The `cst.arj` created will then be located in the same directory as the generation tools.
6. As of SW 6.3, a further file, `cst.idx`, is also created in this directory. This is integrated into the HMI software along with `cst.arj`.

### References:

/BEM/, HMI Embedded Operator's Guide

/IAM/, HMI Installation and Startup Guide: IM2 "Startup of HMI Embedded"

## 1.5 Cycle startup

### 1.5.1 Machine data

The following machine data must be taken into account when using cycles: The minimum values for these machine data are given in the table below.

#### Relevant machine data

MD number	MD name	Minimum value
18118	MM_NUM_GUD_MODULES	7
18130	MM_NUM_GUD_NAMES_CHAN	20
18150	MM_GUD_VALUES_MEM	2 * number of channels
18170	MM_NUM_MAX_FUNC_NAMES	40
18180	MM_NUM_MAX_FUNC_PARAM	500
28020	MM_NUM_LUD_NAMES_TOTAL	200
28040	MM_NUM_LUD_VALUES_MEM	25

#### NOTICE

These specifications are valid only for the Siemens standard cycles.

For user cycles, the corresponding values must be added.

When using ShopMill or ShopTurn, comply with the information relevant to these products.

The following machine data settings are also required:

MD number	MD name	Minimum value
20240	CUTCOM_MAXNUM_CHECK_BLOCK	4

The machine data files are delivered with these defaults by the machine manufacturer.

Remember: Any changes you make to these machine data will not apply until you perform a power ON.

#### NOTICE

For the cycle CYCLE840 (tapping with compensating chuck), the axis-specific machine data element MD 30200: NUM\_ENCS must also be taken into account.

## 1.5.2 Definition files for cycles GUD7.DEF and SMAC.DEF

Standard cycles require Global User Data definitions (GUDs) and macro definitions. These are stored in definition files GUD7.DEF and SMAC.DEF, supplied with the standard cycles.

### Relevant definition files

To facilitate the system startup engineer's task of combining GUDs and macros in in one block without editing the original SIEMENS files, the following files are supplied in their entirety along with the "standard cycles":

- GUD7.DEF
- SMAC.DEF

These two files contain no definitions and are simply shortcuts to predefined, product-specific definition files. The call-up mechanisms now included in these cycles allow automatic call-up and compilation of all product-specific GUD and macro definitions.

Each package will now only contain its own definitions. New cycle files GUD7\_xxx.DEF and SMAC\_xxx.DEF, which will be in data management in the definition directory DEF.DIR, will be introduced.

The new files for the standard cycles are:

- GUD7\_SC.DEF and
- SMAC\_SC.DEF.

For other cycle packages the following file IDs are currently occupied by SIEMENS:

(xxx stands for "GUD7" or "SMAC")

File ID	SIEMENS assignment
xxx_JS	JobShop cycles in general
xxx_MC	Measuring cycles
xxx_MJ	Measuring in JOG
xxx_MT	ManualTurn
xxx_SM	ShopMill
xxx_ST	ShopTurn
xxx_ISO	ISO compatibility
xxx_C950	Extended stock removal
xxx_C73	Pocket with islands

---

### Note

Additional IDs not shown here can also be used by the system!

---

According to the definition of Defining user data (see programming guide "Defining user data" production planning), the GUD7 blocks and the SMAC.DEF are NOT available to the machine tool manufacturer/user! MGUD, UGUD, GUD4, 8, 9 and MMAC, UMAC should preferably be employed in user applications.

However, in order to give users the option of integrating their existing definitions in these blocks in this system, the following IDs are kept free:

File ID	Assignment
xxx_CMA	Manufacturer
xxx_CUS	User

### Startup, upgrading for standard cycles

1. If a GUD7.DEF is already active in the control system, select the GUD7 user data via "Services", "Data out", "NC active data" and back up the current values in an archive or on diskette.
2. Read in files GUD7\_SC.DEF and SMAC\_SC.DEF and load them into NCU.
3. Read in and activate GUD7.DEF and SMAC.DEF.
4. Switch the NCU power on.
5. Read in the archive of saved values again.

### Loading an additional cycle package

1. Unload GUD7.DEF and SMAC.DEF (first back up the values, if necessary).
2. Read in files GUD7\_xxx.DEF and SMAC\_xxx.DEF and load them into NCU.
3. Reactivate GUD7.DEF and SMAC.DEF.

---

#### Note

When loading or unloading individual definition files, if a call-up file is already loaded it must be unloaded and then reloaded. Otherwise, the NC retains the previous GUD/macro configuration.

---

### Operation in the HMI Advanced simulation

After upgrading the cycle version in the NCU, machine data adjustment followed by an NC reset of the simulation is necessary as soon as you start the simulation, in order to activate the modified definition files.

### 1.5.3 New delivery forms for cycles in HMI Advanced

As of HMI Advanced 6.3, the delivery form of standard cycles in the HMI will change. The cycle files will no longer be stored as individual files in the corresponding directories of the data management. Instead, they will be available as archive files under:

→ Archives/cycle archives.



This will enable the previously existing cycle version in data management to be retained unchanged when the HMI is upgraded.

For upgrading, these archive files must be read in via "Data in". If these archive files are read in, after the upgrade there are no longer different versions of the cycles in the NCU and on the hard disk. The loaded cycles are overwritten in the NCU, not loaded on the hard drive. The new cycle files are always stored on the hard disk.

**References:** For current information, see:

- "siemensd.txt" file in the supplied software (standard cycles) or
- In the case of HMI Advanced, F:\dh\cst.dir\HLP.dir\siemensd.txt.

#### **1.5.4 Upgrading the cycles with SW 6.4 and higher in HMI Advanced with SW 6.3 and higher**

When upgrading a cycle version with SW 6.4 and higher, the archives in HMI Advanced must first be exchanged under:

→ Archives/cycle archives.

To this end, the supplied software contains several archives for overwriting the existing cycle archives. These archives can be found in the supply software in the path

→ hmi\_adv

**References:** For current information, see:

- "siemensd.txt" file in the supplied software (standard cycles) or
- In the case of HMI Advanced, F:\dh\cst.dir\HLP.dir\siemensd.txt.

Once the cycle archives have been overwritten, they must be loaded as described in the previous section "New delivery forms for cycles in HMI Advanced".

## 1.6 Additional functions for cycles

### Version display

To provide an overview and for diagnosis of the cycle versions and their definition files, it will be possible to display and use version screens.

These can be found in HMI under "Diagnostics => "Service displays" => "Version" => "Cycle Version" or "Definit. Version". A LOG file in ASCII format can be generated and read out under "Services" => "Diagnostics" => "LOG files".

---

#### Note

**This function can only be run with HMI software versions SW 6.3 and higher.**

---

Different overviews are possible with cycle version display:

- Overview of all available cycles.
- Overview of individual directories of data management for user cycles (CUS.DIR), manufacturer cycles (CMA.DIR), and Siemens cycles (CST.DIR).
- Package overview of all cycle packages available in the control system.
- Details of individual packages and cycle files.

#### References:

/BAD/, HMI Advanced Operator's Guide

/BEM/, HMI Embedded Operator's Guide, Service Display section

The version display includes all the \*.SPF cycle files and all the \*.COM files of cycle support.

No additional files are required for version display via directories or all cycles. To display overviews of individual cycle packages, each cycle package must contain a package list of all associated files.

### Package lists

A new file type for packages lists is introduced:

\*.cyp (for cycle package), cycle package list in plain text.

Users can create package lists for their own cycle packages. They must look like this:

Structure of a package list:

1st line	Version entry (after the key word ;VERSION:) and package name (after the key word ;PACKAGE:)
Starting from the 2nd line	List of files associated with the cycle package with their name and type
Last line	M30

## Example

```
%_N_CYC_USER1_CYP
; $PATH=/_N_CUS_DIR
; VERSION: 01.02.03 31.10.2002 ; PACKAGE: $85200
ZYKL1.SPF
ZYKL2.SPF
ZYKL3.COM
M30
```

Input in the text file uc.com:

85200 0 0 "Cycle package 1"

The following is displayed in the package overview:

Diagnosis	CHAN1	AUTO	MPF0
Channel RESET		Program aborted	
		ROV	SBL1

Version data Cycles	
Name	Version
Measuring cycles	06.02.08 Mar 21, 2002
Cycle package	01.02.03 31.10.2002

Package overview

Details

All cycles

User cycles

OEM cycles

Siemens cycles

^

NCU version

HMI version

Cycle version

Definition version

The following is displayed in the detail overview:

Diagnosis	CHAN1	AUTO	MPF0		
Channel RESET			Program aborted		
			ROV SBL1		
Version data ShopMill cycles					
Name	Type	Load	Length	Directory	Version
ZYKL1	SPF		1234	CUS.DIR	01.02.03 31.10.2002
ZYKL2	SPF		778	CUS.DIR	01.02.03 31.10.2002
ZYKL3	COM		521	CUS.DIR	01.02.03 31.10.2002

---

#### Note

The cycle package name behind the keyword PACKAGE can also be written as a string in ". However, it is language-dependent in this case.

---

#### Version entries in cycles

Exactly as in the case of package lists, the entry after the keyword ";VERSION:" is assessed as the version ID.

The version entry has to be located in the first ten lines of the cycle, the search does not go any further.

#### Example

```
%_N_ZYKL1_SPF
; $PATH = /_N_CUS_DIR
; VERSION: 01.02.03 31.10.2002
; Comment
PROC CYCLE1 (REAL PAR1)
...
```

## Drilling cycles and drilling patterns

### 2.1 Drilling cycles

#### 2.1.1 General information

##### Function

Drilling cycles are motional sequences specified according 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.

##### Boring cycles

A total of five cycles is provided for boring. They all follow a different technological procedure and are, therefore, parameterized differently:

Boring cycle	Cycle	Special parameterization features
Boring 1	CYCLE85	Different feed rates for boring and retraction (for reaming)
Boring 2	CYCLE86	Oriented spindle stop, specification of retraction path, retraction in rapid traverse, specification of direction of spindle rotation (for boring)
Boring 3	CYCLE87	Spindle stop M5 and program stop M0 at drilling depth, continue after NC start, retraction in rapid traverse, specification of direction of spindle rotation
Boring 4	CYCLE88	Like CYCLE87 plus dwell time at drilling depth
Boring 5	CYCLE89	Boring and retraction at the same feedrate

The drilling cycles can be modal, that is, they are executed at the end of each block containing motion commands. Other cycles written by the user can also be called modally.

##### Parameters

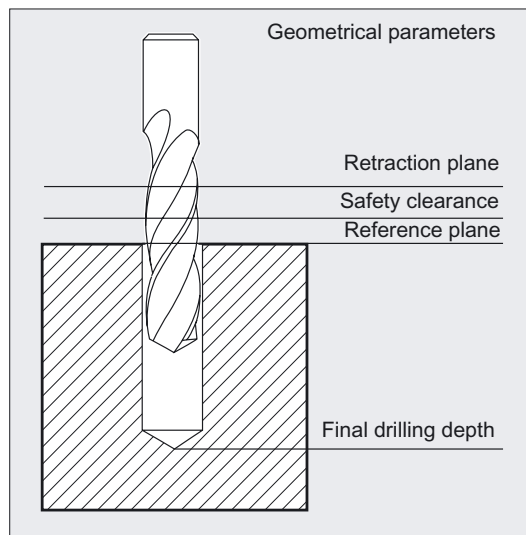
There are two types of parameters:

- Geometrical parameters and
- Machining parameters

## 2.1 Drilling cycles

The geometrical parameters are identical for all drilling cycles, drilling pattern cycles and milling cycles. They define the reference and retraction planes, the safety clearance and the absolute or relative final drilling depth. Geometrical parameters are assigned once during the first drilling cycle CYCLE81.

The machining parameters have a different meaning and effect in the individual cycles. They are therefore programmed in each cycle separately.



### 2.1.2 Requirements

#### 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.

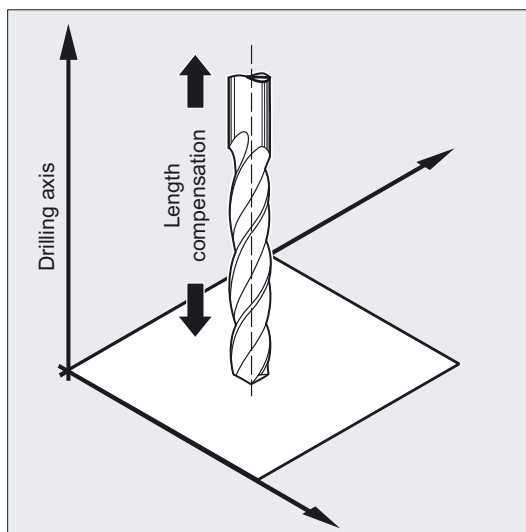
The required values for feedrate, spindle speed and direction of spindle rotation must be programmed in the part program if there are no defining parameters in the drilling cycle.

The G functions and current frame active before the cycle was called remain active beyond the cycle.

#### Plane definition

Drilling cycles generally assume that the current workpiece coordinate system, in which the machining operation is to be performed, has been defined by selecting a plane (G17, G18 or G19) and activating a programmable frame. The drilling axis is always the axis in this coordinate system that is vertical on this 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.



### Spindle handling

The drilling cycles are set up such that the spindle commands they contain always refer to the active master spindle of the control system. If you want to use a drilling cycle on a machine with several spindles, you must first define the spindle with which you will be working as the master spindle.

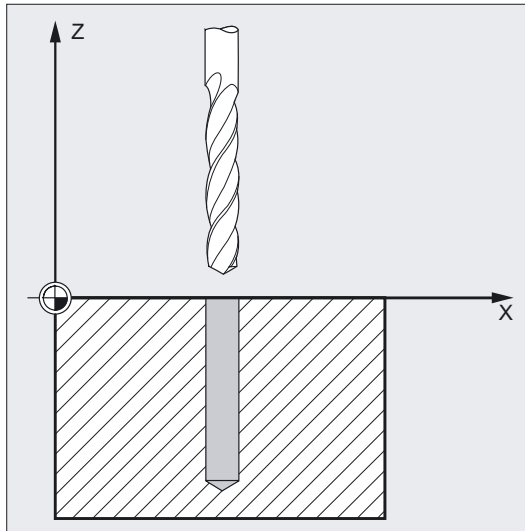
### Dwell time programming

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.

### 2.1.3 Drilling, centering - CYCLE81

#### Function

The tool drills at the programmed spindle speed and feedrate to the specified final drilling depth.



#### Programming

CYCLE81 (RTP, RFP, SDIS, DP, DPR)

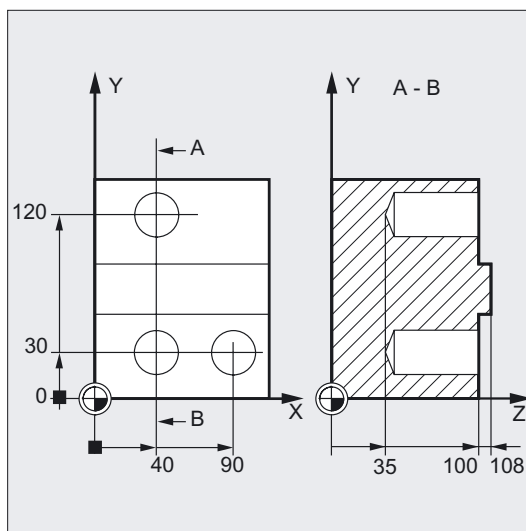
#### Parameters

Parameters	Data type	Meaning
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)



### Example of drilling, centering

Use this program to produce 3 drill holes using the CYCLE81 drilling cycle, whereby this is called using different parameters. The drilling axis is always the Z axis.



```

N10 G0 G90 F200 S300 M3                ;Specification of technology values
N20 D1 T3 Z110                          ;Approach retraction plane
N21 M6
N30 X40 Y120                            ;Approach first drilling position
N40 CYCLE81(110, 100, 2, 35)             ;Cycle call with absolute end drilling
                                         ;depth, safety clearance and incomplete
                                         ;parameter list
N50 Y30                                 ;Approach next drilling position
N60 CYCLE81(110, 102, , 35)             ;Cycle call without safety clearance
N70 G0 G90 F180 S300 M03                ;Specification of technology values
N80 X90                                 ;Approach next position
N90 CYCLE81(110, 100, 2, , 65)          ;Cycle call with relative end drilling
                                         ;depth, and safety clearance
N100 M30                                ;Program end

```

## Sequence

### Position reached prior to cycle start:

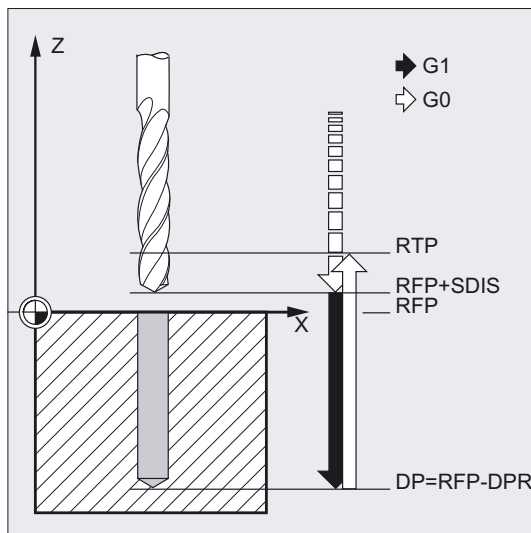
The drilling position is the position in the two axes of the selected plane.

### The cycle creates the following sequence of motions:

Approach of reference plane shifted by the amount of the safety clearance with 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. The cycle assumes that the retraction plane precedes 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.

### Note

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.

**SDIS (safety clearance)**

The safety clearance (SDIS) acts with reference to the reference plane. This is shifted by the amount of the safety clearance.

The direction, in which the safety clearance is active, is automatically determined by the cycle.

**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.

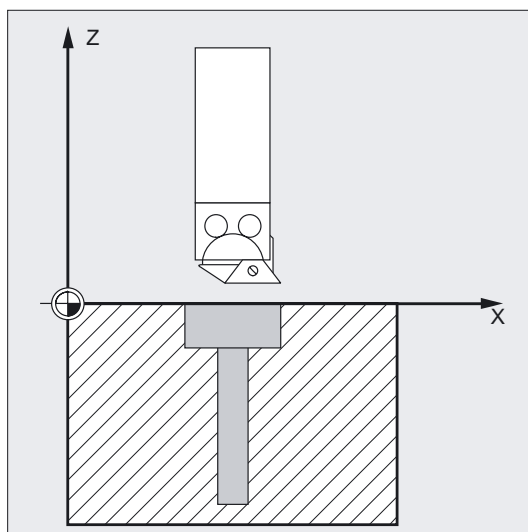
**Note**

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 dialog line.

## 2.1.4 Drilling, counterboring - CYCLE82:

**Function**

The tool drills at the programmed spindle speed and feedrate to the specified final drilling depth. A dwell time can be allowed to elapse when the final drilling depth has been reached.



## Programming

CYCLE82 (RTP, RFP, SDIS, DP, DPR, DTB)

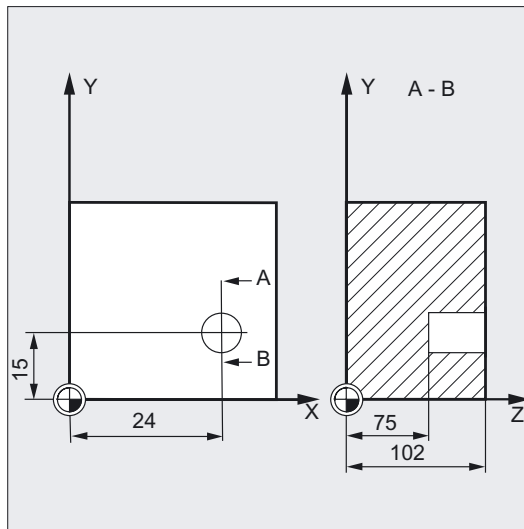
## Parameter

Parameter	Data type	Meaning
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

## Example of drilling, counterboring

The program machines a single hole of a depth of 27 mm at position X24 Y15 in the XY plane with cycle CYCLE82.

The dwell time programmed is 2 s, the safety clearance in the drilling axis Z is 4 mm.



```

N10 G0 G90 F200 S300 M3           ;Specification of technology values
N20 D1 T3 Z110                     ;Approach retraction plane
N21 M6
N30 X24 Y15                         ;Approach drilling position
N40 CYCLE82 (110, 102, 4, 75, , 2) ;Cycle call with absolute end drilling depth,
                                     ;and safety clearance
N50 M30                             ;Program end

```

## Sequence

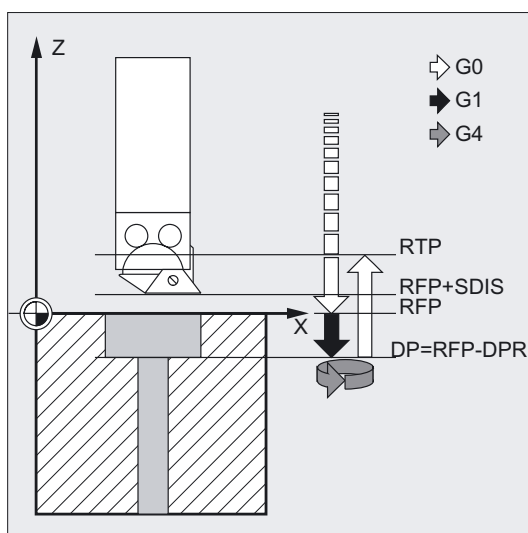
### Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

### The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0
- Traversing to the final drilling depth at the feedrate programmed in the calling program (G1).
- Execution of dwell time at final drilling depth.
- Retraction to the retraction plane with G0.

## Explanation of the parameters



### DTB (dwell time)

The dwell time to the final drilling depth (chip breaking) is programmed under DTB in seconds.

### Note

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, see Drilling, centering – CYCLE81.

## See also

Drilling, centering - CYCLE81 (Page 44)

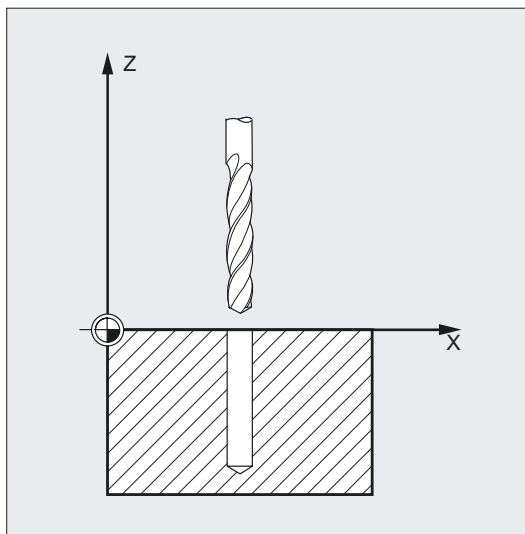
## 2.1.5 Deep-hole drilling - CYCLE83

### 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.

Optionally, the drilling machine can be retracted after each infeed depth either to the reference plane + safety clearance for chip removal or by the length of the programmed retraction path for chip breakage.



### Programming

CYCLE83 (RTP, RFP, SDIS, DP, DPR, FDEP, FDPR, DAM, DTB, DTS, FRF, VARI, \_AXN, \_MDEP, \_VRT, \_DTD, \_DIS1)

### Parameter

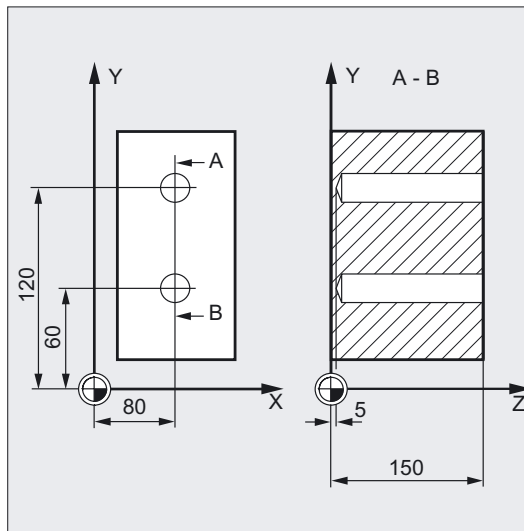
Parameter	Data type	Meaning
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
FDEP	real	First drilling depth (absolute)
FDPR	real	First drilling depth relative to the reference plane (enter without sign)

Parameter	Data type	Meaning
DAM	real	Degression (enter without sign)
		Values: > 0: degression as a quantity < 0: degression factor = 0: no degression
DTB	real	Dwell time at drilling depth (chip breakage)
		Values: > 0: in seconds < 0: in revolutions
DTS	real	Dwell time at starting point and for chip removal
		Values: > 0: in seconds < 0: in revolutions
FRF	real	Feedrate factor for first drilling depth (enter without sign) Value range: 0.001...1
VARI	integer	Machining type
		Values: 0: chip breakage 1: chip removal
_AXN	integer	Tool axis
		Values: 1: 1st geometrical axis 2: 2nd geometrical axis otherwise 3rd geometrical axis
_MDEP	real	Minimum drilling depth (only in connection with degression factor)
_VRT	real	Variable retraction value for chip breakage (VARI=0)
		Values: > 0: if retraction value = 0: retraction value 1 mm set
_DTD	real	Dwell time at final drilling depth
		Values: > 0: in seconds < 0: in revolutions = 0: value same as DTB
_DIS1	real	Programmable limit distance for reimmersion in the drill hole (for chip removal VARI=1)
		Values: > 0: programmable value applies = 0: automatic calculation

This program executes the cycle CYCLE83 at the positions X80 Y120 and X80 Y60 in the XY plane. The first drill hole is drilled with a dwell time zero and machining type chip breaking.

The final drilling depth and the first drilling depth are entered as absolute values. In the second cycle call, a dwell time of 1 s is programmed. Machining type chip removal is selected, the final drilling depth is relative to the reference plane. The drilling axis in both cases is the Z axis.

The drilling stroke is calculated on the basis of a degression factor and must not become shorter than the minimum drilling depth of 8 mm.



DEF REAL RTP=155, RFP=150, SDIS=1,	;Definition of parameters
DP=5, DPR=145, FDEP=100, FDPR=50,	
DAM=20, DTB=1, FRF=1, VARI=0, _VRT=0.8,	
_MDEP=8, _DIS1=0.4	
N10 G0 G17 G90 F50 S500 M4	;Specification of technology values
N20 D1 T42 Z155	;Approach retraction plane
N30 X80 Y120	;Approach first drilling position
N40 CYCLE83 (RTP, RFP, SDIS, DP, ,->	;Call of cycle depth parameters with
-> FDEP, , DAM, , , FRF, VARI, , , _VRT)	;absolute values
N50 X80 Y60	;Approach next drilling position
N55 DAM=-0.6 FRF=0.5 VARI=1	;Value assignment
N60 CYCLE83 (RTP, RFP, SDIS, , DPR, , ->	;Call of cycle with relative
-> FDPR, DAM, DTB, , FRF, VARI, , _MDEP,	;specifications of final drilling depth
-> , , _DIS1)	;and 1st drilling depth, the safety
	;clearance is 1 mm and the feedrate
	;factor is 0.5
N70 M30	;Program end

-> means: it must be programmed in a block.



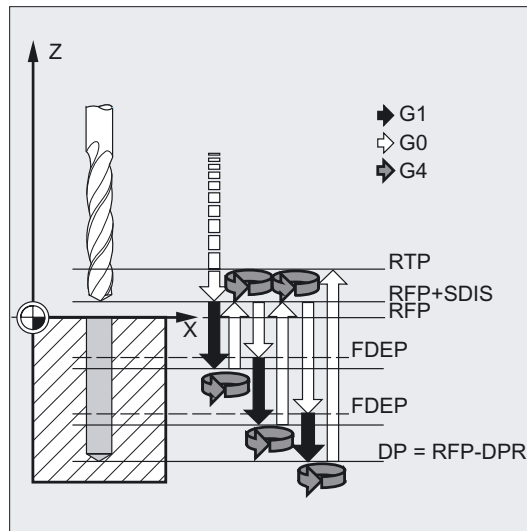
## Sequence

### Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

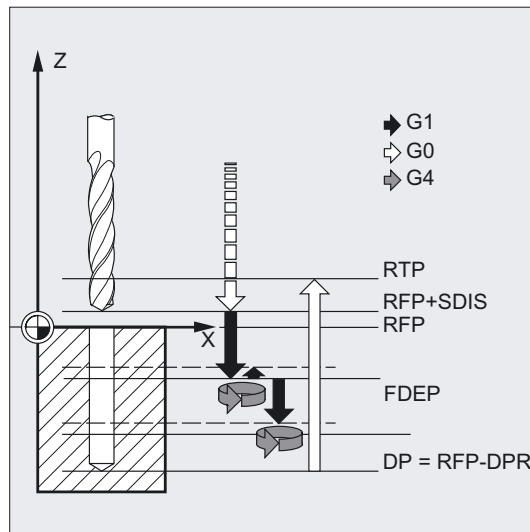
### The cycle creates the following sequence:

#### Deep-hole drilling with chip removal (VARI=1):



- Approach of reference plane shifted by the amount of the safety clearance with G0
- Traversing to the first drilling depth with G1, whereby the feedrate results from the feedrate programmed during the cycle call that is compensated with the FRF (feedrate factor) parameter.
- Execution of dwell time at final drilling depth (DTB parameter).
- Retraction to the reference plane shifted by the amount of the safety clearance with G0 for chip removal.
- Execute dwell time at starting point (DTS parameter).
- Approach of the last drilling depth reached, minus the limit distance, programmable or calculated within the cycle, with G0.
- Traversing to next drilling depth with G1 (motion sequence is continued until the final drilling depth is reached).
- Retraction to the retraction plane with G0.

**Deep-hole drilling with chip breakage (VARI=0):**



- Approach of reference plane shifted by the amount of the safety clearance with G0
- Traversing to the first drilling depth with G1, whereby the feedrate results from the feedrate programmed during the cycle call that is compensated with the FRF (feedrate factor) parameter.
- Execution of dwell time at final drilling depth (DTB parameter).
- Variable retraction (Parameter \_VRT) from the current drilling depth with G1 and the feedrate programmed in the calling program (for chip breakage).
- Traversing to next drilling depth with G1 and the programmed feedrate (motion sequence is continued until the final drilling depth is reached).
- Retraction to the retraction plane with G0.

**Explanation of the parameters**

**FDEP and FDPR (first drilling depth absolute and relative)**

The first drilling depth is programmed by either one of these two parameters. The FDPR parameter has the same effect in the cycle as the DPR parameter. If the values for the reference and retraction planes are identical, the first drilling depth can be defined as a relative value.

**DAM (Degression)**

With deep holes, drilled in several steps, it is useful to work with degressive values for the individual drilling strokes (deggression). This allows for removal of the chips and there is no tool breakage.

In the parameter, either program an incremental degression value in order to reduce the first drilling depth step by step or a % value to act as a degression factor.

**DAM=0 no degression**

**DAM>0 degression as value**

The current depth is derived in the cycle as follows:

- In the first step, the depth parameterized with the first drilling depth FDEP or FDPR 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.

If a value smaller than the programmed amount of degression has already been obtained for the second drilling stroke, this is executed in one cut.

- 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.

#### Example of drilling strokes:

Programming the values RTP=0, SDIS=0, DP=-40, FDEP=-12, and DAM=3 results in the following drilling strokes:

Value	Meaning
-12	Corresponds to the first drilling depth
-21	The incremental difference 9 results from the first drilling depth 12 minus the degression value 3
-27	Previous drilling depth minus degression value 3
-30, -33, -36	Degression value
-38, -40	Remaining depth divided into two cuts

#### DAM<0 (-0.001 to -1) degression factor

The current depth is derived in the cycle as follows:

- The first step is traversing the depth parameterized with the first drilling depth FDED or FDPR, provided it does not exceed the total drilling depth.
- The next drilling strokes are calculated from the last drilling stroke multiplied by the degression factor, as long as the stroke does not fall below the minimum drilling depth.
- The last two drilling strokes are divided equally and traversed and are, therefore, always greater than half of the minimum drilling depth.
- 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.
- If the value DAM<-1 is programmed, the cycle aborts and generates error message 61019 "Parameter DAM defined incorrectly".

### Example of drilling strokes:

Programming the values RTP=0, SDIS=0, DP=-40, FDEP=-10, DAM=-0.8, and MDEP=5 results in the following drilling strokes:

Value	Meaning
-10	Corresponds to the first drilling depth
-18	The incremental difference 8 corresponds to $0.8 \cdot$ the first drilling depth
-24.4, -29.52	The relevant previous drilling depth $\cdot$ degression factor
-34.52	Minimum drilling depth MDEP is in effect
-37.26, -40	Remaining depth divided into two cuts

### DTB (dwell time)

The dwell time at final drilling depth (chip breaking) is programmed in DTB in seconds or revolutions of the main spindle.

### DTS (dwell time)

The dwell time at the starting point is only performed if VARI=1 (chip removal).

### FRF (feedrate factor)

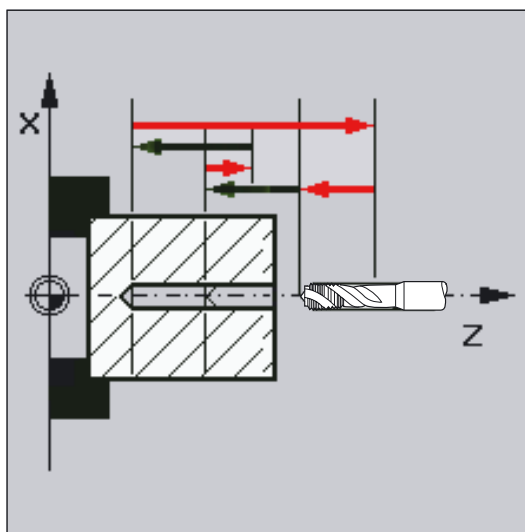
With this parameter, you can specify a reduction factor for the active feedrate which only applies to the approach to the first drilling depth in the cycle.

If the FRF programmed is too large, no alarm is output. The factor is limited to 1 within the cycle.

### VARI (machining type)

If parameter VARI=0 is set, the drill retracts by the programmed retraction distance after reaching each drilling depth for chip breaking. If VARI=1 (for chip removal), the drill traverses in each case to the reference plane shifted by the amount of the safety clearance.

### \_AXN (tool axis)



By programming the drilling axis via `_AXN`, it is possible to omit the switchover from plane G18 to G17 when the deep-hole drilling cycle is used on turning machines.

The identifiers have the following meanings:

<code>_AXN=1</code>	1 <sup>st</sup> axis of the current plane
<code>_AXN=2</code>	2 <sup>nd</sup> axis of the current plane
<code>_AXN=3</code>	3 <sup>rd</sup> axis of the current plane

For example, to machine a center hole (in Z) in the G18 plane, you program:

G18

`_AXN=1`

#### **`_MDEP` (minimum drilling depth)**

You can define a minimum drilling depth for drill stroke calculations based on a degression factor. If the calculated drilling stroke becomes shorter than the minimum drilling depth, the remaining depth is machined in strokes equaling the length of the minimum drilling depth.

#### **`_VRT` (variable retraction value for chip breakage with `VARI=0`)**

You can program the retraction path for chip breaking.

#### **`_DTD` (dwell time at final drilling depth)**

The dwell time at final drilling depth can be entered in seconds or revolutions.

#### **`_DIS1` (programmable limit distance for `VARI=1`)**

The limit distance after re-insertion in the hole can be programmed.

The limit distance is calculated within the cycle as follows:

- Up to a drilling depth of 30 mm, the value is set to 0.6 mm.
- For larger drilling depths, the limit distance is the result of  $(RFP + SDIS - \text{current depth}) / 50$ . If this calculated value  $> 7$ , a limit of 7 mm, maximum, is applied.

---

#### **Note**

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, see Drilling, centering – CYCLE81.

---

#### **See also**

Drilling, centering - CYCLE81 (Page 44)

## 2.1.6 Rigid tapping - CYCLE84

### Function

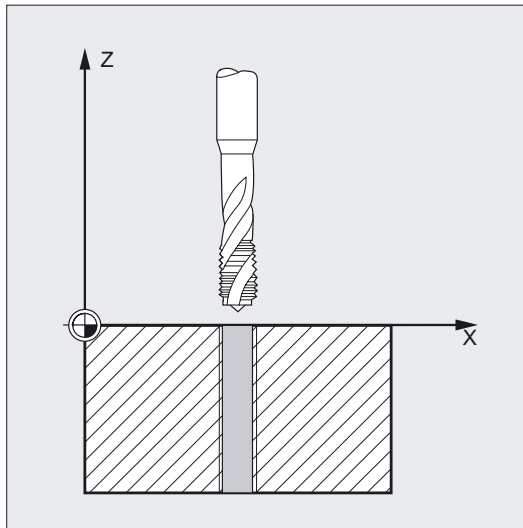
The tool drills at the programmed spindle speed and feedrate to the entered final thread depth.

CYCLE84 can be used to make tapped holes without compensating chuck.

The cycle is also capable of performing tapping operations in several stages (deep-hole drilling).

#### NOTICE

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.



### Programming

CYCLE84 (RTP, RFP, SDIS, DP, DPR, DTB, SDAC, MPIT, PIT, POSS, SST, SST1, \_AXN, \_PTAB, \_TECHNO, \_VARI, \_DAM, \_VRT)

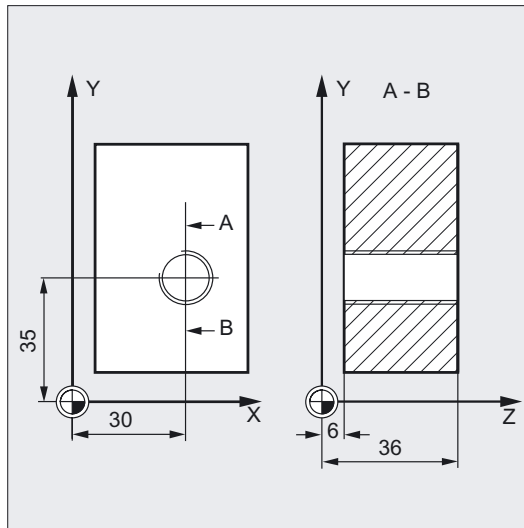
### Parameter

Parameter	Data type	Meaning
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DTB	real	Dwell time at thread depth (chip breaking)

Parameter	Data type	Meaning
SDAC	integer	Direction of rotation after end of cycle
		Values: 3, 4 or 5
MPIT	real	Pitch as a thread size (signed):
		Range of values: 3: (for M3) to 48: (for M48), the sign determines the direction of rotation in the thread
PIT	real	Pitch as a value (signed)
		Range of values: 0.001 ... 2000.000 mm), the sign determines the direction of rotation in the thread: if _PTAB=0 or 1: in mm (as previously) if _PTAB=2: in thread grooves per inch if _PTAB=3: in inches/rotation
POSS	real	Spindle position for oriented spindle stop in the cycle (in degrees)
SST	real	Speed for tapping
SST1	real	Speed for retraction
_AXN	integer	Tool axis
		Values: 1: 1st geometrical axis 2: 2nd geometrical axis otherwise 3rd geometrical axis
_PTAB	integer	Evaluation of thread pitch PIT
		Values: 0: corresponds to programmed measuring system inch/metric 1: pitch in mm 2: pitch in thread grooves per inch 3: pitch in inches/rotation
_TECHNO	integer	Technological settings
		Values: UNITS DIGIT: Exact stop behavior 0: as programmed before cycle call 1: (G601) 2: (G602) 3: (G603)
		TENS DIGIT: Feed-forward control 0: as programmed before cycle call 1: with feed-forward control (FFWON) 2: without feed-forward control (FFWOF)
		HUNDREDS DIGIT: Acceleration 0: as programmed before cycle call 1: axis acceleration with jerk limitation (SOFT) 2: abrupt axis acceleration (BRISK) 3: reduced axis acceleration (DRIVE)
		THOUSANDS DIGIT: Behavior with MCALL 0: reactivate spindle operation (for MCALL) 1: remain in position-controlled operation (for MCALL)
_VARI	integer	Machining type
		Values: 0: tapping in one pass 1: deep-hole tapping with chip breakage 2: deep-hole tapping with chip removal
_DAM	real	Incremental drilling depth Range of values: 0 ≤ max. value
_VRT	real	Variable retraction value for chip breakage Range of values: 0 ≤ max. value

### Example of rigid tapping

A thread is tapped without compensating chuck at position X30 Y35 in the XY plane; the tapping axis is the Z axis. No dwell time is programmed; the depth is programmed as a relative value. The parameters for the direction of rotation and for the pitch must be assigned values. A metric thread M5 is tapped.



N10 G0 G90 T4 D1	;Specification of technology values
N20 G17 X30 Y35 Z40	;Approach drilling position
N30 CYCLE84 (40, 36, 2, , 30, , 3, 5, ->	;Cycle call, parameter PIT has been
->, 90, 200, 500)	;omitted, no indication of absolute
	;depth, no dwell time,
	;Spindle stop at 90 degrees,
	;rotational speed during tapping is 200,
	;Speed for retraction is 500
N40 M30	;Program end

#### Note

-> means: it must be programmed in a single block

#### Note

For tapping with compensating chuck, a separate cycle CYCLE840 is provided.

### See also

Tapping with compensating chuck - CYCLE840 (Page 65)



## Sequence

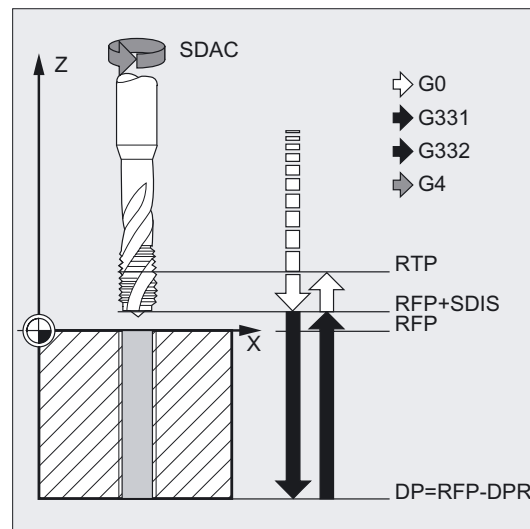
### Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

### The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0
- Oriented spindle stop with SPOS (value in POSS parameter) and transfer of spindle to axis operation.
- Tapping to final drilling depth with G331 and speed of rotation SST.
- Execute dwell time at thread depth (DTB parameter).
- Retraction to the reference plane brought forward by the safety clearance with G332, speed of rotation SST1 and direction of rotation reversal.
- Retraction of the retraction plane with G0, by rewriting the last programmed spindle speed of rotation before the cycle call and the direction of rotation programmed under SDAC, the spindle operation is reinitiated (if this is programmed as such in the \_TECHNO parameter).

## Explanation of the parameters



### DTB (dwell time)

The dwell time must be programmed in seconds. When tapping blind holes, it is recommended that you omit the dwell time.

### SDAC (direction of rotation after end of cycle)

Under SDAC, the direction of rotation after end of cycle is programmed. For tapping, the direction is changed automatically by the cycle.

### MPIT and PIT (as thread size and as 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 → right (same as M3)
- Negative value → left (same as 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)

In the cycle, the spindle is oriented and stopped, and brought into position control before tapping with the SPOS command.

The spindle position for this spindle stop is programmed under POSS.

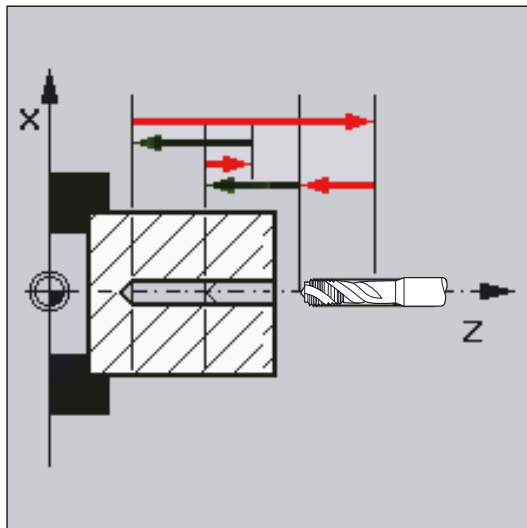
### SST (speed)

Parameter SST contains the spindle speed for the tapping block with G331.

### SST1 (speed for retraction)

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.

### \_AXN (tool axis)



By programming the drilling axis via \_AXN, it is possible to omit the switchover from plane G18 to G17 when the deep-hole tapping cycle is used on turning machines.

The identifiers have the following meanings:

_AXN=1	1st axis of the current plane
_AXN=2	2nd axis of the current plane
_AXN=3	3rd axis of the current plane

For example, to machine a tapped hole (in Z) in the G18 plane, you program:

G18

\_AXN=1

#### **\_PTAB (Evaluation of thread pitch PIT)**

Parameter \_PTAB determines the unit of measurement of the thread pitch.

- 0 = According to programmed system of units inch/metric
- 1 = Pitch in mm
- 2 = Pitch in thread starts per inch
- 3 = Pitch in inches/revolution

This parameter is needed in connection with the option for selecting different thread tables in the cycles support.

#### **\_TECHNO (Technological settings)**

Parameter \_TECHNO can be set to select technological characteristics for tapping operations.

Possible settings: see CYCLE84 parameter table

With the "Technology" "Yes" input box, both the machine tool manufacturer and the operator/programmer can make technology adjustments for tapping.

- **Customization by the machine manufacturer**
  - Requirements: Manufacturer password is set, input box "Technology" = "Yes"
  - When the input cycle screen form CYCLE84 is opened, the parameters are preassigned with the value of GUD7 variable \_SC\_MASK[0]. When the parameters are changed, the values are written directly to this GUD7 variable (see "Cycle support in program editor" in General section)
  - This enables the machine manufacturer to customize the default settings to the characteristics of his machine.
- **Customization by the operator/programmer**
  - Requirements: Manufacturer password is deleted, input box "Technology" = "Yes"
  - The modified data are used to generate CYCLE84. The machine manufacturer settings will take effect the next time a tapping cycle is called.

Input field "Technology" and the following input fields are not shown if the keyswitch is in position 0 or 1.

#### **Deep-hole tapping \_VARI, \_DAM, \_VRT**

With the \_VARI parameter, it is possible to distinguish between simple tapping (\_VARI = 0) and deep-hole tapping (\_VARI ≠ 0).

## 2.1 Drilling cycles

In conjunction with deep-hole tapping, it is possible to choose between chip breaking (retraction by variable distance from current drilling depth, parameter `_VRT`, `_VARI` = 1) and chip removal (withdrawal from reference plane `_VARI` = 2). These functions work analogously to the normal deep-hole drilling cycle `CYCLE83`.

The incremental drilling depth for one pass is specified via parameter `_DAM`. The cycle internally calculates the intermediate depth as follows:

- The programmed incremental drilling depth is executed in each step until the rest up to the final drilling depth is less than ( $<$ )  $2 \cdot \text{\_DAM}$
- The remaining drilling depth is halved and executed in two steps. Thus, the minimum drilling depth is not smaller than  $\text{\_DAM}/2$ .

---

### Note

The direction of rotation when tapping in the cycle is always reversed automatically.

Within the cycle, the programmed direction of rotation in the thread can be reversed via `_ZSD[7]`. With this setting, the cycle on turning machines can be adjusted to the C-axis direction described in DIN 66217,

`_ZSD[7]=1`: The programmed sign of the pitch is reversed within the cycle.

`_ZSD[7]=0`: The programmed sign of the pitch is retained within the cycle.

---

### Note

For an explanation of the parameters `RTP`, `RFP`, `SDIS`, `DP`, and `DPR`, see Drilling, centering – `CYCLE81`.

---

## See also

Drilling, centering - `CYCLE81` (Page 44)

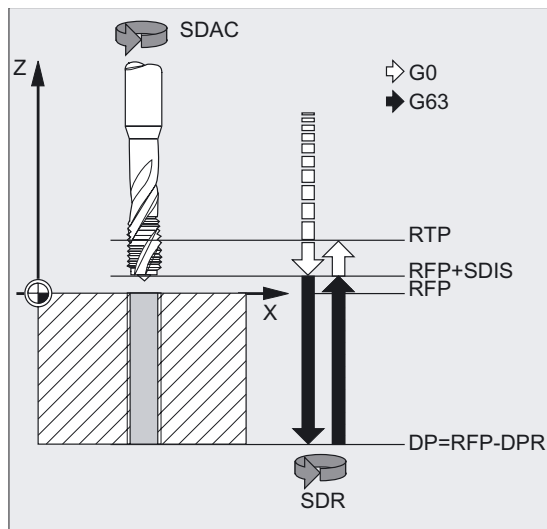
### 2.1.7 Tapping with compensating chuck - CYCLE840

#### Function

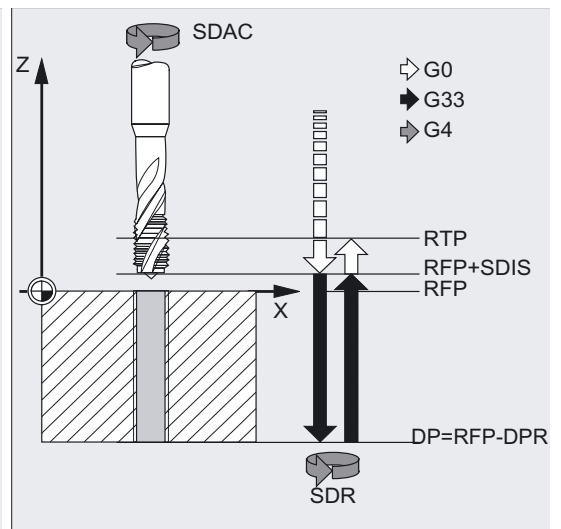
The tool drills at the programmed spindle speed and feedrate to the entered final thread depth.

With this cycle, tapped holes with compensating chuck can be made as follows:

#### Without encoder



#### With encoder



#### Programming

CYCLE840 (RTP, RFP, SDIS, DP, DPR, DTB, SDR, SDAC, ENC, MPIT, PIT, \_AXN, \_PTAB, \_TECHNO)

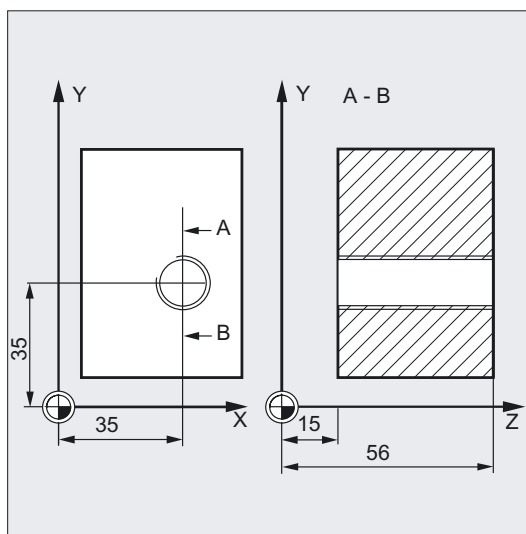
#### Parameter

Parameter	Data type	Meaning
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DTB	real	Dwell time at thread depth: always in effect if >0 programmed Value range: 0<=DTB
SDR	real	Direction of rotation for retraction
		Values: 0: (automatic reversal of direction of rotation) 3 or 4: (for M3 or M4)

2.1 Drilling cycles

Parameter	Data type	Meaning
SDAC	integer	Direction of rotation after end of cycle
		Values: 3, 4 or 5: (for M3, M4, or M5)
ENC	integer	Tapping with/without encoder
		Values: 0: with encoder, without dwell time 1: without encoder, program feed rate before cycle 11: without encoder, calculate feed rate during cycle 20: with encoder, with dwell time
MPIT	real	Thread pitch as thread size
		Range of values: 3 (for M3) ... 48 (for M48)
PIT	real	Thread pitch as a value
		Range of values: 0.001 ... 2000.000 mm), the sign determines the direction of rotation in the thread: if _PTAB=0 or 1: in mm if _PTAB=2: in thread grooves per inch if _PTAB=3: in inches/rotation
_AXN	integer	Tool axis
		Values: 1: 1st geometrical axis 2: 2nd geometrical axis otherwise 3rd geometrical axis
_PTAB	integer	Evaluation of thread pitch PIT
		Values: 0: corresponds to programmed measuring system inch/metric 1: pitch in mm 2: pitch in thread grooves per inch 3: pitch in inches/rotation
_TECHNO	integer	Technological settings
		Values: UNITS DIGIT: Exact stop behavior 0: as programmed before cycle call 1: (G601) 2: (G602) 3: (G603)  TENS DIGIT: Feed-forward control 0: as programmed before cycle call 1: with feed-forward control (FFWON) 2: without feed-forward control (FFWOF)

## Example



### Tapping without encoder

In this program, a thread is tapped without encoder at position X35 Y35 in the XY plane; the tapping axis is the Z axis. The parameters SDR and SDAC for the direction of rotation must be assigned; parameter ENC is assigned the value 1, the value for the depth is the absolute value. Pitch parameter PIT can be omitted. A compensating chuck is used in machining.

```

N10 G90 G0 D2 T2 S500 M3           ;Specification of technology values
N20 G17 X35 Y35 Z60                 ;Approach drilling position
N30 G1 F200                          ;Setting the path feedrate
N40 CYCLE840 (59, 56, , 15, , 1, 4, ;Cycle call, dwell time 1 s, SDR=4, SDAC=3,
3, 1)                               ;no safety clearance, parameters MPIT, PIT
                                   ;are not programmed, i.e., the pitch
                                   ;results from the relationship of
                                   ;user-programmed F and S values
N50 M30                             ;Program end

```

### Tapping with encoder

In this program, a thread is tapped with encoder at position X35 Y35 in the XY plane. The drilling axis is the Z axis. The pitch parameter must be defined, automatic reversal of the direction of rotation is programmed. A compensating chuck is used in machining.

```

DEF INT SDR=0                       ;Definition of parameters with
DEF REAL PIT=3.5                     ;Value assignments
N10 G90 G0 D2 T2 S500 M4             ;Specification of technology values
N20 G17 X35 Y35 Z60                 ;Approach drilling position
N30 CYCLE840 (59, 56, , 15, , , , , ;Cycle call, without safety clearance, with
, , PIT)                             ;absolute depth specification, SDAC ENC, MPIT
                                   ;omitted (e.g., have a value of zero)
N40 M30                             ;Program end

```

## Sequence

### Tapping with compensating chuck without encoder (ENC = 1/11)

#### 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 reference plane shifted by the amount of the safety clearance with G0
- Tapping to final drilling depth with G63.
- Execute dwell time at thread depth (DTB parameter).
- Retraction to the reference plane brought forward by the safety clearance with G63.
- Retraction to the retraction plane with G0.

<b>NOTICE</b>
The spindle override must be set to 100%!

### Tapping with compensating chuck with encoder (ENC = 0/20)

#### 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 reference plane shifted by the amount of the safety clearance with G0
- Tapping to final drilling depth with G33.
- Retraction to the reference plane brought forward by the safety clearance with G33.
- Execute dwell time at reference plane + safety clearance (DTB parameter) (only when ENC=20).
- Retraction to the retraction plane with G0.

## Explanation of the parameters

### DTB (dwell time)

The dwell time must be programmed in seconds. It works in parameter ENC depending on the technological variant you selected.

### SDR (direction of rotation for retraction)

The direction of rotation for retraction when tapping is programmed in parameter SDR. When SDR=0, the spindle direction is reversed automatically.

### SDAC (direction of rotation)

Because the cycle can also be called modally, it needs a direction of rotation for execution of additional tapped 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 is of no significance in the cycle and can be omitted from the parameterization.



**Note**

For information about modal calling of drilling cycles, see Section 2.2.

---

**ENC (tapping)**

To perform tapping without encoder although an encoder exists, parameter ENC must be set to x1.

However, if an encoder is not present, and the parameter has a value of x0, it is not taken into account during the cycle.

- **Tapping without encoder, with indication of pitch**

You can calculate the correlation between feedrate and speed via the programmed pitch within the cycle for tapping without encoder (G63 thread). The speed must be programmed before the cycle is called.

You can specify the pitch either via MPIT (metric thread unit) or via PIT (thread pitch as value) just as for tapping without encoder.

The feedrate is then calculated in the cycle from the pitch and the speed. After the cycle has ended, the last programmed feedrate applies again.

Programming:

ENC=11, program the pitch in MPIT or PIT

- **Tapping with encoder, with dwell time**

As an alternative, you can program a dwell time in the parameter DTB for tapping with encoder (G33 thread). It applies after tapping and before retraction to the retraction plane RTP, and is required for machines with unfavorable spindle dynamics.

Programming:

ENC=20, enter the dwell time in parameter DTB

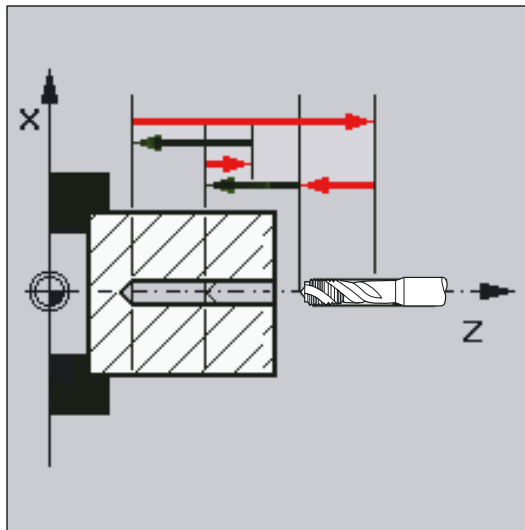
**MPIT and PIT (as thread size and as value)**

The parameter for the 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.

### AXN (tool axis)



By programming the drilling axis via `_AXN`, it is possible to omit the switchover from plane G18 to G17 when the deep-hole tapping cycle is used on turning machines. The identifiers have the following meanings:

<code>_AXN=1</code>	1 <sup>st</sup> axis of the current plane
<code>_AXN=2</code>	2 <sup>nd</sup> axis of the current plane
<code>_AXN=3</code>	3 <sup>rd</sup> axis of the current plane

For example, to machine a center hole (in Z) in the G18 plane, you program:

G18

`_AXN=1`

### `_PTAB` (evaluation of thread pitch PIT)

Parameter `_PTAB` determines the unit of measurement of the thread pitch.

- 0 = According to programmed system of units inch/metric
- 1 = Pitch in mm
- 2 = Pitch in thread starts per inch
- 3 = Pitch in inches/revolution

This parameter is needed in connection with the option for selecting different thread tables in the cycles support.

### `_TECHNO` (technological settings)

Parameter `_TECHNO` can be set to select technological characteristics for tapping operations.

Possible settings are: see CYCLE840 parameter table

With the "Technology" "Yes" input box, both the machine tool manufacturer and the operator/programmer can make technology adjustments for tapping.

- **Customization by the machine manufacturer**

- Requirements: Manufacturer password is set, input box "Technology" = "Yes".
- When the input cycle screen form CYCLE840 is opened, the parameters are preassigned with the value of GUD7 variable \_SC\_MASK[1]. When the parameters are changed, the values are written directly to this GUD7 variable.
- This enables the machine manufacturer to customize the default settings to the characteristics of his machine.

- **Customization by the operator/programmer**

- Requirements: Manufacturer password is deleted, input box "Technology" = "Yes"
- The modified data are used to generate CYCLE840. The machine manufacturer settings will take effect the next time a tapping cycle is called.

Input field "Technology" and the following input fields are not shown if the keyswitch is in position 0 or 1.

---

**Note**

Depending on the setting in machine data 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.

---

**Note**

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, see Drilling, centering – CYCLE81.

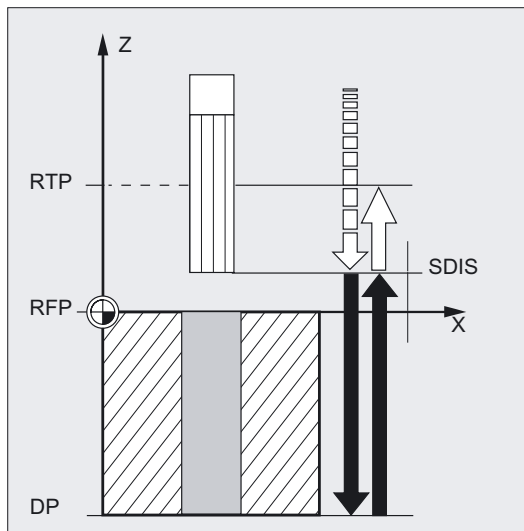
---

**See also**

Drilling, centering - CYCLE81 (Page 44)

## Function

This cycle can be used for reaming of bore holes.



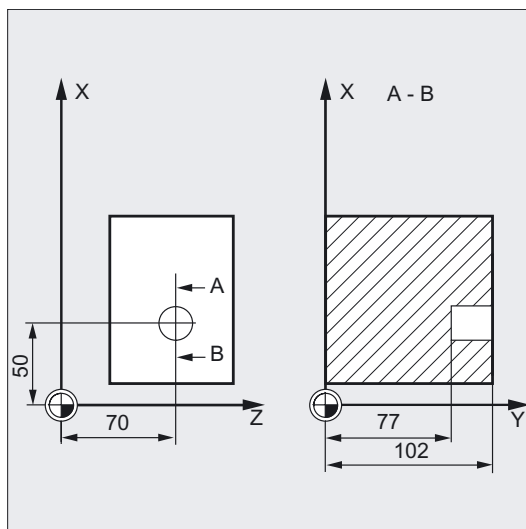
CYCLE85 (RTP, RFP, SDIS, DP, DPR, DTB, FFR, RFF)

## Parameter

Parameter	Data type	Meaning
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DTB	real	Dwell time at thread depth (chip breaking)
FFR	real	Feedrate
RFF	real	Retraction feedrate

### Example of first boring

CYCLE85 is called at position Z70 X50 in the ZX plane. The boring axis is the Y axis. The value for the final drilling depth in the cycle call is programmed as a relative value; no dwell time is programmed. The workpiece upper edge is at Y102.



```

DEF REAL FFR, RFF, RFP=102,           ;Definition of the parameters and
DPR=25,SDIS=2                         ;value assignments
N10 G0 FFR=300 RFF=1.5*FFR S500 M4    ;Specification of technology values
N20 G18 T1 D1 Z70 X50 Y105            ;Approach drilling position
N21 M6
N30 CYCLE85 (RFP+3, RFP, SDIS, , DPR, ,-> ;Cycle call, no dwell time
-> FFR, RFF)                          ;programmed
N40 M30                               ;Program end

```

### Note

-> means: it must be programmed in a block

### Sequence

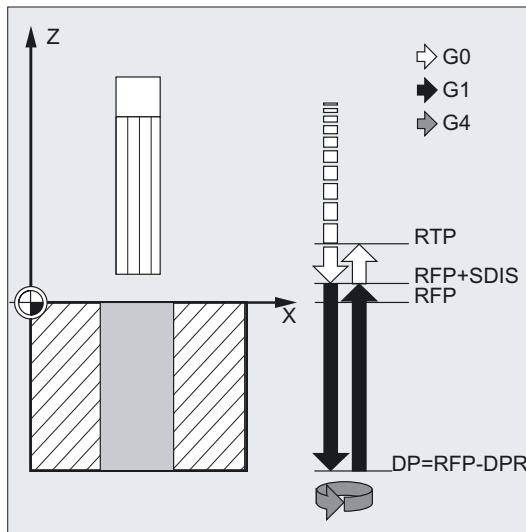
#### Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

#### The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0
- Traversing to the final drilling depth with G1 and at the feedrate programmed under the parameter FFR
- Execution of dwell time at final drilling depth.
- Retraction to the reference plane brought forward by the safety clearance with G1, and the retraction feedrate assigned under parameter RFF.
- Retraction to the retraction plane with G0.

### Explanation of the parameters



#### DTB (dwell time)

The dwell time to the final drilling depth (chip breaking) is programmed under DTB in seconds.

#### FFR (feedrate)

The feedrate value programmed under FFR is active in drilling.

#### RFF (retraction feedrate)

The feedrate value assigned to RFF is active for retraction from the plane.

---

#### Note

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, see Drilling, centering – CYCLE81.

---

### See also

Drilling, centering - CYCLE81 (Page 44)

## 2.1.9 Boring 2 - CYCLE86

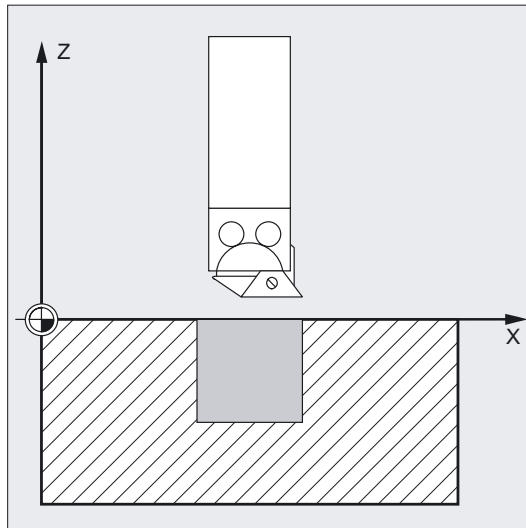
### Function

The tool drills at the programmed spindle speed and feedrate velocity up to the entered drilling depth.

With Boring 2, oriented spindle stop is activated with the SPOS command once the drilling depth has been reached. Then, the programmed retraction positions are approached in rapid traverse and, from there, the retraction plane.

#### NOTICE

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

CYCLE86 (RTP, RFP, SDIS, DP, DPR, DTB, SDIR, RPA, RPO, RPAP, POSS)

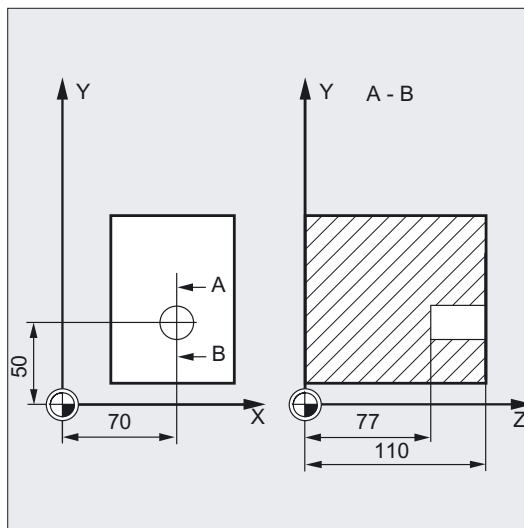
### Parameter

Parameter	Data type	Meaning
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
DTB	real	Dwell time at thread depth (chip breaking)
SDIR	integer	Direction of rotation

Parameter	Data type	Meaning
		Values: 3: (for M3) 4: (for M4)
RPA	real	Retraction path along the abscissa of the active plane (incremental, enter with sign)
RPO	real	Retraction path along the ordinate of the active plane (incremental, enter with sign)
RPAP	real	Retraction path along the boring axis (incremental, enter with sign)
POSS	real	Spindle position for oriented spindle stop in the cycle (in degrees)

### Example of second boring

CYCLE86 is called at position X70 Y50 in the XY 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 sec. The top edge of the workpiece is positioned at Z110. In the cycle, the spindle is to rotate with M3 and to stop at 45 degrees.



```

DEF REAL DP, DTB, POSS          ;Definition of parameters
N10 DP=77 DTB=2 POSS=45         ;Value assignments
N20 G0 G17 G90 F200 S300        ;Specification of technology values
N30 D1 T3 Z112                  ;Approach retraction plane
N40 X70 Y50                     ;Approach drilling position
N50 CYCLE86 (112, 110, , DP, , DTB, 3,-> ;Cycle call with absolute drilling depth
-> -1, -1, +1, POSS)
N60 M30                         ;Program end

```

### Note

-> means: it must be programmed in a block



## Sequence

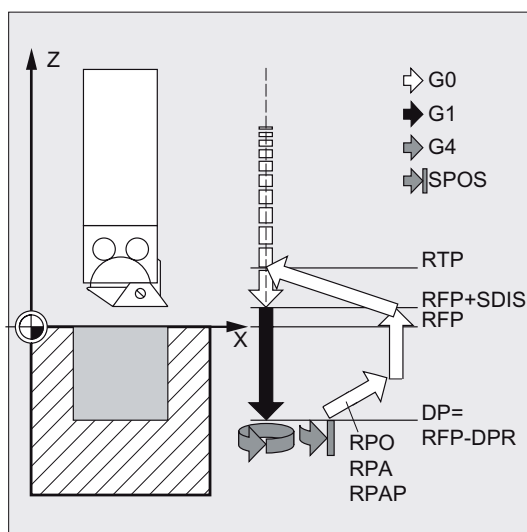
### Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

### The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0
- Traversing to the final drilling depth at G1 and the feedrate programmed before the cycle call.
- Dwell time at final drilling depth is executed.
- Oriented spindle hold at the spindle position programmed under POSS.
- Traverse retraction path in up to 3 axes at G0.
- Retraction to the reference plane brought forward by the safety clearance with G0.
- Retraction to the retraction plane at G0 (initial drilling position in both axes of the plane).

## Explanation of the parameters



### 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 abscissa)

Under this parameter, you define a retraction movement in the abscissa, which is executed after the final drilling depth has been reached and oriented spindle stop has been performed.

## 2.1 Drilling cycles

### **RPO (retraction path, in ordinate)**

With this parameter, you define a retraction movement in the ordinate, which is executed after the final drilling depth has been reached and oriented spindle stop has been performed.

### **RPAP (retraction path, in drilling axis)**

You use this parameter to define a retraction movement along the drilling 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**

With the SPOS command, you can perform an oriented spindle stop of the active master spindle. The angular value is programmed using a transfer parameter.

### **Turning machine without Y axis**

You can now use the cycle CYCLE86 for turning machines without a Y axis. Retraction to drilling depth is then traversed in two axes only. If a retraction path was programmed for the 3rd axis, it would be ignored.

If you call the cycle without Y axis in the G18 plane, the following alarm appears: 61005 "3rd geometry axis does not exist", as the Y axis would be the drilling axis in this case.

---

#### **Note**

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, see Drilling, centering – CYCLE81.

---

## **See also**

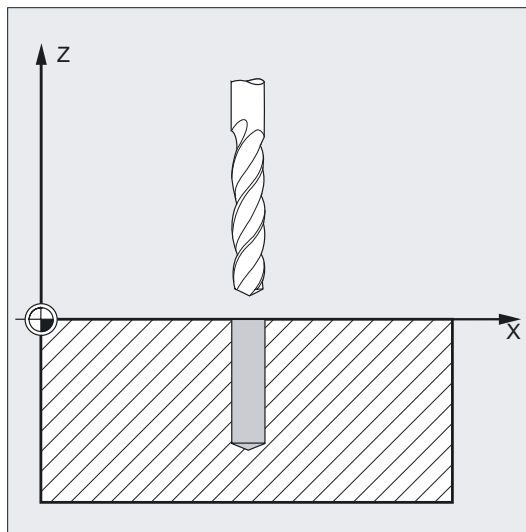
Drilling, centering - CYCLE81 (Page 44)

### 2.1.10 Boring 3 - CYCLE87

#### 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.



#### Programming

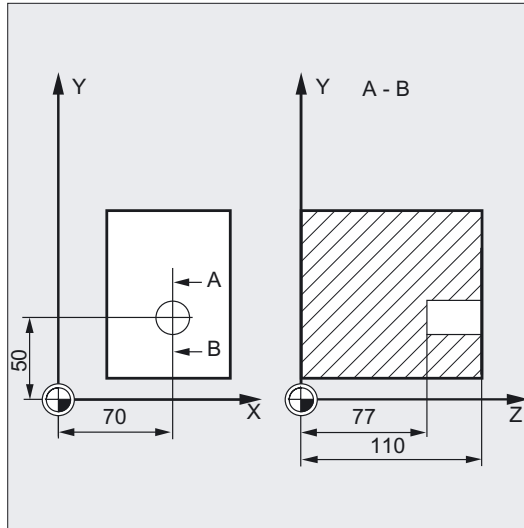
CYCLE87 (RTP, RFP, SDIS, DP, DPR, SDIR)

#### Parameter

Parameter	Data type	Meaning
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to the reference plane (enter without sign)
SDIR	integer	Direction of rotation
		Values: 3: (for M3) 4: (for M4)

### Example of 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.



DEF REAL DP, SDIS	;Definition of parameters
N10 DP=77 SDIS=2	;Value assignments
N20 G0 G17 G90 F200 S300	;Specification of technology values
N30 D1 T3 Z113	;Approach retraction plane
N40 X70 Y50	;Approach drilling position
N50 CYCLE87 (113, 110, SDIS, DP, , 3)	;Cycle call with programmed
	;direction of spindle rotation M3
N60 M30	;Program end

### Sequence

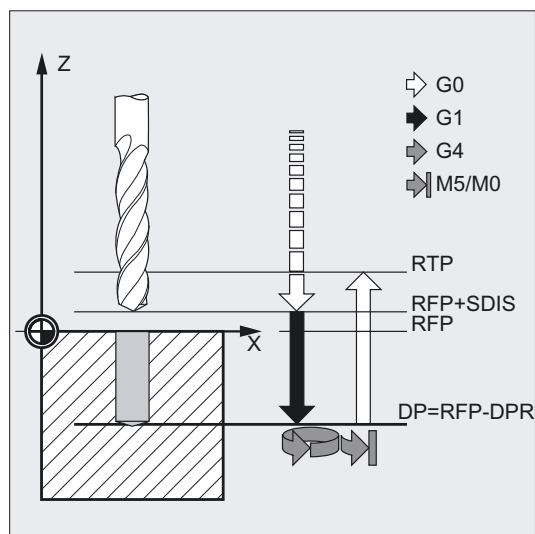
#### Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

#### The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0
- Traversing to the final drilling depth at G1 and the feedrate programmed before the cycle call.
- Spindle stop with M5
- Press NC-START button.
- Retraction to the retraction plane with G0.

## Explanation of the parameters



### 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.

### Note

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, see Drilling, centering – CYCLE81.

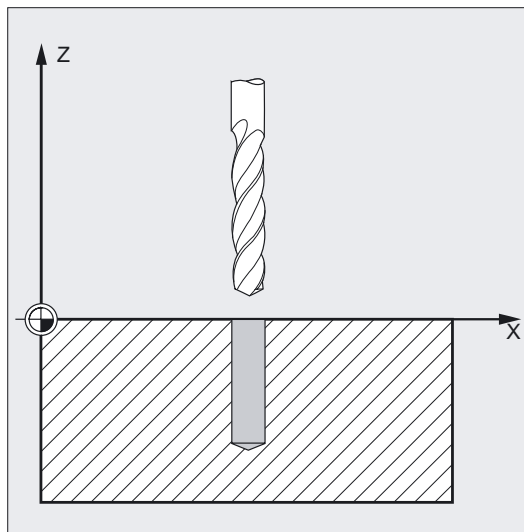
## See also

Drilling, centering - CYCLE81 (Page 44)

### 2.1.11 Boring 4 - CYCLE88

#### 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 traverses the outward movement at rapid traverse until the retraction plane is reached.



#### Programming

CYCLE88 (RTP, RFP, SDIS, DP, DPR, DTB, SDIR)

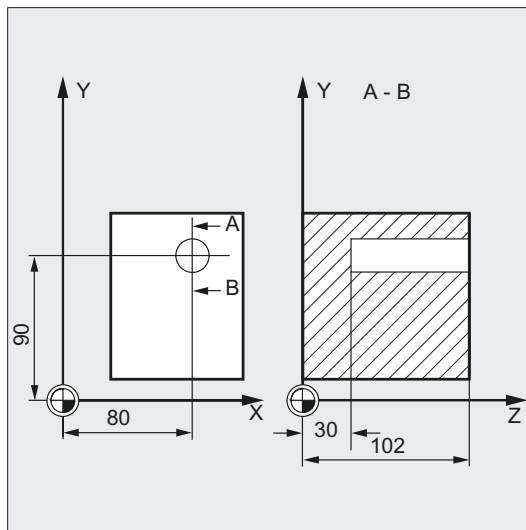
#### Parameter

Parameter	Data type	Meaning
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
SDIR	integer	Direction of rotation
		Values: 3: (for M3) 4: (for M4)

### Example of fourth boring

CYCLE88 is called at position X80 Y90 in the XY plane. The drilling axis is the Z axis. The safety clearance is programmed at 3 mm. The final drilling depth is specified relative to the reference plane.

M4 is active in the cycle.



```

DEF REAL RFP, RTP, DPR, DTB, SDIS          ;Definition of parameters
N10 RFP=102 RTP=105 DPR=72 DTB=3 ->        ;Value assignments
-> SDIS=3
N20 G17 G90 T1 D1 F100 S450                ;Specification of technology values
N21 M6
N30 G0 X80 Y90 Z105                        ;Approach drilling position
N40 CYCLE88 (RTP, RFP, SDIS, , DPR, ->     ;Cycle call with programmed
-> DTB, 4)                                  ;direction of spindle rotation M4
N50 M30                                     ;Program end

```

### Note

-> means: it must be programmed in a block

### Sequence

#### Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

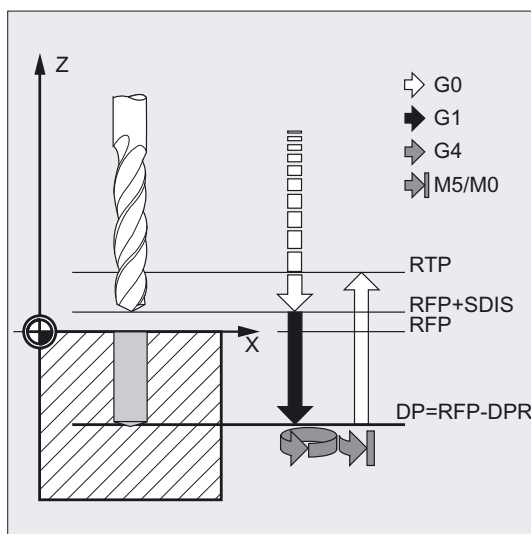
#### The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0
- Traversing to the final drilling depth at G1 and the feedrate programmed before the cycle call.
- Dwell time at final drilling depth.

## 2.1 Drilling cycles

- Spindle stop with M5 ( $\_ZSD[5]=1$ ) or
- Spindle and program stop with M5 M0 ( $\_ZSD[5]=0$ ). After program stop, press the NC START key.
- Retraction to the retraction plane with G0.

### Explanation of the parameters



#### 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.

#### Note

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, see Drilling, centering – CYCLE81.

For an explanation of the cycle setting data  $\_ZSD[5]$ , see Requirements for milling cycles.

### See also

Drilling, centering - CYCLE81 (Page 44)

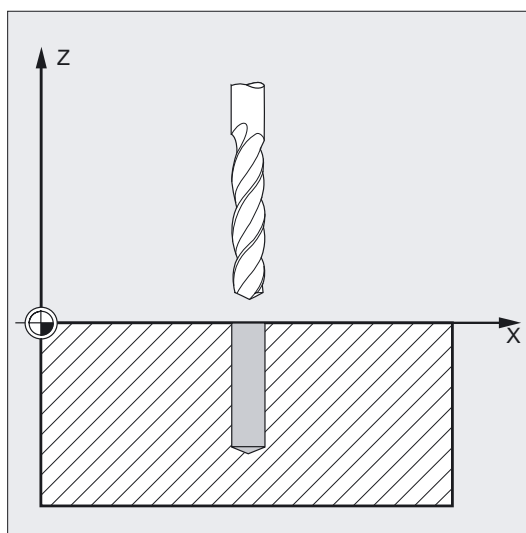
Requirements (Page 101)



## 2.1.12 Boring 5 - CYCLE89

### Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. When the final drilling depth is reached, a dwell time can be programmed.



### Programming

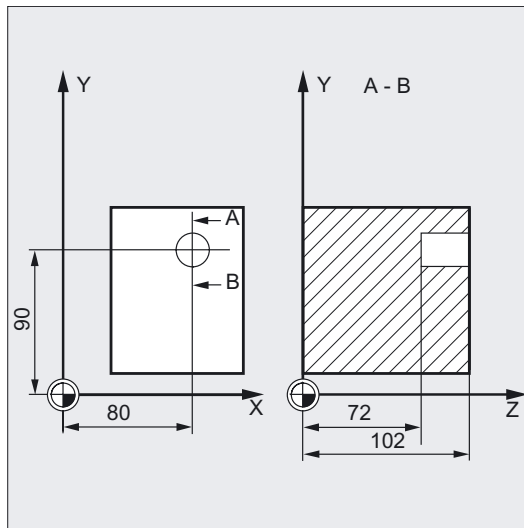
CYCLE89 (RTP, RFP, SDIS, DP, DPR, DTB)

### Parameter

Parameter	Data type	Meaning
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

### Example of 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.



```

DEF REAL RFP, RTP, DP, DTB           ;Definition of parameters
RFP=102 RTP=107 DP=72 DTB=3          ;Value assignments
N10 G90 G17 F100 S450 M4              ;Specification of technology values
N20 G0 T1 D1 X80 Y90 Z107             ;Approach drilling position
N21 M6
N30 CYCLE89 (RTP, RFP, 5, DP, , DTB) ;Cycle call
N40 M30                               ;Program end

```

### Sequence

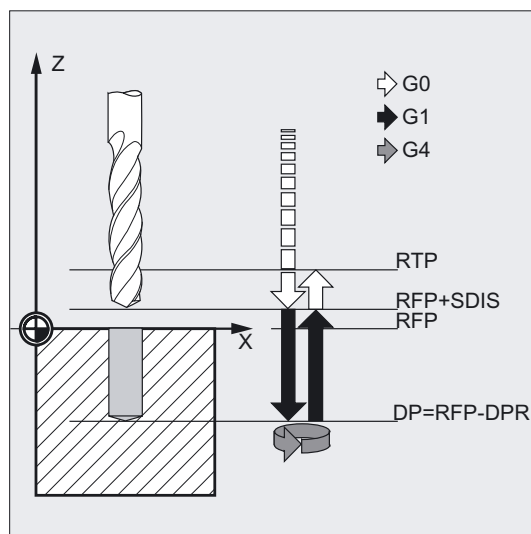
#### Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

#### The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0
- Traversing to the final drilling depth at G1 and the feedrate programmed before the cycle call.
- Dwell time at final drilling depth is executed.
- 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



### DTB (dwell time)

The dwell time to the final drilling depth (chip breaking) is programmed under DTB in seconds.

### Note

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, see Drilling, centering – CYCLE81.

## See also

Drilling, centering - CYCLE81 (Page 44)

## 2.2 Modal call of drilling cycles

### Function

In NC programming, subroutines and cycles can be called modally. This feature is of particular importance for drilling cycles.

You generate a modal subroutine call by programming the keyword MCALL (modal subroutine call) in front of the subroutine name. This function causes the subroutine to be called and executed automatically after each block that contains traversing movement.

The function is deactivated by programming MCALL without a following subroutine name or by modally calling another subroutine.

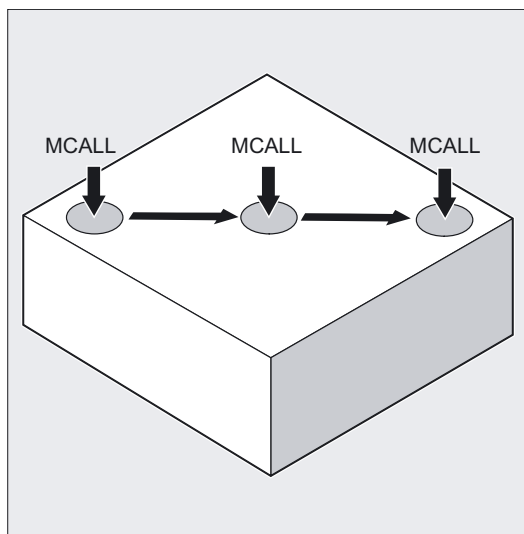
---

### Note

Nesting of modal calls is not permissible, i.e., subroutines that are called modally cannot contain any further modal subroutine calls.

---

Any number of modal drilling cycles can be programmed, the number is not limited to a certain number of G functions reserved for this purpose.



### Programming

Modal call of a subroutine

MCALL

**Example with drilling cycle CYCLE81:**

MCALL CYCLE81 (RTP, RFP, SDIS, DP, DPR)

### Example of row of holes 5

With this program, you can machine a row of holes from 5 tapped holes that are positioned parallel to the Z axis of the ZX plane.

The distance between each of the holes is 20 mm. The row of holes starts at Z20 and X30, the first hole in the row being 20 mm from this point. In this example, the geometry of the row of holes has been programmed without using a cycle.

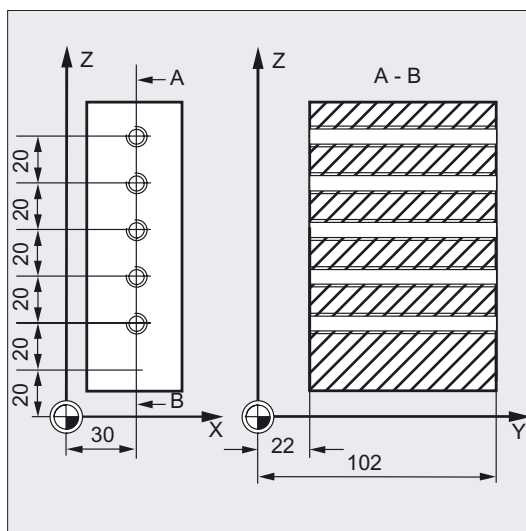
First of all, drilling is performed with cycle CYCLE81 and then with CYCLE84 tapping (rigid). The holes are 80 mm deep. This is the difference between the reference plane and the final drilling depth.

#### Note

##### Explanation of this example

The modal call must be deselected in block N80 because, in the next block, the tool is traversed to a position where no drilling is to be performed.

It is advisable to store the drilling positions for a machining task of this type in a subroutine, which is then called at MA1 or MA2.



```

DEF REAL RFP=102, DP=22, ->           ;Definition of the parameters with
-> RTP=105, PIT=4.2, SDIS             ;value assignments

DEF INT COUNT=1

N10 SDIS=3                             ;Value for safety distance
N20 G90 F300 S500 M3 D1 T1             ;Specification of technology values
N30 G18 G0 Y105 Z20 X30               ;Approach start position
N40 MCALL CYCLE81 (RTP, RFP, ->       ;Modal call of the drilling cycle
-> SDIS, DP)
N50 MA1: G91 Z20                       ;Approach next position

```

N60 COUNT=COUNT+1	; (ZX plane) cycle is executed ; Loop for drilling positions along the row ; of holes
N70 IF COUNT<6 GOTOB MA1	
N80 MCALL	; Deselect modal call
N90 G90 Y105 Z20	; Approach starting position again
N100 COUNT=1	; Set counter to zero
N110 ...	; Change tool
N120 MCALL CYCLE84 (RTP, -> -> RFP, SDIS, DP , , , 3, , -> -> PIT, , 400)	; Modal call of tapping cycle
N130 MA2: G91 Z20	; Next drilling position
N140 COUNT=COUNT+1	; Loop for drilling position of the row of ; holes
N150 IF COUNT<6 GOTOB MA2	
N160 MCALL	; Deselect modal call
N170 G90 X30 Y105 Z20	; Approach starting position again
N180 M30	; Program end

---

**Note**

-> means: it must be programmed in a block.

---

## **2.3 Drilling pattern cycles**

### **2.3.1 Requirements**

#### **Function**

The hole pattern cycles only describe the geometry of an arrangement of drilling holes in the plane. The link to a drilling process is established via the modal call of this drilling cycle before the hole pattern cycle is programmed.

#### **Hole pattern cycles without drilling cycle call**

Hole pattern cycles can also be used for other applications without the first drilling cycle first being called because the hole pattern cycles can be parameterized without reference to the drilling cycle used.

If there was no modal call of the subroutine prior to calling the hole pattern cycle, error message 62100 "No drilling cycle active" appears.

You can acknowledge this error message by pressing the error acknowledgment key and continue the program execution by pressing NC START. The hole pattern cycle will then approach each of the positions calculated from the input data one after the other without calling a subroutine at these points.

#### **Behavior when quantity parameter is zero**

The number of holes in a hole pattern must be parameterized. If the value of the quantity parameter is zero when the cycle is called (or if this parameter is omitted from the parameter list), alarm 61103 "Number of holes is zero" and the cycle is aborted.

#### **Checks in case of limited ranges of input values**

In general, plausibility checks for defining parameters are not made in the hole pattern cycles if this is not declared explicitly for a parameter with a description of the corresponding reaction.

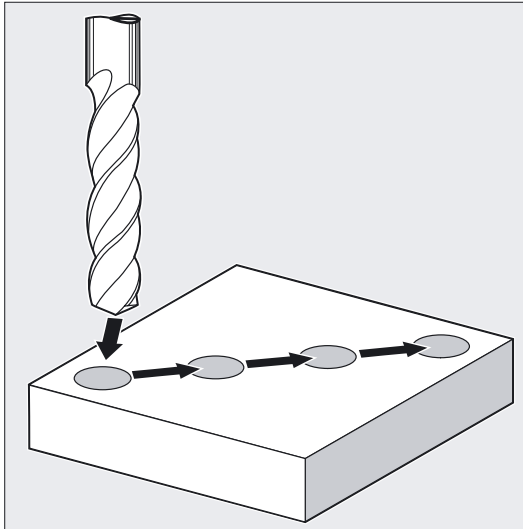
#### **See also**

Modal call of drilling cycles (Page 88)

## 2.3.2 Row of holes - HOLES1

### Function

With this cycle, you can make a row of holes, i.e., a number of drill holes in a straight line.  
The type of drill hole is determined by the drilling cycle that has already been called modally.



### Programming

HOLES1 (SPCA, SPCO, STA1, FDIS, DBH, NUM)

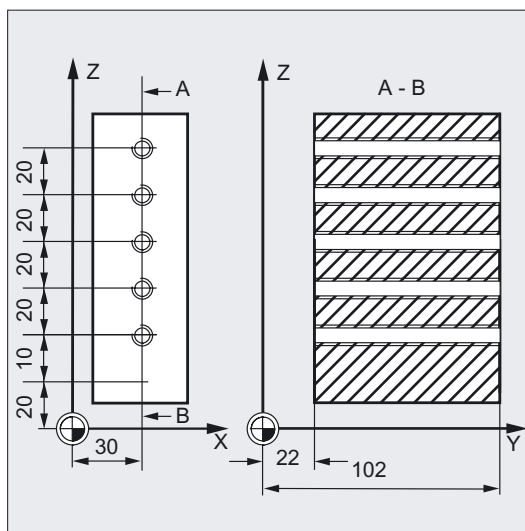
### Parameter

Parameter	Data type	Meaning	
SPCA	real	Abcissa of a reference point on the straight line (absolute)	
SPCO	real	Ordinate of this reference point (absolute)	
STA1	real	Angle to abscissa	
		Range of values:	$-180 < STA1 \leq 180$ degrees
FDIS	real	Distance between the first drill hole and the reference point (enter without sign)	
DBH	real	Distance between the holes (enter without sign)	
NUM	integer	Number of drill holes	



### Example of row of holes

Use this program to machine a row of holes consisting of 5 tapped 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 of all, drilling is performed with cycle CYCLE81 and then with CYCLE84 tapping (rigid). The holes are 80 mm in depth (difference between reference plane and final drilling depth).



```

DEF REAL RFP=102, DP=22, RTP=105          ;Definition of the parameters with
DEF REAL SDIS, FDIS                        ;value assignments

DEF REAL SPCA=30, SPCO=20, ->
-> STA1=0, DBH=20

DEF INT NUM=5

N10 SDIS=3 FDIS=10                        ;Value for safety clearance as well as
                                           ;distance from the first drilling to the
                                           ;reference point

N20 G90 F30 S500 M3 D1 T1                 ;Specification of the technology values for
                                           ;the planing section

N30 G18 G0 Z20 Y105 X30                   ;Approach start position

N40 MCALL CYCLE81 (RTP, RFP, ->           ;Modal call of drilling cycle
-> SDIS, DP)

N50 HOLES1 (SPCA, SPCO, STA1, ->         ;Call of row of holes cycle, beginning with
-> FDIS, DBH, NUM)                       ;the first drill hole, only the drill
                                           ;positions are approached in the cycle

N60 MCALL                                 ;Deselect modal call
...                                       ;Change tool

N70 G90 G0 Z30 Y75 X105                   ;Approach position next to 5th hole

N80 MCALL CYCLE84 (RTP, RFP, ->           ;Modal call of
-> SDIS, DP,, 3, , 4.2, , , 400)         ;tapping cycle

N90 HOLES1 (SPCA, SPCO, STA1, ->         ;Call of row of holes cycle, beginning
-> FDIS, DBH, NUM)                       ;with the 5th drill hole in the row of holes

N100 MCALL                                ;Deselect modal call

N110 M30                                  ;Program end

```

---

**Note**

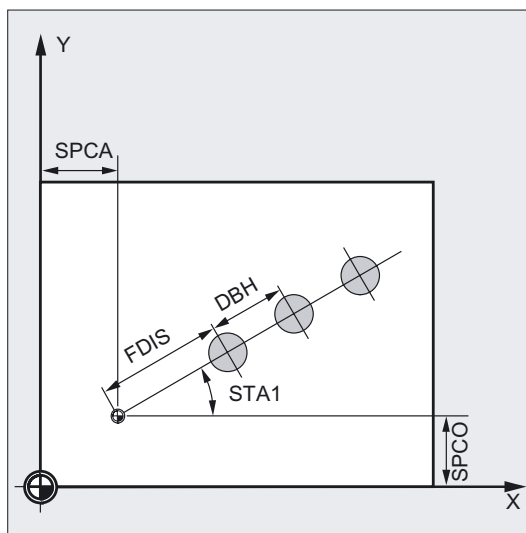
-> means: it must be programmed in a single block

---

**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.

**Explanation of the parameters**



**SPCA and SPCO (reference point abscissa and ordinate)**

One point along the straight line of the row of holes is defined as the reference point for determining the spacing between the holes. The distance to the first hole FDIS is defined from this point.

**STA1 (angle)**

The straight line can be arranged 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 and the abscissa of the workpiece coordinate system that is active when the cycle is called. The angle is entered under STA1 in degrees.

**FDIS and DBH (distance)**

The distance of the first hole and the reference point defined under SPCA and SPCO is programmed with FDIS. The parameter DBH contains the distance between any two holes.

**NUM (number)**

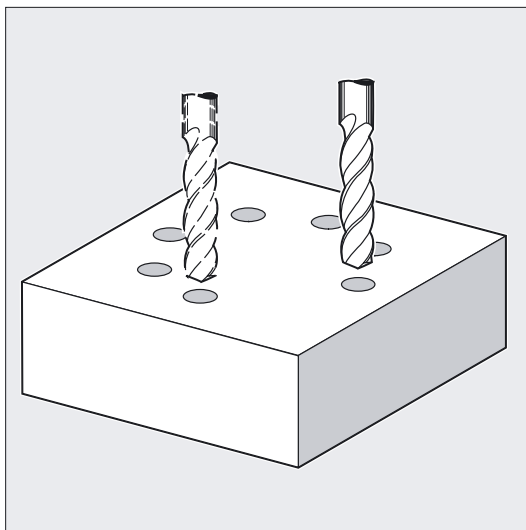
The NUM parameter is used to define the number of holes.

### 2.3.3 Row of holes - HOLES2

#### Function

Use this cycle to machine a circle of holes. The machining plane must be defined before the cycle is called.

The type of drill hole is determined by the drilling cycle that has already been called modally.



#### Programming

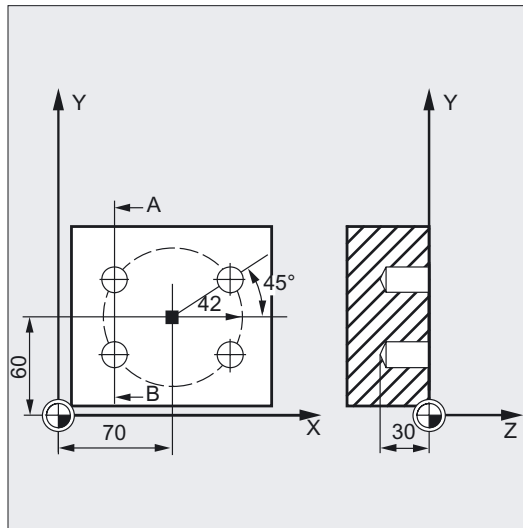
HOLES2 (CPA, CPO, RAD, STA1, INDA, NUM)

#### Parameter

Parameter	Data type	Meaning	
CPA	real	Center point of circle of holes, abscissa (absolute)	
CPO	real	Center point of circle of holes, ordinate (absolute)	
RAD	real	Radius of circle of holes (enter without sign)	
STA1	real	Starting angle	
		Range of values:	$-180 < STA1 \leq 180$ degrees
INDA	real	Incrementing angle	
NUM	integer	Number of drill holes	

### Example of row of holes

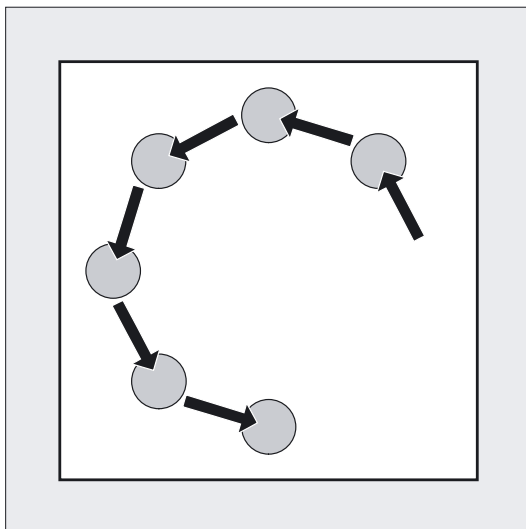
The program uses CYCLE82 to produce 4 holes having a depth of 30 mm. The final drilling depth is specified as a relative value to the reference plane. The circle is defined by the center point X70 Y60 and the radius 42 mm in the XY plane. The starting angle is 45 degrees. The safety clearance in drilling axis Z is 2 mm.



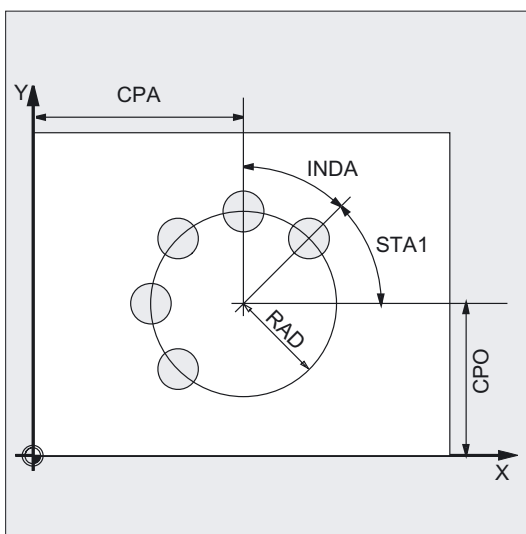
DEF REAL CPA=70,CPO=60,RAD=42,STA1=45	;Definition of parameters with
DEF INT NUM=4	;Value assignments
N10 G90 F140 S710 M3 D1 T40	;Specification of technology values
N20 G17 G0 X50 Y45 Z2	;Approach start position
N30 MCALL CYCLE82 (2, 0,2, , 30)	;Modal call of drilling cycle, without ;dwell time, DP is not programmed
N40 HOLES2 (CPA, CPO, RAD, STA1, , NUM)	;Call of circle of holes, the ;incrementing angle is calculated in ;the cycle, as the INDA parameter has ;been omitted
N50 MCALL	;Deselect modal call
N60 M30	;Program end

## 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 and radius, abscissa, ordinate)

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 defines the angle of rotation between the positive direction of the abscissa in the 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 the INDA parameter 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)**

The NUM parameter defines the number of holes.

**2.3.4 Dot matrix - CYCLE801****Function**

Cycle CYCLE801 can be used to machine a "grid of holes" drilling pattern. The type of drill hole is determined by the drilling cycle that has already been called modally.

**Programming**

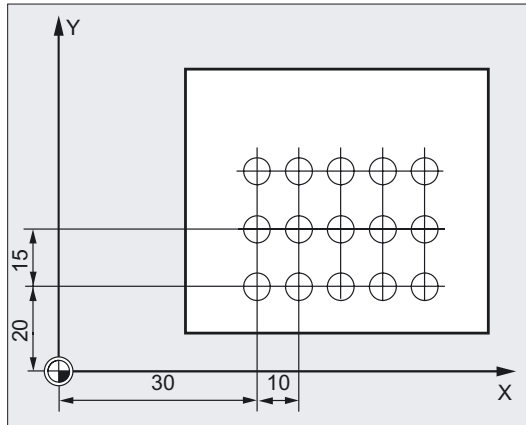
CYCLE801 (\_SPCA, \_SPCO, \_STA, \_DIS1, \_DIS2, \_NUM1, \_NUM2)

**Parameter**

Parameter	Data type	Meaning
_SPCA	real	Reference point for grid of holes in the 1st axis, abscissa (absolute)
_SPCO	real	Reference point for grid of holes in the 2nd axis, ordinate (absolute)
_STA	real	Angle to abscissa
_DIS1	real	Distance between columns (without sign)
_DIS2	real	Distance between rows (without sign)
_NUM1	integer	Number of columns
_NUM2	integer	Number of rows

### Example of dot matrix

Use the CYCLE801 cycle to machine a dot matrix consisting of 15 drill holes in 3 rows and 5 columns. The associated drilling program is called modally beforehand.



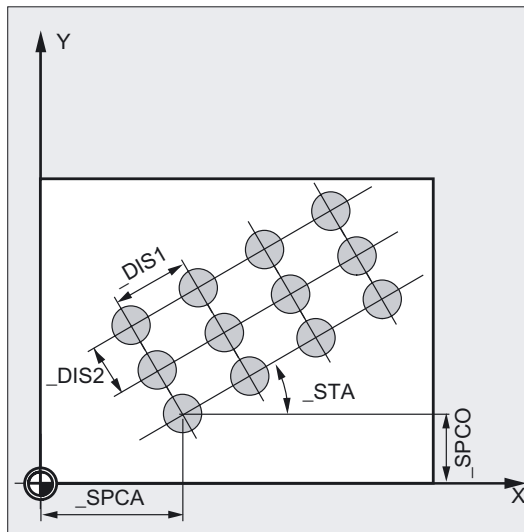
```
N10 G90 G17 F900 S4000 M3 T2 D1 ;Specification of technology values
N15 MCALL CYCLE82(10,0,1,-22,0,0) ;Modal call of a drilling cycle
N20 CYCLE801(30,20,0,10,15,5,3) ;Call dot matrix
N25 M30 ;End of program
```

### Sequence

The cycle calculates the sequence of holes such that the empty paths between them are kept as short as possible. The starting position of the machining operation is defined according to the last position reached in the plane prior to the cycle call.

In each case, starting positions are one of the four possible corner positions.

### Explanation of the parameters



#### **\_SPCA and \_SPCO (reference point abscissa and ordinate)**

These two parameters determine the first point of the grid of holes. The row and column distances are specified in relation to this point.

#### **\_STA (angle)**

The grid of holes can be positioned at any angle in the plane. This angle is programmed in degrees in **\_STA** and refers to the abscissa of the workpiece coordinate system active as the cycle is called.

#### **\_DIS1 and \_DIS2 (column and row distances)**

The distances must be entered without sign. To avoid unnecessary empty travel, the dot matrix is machined line by line or column by column, based on a comparison of distance measurements.

#### **\_NUM1 and \_NUM2 (number)**

With this parameter, you determine the number of columns or lines.



## Milling cycles

### 3.1 General information

The following sections describe how milling cycles are programmed.

This section is intended to guide you in selecting cycles and assigning parameters to them. In addition to a detailed description of the function of each cycle and the associated parameters, you will find in each section programming examples that make it easier for you to work with cycles.

### 3.2 Requirements

#### Programs required in the control system

The milling cycles call the following program internally as a subroutine:

- PITCH.SPF

In addition, you must also have the data block GUD7.DEF and the macro definition file SMAC.DEF.

Load them into the parts program memory of the control unit before executing the milling cycles.

#### Call and return conditions

Milling cycles are programmed independently of the particular axis name. Before you call the milling cycles, a tool compensation must be activated.

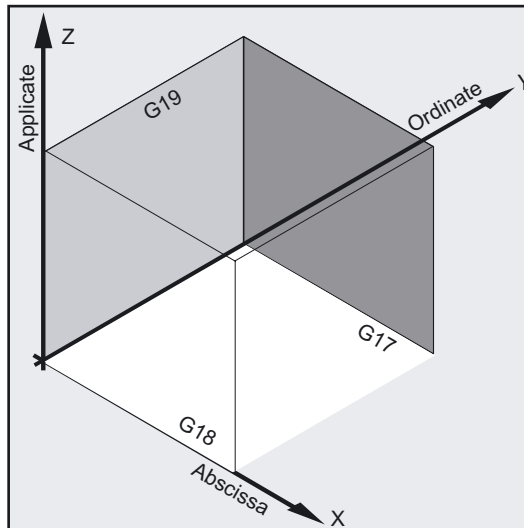
The appropriate values for feedrate, spindle speed and direction of rotation of spindle must be programmed in the part program if the appropriate parameters are not provided in the milling cycle.

The center point coordinates for the milling pattern or the pocket to be machined are programmed in a rectangular coordinate system.

The G functions active prior to the cycle call and the current programmable frame remain active beyond the cycle.

### Plane definition

Milling cycles generally assume that the current workpiece coordinate system has been defined by selecting a plane (G17, G18 or G19) and activating a programmable frame (if necessary). The infeed axis is always the 3rd axis of this coordinate system.



#### References:

/PG/, Programming Guide Fundamentals

/PG/, Programming Guide Production Planning

### Spindle handling

The spindle commands in the cycles always refer to the active master spindle of the control.

If a cycle is to be used on a machine with several spindles, the spindle with which you are working must first be defined as the master spindle with the SETMS command.

### Machining status message

During the execution of the milling cycles, various messages that refer to the machining status are displayed on the screen.

These messages do not interrupt the program execution and continue to be displayed until the next message is displayed or the cycle is completed.

### Cycle setting data

Some of the parameters and the behavior of milling cycles can be altered using cycle settings.

The cycle setting data is defined in data block GUD7.DEF. These values are used in part to establish program compatibility in the event of changes or function expansions in the cycles.

The following cycle setting data is introduced:

<b>_ZSD[x]</b>	<b>Value</b>	<b>Meaning</b>	<b>Cycles affected</b>
<b>_ZSD[1]</b>	0	Depth calculation in the new cycles is carried out between the reference plane + safety clearance and depth ( $\_RFP + \_SDIS - \_DP$ )	POCKET1 to POCKET4, LONGHOLE, CYCLE71, SLOT1, CYCLE72, SLOT2
	1	Depth calculation is carried out without including safety distance	
<b>_ZSD[2]</b>	0	Dimensioning of rectangular pocket or rectangular spigot from the center point	POCKET3, CYCLE76
	1	Dimensioning of rectangular pocket or rectangular spigot from a corner	
<b>_ZSD[9]</b>	0	In the case of dimensioning of the rectangular pocket via a corner point (i.e. $\_ZSD[2]=1$ ), $\_STA$ refers to this reference point	POCKET3
	1	In the case of dimensioning of the rectangular pocket via a corner point (i.e. $\_ZSD[2]=1$ ), $\_STA$ refers to the pocket center point	
<b>_ZSD[10]</b>	0	The programmed feedrate FFR acts in/on the thread	CYCLE90
	1	The programmed feedrate FFR is converted to the center-point path of the tool	
<b>_ZSD[3]</b>		Call from JobShop	

#### **Note**

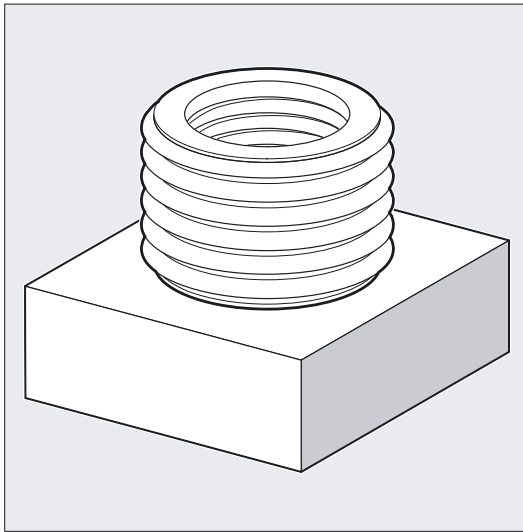
If the cycle setting data is used (e.g. also for JobShop), an incalculable program termination (reset) can result in an unsuitable or undesired setting taking effect. Therefore, the desired settings should be made for self-programmed cycle calls.

### 3.3 Thread milling - CYCLE90

#### Function

By using the cycle CYCLE90, you can produce internal or external threads. The path when milling threads is based on a helix interpolation. All three geometry axes of the current plane, which you will define before calling the cycle, are involved in this motion.

The programmed feedrate F acts according to the the axis grouping defined in the FGROUPO instruction before the call.



#### Programming

CYCLE90 (RTP, RFP, SDIS, DP, DPR, DIATH, KDIAM, PIT, FFR, CDIR, TYPH, CPA, CPO)

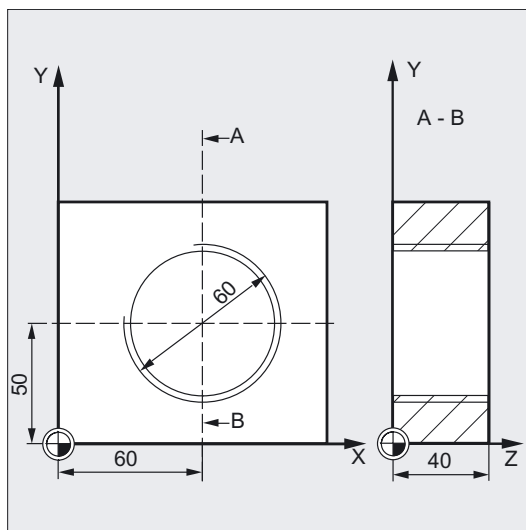
#### Parameter

Parameter	Data type	Meaning	
RTP	real	Retraction plane (absolute)	
RFP	real	Reference plane (absolute)	
SDIS	real	Safety clearance (enter without sign)	
DP	real	Final drilling depth (absolute)	
DPR	real	Final drilling depth relative to the reference plane (enter without sign)	
DIATH	real	Nominal diameter, outer diameter of the thread	
KDIAM	real	Core diameter, internal diameter of the thread	
PIT	real	Pitch	
		Range of values:	0.001 ... 2000.000 mm
FFR	real	Feedrate for thread milling (enter without sign)	
CDIR	integer	Direction of rotation for thread milling	
		Values:	2: (for thread milling at G2)
			3: (for thread milling at G3)

Parameter	Data type	Meaning
TYPH	integer	Thread type
		Values:
		0: Internal thread
		1: External thread, diameter programming via DIATH
		2: External thread, diameter programming via KDIAM
CPA	real	Center point of circle, abscissa (absolute)
CPO	real	Center point of circle, ordinate (absolute)

### Example of internal thread

By using this program, you can mill an internal thread at point X60 Y50 of the G17 plane.



```

DEF REAL RTP=48, RFP=40, SDIS=5, ->           ;Definition of variables with
-> DPR=40, DIATH=60, KDIAM=50                 ;Value assignments
DEF REAL PIT=2, FFR=500, CPA=60,CPO=50
DEF INT CDIR=2, TYPH=0
N10 G90 G0 G17 X0 Y0 Z80 S200 M3              ;Approach start position
N20 T5 D1                                       ;Specification of technology values
N30 CYCLE90 (RTP, RFP, SDIS, DPR, ->          ;Cycle call
-> DIATH, KDIAM, PIT, FFR, CDIR, ->
-> TYPH, CPA, CPO)
N40 G0 G90 Z100                                ;Approach position after cycle
N50 M02                                         ;Program end

```

### Note

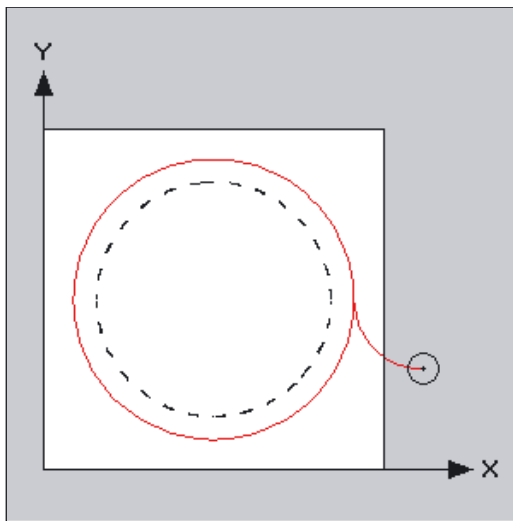
-> means: it must be programmed in a single block

### Sequence for external thread

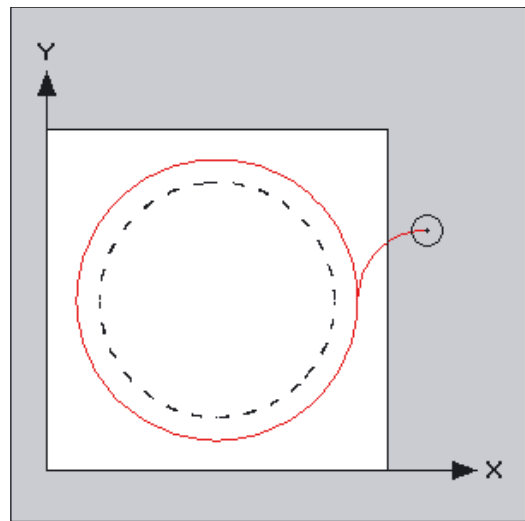
#### Position reached prior to cycle start:

The starting position is any position from which the starting position at the outside diameter of the thread at the height of the retraction plane can be reached without collision.

This start position for thread milling with G2 lies between the positive abscissa and the positive ordinate in the current level (i.e., in the 1st quadrant of the coordinate system). For thread milling with G3, the start position lies between the positive abscissa and the negative ordinate (i.e., in the 4th quadrant of the coordinate system).

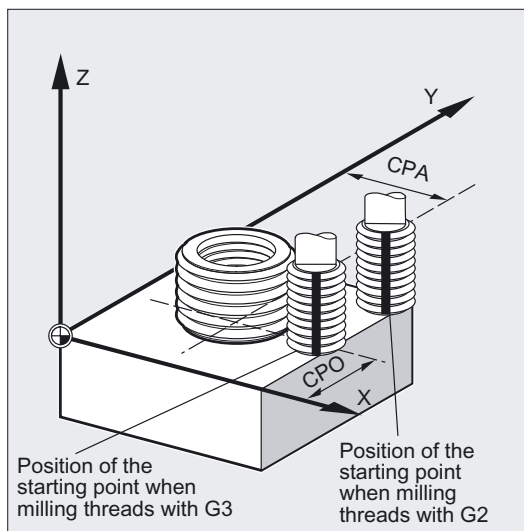


Position of starting point with G2 ... and



... with G3

The distance from the thread diameter depends on the size of the thread and the tool radius used.



**The cycle creates the following sequence of motions:**

- Positioning to the starting point with G0 at the level of the retraction plane in the tool axis of the current plane.
- Infeed to the reference plane brought shifted by the amount of the safety clearance with G0.
- Travel-in movement to the thread diameter on a circular path against the direction G2/G3 programmed under CDIR.
- Thread milling on a helical path with G2/G3 and the feedrate value FFR.
- Travel-out movement on a circular path with counterrotational operation G2/G3 and the reduced feedrate FFR.
- Retraction to the retraction plane in the tool axis with G0.

**Sequence for internal thread****Position reached prior to cycle start:**

The starting position is any position from which the center point of the thread at the height of the retraction plane can be reached without collision.

**The cycle creates the following sequence of motions:**

- Positioning to the center point of the thread with G0 at the level of the retraction plane in the tool axis of the current plane.
- Infeed to the reference plane brought forward by the safety clearance with G0.
- Approach a travel-in circle calculated within the cycle with G1 and the reduced feedrate FFR.
- Travel-in movement to the thread diameter on a circular path corresponding to the direction G2/G3 programmed under CDIR.
- Thread milling on a helical path with G2/G3 and the feedrate value FFR.
- Travel-out movement on a circular path with the same direction of rotation and the reduced feedrate FFR.
- Retraction to the center point of the thread with G0.
- Retraction to the retraction plane in the tool axis with G0.

**Thread sequence from bottom to top**

For technological reasons, it can also be reasonable to machine a thread from bottom to top. In this case, the retraction plane RTP will be behind the thread depth DP.

This machining is possible, but the depth specifications must be programmed as absolute values and the retraction plane must be approached before calling the cycle or a position after the retraction plane must be approached.

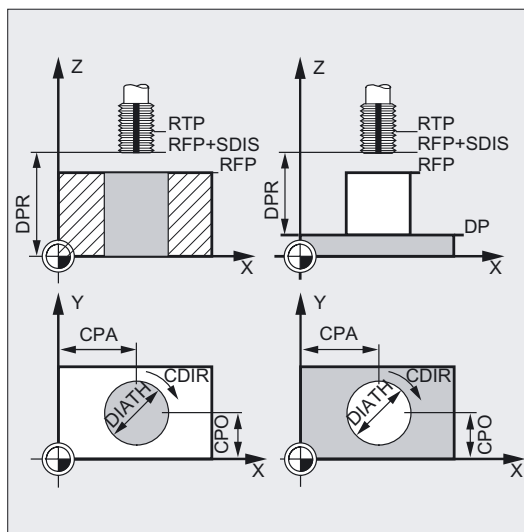
**Example of thread from bottom to top**

A thread beginning at Z-20 to Z0 with pitch 3 is to be milled. The retraction plane is at Z8.

```

N10 G17 X100 Y100 S300 M3 T1 D1 F1000
N20 Z8
N30 CYCLE90 (8, -20, 0, 0, 0, 46, 40, 3, 800, 3, 0, 50, 50)
N40 M2

```

**Explanation of the parameters****DIATH, KDIAM, and PIT (nominal diameter, core diameter, and thread pitch)**

These parameters are used to determine the thread data nominal diameter, core diameter, and pitch. The parameter DIATH is the external, and KDIAM is the internal diameter of the thread. The travel-in / travel-out movements are created internally in the cycle, based on these parameters.

**FFR (feedrate)**

The value of the parameter FFR is specified as the current feedrate value for thread milling. It is effective when thread milling on a helical path.

This value will be reduced in the cycle for the travel-in / travel-out movements. The retraction is performed outside the helix path using G0.

**CDIR (direction of rotation)**

This parameter is used to specify the value for the machining direction of the thread.

If the parameter has an impermissible value, the message "Wrong milling direction, G3 will be generated".

In this case, the cycle is continued and G3 is automatically generated.



**TYPTH (thread type)**

The parameter TYPTH is used to define whether you want to machine an external or an internal thread. With SW 6.4 and higher, the thread diameter for the external thread can be programmed via the nominal diameter DIATH (TYPTH=1) or as usual in many cases via the core diameter KDIAM (TYPTH=2). In this case, the DIATH parameter is irrelevant.

**CPA and CPO (center point)**

These parameters are used to define the center point of the drill hole or of the spigot on which the thread will be produced.

**Overshooting in the direction of the thread length**

For thread milling, the travel-in and travel-out movements occur along all three axes concerned. This means that the travel-out movement includes a further step in the vertical axis, beyond the programmed thread depth.

The overshoot is calculated as follows:

$$\Delta z = \frac{p}{4} \cdot \frac{2 \cdot WR + RDIFF}{DIATH}$$

Where:

$\Delta z$	Overshoot, internal
p	Pitch
WR	Tool radius
DIATH	External diameter of the thread
RDIFF	Radius difference for travel-out circle

For internal threads,  $RDIFF = DIATH/2 - WR$ .

For external threads,  $RDIFF = DIATH/2 + WR$ .

The feedrate is converted to the tool cutting edge. For the inside thread, the travel-in and travel-out movement now takes a more technologically favorable semicircular path, whose radius is calculated internally by the cycle to match the tool. This results in the overshoot being calculated within the cycle, using the formula below:

- For tool diameters  $< 2/3$  nominal diameter

$$\Delta z = \frac{p}{2} \cdot \frac{WR + DIATH/2}{DIATH}$$

- For tool diameters  $\geq 2/3$  nominal diameter

$$\Delta z = \frac{p/2 \cdot KDIAM}{DIATH}$$

Where:

$\Delta z$	Overshoot, internal
p	Pitch
WR	Tool radius
DIATH	External diameter of the thread
KDIAM	Core diameter of the thread

---

**Note**

The cutter radius is calculated internally in the cycle. Therefore, a tool compensation must be programmed before calling the cycle. Otherwise, the alarm 61000 "No tool compensation active" appears and the cycle is aborted.

If the tool radius = 0 or negative, the cycle is also aborted and this alarm is issued.

With internal threads, the tool radius is monitored and alarm 61105 "Cutter radius too large" is output, and the cycle is aborted.

---

**Note**

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, see Drilling, centering – CYCLE81.

---

**See also**

Drilling, centering - CYCLE81 (Page 44)

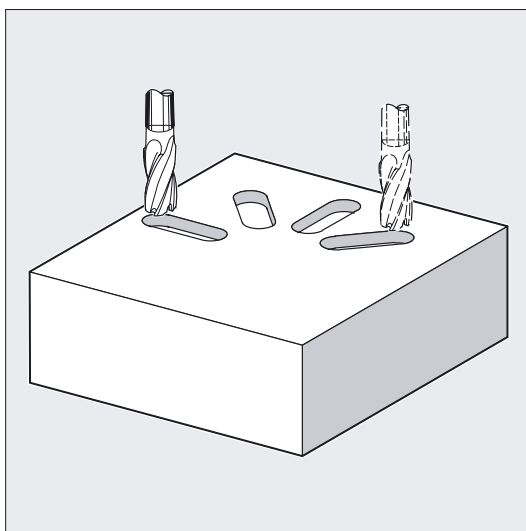
## 3.4 Long holes located on a circle - LONGHOLE

### Function

Use this cycle to machine long holes located on a circle. The longitudinal axis of the long holes is aligned radially.

In contrast to the slot, the width of the long hole is determined by the tool diameter.

Internally in the cycle, an optimum traversing path of the tool is determined, ruling out unnecessary idle passes. If several depth infeeds are required to machine an slot, the infeed is carried out alternately at the end points. The path to be traversed along the longitudinal axis of the long hole will change its direction after each infeed. The cycle will search for the shortest path when changing to the next long hole.



#### NOTICE

The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

#### Note

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

If mutual contour violations of the slots result from incorrect values of the parameters that determine the arrangement and the size of the slots, the cycle will not start the machining. The cycle is aborted and the error message 61104 "Contour violation of slots/long holes" is output.

During the cycle, the workpiece coordinate system is offset and rotated. The values in the workpiece coordinate system are shown on the actual value display such that the longitudinal axis of the long hole being machined is positioned on the first axis of the current machining plane.

After the cycle has been completed, the workpiece coordinate system is in the same position again as it was before the cycle was called.

**Programming**

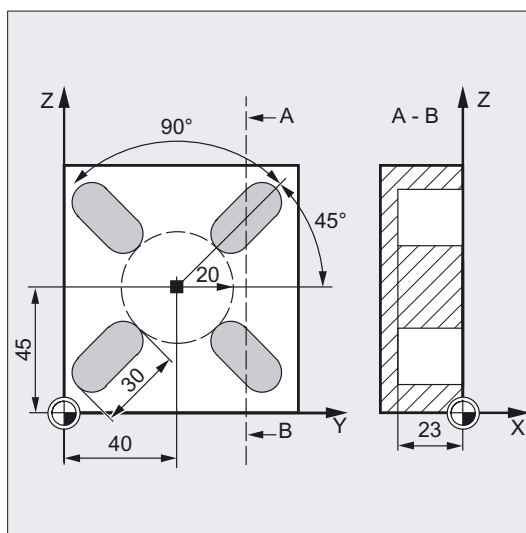
LONGHOLE (RTP, RFP, SDIS, DP, DPR, NUM, LENG, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID)

**Parameter**

Parameter	Data type	Meaning
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Long hole depth (absolute)
DPR	real	Long hole depth relative to the reference plane (enter without sign)
NUM	integer	Number of long holes
LENG	real	Length of long hole (enter without sign)
CPA	real	Center point of circle, abscissa (absolute)
CPO	real	Center point of circle, ordinate (absolute)
RAD	real	Radius of the circle (enter without sign)
STA1	real	Starting angle
INDA	real	Incrementing angle
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for an infeed (enter without sign)

### Example of long hole machining

With this program, you can machine 4 long holes with a length of 30 mm and a relative depth of 23 mm (difference between the reference plane and the long hole base), which are positioned on a circle with the center point Z45 Y40 and a radius of 20 mm in the Y plane (G19). The starting angle is 45 degrees, the incremental angle is 90 degrees. The maximum infeed depth is 6 mm, the safety clearance 1 mm.



```

N10 G19 G90 S600 M3                ;Specification of technology values
T10 D1
M6
N20 G0 Y50 Z25 X5                  ;Approach starting position
N30 LONGHOLE (5, 0, 1, , 23, 4, 30, -> ;Cycle call
-> 40, 45, 20, 45, 90, 100 ,320, 6)
N40 M30                            ;Program end

```

### Note

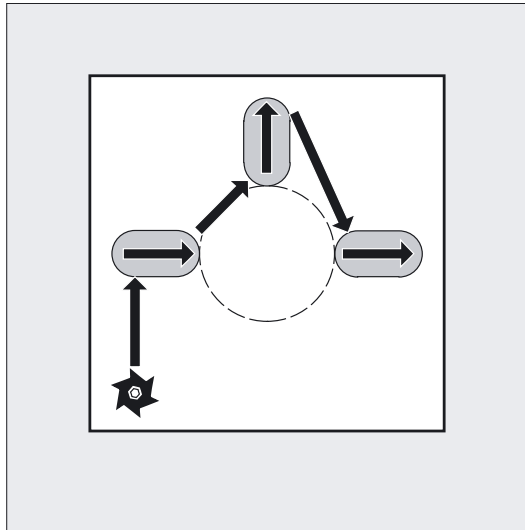
-> means: it must be programmed in a block

## Sequence

### Position reached prior to cycle start:

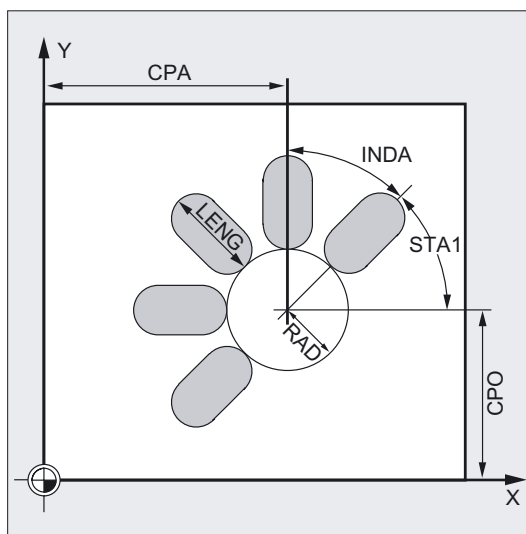
The starting position is any position from which each of the long holes can be approached without collision.

The cycle creates the following sequence of motions:



- Using G0, the starting position for the cycle is approached. In both axes of the current plane, the closest end point of the first long hole to be machined is approached at the level of the retraction plane in the tool plane and then lowered to the reference plane shifted by the amount of the safety clearance.
- Each long hole is milled in a reciprocating motion. The machining in the plane is performed using G1 and the feedrate programmed under FFP1. The infeed to the next machining depth calculated using G1 internally in the cycle and using feedrate is performed at each reversal point until the final depth is reached.
- Retraction to the retraction plane using G0 and approach to the next long hole on the shortest path.
- After the last long hole has been machined, the tool is moved with G0 to the position in the machining plane, which was reached last and which is specified in the diagram below, and the cycle is ended.

## Explanation of the parameters



### DP and DPR (long hole depth)

The depth of the long hole can be specified either absolute (DP) or relative (DPR) to the reference plane.

With relative specification, the cycle will calculate the resulting depth automatically using the positions of reference and retraction planes.

### NUM (number)

Use the parameter NUM to specify the number of long holes.

### LENG (long hole length)

The length of the long hole is programmed under LENG.

If it is detected in the cycle that this length is smaller than the milling diameter, the cycle is aborted with alarm 61105 "Milling radius is too large".

### MID (infeed depth)

Use this parameter to define the maximum infeed depth.

The depth infeed is performed by the cycle in equally sized infeed steps.

Using MID and the total depth, the cycle automatically calculates this infeed which lies between 0.5 x maximum infeed depth and the maximum infeed depth. The minimum possible number of infeed steps is used as the basis. MID=0 means that infeed is carried out in one cut up to the long hole depth.

The depth infeed starts from the reference plane brought forward by the safety clearance (depending on `_ZSD[1]`).

### FFD and FFP1 (feedrate for depth and surface)

The feedrate FFP1 is active for all movements in the plane traversed at feedrate. FFD acts for infeeds vertically to this plane.

**CPA, CPO and RAD (center point and radius)**

You define the position of the circle in the machining plane by the center point (CPA, CPO) and the radius (RAD). Only positive values are permitted for the radius.

**STA1 and INDA (starting and incremental angle)**

The arrangement of the long holes on the circle is defined by these parameters.

If INDA=0, the indexing angle is calculated from the number of long holes, so that they are equally distributed around the circle.

---

**Note**

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, see Drilling, centering – CYCLE81.

For an explanation of the cycle setting data \_ZSD[1], see Requirements for milling cycles.

---

**See also**

Drilling, centering - CYCLE81 (Page 44)

Requirements (Page 101)

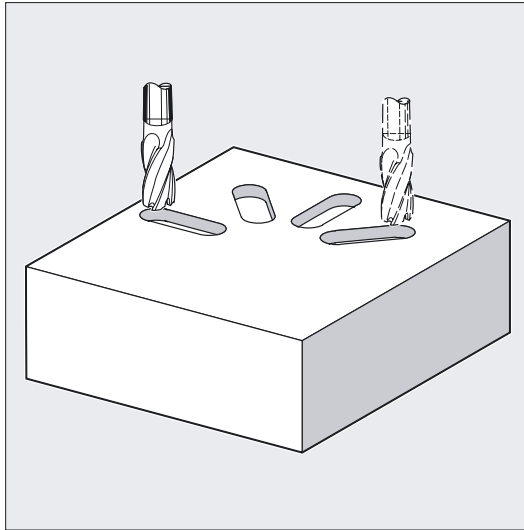


## 3.5 Slots on a circle - SLOT1

### Function

The cycle SLOT1 is a combined roughing-finishing cycle.

Use this cycle to machine slots arranged on a circle. The longitudinal axis of the slots is aligned radially. In contrast to the long hole, a value is defined for the slot width.

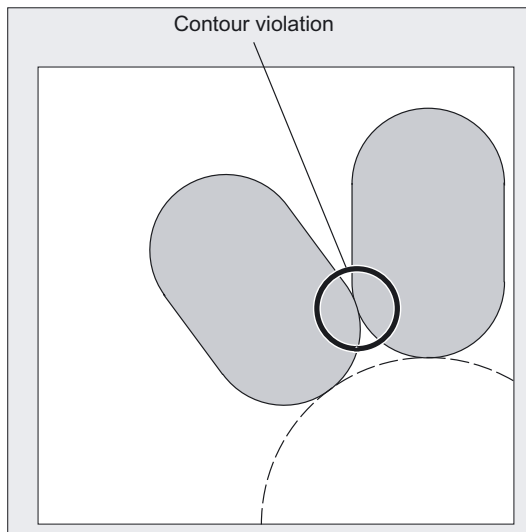


<b>NOTICE</b>
The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

### Note

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

If incorrect values are assigned to the parameters that determine the arrangement and size of the slots and thus cause mutual contour violation of the slots, the cycle is not started. The cycle is aborted and the error message 61104 "Contour violation of slots/long holes" is output.



## Programming

SLOT1 (RTP, RFP, SDIS, DP, DPR, NUM, LENG, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF, \_FALD, \_STA2)

## Parameter

Parameter	Data type	Meaning
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Slot depth (absolute)
DPR	real	Slot depth relative to the reference plane (enter without sign)
NUM	integer	Number of slots
LENG	real	Slot length (enter without sign)
WID	real	Slot width (enter without sign)
CPA	real	Center point of circle, abscissa (absolute)
CPO	real	Center point of circle, ordinate (absolute)
RAD	real	Radius of the circle (enter without sign)
STA1	real	Starting angle
INDA	real	Incrementing angle
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for one infeed (enter without sign)

Parameter	Data type	Meaning
CDIR	integer	Mill direction for machining the slot
		Values: 0: Down-cut milling (corresponds to direction of spindle rotation) 1: Down-cut milling 2: with G2 (independent of direction of spindle rotation) 3: with G3
FAL	real	Finishing allowance at the slot edge (enter without sign)
VARI	integer	Machining type (enter without sign)
		Values: UNITS DIGIT: Machining process 0: Complete machining 1: Roughing 2: Finishing TENS DIGIT: Infeed 0: Perpendicular to G0 1: Perpendicular to G1 3: Oscillation with G1
MIDF	real	Maximum infeed depth for finishing
FFP2	real	Feedrate for finishing
SSF	real	Speed when finishing
_FALD	real	Finishing allowance at the slot base (enter without sign)
_STA2	real	Maximum insertion angle for oscillation movement

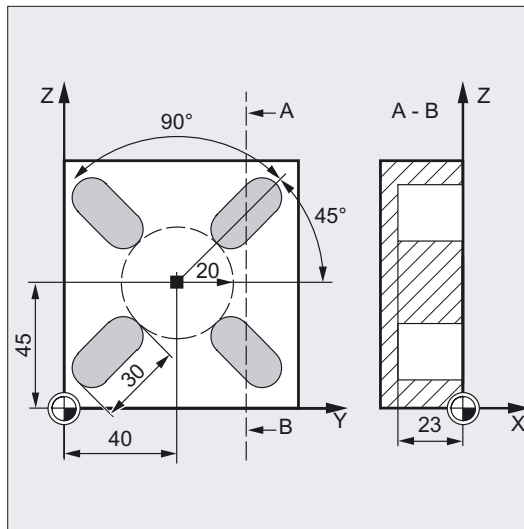
### Example of slot milling

This program realizes the same arrangement of 4 slots located on a circle as the program for long hole machining (see Programming long holes located on a circle - LONGHOLE).

The slots have the following dimensions:

Length	30 mm
Width	15 mm
Depth	23 mm
Safety clearance	1 mm
Finishing allowance	0.5 mm
Milling direction	G2
Maximum infeed in depth	10 mm

The slots must be machined completely with an oscillating insertion motion.



```

N10 G19 G90 S600 M3 ;Specification of technology values
N15 T10 D1
N17 M6
N20 G0 Y20 Z50 X5 ;Approach start position
N30 SLOT1 (5, 0, 1, -23, , 4, 30, 15, -> ;Cycle call, parameters VARI, MIDF,
->40, 45, 20, 45, 90, 100, 320, 10, -> ;FFP2 and SSF omitted
->2, 0.5, 30, 10, 400, 1200, 0.6, 5)
N40 M30 ;Program end

```

### Note

-> means: it must be programmed in a single block

## Sequence

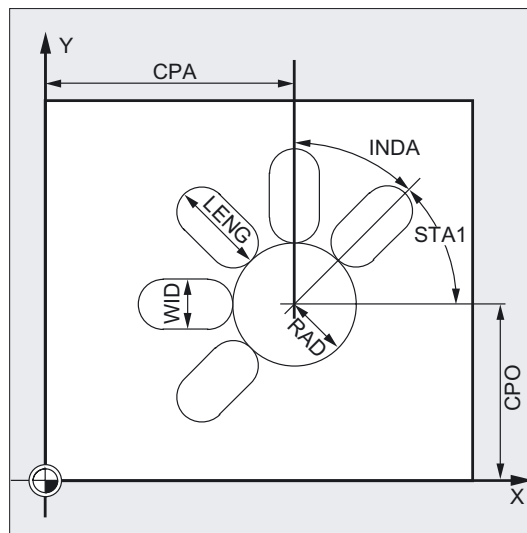
### Position reached prior to cycle start:

The starting position can be any position from which each of the slots can be approached without collision.

### The cycle creates the following sequence of motions:

- Approach of the position at the beginning of the cycle indicated in the figure with G0.
- Complete machining of a slot is carried out in the following steps:
  - Approach of reference plane shifted by the amount of the safety clearance with G0
  - Infeed to next machining depth as programmed under VARI and feedrate FFD.
  - Pocket milling of the slot up to the finishing allowance at the slot base and slot edge with feedrate value FFP1. Then finishing with feedrate value FFP2 and spindle speed SSF along the contour according to the machining direction programmed under CDIR.
  - The vertical depth infeed with G0/G1 always takes place at the same position in the machining plane as long as the slot is reached by the end depth.
  - With oscillating machining the starting point is selected such that the end point always reaches the same position in the machining plane.
- Retract tool to the retraction plane and move to the next slot with G0.
- When machining of the last slot is complete, the tool is traversed to the retraction plane with G0, and the cycle is completed.

## Explanation of the parameters



#### **DP and DPR (slot depth)**

The slot depth can be specified either absolute (DP) or relative (DPR) to the reference plane.

With relative specification, the cycle will calculate the resulting depth automatically using the positions of reference and retraction planes.

#### **NUM (number)**

Use the parameter NUM to specify the number of slots.

#### **LENG and WID (slot length and slot width)**

Use the parameters LENG and WID to define the form of a slot in the plane. The milling cutter diameter must be smaller than the slot width. Otherwise, alarm 61105 "Cutter radius too large" will be activated and the cycle aborted.

The milling cutter diameter must not be smaller than one half the slot width or else residual material will remain in the center of the slot. However, this is not checked.

#### **CPA, CPO and RAD (center point and radius)**

The position of the hole circle in the machining plane is defined by the center point (parameters CPA and CPO) and the radius (RAD). Only positive values are permitted for the radius.

#### **STA1 and INDA (starting and incremental angle)**

The arrangement of the slot on the circle is defined by these parameters.

STA1 defines the angle between the positive direction of the abscissa of the workpiece coordinate system active before the cycle was called and the first slot. Parameter INDA contains the angle from one slot to the next.

If INDA=0, the incrementing angle is calculated from the number of slots so that they are arranged equally around the circle.

#### **FFD and FFP1 (feedrate for depth and surface)**

Feedrate FFD is active for vertical infeed to the machining plane with G1 and for insertion with oscillating motion.

The feedrate FFP1 is active for all movements in the plane traversed at feedrate when roughing.

#### **MID (infeed depth)**

Use this parameter to define the maximum infeed depth. The depth infeed is performed by the cycle in equally sized infeed steps.

Using MID and the total depth, the cycle automatically calculates this infeed which lies between 0.5 x maximum infeed depth and the maximum infeed depth. The minimum possible number of infeed steps is used as the basis. MID=0 means that the cut to slot depth is made with one feed.

The depth infeed starts from the reference plane brought forward by the safety clearance (depending on \_ZSD[1]).

#### **CDIR (milling direction)**

You define the slot machining direction in this parameter.

The direction of milling can be programmed as follows using the CDIR parameter:

- Directly with "2 for G2" and "3 for G3" or
- alternatively, "down-cut" or "up-cut"

Down-cut and up-cut are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

Down-cut	Up-cut
M3 → G3	M3 → G2
M4 → G2	M4 → G3

#### **FAL (finishing allowance at slot edge)**

Use this parameter to program a finishing allowance at the slot edge. FAL does not influence the depth infeed. If the value of FAL is greater than allowed for the specified width and the milling cutter used, FAL is automatically reduced to the maximum possible value. In the case of roughing, milling is performed with a reciprocating movement and depth infeed at both end points of the slot.

#### **VARI, MIDF, FFP2 and SSF (machining type, infeed depth, feedrate and speed)**

##### **Note**

The description of these parameters is also valid for the cycles SLOT2, POCKET1. and POCKET2.

Use the parameter VARI to define the machining type.

Possible settings:

UNITS DIGIT (machining process)

- 0 = complete machining in two sections
  - Solid machining of the slot (SLOT1, SLOT2) or pocket (POCKET1, POCKET2) up to the finishing allowance takes place at the spindle speed programmed before the cycle call and the feedrate FFP1. Depth infeed is defined with MID.
  - Solid machining of the remaining finishing allowance is carried out at the spindle speed defined via SSF and the feedrate FFP2. With the parameter MIDF, another depth infeed can be programmed than for roughing. However, this is only effective for finishing on the edge. If MIDF=0, the infeed is performed right to the final depth.

If FFP2 is not programmed, feedrate FFP1 is active. This also applies analogously if SSF is not specified, i.e. the speed programmed prior to the cycle call will apply.

- 1 = roughing

The slot (SLOT1, SLOT2) or pocket (POCKET1, POCKET2) is solid machined up to the final finishing allowance at the speed programmed before the cycle call and feedrate FFP1. The depth infeed is programmed via MID.

- 2 = finishing

The cycle requires that the slot (SLOT1, SLOT2) or pocket (POCKET1, POCKET2) have already been machined to a remaining final finishing allowance, so it is only necessary to machine the final finishing allowance. If FFP2 and SSF are not programmed, the feedrate FFP1 or the speed programmed before the cycle call is active. For finishing at the edge, it is possible to program a value for the depth infeed in parameter MIDF. In machining mode VARI=30, the edge finishing cut is performed at the last roughing depth.

TENS POINT (infeed)

- 0 = vertical with G0
- 1 = vertical with G1
- 3 = oscillation with G1

If a different value is programmed for the parameter VARI, the cycle is aborted after output of alarm 61102 "Machining type defined incorrectly".

**Milling cutter diameter = slot width (WID)**

- With complete machining, finishing is only performed on the base.
- In machining mode VARI = 32, positioning is paraxial in Z with G1 followed by finish-cutting (infeed possible via MIDF).

**\_FALD (finishing allowance at slot edge)**

When roughing, a separate finishing allowance is taken into account at the base.

**\_STA2 (insertion angle)**

Parameter \_STA2 defines the maximum insertion angle for the reciprocating movement.

- Vertical insertion (VARI = 0X, VARI = 1X)

The vertical depth infeed always takes place at the same position in the machining plane as long as the slot is reached by the end depth.

- Insertion oscillation on center axis of slot (VARI = 3X)

means that the milling center point on a straight line oscillating back and forth is inserted at an angle until it has reached the nearest current depth. The maximum insertion angle is programmed under \_STA2, and the length of the oscillation path is calculated from LENG-WID. The oscillating depth infeed ends at the same point as with vertical depth infeed motions; the starting point in the plane is calculated accordingly. The roughing operation begins in the plane once the current depth is reached. The feedrate is programmed under FFD.

---

**Note**

During the cycle, the workpiece coordinate system is offset and rotated. The values in the workpiece coordinate system displayed on the actual value display are such that the longitudinal axis of the slot that has just been machined corresponds to the first axis of the current machining plane.

After the cycle has been completed, the workpiece coordinate system is in the same position again as it was before the cycle was called.

---

**Note**

For an explanation of the parameters RTP, RFP, and SDIS, see Drilling, centering – CYCLE81.

For an explanation of the cycle setting data \_ZSD[1], see Requirements for milling cycles.

---

**See also**

Drilling, centering - CYCLE81 (Page 44)

Requirements (Page 101)

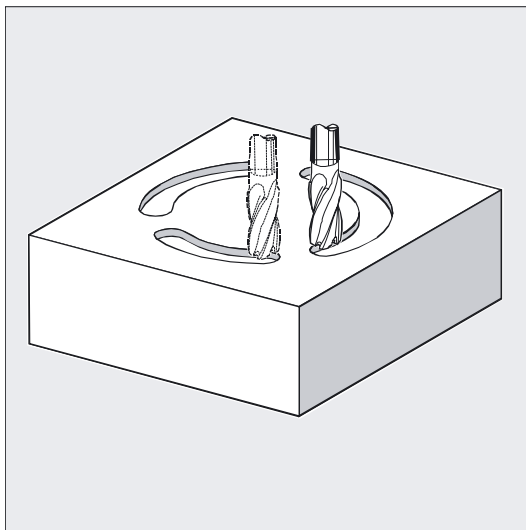


## 3.6 Circumferential slot - SLOT2

### Function

The cycle SLOT2 is a combined roughing-finishing cycle.

Use this cycle to machine circumferential slots arranged on a circle.



### NOTICE

The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

### Programming

SLOT2 (RTP, RFP, SDIS, DP, DPR, NUM, AFSL, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF, \_FFCP)

### Parameter

Parameter	Data type	Meaning
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Slot depth (absolute)
DPR	real	Slot depth relative to the reference plane (enter without sign)
NUM	integer	Number of slots
AFSL	real	Angle for the slot length (enter without sign)

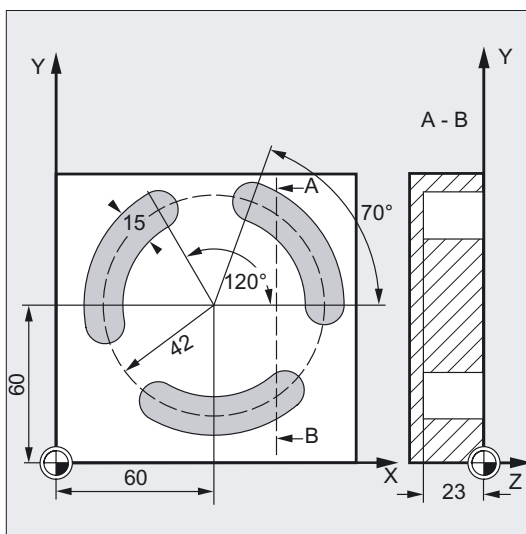
## 3.6 Circumferential slot - SLOT2

Parameter	Data type	Meaning
WID	real	Circumferential slot width (enter without sign)
CPA	real	Center point of circle, abscissa (absolute)
CPO	real	Center point of circle, ordinate (absolute)
RAD	real	Radius of the circle (enter without sign)
STA1	real	Starting angle
INDA	real	Incrementing angle
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for one infeed (enter without sign)
CDIR	integer	Milling direction for machining the circumferential slot
		Values: 2: (for G2) 3: (for G3)
FAL	real	Finishing allowance at the slot edge (enter without sign)
VARI	integer	Machining type
		Values: UNITS DIGIT: Machining process 0: Complete machining 1: Roughing 2: Finishing
		TENS DIGIT infeed 0: Positioning from slot to slot on straight line with G0 1: Positioning from slot to slot on circular path with feedrate
MIDF	real	Maximum infeed depth for finishing
FFP2	real	Feedrate for finishing
SSF	real	Speed when finishing
_FFCP	real	Feedrate for intermediate positioning on a circular path, in mm/min

### Example of slots 2

Use this program to machine 3 circumferential slots arranged at a circle with center point X60 Y60 and radius 42 mm in the XY plane. The circumferential slots have the following dimensions:

Width 15 mm, angle for slot length 70 degrees, depth 23 mm. The initial angle is 0 degrees, the incremental angle is 120 degrees. The slot contours are machined to a final machining allowance of 0.5 mm, the safety clearance in infeed axis Z is 2 mm, the maximum depth infeed is 6 mm. The slots are to be completely machined. Speed and feedrate must be identical for finishing. The infeed when finishing is to be performed to slot depth.



```

DEF REAL FFD=100                                ;Definition of variables with value
                                                ;assignment
N10 G17 G90 S600 M3                              ;Specification of technology values
N15 T10 D1
N17 M6
N20 G0 X60 Y60 Z5                                ;Approach starting position
N30 SLOT2 (2, 0, 2, -23, , 3, 70, ->            ;Cycle call
-> 15, 60, 60, 42, , 120, FFD, ->              ;Parameters VAR, MIDF, FFP2 and SSF
-> FFD+200, 6, 2, 0.5)                          ;have been omitted
N40 M30                                           ;Program end

```

### Note

-> means: it must be programmed in a single block

## Sequence

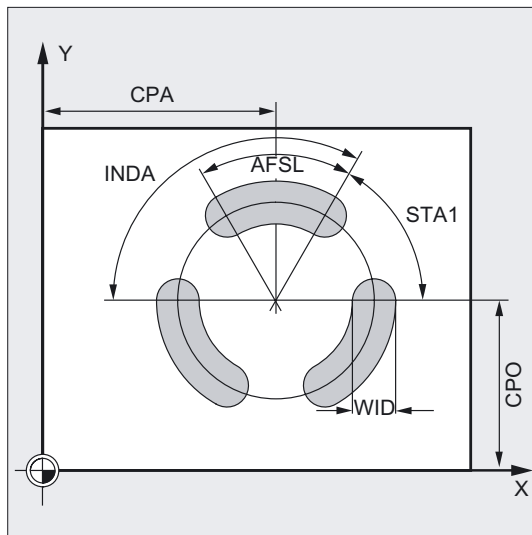
### Position reached prior to cycle start:

The starting position can be any position from which each of the slots can be approached without collision.

### The cycle creates the following sequence of motions:

- With G0, the position indicated in the first figure is approached at the beginning of the cycle.
- The circumferential slot is machined in the same steps as an elongated hole.
- After finishing a circumferential slot, the tool is withdrawn to the retraction plane and the transition to the next slot is made, either on a straight line with G0 or on a circular path at the feedrate programmed in \_FFCP.
- After the last slot has been machined, the tool at the final position in the machining plane specified in the second figure is moved with G0 to the retraction plane, and the cycle is ended.

## Explanation of the parameters



### NUM (number)

Use the parameter NUM to specify the number of slots.

### AFSL and WID (angle and circumferential slot width)

Use the parameters AFSL and WID to define the form of a slot in the plane. The cycle checks whether the slot width is violated with the active tool. Otherwise, alarm 61105 "Cutter radius too large" will be activated and the cycle aborted.

### CPA, CPO and RAD (center point and radius)

You define the position of the circle in the machining plane by the center point (CPA, CPO) and the radius (RAD). Only positive values are permitted for the radius.

### **STA1 and INDA (starting and incremental angle)**

The arrangement of the circumferential slots on the circle is defined by these parameters.

STA1 defines the angle between the positive direction of the abscissa of the workpiece coordinate system active before the cycle was called and the first circumferential slot.

The INDA parameter contains the angle from one circumferential slot to the next.

If INDA=0, the incremental angle is calculated from the number of circumferential slots so that they are arranged equally around the circle.

---

#### **Note**

For an explanation of the parameters RTP, RFP, and SDIS, see Drilling, centering – CYCLE81.

For an explanation of the parameters DP, DPR, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, and SSF, see Slots located on a circle - SLOT1.

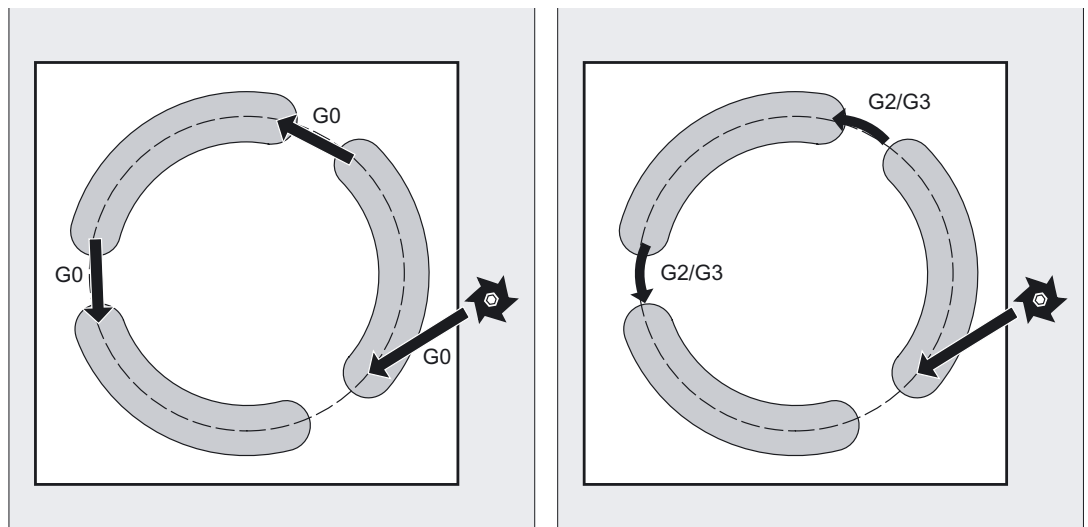
For an explanation of the cycle setting data \_ZSD[1], see Requirements for milling cycles.

---

### **New machining types for Cycles SW 6.3 and higher**

#### **Finishing only edge (VARI = x3)**

- There is a new "Finish edge" selection This is the only machining type where the diameter of the cutter may be less than half the slot width. No check is made of whether it is sufficiently large to execute the final machining allowance FAL.
- A number of depth infeeds are possible. These are programmed as usual by parameter MID. Slot 1 is circumnavigated once at each depth.
- To approach and return from the contour, the cycle generates a smooth approach to the segment of the circular path.

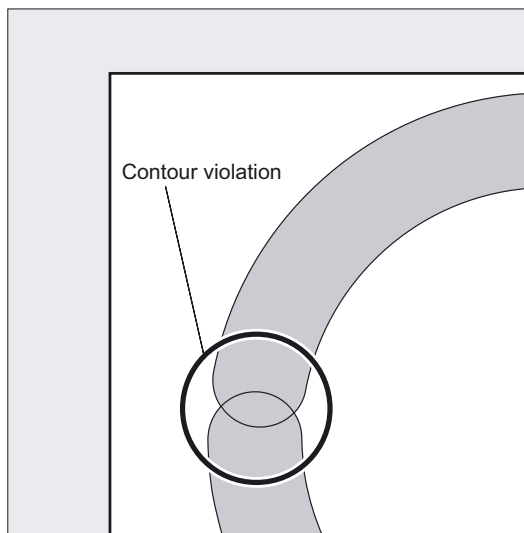


#### Intermediate positioning on circular path (VARI = 1x)

- Particularly when used on turning machines, it can happen that there will be a spigot in the middle of the circle, on which the slots lie, which does not allow direct positioning from one slot to the next with G0.
- The circular path is taken to be the circle, on which the slots lie (determined by parameters CPA, CPO and RAD). Positioning takes place at the same level as intermediate positioning on a straight line with G0. The positioning feedrate for the circular path is programmed under the parameter in mm/min.

#### Additional notes

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

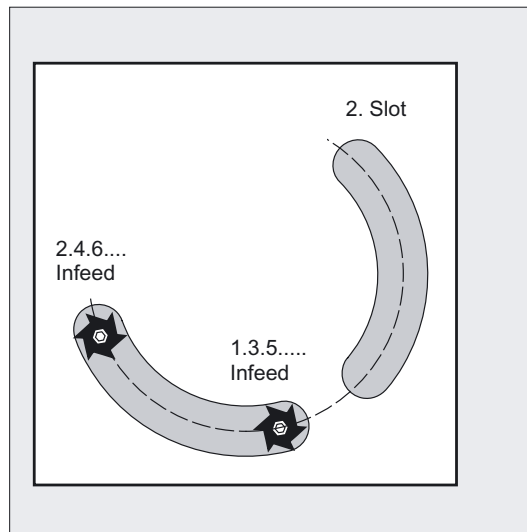


If incorrect values are assigned to the parameters that determine the arrangement and size of the slots and thus cause mutual contour violation of the slots, the cycle is not started.

The cycle is aborted and the error message 61104 "Contour violation of slots/long holes" is output.

During the cycle, the workpiece coordinate system is offset and rotated. The actual value display in the workpiece coordinate system is always shown such that the circumferential slot currently being machined starts on the 1st axis of the current processing level and the zero point of the workpiece coordinate system is in the center of the circle.

After the cycle has been completed, the workpiece coordinate system is in the same position again as it was before the cycle was called.

**Special case: slot width = milling cutter diameter**

- The machining case of slot width = milling cutter diameter is permitted for roughing and finishing. This machining case occurs when slot width  $WID - 2 * \text{finishing allowance } FAL = \text{milling cutter diameter}$ .
- The traversing strategy is then as for the LONGHOLE cycle, i.e., depth infeed occurs alternately at the reversal points, see figure.

**Note**

An annular slot (complete circle) can be machined when  $NUM = 1$  and  $AFSL = 369$  are programmed.

**See also**

Drilling, centering - CYCLE81 (Page 44)

Requirements (Page 101)

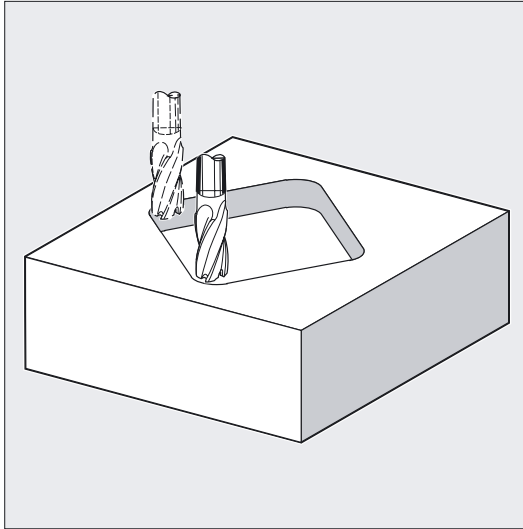
Slots on a circle - SLOT1 (Page 117)

## 3.7 Milling rectangular pockets - POCKET1

### Function

The cycle is a combined roughing-finishing cycle.

With this cycle, you can machine rectangular pockets in any position in the machining plane.



<b>NOTICE</b>
The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

<b>Note</b>
The pocket milling cycle POCKET3 can be performed with any tool.

### Programming

POCKET1 (RTP, RFP, SDIS, DP, DPR, LENG, WID, CRAD, CPA, CPD, STA1, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF)



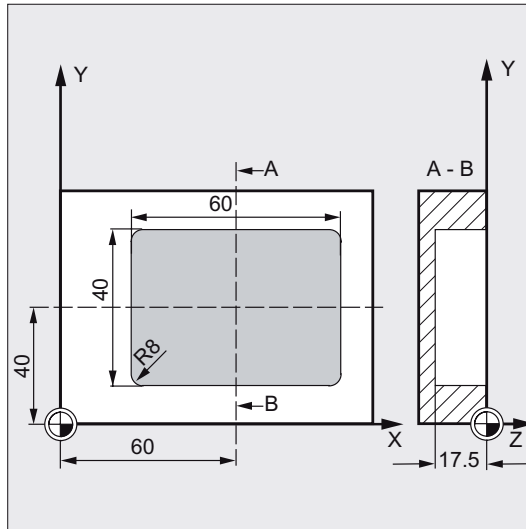
## Parameter

Parameter	Data type	Meaning
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety clearance (enter without sign)
DP	real	Pocket depth (absolute)
DPR	real	Pocket depth relative to the reference plane (enter without sign)
LENG	real	Pocket length (enter without sign)
WID	real	Pocket width (enter without sign)
CRAD	real	Corner radius (enter without sign)
CPA	real	Pocket center point, abscissa (absolute)
CPO	real	Pocket center point, ordinate (absolute)
STA1	real	Angle between longitudinal axis and abscissa
		Range of values: $0 \leq \text{STA1} < 180$ degrees
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for one infeed (enter without sign)
CDIR	integer	Mill direction for machining the pocket
		Values: 2: (for G2) 3: (for G3)
FAL	real	Finishing allowance at the pocket edge (enter without sign)
VARI	integer	Machining type
		Values: 0: Complete machining 1: Roughing 2: Finishing
MIDF	real	Maximum infeed depth for finishing
FFP2	real	Feedrate for finishing
SSF	real	Speed when finishing

### Example of pocket milling

With this program, you can make a pocket with a length of 60 mm, a width of 40 mm, a corner radius of 8 mm and a depth of 17.5 mm (difference between reference plane and pocket base) in the XY plane (G17). The pocket has an angle of 0 degrees to the X axis. The final machining allowance of the pocket edges is 0.75 mm, the safety distance in the Z axis, which is added to the reference plane, is 0.5 mm. The center point of the pocket lies at X60 and Y40, the maximum depth infeed is 4 mm.

Merely a rough machining operation is to be carried out.



```

DEF REAL LENG, WID, DPR, CRAD           ;Definition of variables
DEF INT VARI
N10 LENG=60 WID=40 DPR=17.5 CRAD=8      ;Value assignments
N20 VARI=1
N30 G90 S600 M4                          ;Specification of technology values
N35 T20 D2
N37 M6
N40 G17 G0 X60 Y40 Z5                   ;Approach start position
N50 POCKET1 (5, 0, 0.5, , DPR, ->       ;Cycle call
-> LENG, WID, CRAD, 60, 40, 0, ->      ;Parameters MIDF, FFP2, and SSF are
-> 120, 300, 4, 2, 0.75, VARI)         ;omitted
N60 M30                                  ;Program end

```

### Note

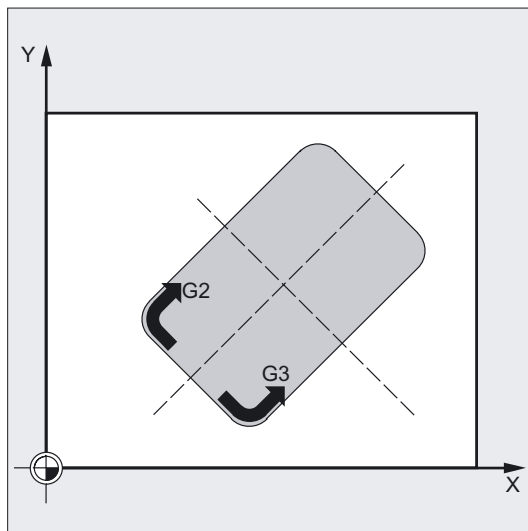
-> means: it must be programmed in a block

## Sequence

### Position reached prior to cycle start:

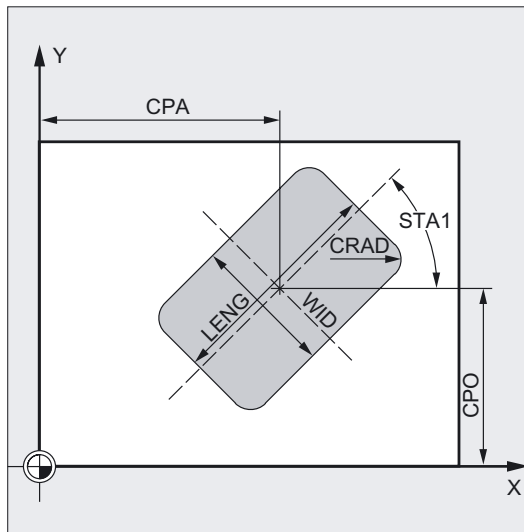
Starting position is any position from which the pocket center point can be approached at the height of the retraction plane without collision.

### The cycle creates the following sequence of motions:



- With G0, the pocket center point is approached at the retraction level, and then, from this position, with G0, too, the reference plane brought forward by the safety clearance is approached. Complete machining of the pocket is performed in the following stages:
  - Infeed to the next machining depth with G1 and the feedrate FFD.
  - Pocket milling of the pocket to the finishing allowance with feedrate FFP1 and the spindle speed in effect before the cycle call.
- After roughing is completed:
  - Infeed to the machining depth defined by MIDF
  - Finishing along the contour with feedrate FFP2 and speed SSF.
  - The machining direction is carried out according to the direction defined under CDIR.
- When machining of the pocket is completed, the tool is traversed to the pocket center point on the retraction plane and the cycle is terminated.

### Explanation of the parameters



#### DP and DPR (pocket depth)

The pocket depth can be defined as either absolute (DP) or relative (DPR) to the reference plane.

If it is entered as a relative value, the cycle automatically calculates the correct depth on the basis of the positions of the reference and retraction planes.

#### LENG, WID and CRAD (length, width and radius)

The shape of a pocket in the plane is determined with parameters LENG, WID and CRAD.

If you cannot traverse the programmed corner radius with the active tool since its radius is larger, then the corner radius of the machine pocket corresponds to the tool radius. If the milling tool radius is larger than half of the length or width of the pocket, then the cycle will be aborted and alarm 61105 "Cutter radius too large" is output.

#### CPA, CPO (center point)

With parameters CPA and CPO, you define the center point of the pocket in the abscissa and ordinate.

#### STA1 (angle)

STA1 defines the angle between the positive abscissa and the longitudinal axis of the pocket.

---

#### Note

For an explanation of the parameters RTP, RFP, and SDIS, see Drilling, centering – CYCLE81.

For an explanation of the parameters FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, and SSF, see Slots located on a circle - SLOT1.

For an explanation of the cycle setting data \_ZSD[1], see Requirements for milling cycles.

---

**Additional notes**

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original coordinate system is active again.

**See also**

Drilling, centering - CYCLE81 (Page 44)

Requirements (Page 101)

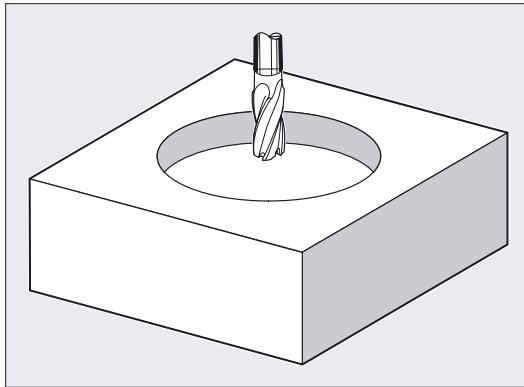
Circumferential slot - SLOT2 (Page 125)

## 3.8 Milling circular pockets - POCKET2

**Function**

The cycle is a combined roughing-finishing cycle.

Use this cycle to machine circular pockets in the machining plane.

**NOTICE**

The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

**Note**

The pocket milling cycle POCKET4 can be performed with any tool.

## Programming

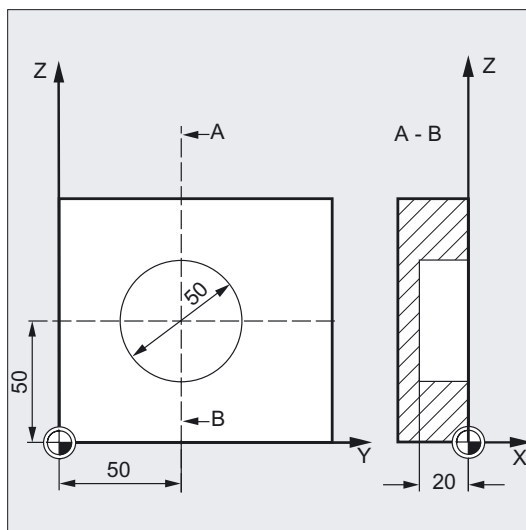
POCKET2 (RTP, RFP, SDIS, DP, DPR, PRAD, CPA, CPO, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF)

## Parameter

Parameter	Data type	Meaning	
RTP	real	Retraction plane (absolute)	
RFP	real	Reference plane (absolute)	
SDIS	real	Safety clearance (enter without sign)	
DP	real	Pocket depth (absolute)	
DPR	real	Pocket depth relative to the reference plane (enter without sign)	
PRAD	real	Pocket radius (enter without sign)	
CPA	real	Pocket center point, abscissa (absolute)	
CPO	real	Pocket center point, ordinate (absolute)	
FFD	real	Feedrate for depth infeed	
FFP1	real	Feedrate for surface machining	
MID	real	Maximum infeed depth for one infeed (enter without sign)	
CDIR	integer	Mill direction for machining the pocket	
		Values:	2: (for G2) 3: (for G3)
FAL	real	Finishing allowance at the pocket edge (enter without sign)	
VARI	integer	Machining type	
		Values:	0: Complete machining 1: Roughing 2: Finishing
MIDF	real	Maximum infeed depth for finishing	
FFP2	real	Feedrate for finishing	
SSF	real	Speed when finishing	

### Example of circular pocket milling

With this program, you can make a circular pocket in the YZ plane (G19). The center point is determined by Y50 Z50. The infeed axis for the depth infeed is the X axis, the pocket depth is entered as an absolute value. Neither finishing dimension nor safety clearance is specified.



```

DEF REAL RTP=3, RFP=0, DP=-20,->           ;Definition of variables with
-> PRAD=25, FFD=100, FFP1, MID=6           ;Value assignments
N10 FFP1=FFD*2
N20 G19 G90 G0 S650 M3                     ;Specification of technology values
N25 T10 D1
N27 M6
N30 Y50 Z50                               ;Approach start position
N40 POCKET2 (RTP, RFP, , DP, , PRAD, ->    ;Cycle call
-> 50, 50, FFD, FFP1, MID, 3, )           ;Parameters FAL, VARI, MIDF, FFP2, ;SSF
                                           ;are omitted
N50 M30                                   ;Program end
DEF REAL RTP=3, RFP=0, DP=-20,->           ;Definition of variables with
-> PRAD=25, FFD=100, FFP1, MID=6           ;Value assignments
N10 FFP1=FFD*2

```

#### Note

-> means: it must be programmed in a single block

## Sequence

### Position reached prior to cycle start:

Starting position is any position from which the pocket center point can be approached at the height of the retraction plane without collision.

---

### Note

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

The depth infeed is always made in the pocket center point. It can be useful to drill there beforehand.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

---

### The cycle creates the following sequence of motions:

- With G0, the pocket center point is approached at the retraction level, and then, from this position, with G0, too, the reference plane brought forward by the safety clearance is approached. Complete machining of the pocket is performed in the following stages:
  - Infeed to position of pocket center point in the next machining depth with feed rate FFD.
  - Pocket milling of the pocket to the finishing allowance with feedrate FFP1 and the spindle speed in effect before the cycle call.
- After roughing is completed:
  - Infeed to the next machining depth defined by MIDF
  - Finishing along the contour with feedrate FFP2 and speed SSF.
  - The machining direction is carried out according to the direction defined under CDIR.
- When machining is completed, the tool is traversed to the pocket center point in the retraction plane and the cycle is terminated.

---

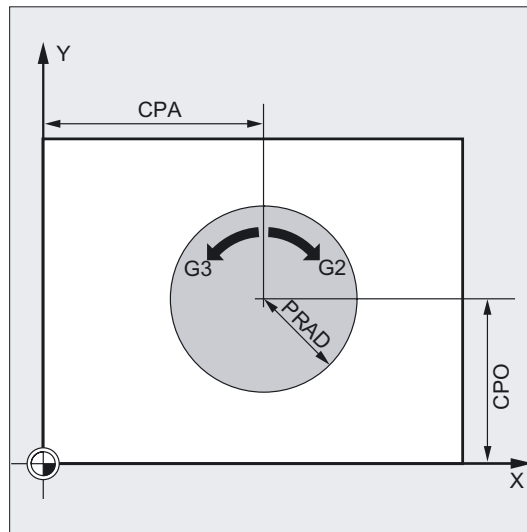
### Note

At the end of the cycle, the original coordinate system is active again.

---



## Explanation of the parameters



### PRAD (pocket radius)

The form of the circular pocket is determined solely by its radius.

If this is smaller than the tool radius of the active tool, then the cycle is aborted and alarm 61105 "Cutter radius too large" is output.

### CPA, CPO (pocket center point)

The center point of the circular pocket in the abscissa and ordinate is defined with parameters CPA and CPO.

---

### Note

For an explanation of the parameters RTP, RFP, and SDIS, see Drilling, centering – CYCLE81.

For an explanation of the parameters DP and DPR, see Milling rectangular pockets - POCKET1.

For an explanation of the parameters FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, and SSF, see Slots located on a circle - SLOT1.

For an explanation of the cycle setting data \_ZSD[1], see Requirements for milling cycles.

---

## See also

Drilling, centering - CYCLE81 (Page 44)

Slots on a circle - SLOT1 (Page 117)

Requirements (Page 101)

Milling rectangular pockets - POCKET1 (Page 132)

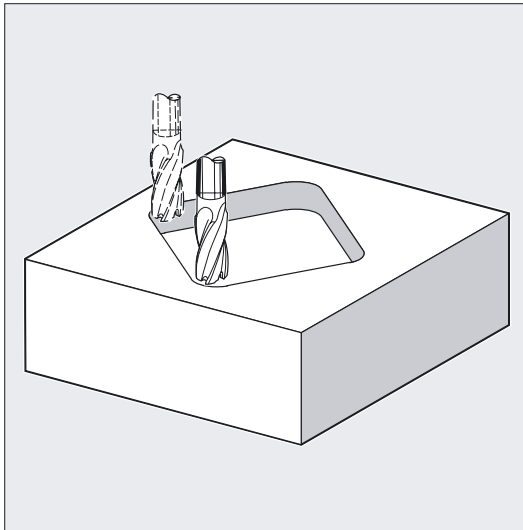
## 3.9 Milling a rectangular pocket - POCKET3

### Function

The cycle can be used for roughing and finishing.

For finishing, a face cutter is required.

The depth infeed will always start at the pocket center point and be performed vertically from there; thus it is practical to predrill at this position.



### New functions compared to POCKET1:

- The milling direction can be specified via a G command (G2/G3) or as up-cut or down-cut milling from the spindle direction.
- For solid machining, the maximum infeed width in the plane can be programmed.
- Finishing allowance also at the base of the pocket.
- Three different insertion strategies:
  - vertically to the pocket center
  - along a helical path around the pocket center
  - oscillation on the center axis of the pocket
- Short paths during approach in the plane when finishing.
- Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).

### Programming

```
POCKET3 (_RTP, _RFP, _SDIS, _DP, _LENG, _WID, _CRAD, _PA, _PO, _STA,  
_MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _MIDA, _AP1, _AP2,  
_AD, _RAD1, _DP1)
```

## Parameter

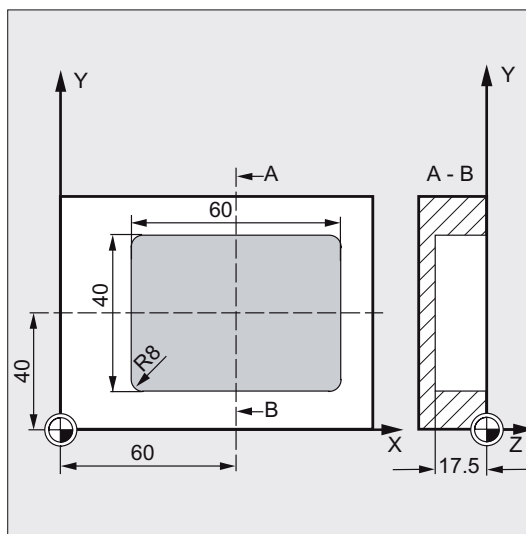
Parameter	Data type	Meaning	
_RTP	real	Retraction plane (absolute)	
_RFP	real	Reference plane (absolute)	
_SDIS	real	Safety clearance (to be added to the reference plane, enter without sign)	
_DP	real	Pocket depth (absolute)	
_LENG	real	Pocket length, for dimensioning from the corner with sign	
_WID	real	Pocket width, for dimensioning from the corner with sign	
_CRAD	real	Pocket corner radius (enter without sign)	
_PA	real	Pocket reference point, abscissa (absolute)	
_PO	real	Pocket reference point, ordinate (absolute)	
_STA	real	Angle between the pocket longitudinal axis and the first axis of the plane (abscissa, enter without sign);	
		Range of values:	$0^{\circ} \leq \text{\_STA} < 180^{\circ}$
_MID	real	Maximum infeed depth (enter without sign)	
_FAL	real	Finishing allowance at the pocket edge (enter without sign)	
_FALD	real	Final machining allowance at base (enter without sign)	
_FFP1	real	Feedrate for surface machining	
_FFD	real	Feedrate for depth infeed	
_CDIR	integer	Milling direction: (enter without sign)	
		Values:	0: Down-cut milling (corresponds to direction of spindle rotation) 1: Down-cut milling 2: with G2 (independent of spindle direction) 3: with G3
_VARI	integer	Machining type: (enter without sign)	
		Values:	UNITS DIGIT: Machining process 1: Roughing 2: Finishing
			TENS DIGIT: Infeed 0: Perpendicular to pocket center with G0 1: Perpendicular to pocket center with G1 2: On helical path 3: Oscillation on pocket longitudinal axis
The other parameters can be selected as options. They define the insertion strategy and overlapping for solid machining:			
_MIDA	real	Maximum infeed width as a value in solid machining in the plane	
_AP1	real	Blank dimension of pocket length	
_AP2	real	Blank dimension of pocket width	
_AD	real	Blank pocket depth dimension from reference plane	
_RAD1	real	Radius of the helical path on insertion (relative to the tool center point path) or maximum insertion angle for reciprocating motion	
_DP1	real	Insertion depth per 360° revolution on insertion along helical path	

### Example of pocket milling

With this program, you can make a pocket with a length of 60 mm, a width of 40 mm, a corner radius of 8 mm and a depth of 17.5 mm in the XY plane (G17). The pocket has an angle of 0 degrees to the X axis. The final machining allowance of the pocket edges is 0.75 mm, 0.2 mm at the base, the safety clearance in the Z axis, which is added to the reference plane, is 0.5 mm. The center point of the pocket lies at X60 and Y40, the maximum depth infeed is 4 mm.

The machining direction results from the direction of rotation of the spindle in the case of down-cut milling.

Merely a rough machining operation is to be carried out.



```

N10 G90 S600 M4                                ;Specification of technology values
N15 T10 D1
N17 M6
N20 G17 G0 X60 Y40 Z5                          ;Approach start position
N25 _ZSD[2]=0                                  ;Dimensioning the pocket via the center
                                                ;point
N30 POCKET3 (5, 0, 0.5, -17.5, 60, ->          ;Cycle call
-> 40, 8, 60, 40, 0, 4, 0.75, 0.2, ->
-> 1000, 750, 0, 11, 5)
N40 M30                                         ;Program end
N10 G90 S600 M4                                ;Specification of technology values
N15 T10 D1

```

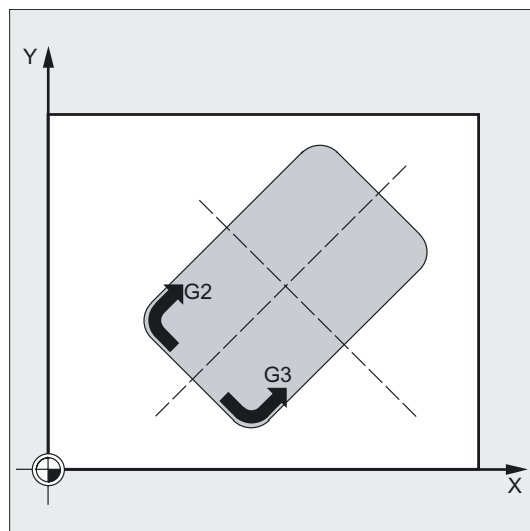
#### Note

-> means: it must be programmed in a single block

## Sequence

### Position reached prior to cycle start:

Starting position is any position from which the pocket center point can be approached at the height of the retraction plane without collision.



### Note

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original coordinate system is active again.

### The cycle creates the following sequence of motions when roughing (VARI = X1):

With G0, the pocket center point is approached at the retraction level, and then, from this position, with G0, too, the reference plane brought forward by the safety clearance is approached. The machining of the pocket is then carried out according to the selected insertion strategy, taking into account the programmed blank dimensions.

#### Insertion strategies:

- **Insertion perpendicular to pocket center (VARI = 0X, VARI = 1X)**  
means that the current infeed depth calculated within the cycle ( $\leq$  maximum infeed depth programmed under \_MID) is carried out in a block with G0 or G1.
- **Insertion on helical path (VARI = 2X)**  
means that the milling center point traverses the helical path specified by the radius \_RAD1 and the depth per revolution \_DP1. In this case, the feedrate programmed under \_FFD is in effect. The direction of rotation of this helical path corresponds to the direction of rotation with which the pocket will be machined.

The insertion depth programmed under `_DP1` is taken into account as the maximum depth and is always calculated as an integer number of revolutions of the helical path.

If the current depth required for an infeed (this can be several revolutions on the helical path) is reached, a full circle is still executed to eliminate the inclined path of insertion.

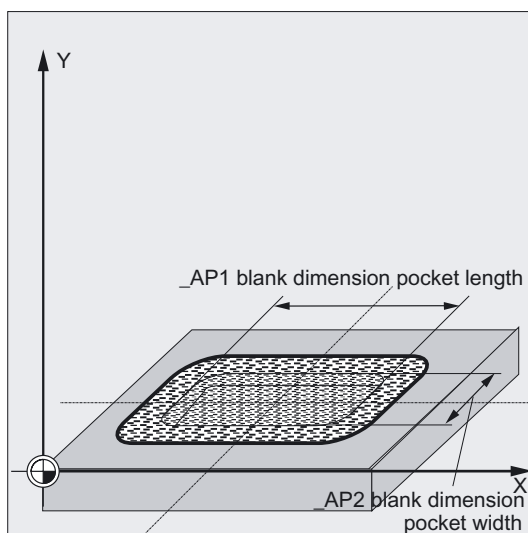
Pocket solid machining then starts in this plane and continues until it reaches the final machining allowance.

The starting point of the described helical path is at the longitudinal axis of the pocket in "plus direction" and is approached with G1.

- **Insertion oscillating on center axis of pocket (VARI = 3X)**

means that the milling center point on a straight line oscillating back and forth is inserted at an angle until it has reached the nearest current depth. The maximum insertion angle is programmed under `_RAD1`, and the length of the oscillation travel is calculated in the cycle. If the current depth is reached, the travel is executed once more without depth infeed in order to eliminate the inclined insertion path. In this case, the feedrate programmed under `_FFD` is in effect.

#### Taking into account the blank dimensions



During solid machining of the pockets, it is possible to take into account blank dimensions (e.g. when machining precast parts).

The blank dimensions for length and width (`_AP1` and `_AP2`) are programmed without sign and are placed by the cycle symmetrically around the pocket center point via calculation. You will define the part of the pocket which is no longer to be machined by solid machining. The blank dimension for the depth (`_AD`) is also programmed without sign and taken into account by the reference plane in the direction of the pocket depth.

The depth infeed when taking into account blank dimensions is carried out according to the programmed type (helical path, reciprocating, vertically). If the cycle detects that there is space enough in the pocket center because of the given blank contour and the radius of the active tool, the infeed is carried out vertically to the pocket center point as long as it is possible in order not to traverse extensive insertion paths in the open.

Solid machining of the pocket is carried out starting from the top downwards.

**The cycle creates the following sequence of motions when finishing (VARI = X2):**

Finishing is performed in the order from the edge until the finishing allowance on the base is reached, and then the base is finished. If one of the finishing allowances is equal to zero, this part of the finishing process is skipped.

- **Finishing on the edge**

While finishing on the edge, the tool will traverse around the pocket contour only once.

For finishing on the edge, the path includes one quadrant reaching the corner radius. The radius of this path is normally 2 mm or, if "less space" is provided, equals to the difference between the corner radius and the mill radius.

If the final machining allowance on the edge is larger than 2 mm, the approach radius is increased accordingly.

The depth infeed is performed with G0 in the open towards the pocket center, and the starting point of the approach path is also reached with G0.

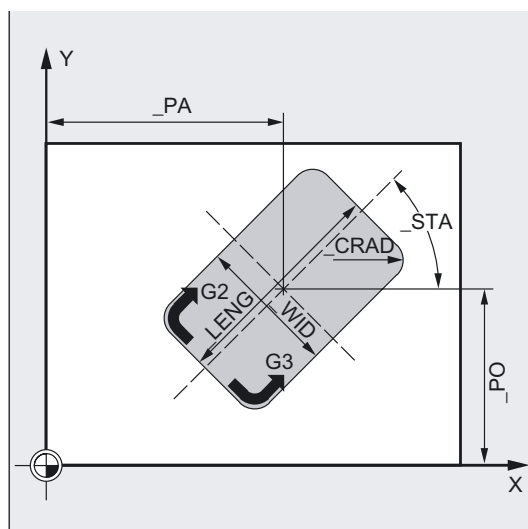
- **Finishing on the base**

During finishing on the base, the machine performs G0 towards the pocket center until reaching a distance equal to pocket depth + finishing allowance + safety clearance. From there, the tool always moves vertically downward at the feedrate for the depth infeed (because a tool that can cut over the face is used for finishing on the base).

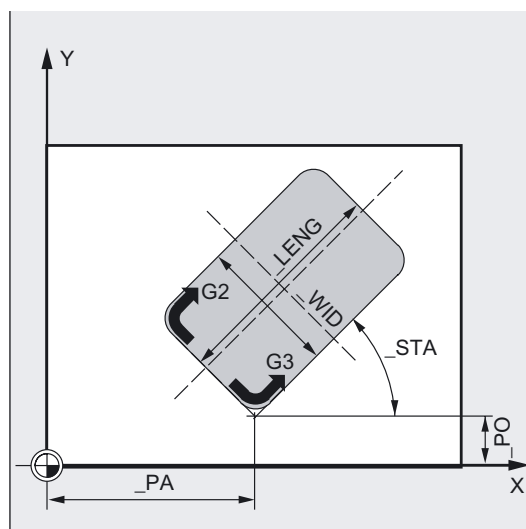
The base surface of the pocket is machined once.

## Explanation of the parameters

**Pocket dimensioned from the center point**



**Pocket dimensioned from the corner**



**\_LENG, \_WID and \_CRAD (pocket length, pocket width and corner radius)**

Use the parameters \_LENG, \_WID and \_CRAD to define the form of a pocket in the plane.

The pocket can be dimensioned from the center or from a corner point. When dimensioning from a corner point, use `_LENG` and `_WID` with sign.

If you cannot traverse the programmed corner radius with the active tool since its radius is larger, then the corner radius of the machine pocket corresponds to the tool radius.

If the milling tool radius is larger than half of the length or width of the pocket, then the cycle will be aborted and alarm 61105 "Cutter radius too large" is output.

#### **`_PA, _PO` (reference point)**

With the parameters `_PA` and `_PO`, you define the reference point of the pocket in the machining plane.

This is either the pocket center point or a corner point. The value of this parameter depends on cycle setting data bit `_ZSD[2]`:

- 0 means pocket center point
- 1 means corner point.

When dimensioning the pocket from a corner, the length and width parameters (`_LENG`, `_WID`) must be entered with sign, thus uniquely defining the position of the pocket.

#### **`_STA` (angle)**

`_STA` indicates the angle between the 1st axis of the plane (abscissa) and the longitudinal axis of the pocket.

The rotation around `_STA` is always carried out at the reference point. (Note setting `_ZSD[2]`).

#### **`_MID` (infeed depth)**

Use this parameter to define the maximum infeed depth.

The depth infeed is performed by the cycle in equally sized infeed steps.

by using `_MID` and the entire depth, the cycle calculates this infeed automatically. The minimum possible number of infeed steps is used as the basis.

`_MID=0` means that infeed is carried out in one cut up to the pocket depth.

#### **`_FAL` (finishing allowance at edge)**

The finishing allowance only affects the machining of the pocket in the plane on the edge.

If the final machining allowance  $\geq$  tool diameter, the pocket will not necessarily be machined completely. The message "Caution: finishing allowance  $\geq$  tool diameter" is displayed.

However, the cycle will continue.

#### **`_FALD` (finishing allowance at the base)**

When roughing, a separate finishing allowance is taken into account at the base.

#### **`_FFD` and `_FFP1` (feedrate for depth and surface)**

The feedrate `_FFD` is effective when inserting into the material.

The feedrate `_FFP1` is active for all movements in the plane which are traversed at feedrate during machining.



**\_CDIR (milling direction)**

Use this parameter to specify the machining direction for the pocket.

The direction of milling can be programmed as follows using the \_CDIR parameter:

- Directly with "2 for G2" and "3 for G3" or
- alternatively, "down-cut" or "up-cut"

Down-cut and up-cut are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

Down-cut	Up-cut
M3 → G3	M3 → G2
M4 → G2	M4 → G3

**\_VARI (machining type)**

Use the parameter \_VARI to define the machining type.

For possible values, see parameter overview.

If a different value is programmed for the parameter \_VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

**\_MIDA (max. infeed width)**

Use this parameter to define the maximum infeed width when solid machining in a plane. Analogously to the known calculation method for the infeed depth (equal distribution of the total depth with the maximum possible value), the width is distributed equally, maximally with the value programmed under \_MIDA.

If this parameter is not programmed or has value 0, the cycle will internally use 80% of the milling tool diameter as the maximum infeed width.

The following parameters must be taken into account when the width infeed calculated from the edge machining is recalculated when reaching the full pocket in depth. Otherwise, the calculated width infeed is retained for the entire cycle.

**AP1, \_AP2, \_AD (blank dimension)**

With the parameters \_AP1, \_AP2 and \_AD, you define the blank dimension (incremental) of the pocket in the plane and depth.

**\_RAD1 (radius)**

Use the parameter \_RAD1 to define the radius of the helical path (referred to the tool center point path) or the max. insertion angle for the reciprocating motion.

**\_DP1 (insertion depth)**

Use the parameter \_DP1 to define the infeed depth when inserting to the helical path.

**Note**

For an explanation of the parameters RTP, RFP, and SDIS, see Drilling, centering – CYCLE81.

For an explanation of the DP parameter, see Milling rectangular pockets - POCKET1

For an explanation of the cycle setting data \_ZSD[1] and \_ZSD[2], see Requirements for milling cycles.

### See also

Drilling, centering - CYCLE81 (Page 44)

Milling rectangular pockets - POCKET1 (Page 132)

Requirements (Page 101)

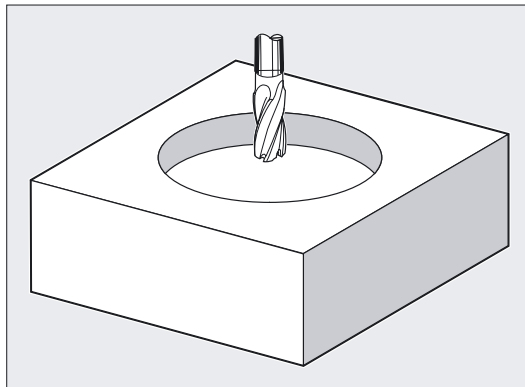
## 3.10 Milling a circular pocket - POCKET4

### Function

Use this cycle to machine circular pockets in the machining plane either "plane-wise" or "helically".

For finishing, a face cutter is required.

The depth infeed will always start at the pocket center point and be performed vertically from there; thus it is practical to predrill at this position.



### New functions compared to POCKET2:

- The milling direction can be specified via a G command (G2/G3) or as up-cut or down-cut milling from the spindle direction.
- For solid machining, the maximum infeed width in the plane can be programmed.
- Finishing allowance also at the base of the pocket.
- Two different insertion strategies:
  - vertically to the pocket center
  - along a helical path around the pocket center
- Short paths during approach in the plane when finishing.
- Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).
- \_MIDA is recalculated during edge machining.
- Helical machining of circular pockets.

## Programming

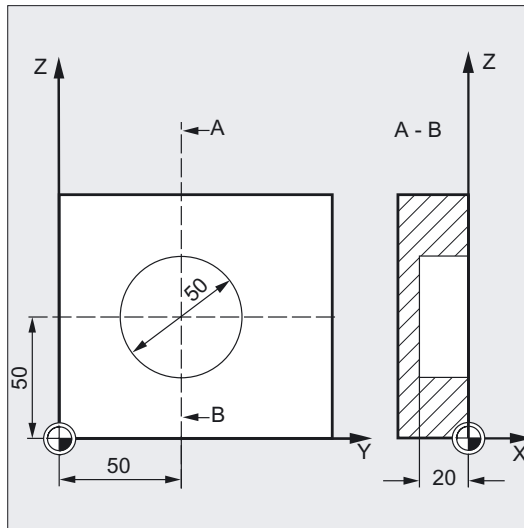
```
POCKET4 (_RTP, _RFP, _SDIS, _DP, _PRAD, _PA, _PO, _MID, _FAL,
_FALD, _FFP1, _FFD, _CDIR, _VARI, _MIDA, _AP1, _AD, _RAD1, _DP1,)
```

## Parameter

Parameter	Data type	Meaning	
_RTP	real	Retraction plane (absolute)	
_RFP	real	Reference plane (absolute)	
_SDIS	real	Safety clearance (to be added to the reference plane, enter without sign)	
_DP	real	Pocket depth (absolute)	
_PRAD	real	Pocket radius	
_PA	real	Pocket center point, abscissa (absolute)	
_PO	real	Pocket center point, ordinate (absolute)	
_MID	real	Maximum infeed depth or maximum pitch with _VARI = helical (enter without sign)	
_FAL	real	Finishing allowance at the pocket edge (enter without sign)	
_FALD	real	Final machining allowance at base (enter without sign)	
_FFP1	real	Feedrate for surface machining	
_FFD	real	Feedrate for depth infeed	
_CDIR	integer	Milling direction: (enter without sign)	
		Values:	0: Down-cut milling (corresponds to direction of spindle rotation) 1: Down-cut milling 2: with G2 (independent of spindle direction) 3: with G3
_VARI	integer	Machining type: (enter without sign)	
		Values:	UNITS DIGIT: Machining process 1: Roughing 2: Finishing
			TENS DIGIT: Infeed 0: Perpendicular to pocket center with G0 1: Perpendicular to pocket center with G1 2: On helical path:
			THOUSANDS DIGIT: Milling technology 0: Plane-wise 1: Helical
The other parameters can be selected as options. They define the insertion strategy and overlapping for solid machining:			
_MIDA	real	Maximum infeed width as a value in solid machining in the plane	
_AP1	real	Blank pocket radius dimension in reference plane (incremental)	
_AD	real	Blank pocket depth dimension from reference plane (incremental)	
_RAD1	real	Radius of the helical path during insertion (referred to the tool center point path)	
_DP1	real	Insertion depth per 360° revolution on insertion along helical path	

### Example of circular pocket milling

With this program, you can make a circular pocket in the YZ plane (G19). The center point is determined by Y50 Z50. The infeed axis for the depth infeed is the X axis. Neither finishing dimension nor safety clearance is specified. The pocket is machined with down-cut milling. Infeed is performed along a helical path.



```

N10 G19 G90 G0 S650 M3                ;Specification of technology values
N15 T20 D1
N17 M6
N20 Y50 Z50                            ;Approach start position
N30 Pocket4(3, 0, 0, -20, 25, 50, ->   ;Cycle call
-> 50, 6, 0, 0, 200, 100, 1, 21, 0, ->
-> 0, 0, 2, 3)
N40 M30                                ;Program end

```

#### Note

-> means: it must be programmed in a single block

## Sequence

### Position reached prior to cycle start:

Starting position is any position from which the pocket center point can be approached at the height of the retraction plane without collision.

---

### Note

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original coordinate system is active again.

---

### The cycle creates the following sequence of motions when roughing (\_VARI = X1):

- Plane-wise (\_VARI = 0XX1)
  - With G0, the pocket center point is approached at the retraction level, and then, from this position, with G0, too, the reference plane brought forward by the safety clearance is approached.
  - The machining of the pocket is then carried out according to the selected insertion strategy, taking into account the programmed blank dimensions.
- Helical (\_VARI = 1XX1)
  - With G0, the pocket center point is approached at the retraction level, and then, from this position, with G0, too, the reference plane brought forward by the safety clearance is approached.
  - The pocket is machined with a helical motion until the pocket depth is reached. A full circle is then machined to remove any residual material.
  - The tool is then retracted along a quadrant (with maximum 2 mm radius) and lifted off simultaneously by the safety clearance. If there is insufficient space in the circular pocket for a 2 mm retraction radius, the radius is reduced to the largest possible value.

### Taking into account the blank dimensions

During solid machining of the pockets, it is possible to take into account blank dimensions (e.g. when machining precast parts).

With circular pockets, the blank dimension \_AP1 is also a circle (with a smaller radius than the pocket radius).

---

### Note

#### Insertion strategies:

See description of POCKET3.

For further information about taking into account the blank dimensions, see POCKET3, as well!

---

**Sequence of motions for finishing (\_VARI = X2):**

Finishing is performed in the order from the edge until the finishing allowance on the base is reached, and then the base is finished. If one of the finishing allowances is equal to zero, this part of the finishing process is skipped.

In machining mode `_VARI= 1XX2` (helical), the tool moves along a helical path down to pocket depth. A full circle is then machined along the edge to remove any residual material.

- **Finishing on the edge:**

While finishing on the edge, the tool will traverse around the pocket contour only once.

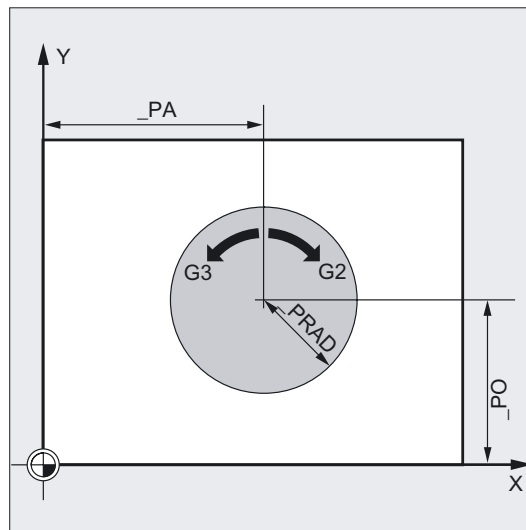
For finishing on the edge, the path includes one quadrant reaching the pocket radius. The radius of this path is 2 mm as the maximum or, if "less space" is provided, equals to the difference between the pocket radius and the mill radius.

The depth infeed is performed with G0 in the open towards the pocket center, and the starting point of the approach path is also reached with G0.

- **Finishing on the base:**

During finishing on the base, the machine performs G0 towards the pocket center until reaching a distance equal to pocket depth + finishing allowance + safety clearance. From there, the tool always moves vertically downward at the feedrate for the depth infeed (because a tool that can cut over the face is used for finishing on the base).

The base surface of the pocket is machined once.

**Explanation of the parameters****\_PRAD (pocket radius)**

The form of the circular pocket is determined solely by its radius.

If this is smaller than the tool radius of the active tool, then the cycle is aborted and alarm 61105 "Cutter radius too large" is output.

**\_PA, \_PO (pocket center point)**

Use the parameters \_PA and \_PO to define the pocket center point. Circular pockets are always dimensioned across the center.

**\_VARI (machining type)**

Use the parameter \_VARI to define the machining type.

For possible values, see parameter overview.

If a different value is programmed for the parameter \_VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

---

**Note****With \_VARI = 1xxx (helical)**

The depth infeed is always entered as a pitch (cursor text: "Max. pitch of helix") in parameter \_MID (text: "Helix infeed").

The infeed value cannot be specified as "Max. insertion angle of helix"!

---

**Edge-only roughing with POCKET4**

With POCKET4 (circular pocket), it is possible to machine the edge only during roughing.

To this end, the blank dimension in depth (parameter \_AD) must be defined such that it is at least as large as the pocket depth (DP) minus the finishing allowance depth (\_FALD).

**Example:**

Requirement: Depth calculation without including safety clearance (\_ZSD[1]=1)

```

_RTP=0      ;Reference plane
_SDIS=2     ;Safety clearance
_DP=-21     ;Pocket depth
_FALD=1.2   ;Depth allowance
→_AD ≥ 19.75 ;Blank dimension for depth must be greater than or equal to the
              ;incremental pocket depth minus depth allowance, i.e., 21-1.25 = 19.75

```

---

**Note**

For an explanation of the parameters RTP, RFP, and SDIS, see Drilling, centering – CYCLE81.

For an explanation of the DP parameter, see Milling rectangular pockets - POCKET1

For an explanation of the parameters MID, FAL, FALD, FFP1, FFD, \_CDIR, \_MIDA, \_AP1, \_AD, \_RAD1, and \_DP1, see Milling a rectangular pocket - POCKET3.

For an explanation of the cycle setting data \_ZSD[1], see Requirements for milling cycles.

---

**See also**

Drilling, centering - CYCLE81 (Page 44)

Milling rectangular pockets - POCKET1 (Page 132)

Milling a rectangular pocket - POCKET3 (Page 142)

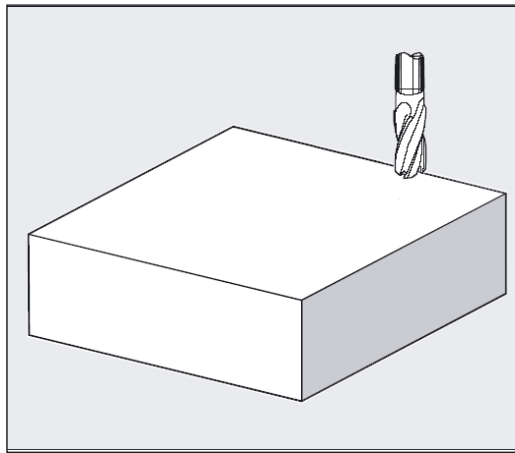
Requirements (Page 101)

## 3.11 Face milling - CYCLE71

### Function

Use CYCLE71 to face mill any rectangular surface. The cycle differentiates between roughing (machining the surface in several steps until reaching the final machining allowance) and finishing (end milling the surface in one step). The maximum infeed in width and depth can be specified.

The cycle operates without cutter radius compensation. The depth infeed is carried out in the open.



### Programming

```
CYCLE71 (_RTP, _RFP, _SDIS, _DP, _PA, _PO, _LENG, _WID, _STA, _MID,
_MIDA, _FDP, _FALD, _FFP1, _VARI, _FDP1)
```

### Parameter

Parameter	Data type	Meaning
_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (to be added to the reference plane, enter without sign)
_DP	real	Depth (absolute)
_PA	real	Starting point, abscissa (absolute)
_PO	real	Starting point, ordinate (absolute)
_LENG	real	Length of the rectangle in the 1st axis, incrementally. The corner from which dimensioning is carried out results from the sign.
_WID	real	Length of the rectangle in the 2nd axis, incrementally. The corner from which dimensioning is carried out results from the sign.
_STA	real	Angle between the longitudinal axis of the rectangle and the first axis of the plane (abscissa, enter without sign);



Parameter	Data type	Meaning	
		Range of values:	$0^\circ \leq \text{\_STA} < 180^\circ$
\_MID	real	Maximum infeed depth (enter without sign)	
\_MIDA	real	Maximum infeed width value for machining in the plane (enter without sign)	
\_FDP	real	Retraction travel in cutting direction (incremental, enter without sign)	
\_FALD	real	Final machining allowance in depth (incremental, enter without sign). In finishing mode, \_FALD refers to the remaining material on the surface.	
\_FFP1	real	Feedrate for surface machining	
\_VARI	integer	Machining type: (enter without sign)	
		Values:	UNITS DIGIT: Machining process 1: Roughing 2: Finishing
			TENS DIGIT: Milling direction 1: parallel to the abscissa, in one direction 2: parallel to the ordinate, in one direction 3: parallel to the abscissa, with alternating direction 4: parallel to the ordinate, with alternating direction
\_FDP1	real	Overrun travel in direction of plane infeed (incremental, enter without sign)	

### Example of face milling a surface

Parameters for the cycle call:

Parameter	Meaning	Value
\_RTP	Retraction plane	10 mm
\_RFP	Reference plane	0
\_SDIS	Safety clearance	2 mm
\_DP	Depth	-11 mm
\_FAL	Finishing allowance in plane	No finishing allowance
\_FALD	Finishing allowance in depth	No finishing allowance
\_PA	Starting point of the rectangle	X = 100 mm
\_PO	Starting point of the rectangle	Y = 100 mm
\_LENG	Rectangle dimensions	X = +60 mm
\_WID	Rectangle dimensions	Y = +40 mm
\_STA	Angle of rotation in the plane	10 degrees
\_MID	Maximum infeed depth	6 mm
\_MIDA	Maximum infeed width	10 mm
\_FDP	Retraction at the end of the milling path	5 mm
\_FFP1	Feedrate in the plane	4000 mm/min
\_VARI	Machining type	Roughing parallel to the X axis with alternating direction
\_FDP1	Overrun on last cut as determined by the cutting edge geometry	2 mm

```

$TC_DP1[1,1] = 120                ;Tool type
$TC_DP6[1.1] =10                  ;Tool radius
N100 T1
N102 M06
N110 G17 G0 G90 G54 G94 F2000 X0 Y0 Z20 ;Approach start position
;
CYCLE71( 10, 0, 2,-11, 100, 100, -> ;Cycle call
-> 60, 40, 10, 6, 10, 5, 0, 4000, 31, 2)
N125 G0 G90 X0 Y0
N130 M30                          ;Program end

```

---

**Note**

-> means: it must be programmed in a single block

---

**Sequence****Position reached prior to cycle start:**

Starting position is any position from which the infeed point can be approached at the height of the retraction plane without collision.

---

**Note**

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

---

**The cycle creates the following sequence of motions:**

- G0 is applied to approach the infeed point at the current position level. The reference plane, brought forward by the safety distance, is then also approached with G0 to this position. then, also with G0, feeding to the machining plane. G0 is possible because the infeed takes place in the open.

There are several roughing strategies (paraxial in one direction or back and forth).

- **Sequence of motions for roughing (VARI = X1):**

According to the programmed values `_DP`, `_MID` and `_FALD`, face milling can be carried out in several planes. Machining is carried out from the top downwards, i.e. one plane each is removed and then the next depth infeed is carried out in the open (parameters `_FDP`).

The traversing paths for solid machining in the plane depend on the values of the parameters `_LENG`, `_WID`, `_MIDA`, `_FDP`, `_FDP1` and the cutter radius of the active tool.

The first path to be milled is always traversed such that the infeed depth exactly corresponds to `_MIDA` so that no width infeed larger than the maximum possible width infeed results. The tool center point thus does not always travel exactly on the edge (only if `_MIDA` = cutter radius).

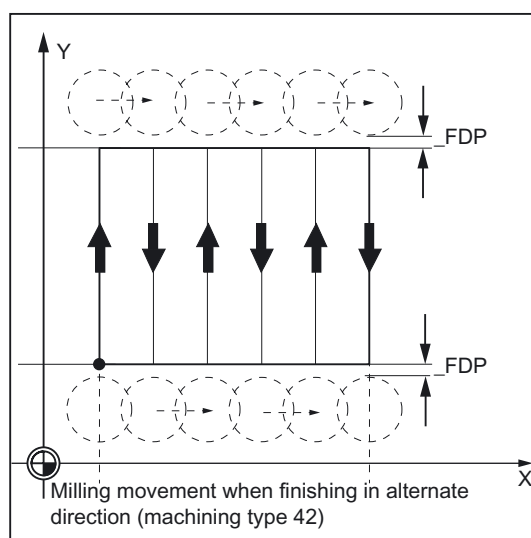
The dimension by which the tool travels outside of the edge is always milling cutter diameter -  $\_MIDA$ , even if only 1 cut is made on the surface, i.e., surface width + overshoot is less than  $\_MIDA$ .

The other paths for width infeed are calculated internally so as to produce a uniform path width ( $\leq \_MIDA$ ).

- **Sequence of motions for finishing (VARI = X2):**

When finishing, the surface is milled in the plane once. This means that the finishing allowance when roughing has to be selected also such that the residual depth can be removed with the finishing tool in one step.

After each surface milling pass in the plane, the tool will retract. The retraction travel is programmed under the parameter  $\_FDP$ .

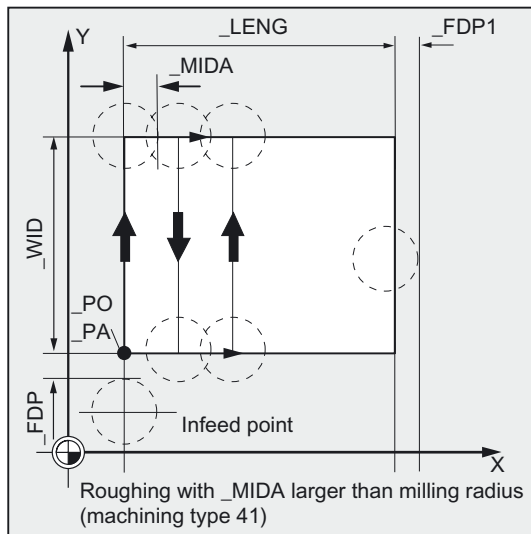


- Machining in one direction stops at the final machining allowance + safety distance and the next starting point is approached in rapid traverse.

Roughing in one direction stops when the calculated infeed depth + safety distance is reached. The depth infeed is performed at the same point as in roughing.

After finishing has been completed, the tool retracts from the last position reached to the retraction plane  $\_RTP$ .

## Explanation of the parameters

**\_DP (depth)**

The depth can be specified as an absolute value (**\_DP**) to the reference plane.

**\_PA, \_PO (starting point)**

With the parameters **\_PA** and **\_PO**, you define the starting point of the surface in the machining plane.

**\_LENG, \_WID (length)**

Use the parameters **\_LENG** and **\_WID** to define the length and width of a rectangle in the plane. The position of the rectangle referred to **\_PA** and **\_PO** results from the sign.

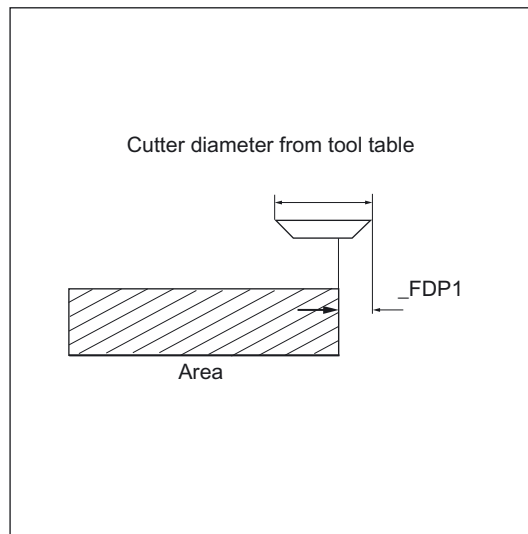
**\_MIDA (max. infeed width)**

Use this parameter to define the maximum infeed width when machining in a plane. Analogously to the known calculation method for the infeed depth (equal distribution of the total depth with the maximum possible value), the width is distributed equally, maximally with the value programmed under **\_MIDA**.

If this parameter is not programmed or has value 0, the cycle will internally use 80% of the milling tool diameter as the maximum infeed width.

**\_FDP (retraction travel)**

Use this parameter to define the dimension for the retraction travel in the plane. This parameter must be programmed with a value greater than zero.

**\_FDP1 (overrun travel)**

Use this parameter to specify an overrun travel in the direction of the plane infeed ( $\_MIDA$ ). Thus, it is possible to compensate the difference between the current cutter radius and the tool nose radius (e.g. cutter radius or cutting tips arranged at an angle). The last cutter center point path thus always results as  $\_LENG$  (or  $\_WID$ ) +  $\_FDP1$  - tool radius (from the compensation table).

**\_FALD (finishing allowance)**

When roughing, a finishing allowance in the depth is taken into account which is programmed under this parameter.

When finishing, the residual material that was programmed as the finishing allowance during roughing and that is still present must be indicated so that the lift-off and subsequent infeed to the starting point of the next cut can be executed without collisions.

**\_VARI (machining type)**

Use the parameter  $\_VARI$  to define the machining type.

Possible settings:

Units digit: (machining process)

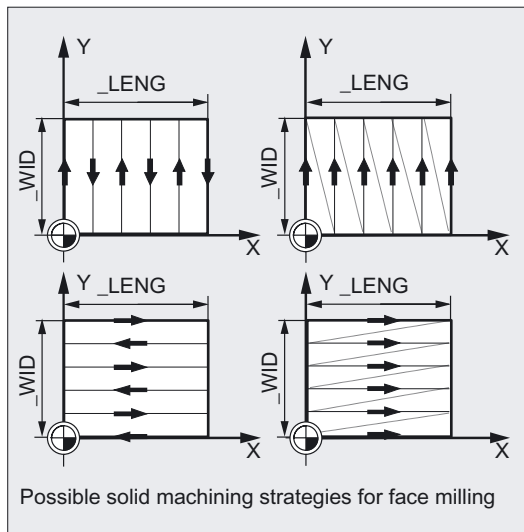
- 1 = roughing to finishing allowance
- 2 = finishing

Tens digit: (milling direction)

The milling cutter travels:

- 1 = parallel to the abscissa, in one direction
- 2 = parallel to the ordinate, in one direction
- 3 = parallel to the abscissa, with alternating direction

- 4 = parallel to the ordinate, with alternating direction



If a different value is programmed for the parameter `_VARI`, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

#### Note

For an explanation of the parameters `RTP`, `RFP`, and `SDIS`, see Drilling, centering – `CYCLE81`.

For an explanation of the parameters `_STA`, `_MID`, and `_FFP1`, see Milling rectangular pockets - `POCKET3`.

For an explanation of the cycle setting data `_ZSD[1]`, see Requirements for milling cycles.

## 3.12 Path milling - CYCLE72

### Function

With CYCLE72, you mill along any user-defined contour. The cycle operates with or without cutter radius compensation.

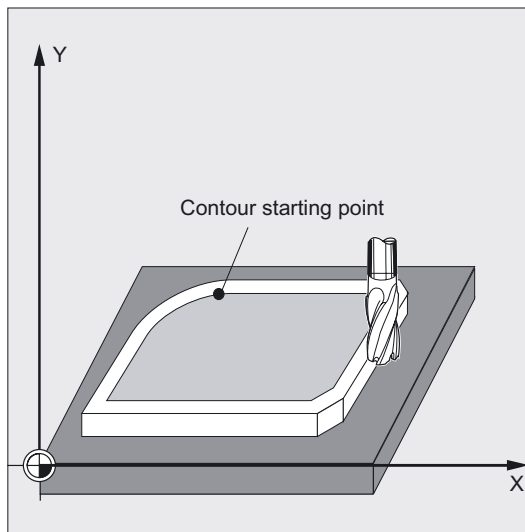
The contour does not need to be closed; internal or external machining is defined by the position of the cutter radius compensation (central, left or right of contour).

The contour must be programmed in the direction in which it is to be milled and must be located in one plane.

In addition, it must also consist of at least 2 contour blocks (starting and finishing point) because the contour subroutine is called directly within the cycle.

#### NOTICE

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.



#### Functions of the cycle:

- Selection of roughing (one-time circumnavigation parallel to the contour while taking into account a finishing allowance at several depths, if necessary, up to the finishing allowance) and smoothing (one-time circumnavigation of the final contour at several depths, if necessary).
- Soft approach and return from the contour, either tangentially or radially (quadrant or semicircle).
- Depth infeeds can be programmed.
- Intermediary motions can be carried out in rapid traverse or with feedrate.

## Programming

CYCLE72 ( \_KNAME, \_RTP, \_RFP, \_SDIS, \_DP, \_MID, \_FAL, \_FALD, \_FFP1, \_FFD, \_VARI, \_RL, \_AS1, \_LP1, \_FF3, \_AS2, \_LP2 )

## Parameter

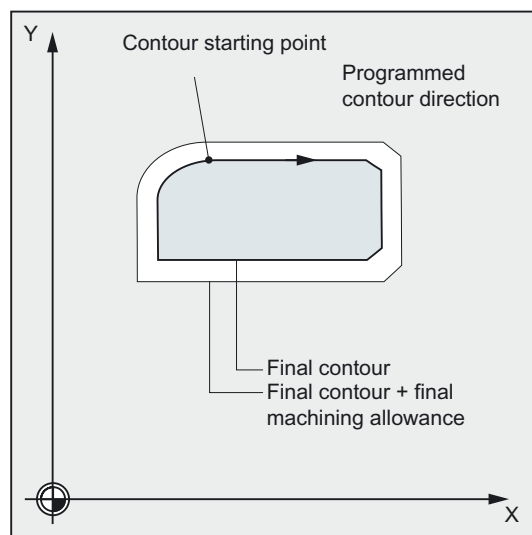
Parameter	Data type	Meaning
_KNAME	string	Name of contour subroutine
_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (to be added to the reference plane, enter without sign)
_DP	real	Depth (absolute)
_MID	real	Maximum infeed depth (incremental, enter without sign)
_FAL	real	Finishing allowance at the edge contour (enter without sign)
_FALD	real	Finishing allowance at the base (incremental, enter without sign)
_FFP1	real	Feedrate for surface machining
_FFD	real	Feedrate for depth infeed (enter without sign)
_VARI	integer	Machining type: (enter without sign)
		Values: UNITS DIGIT: Machining process 1: Roughing 2: Finishing
		TENS DIGIT: Intermediate paths 0: Intermediate paths with G0 1: Intermediate paths with G1
		HUNDREDS DIGIT: Retraction 0: Retraction at contour end to _RTP 1: Retraction at contour end to _RFP + _SDIS 2: Retraction at contour end by _SDIS 3: No retraction at contour end
_RL	integer	Traveling around the contour either centrally, to the right or to the left (with G40, G41 or G42; enter without sign)
		Values: 40: G40 (approach and return only in straight line) 41: G41 42: G42
_AS1	integer	Specification of the approach direction/approach path: (enter without sign)
		Values: UNITS DIGIT: Approach path 1: Straight line tangential 2: Quadrant 3: Semicircle
		TENS DIGIT: Plane/3-D 0: Approach contour in the plane 1: Approach contour on a three-dimensional path
_LP1	real	Length of the approach travel (with straight-line) or radius of the approach arc (with circle) (enter without sign)
The other parameters can be selected as options.		
_FF3	real	Retraction feedrate and feedrate for intermediate positions in the plane (in the open)



Parameter	Data type	Meaning
_AS2	integer	Specification of the return direction/retraction path: (enter without sign)
		Values: UNITS DIGIT: Approach path 1: Straight line tangential 2: Quadrant 3: Semicircle
		TENS DIGIT: Plane/3-D 0: Return from contour in the plane 1: Return from contour on a three-dimensional path
_LP2	real	Length of the retraction travel (with straight-line) or radius of the retraction arc (with circle) (enter without sign)

### Example 1 Milling around a closed contour externally

This program is used to mill a contour as shown in the figure.



### Parameters for the cycle call

Parameter	Meaning	Value
_RTP	Retraction plane	250 mm
_RFP	Reference plane	200
_SDIS	Safety clearance	3 mm
_DP	Depth	175 mm
_MID	Maximum depth infeed	10 mm
_FAL	Finishing allowance in plane	1 mm
_FALD	Finishing allowance in depth	1.5 mm
_FFP1	Feedrate in the plane	800 mm/min
_FFD	Feedrate depth infeed	400 mm/min

Parameter	Meaning	Value
_VARI	Machining type	Roughing up to finishing allowance; intermediate paths with G1, for intermediate paths retraction in Z to _RFP + _SDIS

**Parameters for approach:**

Parameter	Meaning	Value
_RL	G41 - left of the contour, i.e. external machining	41
_LP1	Approach and return in a quadrant in the plane	20 mm radius
_FF3	Retraction feedrate	1,000 mm/min

```

N10 T20 D1 ;T20: Milling cutter with radius 7
N15 M6 ;Changing tool T20,
N20 S500 M3 F3000 ;Program feedrate and spindle speed
N25 G17 G0 G90 X100 Y200 Z250 G94 ;Approach start position
N30 CYCLE72 ("MYCONTOUR", 250, 200, -> ;Cycle call
-> 3, 175, 10,1, 1.5, 800, 400, 111, ->
-> 41, 2, 20, 1000, 2, 20)
N90 X100 Y200
N95 M02 ;Program end

```

```

%_N_MYCONTUR_SPF ;Subroutine for contour milling (for
;example)
; $PATH=/_N_SPF_DIR
N100 G1 G90 X150 Y160 ;Starting point of contour
N110 X230 CHF=10
N120 Y80 CHF=10
N130 X125
N140 Y135
N150 G2 X150 Y160 CR=25
N160 M17

```

**Note**

-> means: it must be programmed in a single block

**Example 2 Milling around a closed contour externally**

With this program, the same contour is milled as in example 1. The difference is that the contour programming is now in the calling program.

```

$TC_DP1[20,1]=120 $TC_DP6[20,1]=7
N10 T20 D1 ;T20: Milling cutter with radius 7
N15 M6 ;Changing tool T20,
N20 S500 M3 F3000 ;Program feedrate and spindle speed
N25 G17 G0 G90 G94 X100 Y200 Z250 -> ;Approach start position,
-> CYCLE72 ("START:END", 250, 200, -> ;cycle call
-> 3, 175, 10, 1, 1.5, 800, 400, 11, ->
-> 41, 2, 20, 1000, 2, 20)
N30 G0 X100 Y200
N35 GOTOF END
START:
N100 G1 G90 X150 Y160
N110 X230 CHF=10
N120 Y80 CHF=10
N130 X125
N140 Y135
N150 G2 X150 Y160 CR=25
END:
N160 M02

```

---

### Note

-> means: it must be programmed in a single block

---

## Sequence

### Position reached prior to cycle start:

Starting position is any position from which the contour starting point can be approached at the height of the retraction plane without collision.

### The cycle creates the following sequence of motions when roughing (VARI = XX1):

The depth infeeds are distributed equally with the maximum possible value of the specified parameters.

- Travel to the starting point for the first roughing with G0/G1 (and \_FF3). This point is calculated within the control system and depends on;
  - the contour starting point (first point in the subroutine),
  - the direction of the contour at the starting point,
  - the approach mode and its parameters and
  - the tool radius.

The cutter radius compensation is activated in this block.

- Depth infeed to first or next machining depth plus programmed safety distance (parameter \_SDIS) with G0/G1. The first machining depth results from

- the total depth,
- the finishing allowance and
- the maximum possible depth infeed.
- Approach the contour vertically with feedrate for depth infeed `_FFD` and then in the plane with the feedrate `_FFP1` programmed for surface milling or 3D with the feedrate programmed under `_FFD` corresponding to the programming for soft approach.
- Milling along the contour with `G40/G41/G42`.
- Soft retraction from the contour with `G1`, still with the feedrate for surface machining by a lift-off amount calculated within the cycle.
- Retraction with `G0 /G1` (and feedrate for intermediate paths `_FF3`), depending on the programming.
- Retraction to the depth infeed point with `G0/G1` (and `_FF3`).
- This sequence is repeated on the next machining plane up to finishing allowance in the depth.

A completion of roughing, the tool stands above the point (calculated internally in the control system) of retraction from the contour at the height of the retraction plane.

**The cycle creates the following sequence of motions when finishing (VARI = XX2):**

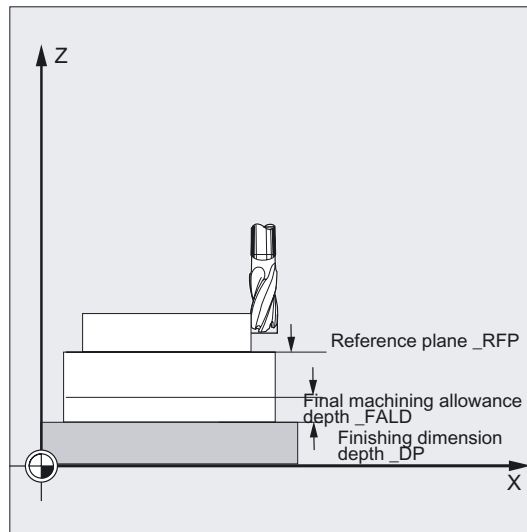
- When finishing, the milling is carried out with the relevant infeed in depth along the contour until the final dimension `_DP` is reached.
- The appropriate path is calculated internally in the control system.
- At the end of the cycle, the tool is positioned at the contour retraction point at the height of the retraction level.

#### **Contour programming**

When programming the contour, please observe the following:

- No programmable frame (TRANS, ROT, SCALE, MIRROR) may be selected in the subroutine before the first programmed position.
- The first block of the contour subroutine is a straight line block containing `G90` and `G0` and defines the contour start.
- The cutter radius compensation is selected / deselected by the higher-level cycle; therefore, no `G40`, `G41`, `G42` is programmed in the contour subroutine.

## Explanation of the parameters



### **\_KNAME (name)**

The contour to be milled is programmed completely in a subroutine. **\_KNAME** defines the name of the contour subroutine.

The milling contour can also be a section of the calling routine or of any other program. The section is identified by start or end labels or by block numbers. Program name and labels/block number are indicated with ":".

### **Examples:**

Name	Meaning
<b>_KNAME="KONTUR_1"</b>	The milling contour is the complete program Kontur_1.
<b>_KNAME="ANFANG:ENDE"</b>	The milling contour is defined as a section in the calling program, which starts from the block containing label ANFANG (START) to the block containing label ENDE (END).
<b>_KNAME=</b> <b>"/_N_SPF_DIR/_N_KONTUR_1_SPF:N130:N210"</b>	The milling contour is defined in blocks N130 to N210 in the program KONTUR_1. The program name must be written out completely with the path and extension. See call description in the reference: /PGA/ Programming Guide Production Planning

### **NOTICE**

If the section is defined with block numbers, then you must note that with **\_KNAME**, the block numbers for the section must also be adjusted after a program change with a subsequent "renumbering" operation.

**\_VARI (machining type)**

Use the parameter \_VARI to define the machining type. For possible values, see "Parameters for CYCLE72".

If a different value is programmed for the parameter \_VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

**\_RL (travel around the contour)**

With the parameter \_RL, you will program traveling around the contour centrally, to the right or to the left with G40, G41 or G42. For possible values, see "Parameters for CYCLE72".

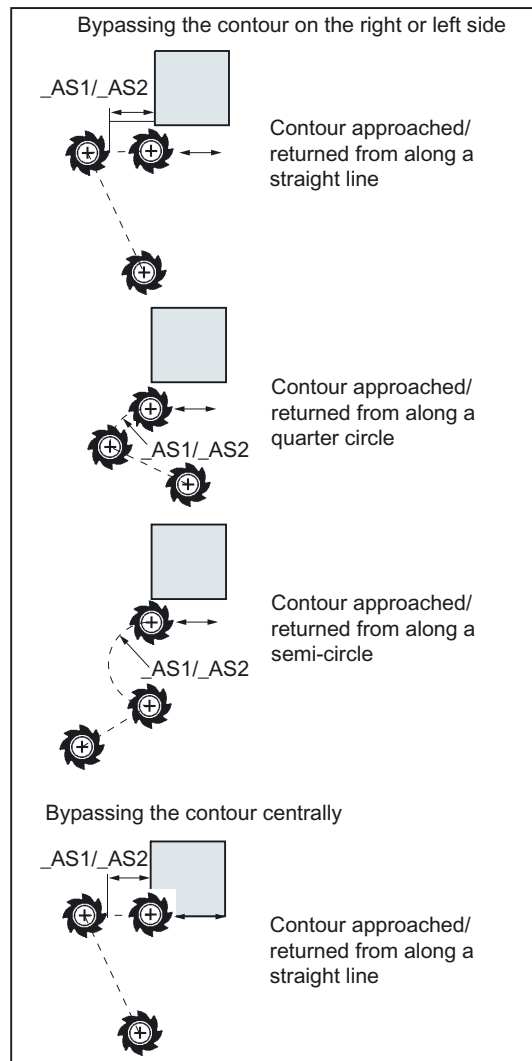
---

**Note**

If the contour is circumnavigated with G40 during roughing (\_RL = 0), the cycle does not provide for any finishing allowance in the planes (parameter \_FAL). \_FAL = 0 must be programmed.

---

**\_AS1, \_AS2 (Approach direction/path, return direction/path)**



Use the parameter `_AS1` to program the specification of the approach path and `_AS2` to program that of the retraction path. For possible values, see "Parameters for CYCLE72". If `_AS2` is not programmed, then the behavior of the retraction path is analogously to that of the approach path.

Smooth approach of the contour along a spatial path (helix or straight line) should only be programmed if the tool is not yet in mesh or is suitable for this type of approach.

**Note**

In the case of central (G40) approach and retraction, only possible along a straight line.

**\_LP1, \_LP2 (length, radius)**

Use parameter \_LP1 to program the approach travel or approach radius (distance from the tool external edge to the contour starting point), and \_LP2 to program the retraction travel or retraction radius (distance from the tool external edge to the contour end point).

Parameters \_LP1, \_LP2 must be set to >0. When the value is 0, error 61116 "Approach or return path = 0" is displayed.

---

**Note**

When using G40, the approach or retraction travel is the distance from the tool center point to the starting or end point of the contour.

---

**\_FF3 (retraction feedrate)**

Use parameter \_FF3 to define a retraction feedrate for intermediate positions in the plane (in the open) if the intermediate motions are to be carried out with feedrate (G01). If no feedrate value is programmed, the intermediate motions with G01 are carried out at surface feedrate.

---

**Note**

For an explanation of the parameters RTP, RFP, and SDIS, see Drilling, centering – CYCLE81.

For an explanation of the parameters \_MID, \_FAL, \_FALD, and \_FFP1, see Milling rectangular pockets - POCKET3.

For an explanation of the parameter \_DP, see Face milling - CYCLE71

For an explanation of the cycle setting data \_ZSD[1], see Requirements for milling cycles.

---

**See also**

Drilling, centering - CYCLE81 (Page 44)

Milling a rectangular pocket - POCKET3 (Page 142)

Face milling - CYCLE71 (Page 156)

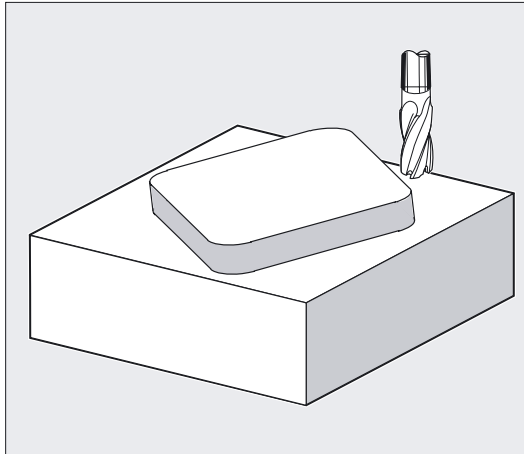
Requirements (Page 101)



### 3.13 Rectangular spigot milling - CYCLE76

#### Function

Use this cycle to machine rectangular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always carried out in the position upstream of the semicircle style approach to the contour.



#### Programming

CYCLE76 ( \_RTP, \_RFP, \_SDIS, \_DP, \_DPR, \_LENG, \_WID, \_CRAD, \_PA, \_PO, \_STA, \_MID, \_FAL, \_FALD, \_FFP1, \_FFD, \_CDIR, \_VARI, \_AP1, \_AP2 )

#### Parameter

The following input parameters are always required:

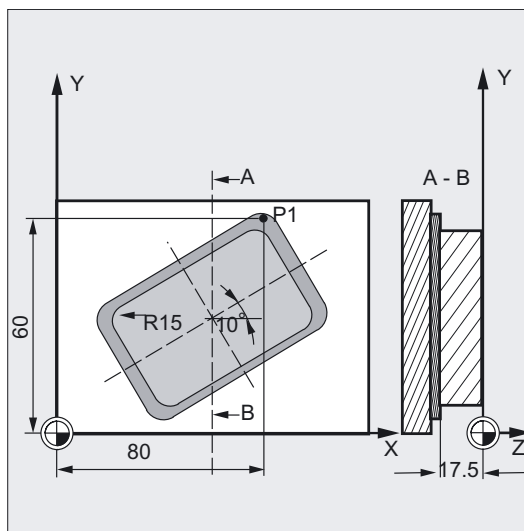
Parameter	Data type	Meaning
_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (to be added to the reference plane, enter without sign)
_DP	real	Depth (absolute)
_DPR	real	Depth relative to the reference plane (enter without sign)
_LENG	real	Spigot length, for dimensioning from corner with sign
_WID	real	Spigot width, for dimensioning from corner with sign
_CRAD	real	Spigot corner radius (enter without sign)
_PA	real	Reference point of spigot, abscissa (absolute)
_PO	real	Reference point of spigot, ordinate (absolute)
_STA	real	Angle between longitudinal axis and 1st axis of plane

## 3.13 Rectangular spigot milling - CYCLE76

Parameter	Data type	Meaning
_MID	real	Maximum depth infeed (incremental; enter without sign)
_FAL	real	Final machining allowance at the margin contour (incremental)
_FALD	real	Finishing allowance at the base (incremental, enter without sign)
_FFP1	real	Feedrate on contour
_FFD	real	Feedrate for depth infeed
_CDIR	integer	Milling direction: (enter without sign)
		Values: 0: Down-cut milling 1: Down-cut milling 2: with G2 (independent of direction of spindle rotation) 3: with G3
_VARI	integer	Machining type:
		Values: 1: Roughing to finishing allowance 2: Finishing
_AP1	real	Length of blank spigot
_AP2	real	Width of blank spigot

## Example of spigot milling

With this program, you can make a spigot with a length of 60 mm, a width of 40 mm, and a corner radius of 15 mm in the XY plane (G17). The spigot has an angle of 10 degrees in relation to the X axis and is programmed from a corner point P1. When corners are used for dimensioning, the length and width of the spigot must be specified with a sign to define the spigot position uniquely. The spigot is premachined with an allowance of 80 mm in length and 50 mm in width.



```

N10 G90 G0 G17 X100 Y100 T20 D1 S3000      ;Specification of technology values
M3
N11 M6
N20 _ZSD[2]=1                               ;Dimensioning of spigot with reference to
                                           ;corners
N30 CYCLE76 (10, 0, 2, -17.5, , -60, ->    ;Cycle call
-> -40, 15, 80, 60, 10, 11, , , 900, ->
-> 800, 0, 1, 80, 50)
N40 M30                                     ;Program end

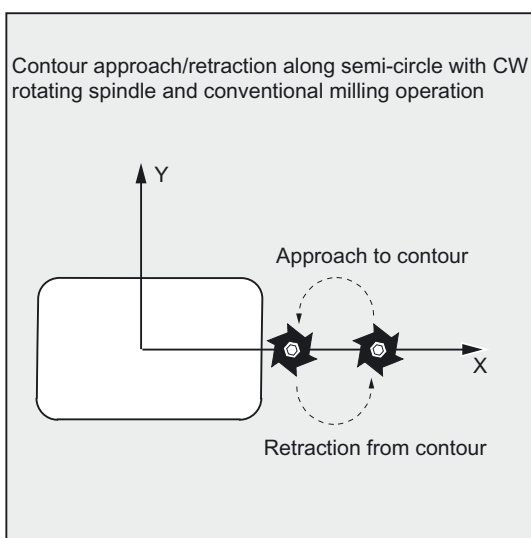
```

**Note**

-> means: it must be programmed in a block

**Sequence****Position reached prior to cycle start:**

Within the cycle, a position located on the first axis in the plane in the positive direction of this axis is calculated as the starting point. It depends on the blank dimension of the spigot and the tool radius, and must be approachable without collisions.

**NOTICE**

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is canceled and alarm 61009 "Active tool number=0" is issued.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original coordinate system is active again.

**The cycle creates the following sequence of motions when roughing (\_VARI = 1):**

- **Approach/retraction from contour:**

The retraction plane (\_RTP) is approached at rapid traverse rate to be able to position then at this height to the starting point in the machining plane.

The tool is feed to the safety clearance (\_SDIS) at rapid traverse with subsequent traversing to the machining depth at feedrate.

To approach the spigot contour, the tool will travel along a semicircle path.

The milling direction can be determined either as up-cut milling or down-cut milling with reference to the spindle direction.

If the spigot is bypassed once, the contour is left along a semi-circle in the plane, and the tool is fed to the next machining depth.

The contour is then reapproached along a semicircle and the spigot is circumnavigated once. This process is repeated until the programmed spigot depth is reached. Then, the retraction plane (\_RTP) is approached at rapid traverse rate.

- **Depth infeed:**

- Feeding to the safety clearance

- Insertion to machining depth

The first machining depth is calculated from:

- the total depth,

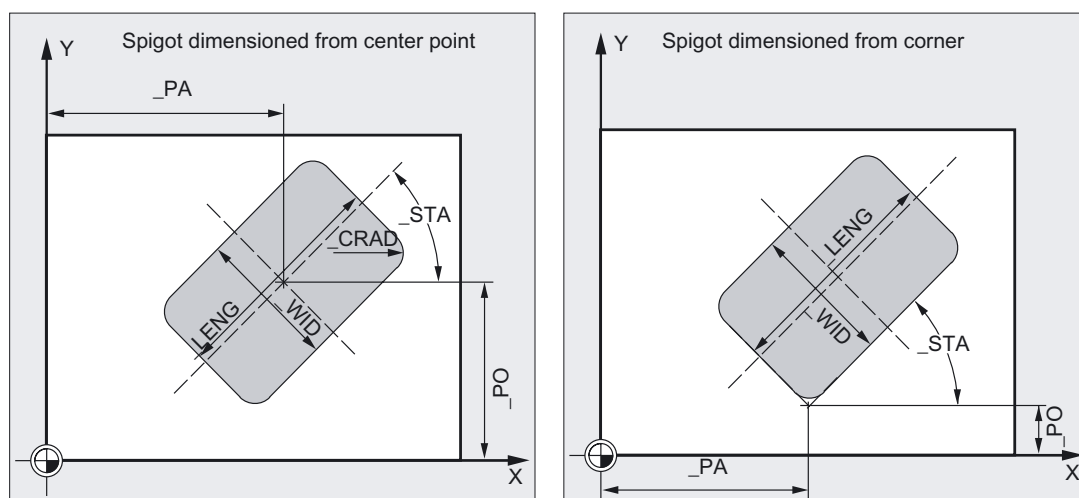
- the finishing allowance and

- the maximum possible depth infeed.

**The cycle creates the following sequence of motions when smoothing (\_VARI = 2)**

Depending on the programmed parameters \_FAL and \_FALD, smoothing is carried out on the peripheral contour, on the base, or both. The approach strategy corresponds to the motions in the plane as with roughing.

## Explanation of the parameters



### **\_LENG, \_WID and \_CRAD (spigot length, spigot width, and corner radius)**

Use the parameters \_LENG, \_WID, and \_CRAD to define the form of a spigot in the plane.

The spigot can be dimensioned from the center or from a corner point. When dimensioning from a corner point, use \_LENG and \_WID with sign.

The length value (\_LENG) always refers to the first axis (zero degrees for plane angle).

### **\_PA, \_PO (reference point)**

With the parameters \_PA and \_PO, you define the reference point of the spigot in the machining plane.

This is either the spigot center point or a corner point. The value of this parameter depends on cycle setting data bit \_ZSD[2]:

- 0 means spigot center point
- 1 means corner point.

When the spigot is dimensioned from a corner, the length and width parameters (\_LENG, \_WID) must be entered with sign so that a unique position for the spigot is defined.

### **\_STA (angle)**

\_STA specifies the angle between the 1st axis of the plane (abscissa) and the longitudinal axis of the spigot.

The rotation around \_STA is always carried out at the reference point. (Note setting \_ZSD[2]).

### **\_CDIR (milling direction)**

Use this parameter to specify the machining direction for the spigot. With the parameter \_CDIR, the milling direction can be programmed directly as "2 for G2" and "3 for G3" or alternatively as "down-cut" or "up-cut". Down-cut and up-cut milling are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

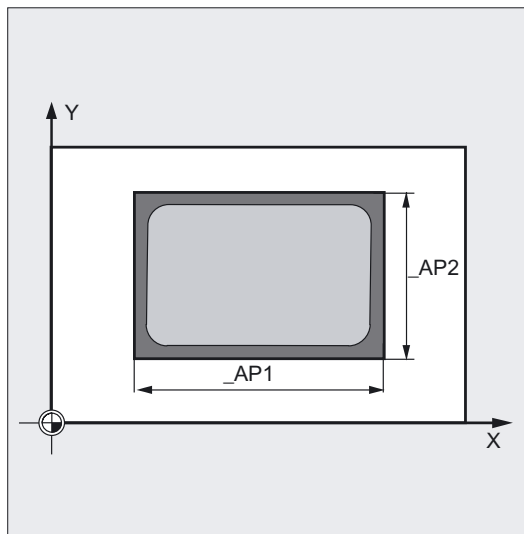
Down-cut	Up-cut
M3 → G3	M3 → G2
M4 → G2	M4 → G3

**\_VARI (machining type)**

Use the parameter \_VARI to define the machining type.

Possible settings:

- 1 = roughing
- 2 = finishing

**\_AP1, \_AP2 (blank dimensions)**

When machining the spigot, it is possible to take into account blank dimensions (e.g. when machining precast parts). The basic sizes for the length and width (\_AP1 and \_AP2) are programmed without sign and their symmetrical positions around the spigot center are computed in the cycle. The internally calculated radius of the approach semicircle depends on this dimension.

**Note**

For an explanation of the parameters RTP, RFP, SDIS, \_DP, and \_DPR, see Drilling, centering – CYCLE81.

For an explanation of the parameters \_MID, \_FAL, \_FALD, \_FFP1, and \_FFD, see Milling rectangular pockets - POCKET3.

For an explanation of the cycle setting data \_ZSD[2], see Requirements for milling cycles.

**See also**

Drilling, centering - CYCLE81 (Page 44)

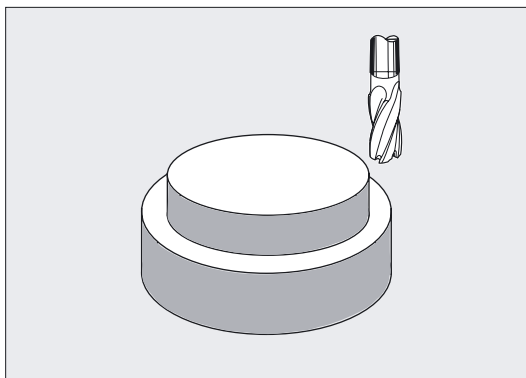
Milling a rectangular pocket - POCKET3 (Page 142)

Requirements (Page 101)

## 3.14 Circular spigot milling - CYCLE77

### Function

Use this cycle to machine circular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always carried out in the position upstream of the semicircle style approach to the contour.



### Programming

CYCLE77 (\_RTP, \_RFP, \_SDIS, \_DP, \_DPR, \_PRAD, \_PA, \_PO, \_MID, \_FAL, \_FALD, \_FFP1, \_FFD, \_CDIR, \_VARI, \_AP1)

### Parameter

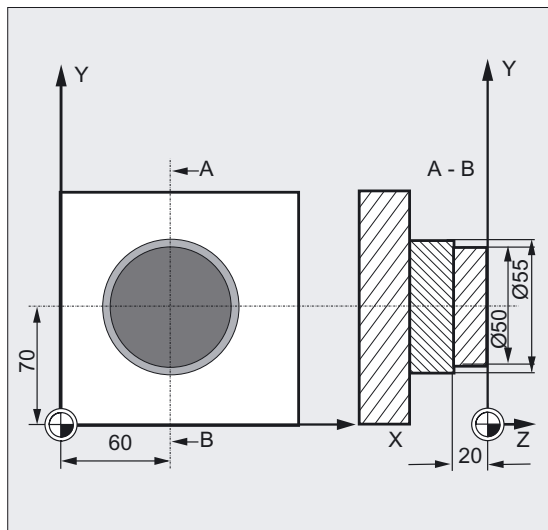
The following input parameters are always required:

Parameter	Data type	Meaning
_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (to be added to the reference plane, enter without sign)
_DP	real	Depth (absolute)
_DPR	real	Depth relative to the reference plane (enter without sign)
_PRAD	real	Spigot diameter (enter without sign)
_PA	real	Center point of spigot, abscissa (absolute)
_PO	real	Center point of spigot, ordinate (absolute)
_MID	real	Maximum depth infeed (incremental; enter without sign)
_FAL	real	Final machining allowance on edge contour (incremental)
_FALD	real	Finishing allowance at the base (incremental, enter without sign)
_FFP1	real	Feedrate on contour
_FFD	real	Feedrate for depth infeed (or spatial infeed)
_CDIR	integer	Milling direction: (enter without sign)

Parameter	Data type	Meaning
		Values: 0: Down-cut milling 1: Down-cut milling 2: with G2 (independent of direction of spindle rotation) 3: with G3
_VARI	integer	Machining type
		Values: 1: Roughing to finishing allowance 2: Smoothing (allowance X/Y/Z=0)
_AP1	real	Diameter of blank spigot

### Example of circular spigot milling

Machining a spigot from a blank with a diameter of 55 mm and a maximum infeed of 10 mm per cut; specification of a final machining allowance for subsequent finishing of the spigot surface. The whole machining is performed with reverse rotation.



```

N10 G90 G17 G0 S1800 M3 D1 T1          ;Specification of technology values
N11 M6
N20 CYCLE77 (10, 0, 3, -20, ,50, 60, -> ;Roughing cycle call
-> 70, 10, 0.5, 0, 900, 800, 1, 1, 55)
N30 D1 T2 M6                            ;Change tool
N40 S2400 M3                             ;Specification of technology values
N50 CYCLE77 (10, 0, 3, -20, , 50, 60, -> ;Finishing cycle call
-> 70, 10, 0, 0, 800, 800, 1, 2, 55)
N60 M30                                 ;Program end

```

### Note

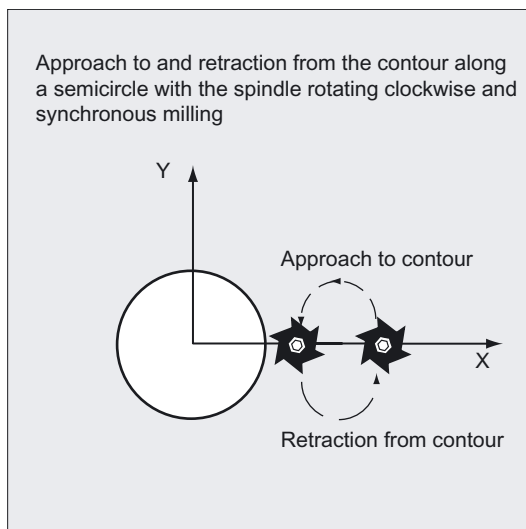
-> means: it must be programmed in a block



## Sequence

### Position reached prior to cycle start:

The starting position is calculated within the cycle exactly as in CYCLE76 (see Section 3.13, Milling a rectangular spigot - CYCLE76).



### NOTICE

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is canceled and alarm 61009 "Active tool number=0" is issued.

Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

At the end of the cycle, the original coordinate system is active again.

### The cycle creates the following sequence of motions when roughing ( $\_VARI = 1$ ):

- **Approach/retraction from contour:**

The retraction plane ( $\_RTP$ ) is approached at rapid traverse rate to be able to position then at this height to the starting point in the machining plane.

The tool is feed to the safety clearance ( $\_SDIS$ ) at rapid traverse with subsequent traversing to the machining depth at feedrate. To approach the spigot contour, the tool is approached along a semicircle path, taking into account the programmed blank spigot.

The milling direction can be determined either as up-cut milling or down-cut milling with reference to the spindle direction.

If the spigot is bypassed once, the contour is left along a semi-circle in the plane, and the tool is fed to the next machining depth.

The contour is then reapproached along a semicircle and the spigot is circumnavigated once. This process is repeated until the programmed spigot depth is reached.

Then, the retraction plane ( $\_RTP$ ) is approached at rapid traverse rate.

- **Depth infeed:**

- Feeding to the safety clearance
- Insertion to machining depth

The first machining depth is calculated from:

- the total depth,
- the finishing allowance and
- the maximum possible depth infeed.

**The cycle creates the following sequence of motions when smoothing (\_VARI = 2):**

Depending on the programmed parameters \_FAL and \_FALD, smoothing is carried out on the peripheral contour, on the base, or both. The approach strategy corresponds to the motions in the plane as with roughing.

### Explanation of the parameters

#### **PRAD (diameter of spigot)**

Enter the diameter without sign.

#### **\_PA, \_PO (spigot center point)**

Use the parameters \_PA and \_PO to define the reference point of the spigot.

Circular spigots are always measured from the center.

#### **\_CDIR (milling direction)**

Use this parameter to specify the machining direction for the spigot.

With the parameter \_CDIR, the milling direction can be programmed directly as "2 for G2" and "3 for G3" or alternatively as "down-cut" or "up-cut". Down-cut and up-cut milling are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

Down-cut	Up-cut
M3 → G3	M3 → G2
M4 → G2	M4 → G3

#### **\_VARI (machining type)**

Use the parameter \_VARI to define the machining type.

Possible settings:

- 1 = roughing
- 2 = finishing

#### **\_AP1 (diameter of blank spigot)**

Use this parameter to define the blank dimension of the spigot (without sign). The internally calculated radius of the approach semicircle depends on this dimension.

---

**Note**

For an explanation of the parameters RTP, RFP, SDIS, \_DP, and \_DPR , see Drilling, centering – CYCLE81.

For an explanation of the parameters \_MID, \_FAL, \_FALD, \_FFP1, and \_FFD, see Milling rectangular pockets - POCKET3.

---

**See also**

Drilling, centering - CYCLE81 (Page 44)

Milling a rectangular pocket - POCKET3 (Page 142)

## 3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75

### 3.15.1 General information

---

**Note**

Pocket milling with islands is an option that requires SW6 in both the NCK and HMI Advanced.

---

#### Function

Cycles CYCLE73, CYCLE74 and CYCLE75 enable you to machine pockets with islands.

The contours of the pocket and islands are defined in DIN code in the same program as the pocket machining operation or as a subroutine.

Cycles CYCLE74 and CYCLE75 transfer the pocket edge contour or island contours to CYCLE73, which is the actual pocket milling cycle.

CYCLE73 uses a geometry processor to generate and execute one or more machining programs. To ensure correct program processing, it is important to program cycle calls in the proper sequence.

- CYCLE74( ) ;transfer of edge contour
- CYCLE75( ) ;transfer of island contour 1
- CYCLE75( ) ;transfer of island contour 2
- ...
- CYCLE73( ) ;machining of pocket

**Requirement**

To use the pocket milling cycle with islands, you need specific machine data settings.

**References:**

For current information, see:

- "siemensd.txt" file in the supplied software (standard cycles)

### 3.15.2 Transfer pocket edge contour - CYCLE74

#### Function

Cycle CYCLE74 transfers the pocket edge contour to pocket milling cycle CYCLE73. This is achieved by creating a temporary internal file in the standard cycles directory and storing the transferred parameter values in it.

If a file of this type already exists, it is deleted and set up again.

For this reason, a program sequence for milling pockets with islands must always begin with a call for CYCLE74.

#### Programming

```
CYCLE74 (_KNAME, _LSANF, _LSEND)
```

#### Example

See CYCLE75.

#### Parameter

Parameter	Data type	Meaning
_KNAME	string	Name of contour subroutine of pocket edge contour
_LSANF	string	Block number/label identifying start of contour definition
_LSEND	string	Block number/label identifying end of contour definition

#### Explanation of the parameters

The edge contour can be programmed either in a separate program or in the main program that calls the routine. Transfer to the cycle takes place via the \_KNAME parameter, the name of the program, and \_LSANF, LSEND, identification of the program section from...to by block numbers or labels, whereby not all of these need to be programmed.

The following options are available for contour programming:

- **Contour is in its own program**, in this case, only \_KNAME must be programmed; e.g. CYCLE74 ("EDGE", "", "")
- **Contour is in the calling program**, in this case, only \_LSANF and \_LSEND must be programmed; e.g. CYCLE74 ("", "N10", "N160")
- **The edge contour is a section of a program, but not of the program calling the cycle**, in this case, all three parameters must be programmed. e.g. CYCLE74("EDGE", "LABEL\_START", "LABEL\_END")

The program name can be described by its path name and program type.

#### Example:

```
_KNAME=" /N_WKS_DIR/ _N_EXAMPLE3_WPD/ _N_EDGE_MPF"
```

### 3.15.3 Transfer island contour - CYCLE75

#### Function

##### Note

Pocket milling with islands is an option that requires SW6 in both the NCK and HMI Advanced.

Cycle CYCLE75 transfers the island contours to pocket milling cycle CYCLE73. The cycle is called once per island contour. It does not need to be called if no island contours are programmed.

The transferred parameter values are written to the temporary file opened by CYCLE74.

#### Programming

```
CYCLE75 (_KNAME, _LSANF, _LSEND)
```

#### Parameter

Parameter	Data type	Meaning
_KNAME	string	Name of contour subroutine of island contour
_LSANF	string	Block number/label identifying start of contour definition
_LSEND	string	Block number/label identifying end of contour definition

#### Contour programming

Pocket edge and island contours must always be closed, i.e., the start and end points are identical.

The starting point, i.e., the first point in every contour, must always be programmed with G0. All other contour elements are programmed via G1 to G3.

When programming the contour, the last contour element must not contain a radius or chamfer.

The tool may not be located at a starting position of the programmed contour element before CYCLE73 is invoked.

The necessary programs must always be stored in one directory only (workpiece or parts program). It is permissible to use the global subroutine memory for pocket edge or island contours.

Workpiece-related geometric dimensional data may be programmed in either metric or imperial dimensions. Switching between these units of measurement within individual contour programs causes errors in the machining program.

When G90/G91 are programmed alternately in contour programs, care must be taken to program the correct dimensional command at the start of the program in the sequence of contour programs to be executed.

Only the geometries in the plane are taken into account when calculating the machining program for the pocket.

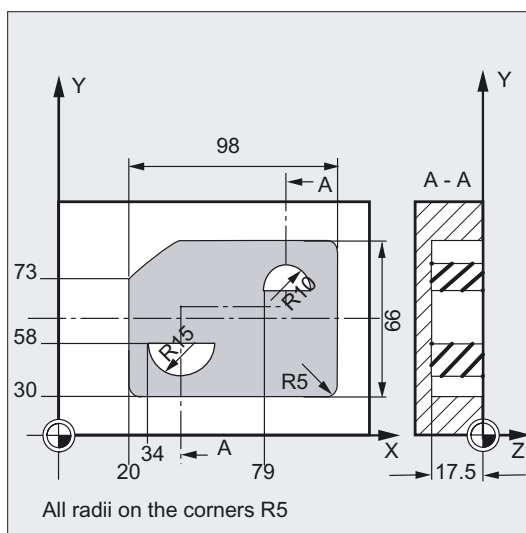
If other axes or functions (T., D., S., M., etc.) are programmed in the contour sections, these are skipped when the contour is preprocessed in the cycle.

All technical machine-specific program commands (e.g., tool call, speed, M command) must be programmed before the cycle commences. Feedrates must be set as parameters in CYCLE73.

The tool radius must be greater than zero.

It is not possible to repeat island contours by offsets implemented by suitable control commands (e.g., zero offset, frames, etc.). Every island to be repeated always has to be programmed again with the offsets calculated in the coordinates.

### Example of contour programming for pocket milling



```
%_N_SAMPLE1_MPF
; $PATH=_N_MPF_DIR
; Example_1: Pocket with islands
;
$TC_DP1[5,1]=120 $TC_DP3[5,1]=111 $TC_DP6[5,1]=6      ;Tool offset cutter T5 D1
$TC_DP1[2,2]=120 $TC_DP3[2,2]=130 $TC_DP6[2,2]=5
N100 G17 G40 G90                                       ;Initial conditions G code
N110 T5 D1                                             ;Load milling tool
N120 M6
N130 S500 M3 F2000 M8
```

```

GOTOF _MACHINE
;
N510 _EDGE:G0 G64 X25 Y30 F2000 ;Define edge contour
N520 G1 X118 RND=5
N530 Y96 RND=5
N540 X40 RND=5
N545 X20 Y75 RND=5
N550 Y35
N560 _ENDEGE:G3 X25 Y30 CR=5
;
N570 _ISLAND1:G0 X34 Y58 ;Define bottom island
N580 G1 X64
N590 _ENDISLAND1:G2 X34 Y58 CR=15
;
N600 _ISLAND2:G0 X79 Y73 ;Define top island
N610 G1 X99
N620 _ENDISLAND2:G3 X79 Y73 CR=10
;
_MACHINE:
;Programming of contours
SAMPLE_CONT:
CYCLE74 ("Example1", "_EDGE", "_FINAL_EDGE") ;Transfer edge contour
CYCLE75 ("Example1", "_ISLAND1", "_FINAL_ISLAND1") ;Transfer island contour 1
CYCLE75 ("Example1", "_ISLAND2", "_FINAL_ISLAND2") ;Transfer island contour 2
ENDLABEL:
M30

```

**Explanation of the parameters**

The number and meaning of parameters are the same as for CYCLE74.

**See also**

Transfer pocket edge contour - CYCLE74 (Page 185)



### 3.15.4 Pocket milling with islands - CYCLE73

#### 3.15.4.1 General information

##### Function

---

**Note**

Pocket milling with islands is an option that requires SW6 in both the NCK and HMI Advanced.

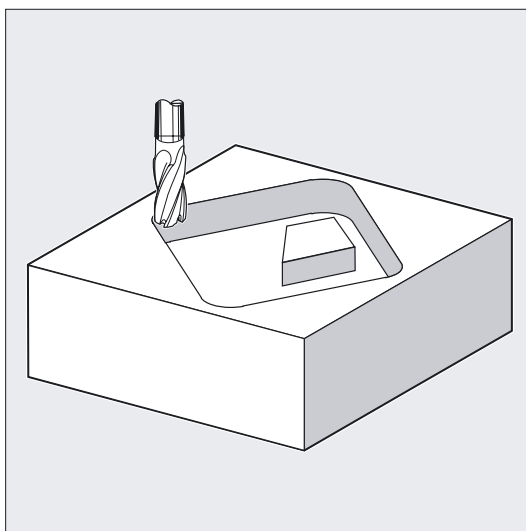
---

CYCLE73 is a machining cycle used for machining any contour axes with or without islands. It supports complete machining of this type of pocket and offers the following machining operations:

- Predrilling
- Solid machine pocket
- Machine residual material
- Finish edge
- Finish base

Pocket and island contours are freely programmed in DIN code supported, for example, by the geometry processor.

The cycle is executed once for each operation according to the programmed machining type (`_VARI`). In other words, in applications requiring roughing and finishing, or an additional rough-cut residual material operation, CYCLE73 must be called a second time.



**Solid machine pocket**

When a pocket is solid machined, it is machined with the active tool down to the programmed final machining allowances. The insertion strategy for milling can be selected. The cutting operation is segmented in the pocket depth direction (tool axis) in accordance with the specified values.

**Machine residual material**

The cycle allows material to be removed with a smaller milling tool. The traversing motions defined by the residual material of the last milling operation and the current tool radius are output in the generated program. The residual material technology can be programmed repeatedly with a succession of decreasing tool radii.

There is no check to determine whether residual material remains in the pocket.

**Edge/base finishing**

Another function of the cycle is to finish the pocket base or circumnavigate the pocket and individual islands in a finish operation.

**Predrilling**

Depending on the milling tool used, it may be necessary to drill before solid machining the workpiece. The cycle automatically calculates the predrilling positions as a function of the solid machining operation to be performed subsequently. The drilling cycle called modally beforehand is executed at each of these positions. Predrilling can be executed in a number of technological machining operations (e.g., 1. centering, 2. drilling).

**Programming**

CYCLE73 (\_VARI, \_BNAME, \_PNAME, \_TN, \_RTP, \_RFP, \_SDIS, \_DP, \_DPR, \_MID, \_MIDA, \_FAL, \_FALD, \_FFP1, \_FFD, \_CDIR, \_PA, \_PO, \_RAD, \_DP1, \_DN)

**Parameter**

Parameter	Data type	Meaning
_VARI	integer	Machining type: (enter without sign)
		Values:
		UNITS DIGIT: Machining process 1: Roughing (solid machining) from solid material 2: Roughing of residual material 3: Finishing of edge 4: Finishing of base 5: Predrilling
		TENS DIGIT: Insertion strategy 1: Perpendicular to G1 2: On helical path 3: Oscillation
		HUNDREDS DIGIT: Lift-off mode 0: on retraction plane (_RTP) 1: by amount of safety clearance (_SDIS) over reference plane (_RFP)
_BNAME	string	THOUSANDS DIGIT: Start point 1: automatic 2: manual
		Name for program of drill positions

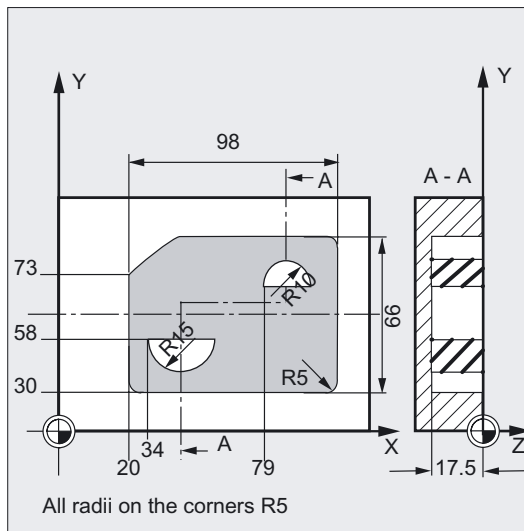
## 3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75

Parameter	Data type	Meaning
_PNAME	string	Name for pocket milling machining program
_TN	string	Name of solid machining tool
_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (to be added to the reference plane, enter without sign)
_DP	real	Pocket depth (absolute)
_DPR	real	Pocket depth (incremental)
_MID	real	Maximum infeed depth for one infeed (enter without sign)
_MIDA	real	Maximum infeed width in the plane (enter without sign)
_FAL	real	Final machining allowance in the plane (enter without sign)
_FALD	real	Final machining allowance on base (enter without sign)
_FFP1	real	Feedrate for surface machining
_FFD	real	Feedrate for depth infeed
_CDIR	integer	Mill direction for machining the pocket: (enter without sign)
		Values: 0: Down-cut milling (corresponds to direction of spindle rotation) 1: Down-cut milling 2: with G2 (independent of spindle direction) 3: with G3
_PA	real	Start point in first axis (only with manual selection of start point)
_PO	real	Start point in second axis (only with manual selection of start point)
_RAD	real	Radius of center point path for insertion along helical path or max. insertion angle for oscillating insertion
_DP1	real	Insertion depth per 360° revolution on insertion along helical path
_DN	integer	Tool offset number of stock removal tool (D number)

## 3.15.4.2 Examples

## Example 1

The machining task consists of milling a pocket with 2 islands from solid material, followed by finishing in the plane X,Y.



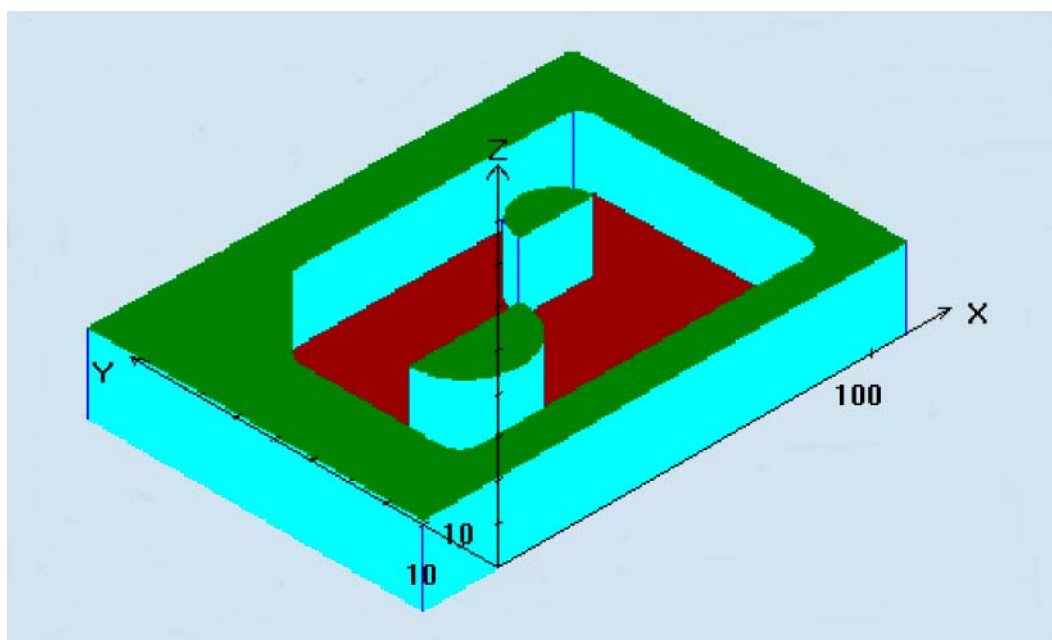
```
%_N_SAMPLE1_MPF
;$PATH=/_N_WKS_DIR/_N_CC73BEI1_WPD
;Example_1: Pocket with islands
;Solid machine and finish
$TC_DP1[5,1]=120 $TC_DP3[5,1]=111 $TC_DP6[5,1]=4 ;Tool offset cutter T5 D1
$TC_DP1[2,1]=120 $TC_DP3[2,1]=130 $TC_DP6[2,1]=5 ;Tool offset cutter T2 D1
N100 G17 G40 G90 ;Initial conditions G code
N110 T5 D1 ;Load milling tool
N120 M6
N130 M3 F2000 S500 M8
N140 GOTOF _MACHINE
;
N510 _EDGE:G0 G64 X25 Y30 ;Define edge contour
N520 G1 X118 RND=5
N530 Y96 RND=5
N540 X40 RND=5
N545 X20 Y75 RND=5
N550 Y35
N560 _ENDEGE:G3 X25 Y30 CR=5
;
N570 _ISLAND1:G0 X34 Y58 ;Define bottom island
```

```

N580 G1 X64
N590 _ENDISLAND1:G2 X34 Y58 CR=15
;
N600 _ISLAND2:G0 X79 Y73 ;Define top island
N610 G1 X99
N620 _ENDISLAND2:G3 X79 Y73 CR=10
;
;Programming of contours
_MACHINE:
GOTOF ENDLABEL
SAMPLE1_CONT:
CYCLE74(,"_EDGE","_ENDEGE")
CYCLE75(,"_ISLAND1","_ENDISLAND1")
CYCLE75(,"_ISLAND2","_ENDISLAND2")
ENDLABEL:
;
; Programming of pocket milling
REPEAT EXAMPLE1_CONT ENDLABEL
CYCLE73(1021,"EXAMPLE1_DRILL","EXAMPLE1_MILL1","5",10,0,1,-
17.5,0,0,2,0.5,0,9000,3000,0,0,0,4,3)
;
; Milling cutter for finishing
T2 D1 M6
S3000 M3
;
; Programming of pocket finishing
REPEAT EXAMPLE1_CONT ENDLABEL
CYCLE73(1023,"EXAMPLE1_DRILL","EXAMPLE1_MILL3","5",10,0,1,-
17.5,0,0,2,0,0,9000,3000,0,0,0,4,2)
M30

```

**Machining result:**



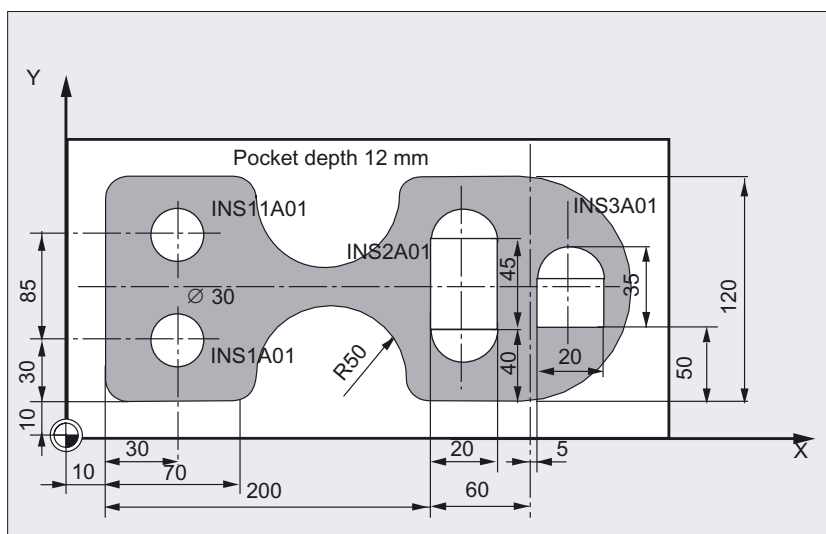
## Example 2

### Machining task:

Before the pocket is milled, the workpiece must be predrilled to ensure optimum insertion of the milling tool.

- Predrilling
- Solid machine pocket with islands, cutter radius 12 mm
- Solid machine residual material, cutter radius 6 mm
- Finish pocket, cutter radius 5 mm

### Machining outline:



### Machining program:

```
%_N_SAMPLE2_MPF
; $PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
; Example_2: Pocket with islands
; Predrilling, solid machining, solid machining of residual material, finishing
;
; Drill 20 mm, boring
$TC_DP1[2,1]=220 $TC_DP3[2,1]=111 $TC_DP6[2,1]=10 ;Drill
; Milling tool 24 mm, solid machining
$TC_DP1[3,1]=120 $TC_DP3[3,1]=150 $TC_DP6[3,1]=12 ;Milling tool, roughing
; Milling tool 10 mm, finishing
$TC_DP1[5,1]=120 $TC_DP3[5,1]=150 $TC_DP6[5,1]=5 ;Milling cutter for finishing
; Milling tool 12 mm, residual material
$TC_DP1[6,1]=120 $TC_DP3[6,1]=150 $TC_DP6[6,1]=6 ;Milling tool, residual material
; machining
;
TRANS X10 Y10
G0 G17 G500 G40 G90
```

## 3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75

```

;
;Define contours
GOTOF ENDLABEL
ACCEPTANCE4_CONT:
CYCLE74("EDGEA01", , )
CYCLE75("ISL11A01",,,)
CYCLE75("ISL1A01",,,)
CYCLE75("ISL2A01",,,)
CYCLE75("ISL3A01",,,)
ENDLABEL:
;
; Programming of predrilling
T2 M6
D1 M3 F2222 S3000
MCALL CYCLE81 (10,0,1,-12,)
REPEAT ACCEPTANCE4_MACH ACCEPTANCE4_MACH_END
MCALL
;
; Programming of solid machining
T3 M6
D1 M3 F3000 S4000
GOTOF ACCEPTANCE4_MACH_END
ACCEPTANCE4_MACH:
REPEAT ACCEPTANCE4_CONT ENDLABEL
CYCLE73(1015,"ACCEPTANCE4_DRILL","ACCEPTANCE4_MILL1","3",10,0,1,-
12,0,0,2,0.5,0,2000,400,0,,,,)
ACCEPTANCE4_MACH_END:
REPEAT ACCEPTANCE4_CONT ENDLABEL
CYCLE73(1011,"ACCEPTANCE4_DRILL","ACCEPTANCE4_MILL1","3",10,0,1,-
12,0,0,2,0.5,0,2000,400,0,,,,)
;
; Programming of solid machining of residual
material
T6 M6
D1 M3 S4000
REPEAT ACCEPTANCE4_CONT ENDLABEL
CYCLE73(1012,"","ACCEPTANCE4_3_MILL2","3",10,0,1,-12,0,0,2,0.5,0,1500,800,0,,,,)
;
; Programming of edge finishing
T5 M6
D1 M3 S500
G0 Z120
REPEAT ACCEPTANCE4_CONT ENDLABEL
CYCLE73(1013,"ACCEPTANCE4_DRILL","ACCEPTANCE4_MILL3","6",10,0,1,-
12,0,0,2,0,0,3000,700,0,,,,)
M30

```



## Edge contour programming example 2:

```

%_N_EDGEA01_MPF
; $PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
; Ste 17.05.99
; Edge contour programming example 2

N5 G0 G90 X260 Y0
N7 G3 X260 Y120 CR=60
N8 G1 X170 RND=15
N9 G2 X70 Y120 CR=50
N10 G1 X0 RND=15
N11 Y0 RND=15
N35 X70 RND=15
N40 G2 X170 Y0 CR=50
N45 G1 X260 Y0
N50 M30

```

## Island contours programming example 2:

```

%_N_ISL1A01_MPF
; $PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
; Ste 18.06.99
; Island contour sample program 2

N5 G90 G0 X30 Y15
N10 G91 G3 X0 Y30 CR=15
N12 X0 Y-30 CR=15
N15 M30

%_N_ISL11A01_MPF
; $PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
; Ste 18.06.99
; Island contour sample program 2

N5 G90 G0 X30 Y70
N10 G91 G3 X0 Y30 CR=15
N12 X0 Y-30 CR=15
N15 M30

%_N_ISL2A01_MPF
; $PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
; Ste 18.06.99
; Island contour sample program 2

N5 G90 G0 X200 Y40

```

```

N10 G3 X220 Y40 CR=10
N15 G1 Y85
N20 G3 X200 Y85 CR=10
N25 G1 Y40
N30 M30

%_N_ISL3A01_MPF
; $PATH=_N_WKS_DIR/_N_CC73BEI2_WPD
; Ste 18.06.99
; Island contour sample program 2

N5 G0 G90 X265 Y50
N10 G1 G91 X20
N15 Y25
N20 G3 X-20 I-10
N25 G1 Y-25
N30 M30

```

**Machining result:**

→ See explanations of technology using example 2

**Example 3****Machining task:**

Show the program sequence of a machining task, illustrated by two different pockets with islands. The machining process is tool-oriented, i.e., each time a new tool becomes available, all machining tasks requiring this particular tool are completely performed on both pockets before the next tool is used.

- Predrilling
- Solid machine pocket with islands
- Solid machine residual material

```

%_N_SAMPLE3_MPF
; $PATH=_N_WKS_DIR/_N_CC73BEI3_WPD
; Sample3
; 07.04.2000

; Tool offset data
$TC_DP1[2,1]=220 $TC_DP3[2,1]=330 $TC_DP6[2,1]=10
$TC_DP1[3,1]=120 $TC_DP3[3,1]=210 $TC_DP6[3,1]=12
$TC_DP1[6,1]=120 $TC_DP3[6,1]=199 $TC_DP6[6,1]=6

; Workpiece zero
; G54

```

## 3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75

```

$P_UIFR[1,X,TR]=620
$P_UIFR[1,Y,TR]=50
$P_UIFR[1,Z,TR]=-320
; G55
$P_UIFR[2,X,TR]=550
$P_UIFR[2,Y,TR]=200
$P_UIFR[2,Z,TR]=-320
;
N10 G0 G17 G54 G40 G90
N20 T2
M6
D1 M3 F2000 S500
N30 G0 Z20
;Machining contours pocket 1
GOTOF ENDLABEL
POCKET1_CONT:
CYCLE74("EDGE"," "," ")
CYCLE75("ISLAND1"," "," ")
CYCLE75("ISLAND2"," "," ")
ENDLABEL:
;Machining contours pocket 2
GOTOF ENDLABEL
SAMPLE2_CONT:
CYCLE74("EDGEA01", , )
CYCLE75("ISL11A01", , )
CYCLE75("ISL1A01", , )
CYCLE75("ISL2A01", , )
CYCLE75("ISL3A01", , )
ENDLABEL:
;Drill
T2 M6
D1 M3 F6000 S4000
MCALL CYCLE81 (10,0,1,-8,)
REPEAT POCKET1_MACH POCKET1_MACH_END
MCALL

G55
MCALL CYCLE81 (10,0,1,-8,)
REPEAT SAMPLE2_MACH SAMPLE2_MACH_END
MCALL
;Solid machining pocket 1
T3 M6
G54 D1 M3 S3300
GOTOF POCKET1_MACH_END
POCKET1_MACH:

```

## 3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75

```

REPEAT POCKET1_CONT ENDLABEL
CYCLE73(1015,"POCKET1_DRILL","POCKET1_MILL1","3",10,0,1,
-8,0,0,2,0,0,2000,400,0,0,0,1,4)
POCKET1_MACH_END:
REPEAT POCKET1_CONT ENDLABEL
CYCLE73(1011,"POCKET1_DRILL","POCKET1_MILL1","3",10,0,1,
-8,0,0,2,0,0,2000,400,0,0,0,1,4)

;Solid machining pocket 2
G55
GOTOF SAMPLE2_MACH_END

SAMPLE2_MACH:
REPEAT SAMPLE2_CONT ENDLABEL
CYCLE73(1015,"SAMPLE2_DRILL","SAMPLE2_MILL1","3",10,0,1,
-8,0,0,2,0,0,2000,400,0,0,0,1,4)
SAMPLE2_MACH_END:
REPEAT SAMPLE2_CONT ENDLABEL
CYCLE73(1011,"SAMPLE2_DRILL","SAMPLE2_MILL1","3",10,0,1,
-8,0,0,2,0,0,2000,400,0,0,0,1,4)

;Solid machining residual material pocket 1 and pocket 2
T6 M6
D1 G54 M3 S222
REPEAT POCKET1_CONT ENDLABEL
CYCLE73(1012,"","POCKET1_3_MILL2","3",10,0,1,-8,0,,2,,,2500,800,0,,,,)

G55
REPEAT SAMPLE2_CONT ENDLABEL
CYCLE73(1012,"","SAMPLE2_3_MILL2","3",10,0,1,-8,0,,2,,,2500,800,0,,,,)
G0 Z100
M30

;Edge and island contours
;Pocket 2 corresponds to programming example 2
Pocket 1:
%_N_Edge_MPF
;$PATH=/_N_WKS_DIR/_N_CC73BEI3_WPD
;29.03.99
N1 G0 X0 Y0 G90
N3 G1 X200 Y0
N5 X200 Y100
N10 X0 Y100
N20 X0 Y0
M30

```

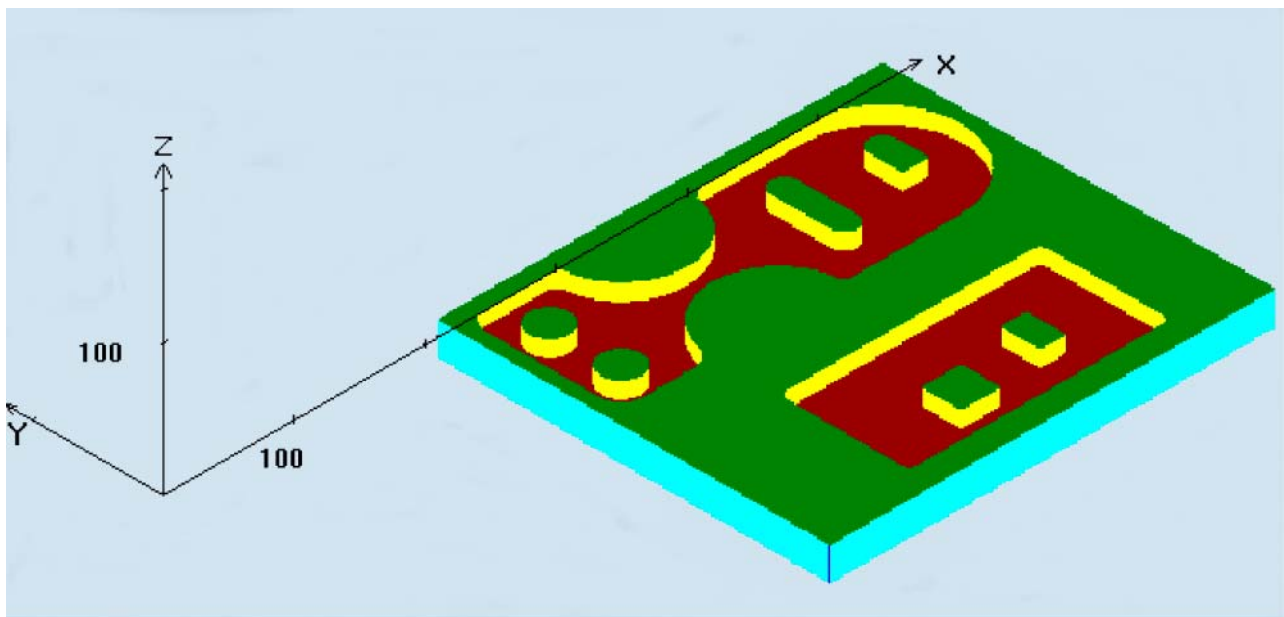
```

%_N_ISLAND1_MPF
; $PATH=/_N_WKS_DIR/_N_CC73BEI3_WPD
; 29.03.99
N100 G0 X130 Y30 Z50 G90
N110 G1 X150 Y30
N120 X150 Y60
N130 X130 Y60
N200 X130 Y30
M30

%_N_ISLAND2_MPF
; $PATH=/_N_WKS_DIR/_N_CC73BEI3_WPD
; 29.03.99
N12 G0 X60 Y20
N13 G1 X90 Y20
N14 X90 Y50
N30 X60 Y50
N40 X60 Y20
M30

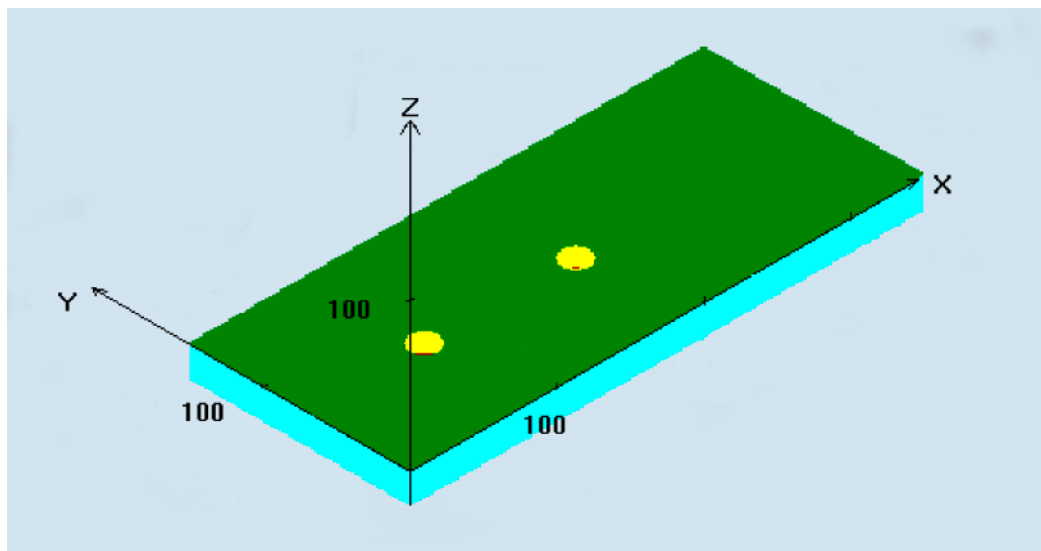
```

**Machining result:**



### 3.15.4.3 Description of technology in Example 2

#### Predrilling sequence



In the first machining section of the predrilling operation, a REPEAT command must be used after a modal call of the drilling cycle to call a sequence of machining steps with the contents of CYCLE73 and the contour repetition. The drilling cycle must be deselected modally before the next tool change. Other drilling technologies can be programmed subsequently.

The next program section contains CYCLE73, which contains all necessary parameters and programs for solid machining and drilling.

Parameter \_VARI is the only one to define all solid machining parameters and, therefore, must always be programmed.

The cycle now generates the solid machining and drilling position programs for the pocket. It then calls the drilling position program and executes it.

If the operation involves several different pockets, it will be necessary to call the associated contours again in this section. This block can be omitted if there is only one pocket.

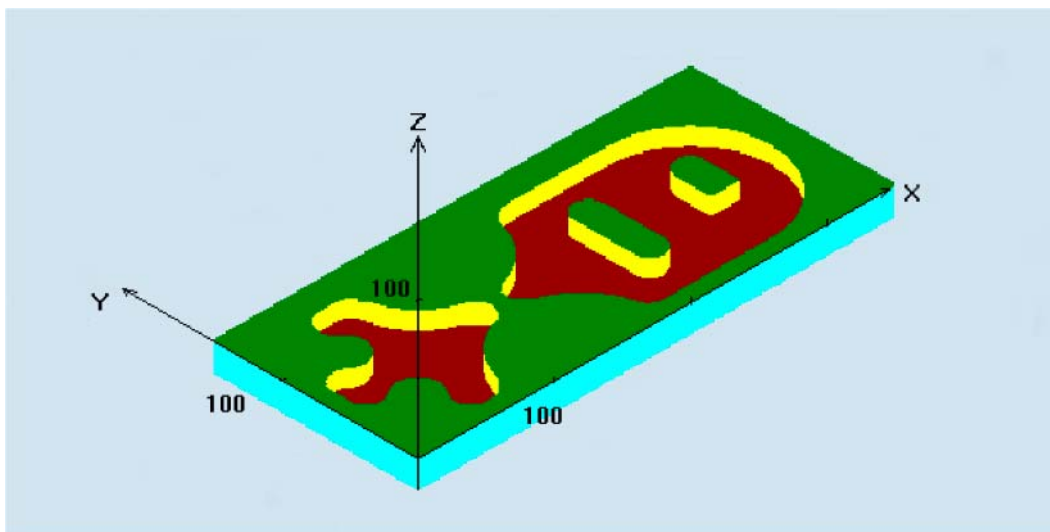
This entire machining section must be marked by a skip command to the following "Solid machine pocket" section.

---

#### Note

The programming can be reviewed in Examples 2 and 3.

---

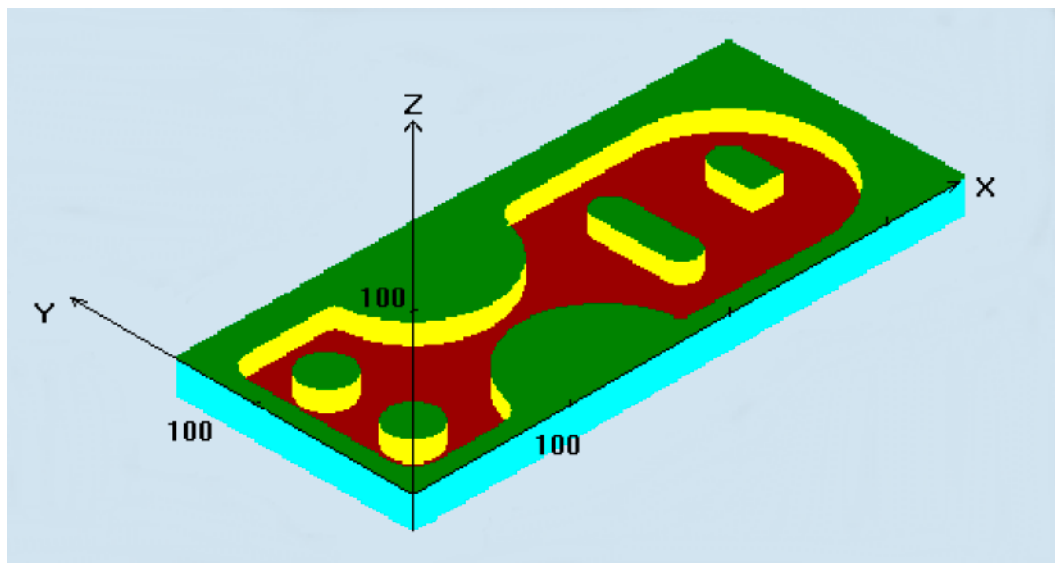
**Sequence for roughing, solid machining (\_VARI=XXX1)**

All parameters must be written to the CYCLE73 command again.

The program performs the following machining steps:

- Approach a manually calculated or automatically generated start point located on the return plane level. G0 is then used to traverse the axis to a reference plane shifted by the amount of the safety clearance.
- Infeed to the current machining depth according to the selected insertion strategy (\_VARI) with feed value \_FFD.
- Mill pocket with islands down to final machining allowance with feedrate \_FFP1. The machining direction corresponds to the setting in \_CDIR.
- Lift off in accordance with selected retraction mode and return to start point for next plane infeed.
- When the pocket has been machined, the tool is retracted either to the retraction plane or by the safety clearance via the reference plane, depending on the selected lift-off mode. The tool position in the plane is above the pocket surface, as determined by the generated program.

## Sequence for finishing (\_VARI = XXX3)



- The pocket and island contours are each circumnavigated once during the edge finishing operation. Vertical insertion with G1 (\_VARI) must be programmed as the insertion strategy. Approach and retraction at the start and end points of the finishing operation respectively are executed along a tangential circle segment.
- To finish the base, the tool is inserted to pocket depth + final machining allowance + safety clearance with G0. From this position, the tool is fed in vertically at the feedrate for depth infeed. The base surface of the pocket is machined once.
- Lift-off and retraction as for solid machining.
- Parameters \_FAL, \_FALD and \_VARI=XXX4 must be assigned for simultaneous finishing in the plane and on the base.

## Explanation of the parameters

**\_VARI (machining type)**

You can define the type of machining with parameter \_VARI. Possible settings are:

- **Units digit:** (selection of machining)
  - 1 = Roughing (solid machining) from solid material
  - 2 = Roughing residual material
  - 3 = Finishing edge
  - 4 = Finishing base
  - 5 = Predrilling

When "Roughing from solid material" is specified, the machining program solid-machines the pocket completely down to the final machining allowance.



If it is not possible to machine areas of the edge surfaces with the selected cutter diameter, then setting "2" can be selected in order to machine them with a smaller milling tool afterwards. To do this, cycle CYCLE73 must be called again.

- **Tens digit:** (selection of insertion strategy)

- 1 = Vertically with G1
- 2 = Along a helical path
- 3 = Oscillating

Insert vertically (\_VARI = XX1X)

means that the current infeed depth calculated internally is executed in one block.

Inset along helical path (\_VARI = XX2X)

means that the cutter center traverses along the helical path determined by the radius \_RAD and the depth per revolution \_DP1. The feedrate is also programmed under \_FFD. The direction of rotation of this helical path corresponds to the direction of rotation with which the pocket will be machined.

The insertion depth programmed under \_DP1 is taken into account as the maximum depth and is always calculated as an integer number of revolutions of the helical path.

If the current depth required for an infeed (this can be several revolutions on the helical path) is reached, a full circle is still executed to eliminate the inclined path of insertion.

Pocket solid machining then starts in this plane and continues until it reaches the final machining allowance.

Insert with oscillation (\_VARI = XX3X)

means that the milling center point on a straight line oscillating back and forth is inserted at an angle until it has reached the nearest current depth. The maximum insertion angle is programmed under \_RAD, the position of the oscillation path is calculated within the cycle. If the current depth is reached, the travel is executed once more without depth infeed in order to eliminate the inclined insertion path. The feedrate is programmed under \_FFD.

- **Hundreds digit:** (selection of lift-off mode)

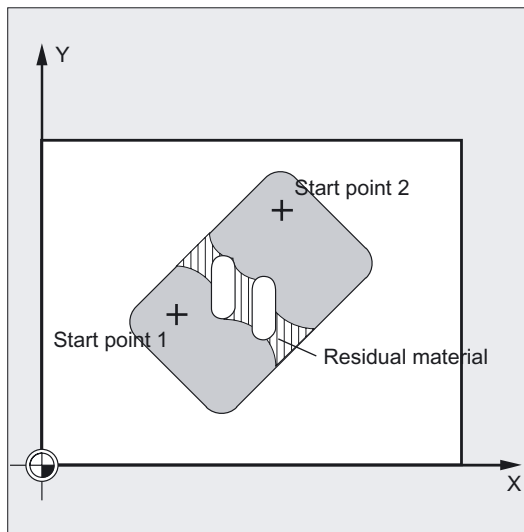
- 0 = To retraction plane (\_RTP)
- 1 = By amount of safety clearance (\_SDIS) over reference plane (\_RFP)

- **Thousands digit:** (selection of starting point)

- 1 = Automatic starting point
- 2 = Manual starting point

When the starting point is selected automatically, the cycle internally calculates the starting point for machining itself.

If the pocket has to be split as a result of the island position and the cutter diameter used, several start points are calculated automatically.

**NOTICE**

Manually specified start positions must not be too close to the island surface. Manually specified start positions are not internally monitored.

With manual start point selection, parameters `_PA` and `_PO` must also be programmed. However, these can only define one start point.

If the pocket has to be split, the required start points are calculated automatically.

**`_BNAME` (name for drilling position program)**

**`_PNAME` (name for pocket milling program)**

The pocket milling cycle generates programs with traversing blocks required to predrill or mill the workpiece. These programs are stored in the same directory as the calling program in the parts program memory, i.e., in the "Parts programs" directory (MPF.DIR), if the cycle is called from there, or in the relevant workpiece directory. The programs are always main programs (type MPF).

The names of these programs are defined by parameters `_BNAME` and `_PNAME`.

A drilling program name is only required when `_VARI=XXX5`.

**Example:**

- No drilling program name:

```
CYCLE73(1011,"",ACCEPTANCE4_MILL,...)
```

**`_TN` (name of solid machining tool), `_DN` (D-number of solid machining tool)**

This parameter must be assigned to the solid machining tool. Depending on whether the tool management function is active or not, the parameter must be set to a tool name or tool number.

**Example:**

- With tool management - MILLINGCUTTER3 D8  
CYCLE73(1015,"PART1\_DRILL","PART1\_MILL","MILLINGCUTTER3",...,8)
- Without tool management - T3 D8  
CYCLE73(1015,"PART1\_DRILL","PART1\_MILL","3",...,8)

The \_TN parameter has a maximum length of 16 characters. If a D-number is not programmed, D1 is automatically used. For a flat D-number, T does not have to be programmed.

**Example:**

- Flat D number - D39  
CYCLE73(1015,"PART1\_DRILL","PART1\_MILL","",...,39)

When the residual material machining operation is used more than once, the tool from the last residual material removal process must be used.

**\_RFP and \_RTP (reference plane and retraction plane)**

Normally, reference plane (RFP) and return plane (RTP) have different values. The cycle assumes that the retraction plane precedes 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 shifted by the amount of the safety clearance.

The direction, in which the safety clearance is active, is automatically determined by the cycle.

**\_DP (absolute pocket depth) and \_DPR (incremental pocket depth)**

The pocket depth can be specified as either an absolute value (\_DP) or an incremental value (\_DPR) in relation to the reference plane. If the incremental option is selected, the cycle automatically calculates the depth on the basis of the reference and retraction plane positions.

**\_MID (maximum infeed depth)**

Use this parameter to define the maximum infeed depth. The depth infeed is performed by the cycle in equally sized infeed steps.

The cycle calculates this infeed automatically on the basis of \_MID and the total depth.

The minimum possible number of infeed steps is used as the basis. \_MID=0 means that infeed is carried out in one cut up to the pocket depth.

**\_MIDA (max. infeed width in the plane)**

Use this parameter to define the maximum infeed width when solid machining in a plane. This value is never exceeded.

If this parameter is not programmed, or if its value is 0, the cycle uses 80% of the cutter radius as the maximum infeed width.

If an infeed width of more than 80% of the cutter diameter is programmed, the cycle is aborted after output of alarm 61982 "Infeed width in plane too large".

**\_FAL (final machining allowance in the plane)**

The finishing allowance only affects the machining of the pocket in the plane on the edge.

If the final machining allowance  $\geq$  tool diameter, the pocket will not necessarily be machined completely.

#### **\_FALD (finishing allowance at the base)**

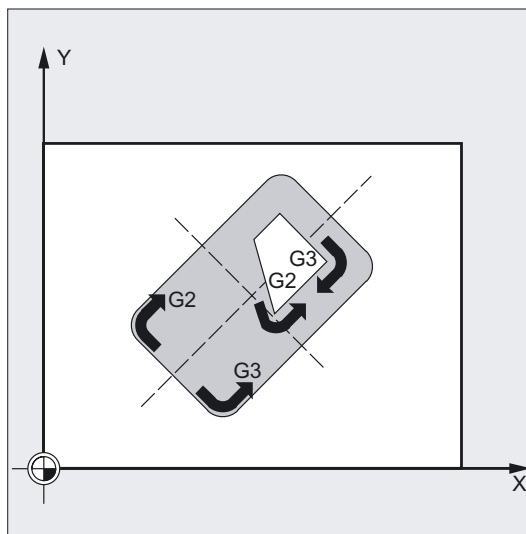
A separate final machining allowance on the base is taken into account in roughing operations.

#### **\_FFD and \_FFP1 (feedrate for depth infeed and surface machining)**

The feedrate `_FFD` is effective when inserting into the material.

The feedrate `_FFP1` is active for all movements in the plane which are traversed at feedrate during machining.

#### **\_CDIR (milling direction)**



Use this parameter to specify the machining direction for the pocket.

The direction of milling can be programmed as follows using the `_CDIR` parameter:

- can be programmed directly with "2 for G2" and "3 for G3" or,
- alternatively, "down-cut" or "up-cut".

Down-cut and up-cut are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

Down-cut	Up-cut
M3 → G3	M3 → G2
M4 → G2	M4 → G3

**\_PA, \_PO (starting point of first and second axes)**

When the start point is selected manually, the start point must be programmed in these parameters such that it can be approached without risk of collision. Please note that only one start point can be programmed (see description of parameter \_VARI).

**\_RAD (center-point path or insertion angle)**

Parameter \_RAD defines the radius of the helical path (with reference to tool center-point path) or the maximum insertion angle for oscillation.

**\_DP1 (insertion depth for helical path)**

Use the parameter \_DP1 to define the infeed depth when inserting to the helical path.

**Name of pocket machining (NAME)**

Pockets are generally machined in several technological machining steps. However, the contours defining the pocket geometries are defined only once.

To ensure that contours can be automatically assigned to the appropriate machining step in the program, the contour definition is marked with labels and this program section is later repeated with the REPEAT instruction.

When programs are written using the cycles support function, a name for the pocket machining program is entered in the relevant screen forms. The name length is restricted to 8 characters.

In programming example 2, this is, for example, "ACCEPTANCE4".

The T number contains the solid-machining milling tool for all machining technologies. When residual material is machined more than once, the tool used previously must always be entered in the T number.

**Explanation of the cycle structure**

Cycle CYCLE73 is used to solve very complex problems associated with solid machining of pockets with islands, which require a high level of computing capacity in the control. For best timing, the calculation is carried out in the HMI.

The calculation is started from the cycle. Its result contains programs with traversing blocks for drilling or milling operations, which are stored in the file system of the control. These are then called by the cycle and executed.

This structure means that it is only necessary to perform the calculation the first time a program is executed with the CYCLE73 call. From the second program run onwards, the generated traversing program is available for immediate call by the cycle.

Recalculation is performed when:

- A finished contour has been modified;
- A transfer parameter of the cycle has changed;
- A tool with different tool offset data has been activated prior to the cycle call;
- Different technologies are involved, such as solid machining and residual material, whose machining programs are generated in different ways.

### Program storage in the file system

If the contours for CYCLE73 are programmed outside the main program that makes the call, the following applies for the search in the file system of the control:

- If the calling program is stored in a workpiece directory, then the programs containing the edge or island contours must be stored in the same workpiece directory.
- If the invoked program is located in the "Parts program" directory (MPF.DIR), the programs are also searched for there.

The programs generated by the cycle are also stored in the same directory as the program containing the cycle call, i.e., in the same workpiece directory or in MPF.DIR or SPF.DIR.

---

#### Note

##### Simulation

In the pocket milling simulation, the generated programs are saved to the NCU file system. Therefore, only the "NC Active Data" setting is practical, since tool offset data are included in the program calculation.

---

#### Note

##### For HMI-Advanced OEM users

The "Pocket milling with islands" package occupies Task 27 in the HMI for communication between the cycle and the HMI.

For this purpose, a file is created in the VADD\_On directory when the package is installed:

- REGIE.INI

As a result, this task is not available to other OEM applications.

---

## 3.16 Swiveling – CYCLE800

### 3.16.1 General information

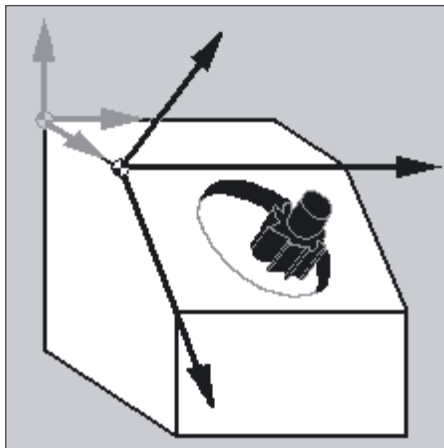
#### Function

The cycle is used to swivel on any type of surface so that it can be machined and/or measured. By calling the appropriate NC functions, the cycle converts the active workpiece zeroes and tool offsets to refer to the inclined surface, taking account of the kinematic chain on the machine, and positions the rotary axes (optional).

Swiveling can be implemented either axially, as a projection angle or as a solid angle.

Before the rotary axes are positioned, the linear axes can be retracted if desired.

Swiveling also includes the "Setting tool" function (see Section 3.16.4).



Swiveling is not an option and is available for NC SW 6.3 (CCU SW 6.3).

The functions below are available in the basic version

- 3/2 axes inclined machining and
- Toolholder with orientation capability

#### References:

Description of Functions 840D/840Di/810D:

- /W1/ "Tool Offset"
- /R2/ "Rotary Axes"
- /K2/ "System Frames"

### Swiveling supports the following machine kinematics

1. Swivel-mounted toolholder (swivel head) → Type T
2. Swivel-mounted workpiece holder (swivel table) → Type P
3. Mixed kinematics from 1 and 2 → Type M

### Requirements before calling the swivel cycle

A tool (tool cutting edge  $D > 0$ ) and the zero offset (ZO), with which the workpiece was scratched or measured, must be programmed before the swivel cycle is first called in the main program. This zero offset is converted to the appropriate machining plane in the swivel cycle. The ZO value remains unchanged. Translational and rotational components are saved in the system frames (swivel frames) below (see HMI → Parameters, Active ZO):

- Tool reference (TOOLFRAME)
- Toolholder (PARTFRAME)
- Workpiece reference (WPFRAME)

The current machining plane (G17, G18, G19) is taken into account by the swivel cycle.

#### Swiveling on a machining or auxiliary surface always involves three steps:

- Offset of reference point prior to rotation (corresponds to TRANS or ATRANS)
- Rotation (corresponds to AROT or AROTS)
- Offset of zero point after rotation (corresponds to ATRANS)

**The offsets or rotations are not machine-specific, but refer to the X, Y, Z coordinate system of the workpiece.**

No programmable frames are used in the swivel cycle. The frames programmed by the user are taken into account in additive swiveling.

Swiveling to a new swivel plane clears the programmable frames (TRANS).

After a program reset or if there is a power failure, the last swivel plane remains active, with the option to set it using machine data. Any type of machining operation can be performed on the swivel plane, e.g., through calling standard or measuring cycles.



## 3.16.2 Programming via screen form

### 3.16.2.1 General information

#### Calling swiveling – CYCLE800

Entry to programs/milling area

Soft key



is displayed if swivel data record is set up (MD 18088: MM\_NUM\_TOOL\_CARRIER > 0).

#### Input screen form CYCLE800 on standard interface

Reference point for rotation around Y axis	
Name:	
Retract:	Z
Schwenken:	Yes
Swivel plane:	new
Ref. point:	
X0	0.000
Y0	25.000
Z0	0.000
Swivel mode:	axis by ax.
Rot. around	X (A) -15.000
Rot. around	Y (B) 0.000
Rot. around	Z (C) 0.000
Zero point:	
X1	0.000
Y1	0.000
Z1	0.000
Direction:	Minus
Tracking TL	No

### 3.16.2.2 Parameters of input screen form

#### Name of swivel data record \_TC

The existing swivel data records (see CYCLE800 startup) can be selected (toggled).

Each swivel data record has a name. If there is only one swivel data record, a name does not have to be declared.

"0" → Swivel data record deselection.

### Retraction \_FR (prior to swiveling rotary axes)

- Do not retract
- Retract axis Z<sup>1)</sup>
- Retract axis Z, XY<sup>1)</sup>
- Maximum retraction in tool direction (with Cycles SW 6.5 and higher)<sup>1) 2)</sup>
- Incremental retraction in tool direction (with Cycles SW 6.5 and higher)<sup>1) 2)</sup>

The incremental value for the traversing path in the tool direction must be entered in the edit box.

The retraction variants and positions can be activated and set up in the CYCLE800 startup menu.

**Note the machine manufacturer's instructions.**

---

#### Note

1) The retraction positions are traversed absolutely. If another sequence or incremental positioning is desired, this can be modified during installation and set up in the TOOLCARR user cycle.

2) See Note 2) when programming using parameters, Section 3.1.6.3.

---

When programming with standard cycles, high settings for the retraction plane and large swivel angles (through 90 degrees with multiface machining), it is possible that the traversing area of the machine may be too small (software end position error), since the order of approach is always the machining plane (for G17 X, Y) first, followed by the infeed axis (Z).

The retraction plane can be reduced to optimize this behavior.

### Swivel, direction \_DIR

- **Swivel yes**

Rotary axes are positioned or the operator can turn the manual rotary axes.

- **Swivel no (calculation only)**

If rotary axes are not to be traversed after activating the swivel cycle, the "Swivel no" selection applies.

Application: Auxiliary swivel levels according to workpiece drawing

- **Direction minus/plus**

Direction reference of traversing direction of rotary axis 1 or 2 of the active swivel data record (machine kinematics). The NC calculates two possible solutions of the rotation / offset programmed in CYCLE800 using the angle traversing range of the rotary axes of the machine kinematics. Usually, only one of these solutions is technologically suitable. The solutions differ by 180 degrees in each case. Selecting the "minus" or "plus" direction determines which of the two possible solutions is to be applied.

– "Minus" → Lower rotary axis value

- "Plus" → Higher rotary axis value

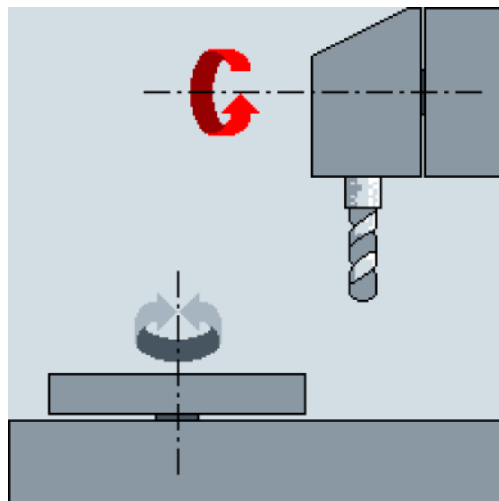
Which of the rotary axes (1 or 2) the two possible solutions are to refer to is specified in the CYCLE800 startup menu.

**Note the machine manufacturer's instructions.**

Example:

- Machine kinematics with swivel head and swivel table.  
Swivel head with rotary axis 1 (B) rotates around machine axis Y.
- Angle traversing range of rotary axis B from -90 to +90 degrees.
- Swivel table with rotary axis 2 (C) rotates around machine axis Z.
- Angle traversing range of rotary axis 2 (C) from 0 to 360 degrees (modulo 360).
- Machine manufacturer has set the direction reference to rotary axis 1 (B) in the Swivel start-up.
- A rotation around X (WCS) of 10 degrees is programmed in the swivel cycle.

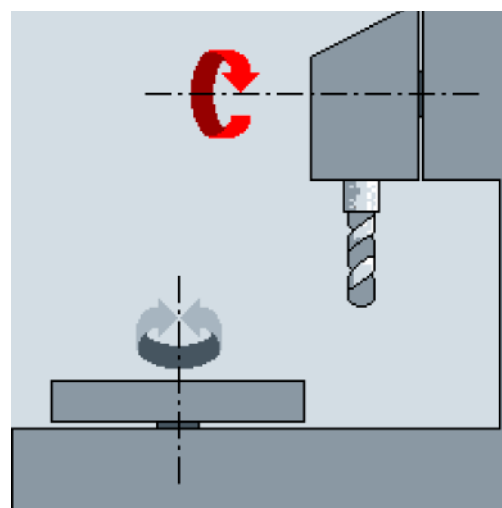
The figures below show the machine in the basic kinematics setting (B=0 C=0).



"Minus" direction

Rotary axis B moves to -10 degrees in the negative direction (red arrow).

Rotary axis C moves to 90 degrees (rotation around X !)



"Plus" direction

Rotary axis B moves to +10 degrees in the positive direction (red arrow).

Rotary axis C moves to 270 degrees

The "Minus" or "Plus" direction settings enable a workpiece to be machined with swiveled planes. The two solutions calculated by the NC differ by 180 degrees (see rotary axis C).

**Note**

In the basic machine kinematics setting, the input screen form for CYCLE800 contains two different direction settings "Plus" and "Minus" if the selection option "Direction rotary axis 1 or 2 optimized" is set in the start-up menu for CYCLE800.

**Note the machine manufacturer's instructions.**

**Swivel plane \_ST**

- **New**

Previous swivel frames and programmed frames are deleted and a new swivel frame is formed according to the values specified in the input screen.

Every main program must begin with a swivel cycle with the new swivel plane, in order to ensure that a swivel frame from another program is not active.

- **Additive**

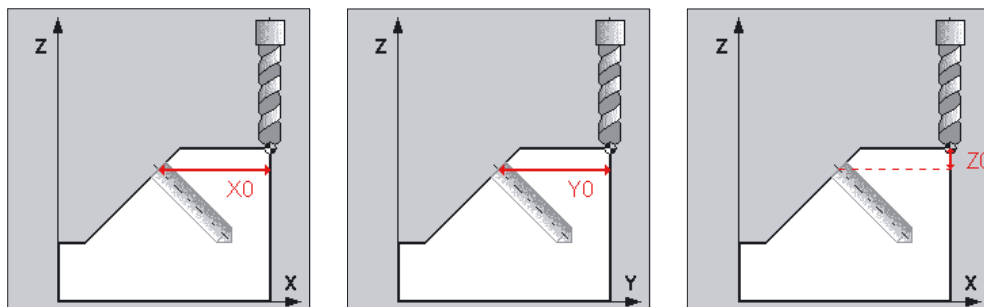
The swivel frame is added to the swivel frame from the last swivel cycle.

If several swivel cycles are programmed in a program and programmable frames are also active between them (e.g., AROT ATRANS), these are taken into account in the swivel frame.

If the currently active ZO contains rotations, e.g., due to previous workpiece gauging, they will be taken into account in the swivel cycle.

**Help screens**

The following help displays relate to machining plane G17 (tool axis Z).

**Reference points before rotation X0, Y0, Z0**

## Swivel mode \_MODE

This parameter defines the swivel mode for the axis.

- Axis by axis
- Projection angle<sup>1)2)</sup>
- Solid angle<sup>1)</sup>
- Rotary axes direct

Swivel mode always refers to the coordinate system of the workpiece and does not, therefore, depend on the machine.

---

### Note

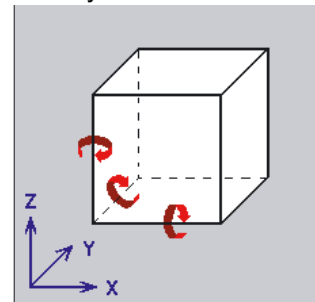
#### For 1: Swiveling as projection or solid angle

Only available if selected by the machine manufacturer in the CYCLE800 startup menu.

---

- With the axial swiveling option, the tool is rotated about the individual axes of the coordinate system in succession, with each rotation starting from the previous rotation. The axis sequence can be freely selected.

Axis by axis

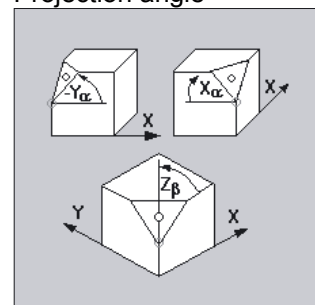


- When swiveling using the projection angle, the angle value of the swiveled surface is projected onto the first two axes of the right-angle coordinate system. The user can select the axis rotation sequence.<sup>2)</sup>

The third rotation starts from the previous rotation. The active plane and the tool orientation must be taken into consideration when the projection angle is applied:

- For G17 projection angle XY, 3rd rotation around Z
- For G18 projection angle ZX, 3rd rotation around Y
- For G19 projection angle YZ, 3rd rotation around X

Projection angle

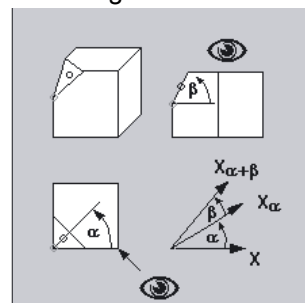


**Note****For 2): Swiveling as projection angle**

- When projection angles around XY and YX are programmed, the new X-axis of the swiveled coordinate system lies in the old ZX plane.
- When projection angles around XZ and ZX are programmed, the new Z-axis of the swiveled coordinate system lies in the old Y-Z plane.
- When projection angles around YZ and ZY are programmed, the new Y-axis of the swiveled coordinate system lies in the old X-Y plane.

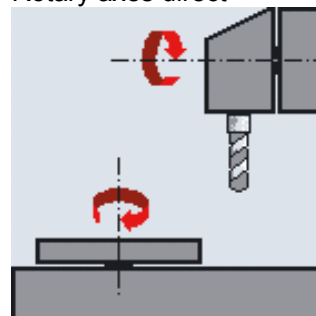
- With the solid angle swiveling option, the tool is first rotated about the Z axis and then about the Y axis. The second rotation starts from the first. The positive direction of each rotation for the different swivel options is shown in the help displays.

Solid angle



- With "Rotary axes direct" swivel mode, the WCS positions of the rotary axes can be specified directly. The target swivel positions are entered in the input fields of the rotary axes of the selected swivel data record (see parameter \_TC).  
A swivel frame is computed for these positions in CYCLE 800. This ensures that the tool orientation is aligned at a perpendicular angle to the swivel plane. It is possible to enter an additional rotation around the tool orientation in input field "Rotation tool". The reference axis is the abscissa (with G17 X). This means that the traversing directions in the plane (with G17 XY) are also precisely defined after swiveling with "Rotary axes direct".

Rotary axes direct



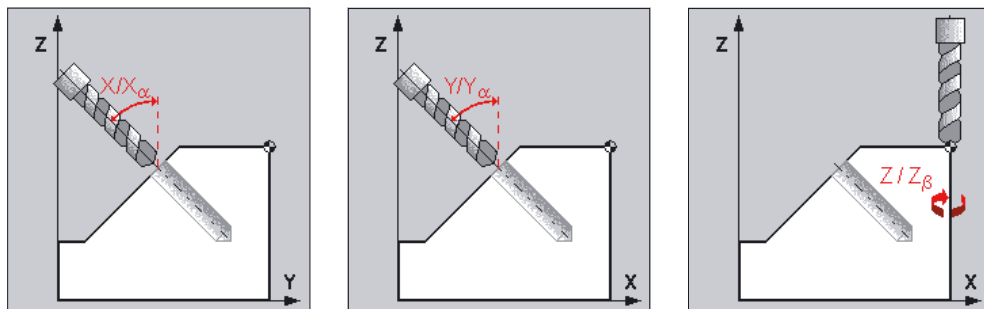
If swivel mode "Rotary axes direct" is applied, the corresponding NC program is machine-dependent, i.e. the NC program can run only on machines with the same swivel kinematics (including the rotary axes identifiers). Manual and semi-automatic rotary axes are also compatible with "Rotary axes direct" swivel mode.

"Rotary axes direct" swivel mode must also be enabled in the start-up menu for CYCLE800.

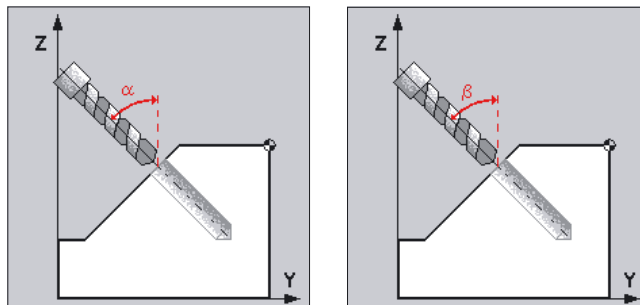
**Note the machine manufacturer's instructions.**

## Rotation about A, B, C

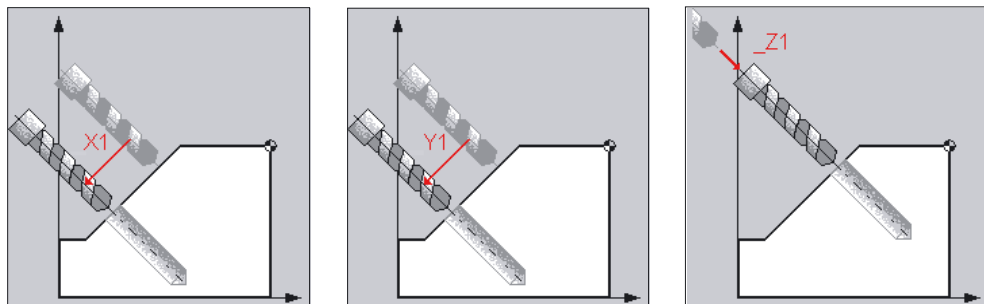
- Rotations (axial, projection angle)



- Rotation (solid angle)



## Zero point after rotation X1, Y1, Z1



**Correct tool**

The Correct display field can be deactivated using the CYCLE800 startup menu.

- **Yes:**

When swiveling onto a machining plane, the linear axes can be corrected to prevent the risk of collision.

Requirements:

1. TRAORI option is required.
2. The machine manufacturer has adapted the TOOLCARR.SPF user cycle appropriately.

- **No:**

Linear axes are not corrected during swiveling.

**3.16.2.3 Operator and programmer instructions**

- In an NC program with swiveled machining planes (CYCLE800), rotary axes are exclusively positioned using CYCLE800 (TOOLCARR.SPF). The only exception is when changing to a 5-axis transformation.

- **Note on calling the 5-axis transformation (TRAORI)**

If a program activating the 5-axis transformation (TRAORI) is to be executed on the swiveled machining plane, the system frames for the swivel head/swivel table must be deactivated before TRAORI is called (see example). The workpiece reference (WPFRAME) is retained.

Example (machine with swivel table)

```

N1 G54
N2 T="MILL_10mm"
N3 M6
N4 CYCLE800(1,"",0,57,0,40,0,-45,0,0,0,0,0,-1)      ;Swivel cycle
N5 CYCLE71(50,24,2,0,0,0,80,60,0,4,10,5,0,2000,31,5) ;Face milling
N6 TCARR=0                                          ;Swivel data record
                                                    ;deselection
N7 PAROTOF
N8 TOROTOF                                          ;(only with
                                                    ;machine kinematic types
                                                    ;"T" and "M")
N9 TRAORI
N10 G54                                             ;Recalculation of
                                                    ;zero offset
N11 EXTCALL "ROLL"                                ;5-axis machining program
                                                    ;with direction vectors
                                                    ;(A3, B3, C3)
N12 M2
.
.
.
```



If the 5-axis transformation with the CYCLE832 "high speed settings" cycle has been activated, blocks N6 to N10 are not required.

- If the rotary axes of machine kinematics are defined as manual axes (CYCLE800 startup menu), the swivel angle to be set is displayed in cancel alarm 62180/62181.

After rotation to the swivel angle, the NC program is continued with NC Start.

- It is possible to traverse the axes in the active swivel plane in JOG mode, if the WCS key is active on the machine control panel. This ensures that the geometry axes are traversed and not the machine axes.
- The swivel data record can be deselected and the swivel frame (WPFRAME, PARTFRAME, TOOLFRAME) can be deleted by programming CYCLE800().
- In CYCLE800, parameters can also be transferred as input values (e.g., result variable of measuring cycles \_OVR[21]).
- The value ranges defined in the Programming Guide Fundamentals apply to the transfer parameters of standard cycles. The value range for angle values is defined as follows (see ROT / AROT in the Programming Guide Fundamentals):

Rotation around 1st geometry axis: -180 degrees to +180 degrees

Rotation around 2nd geometry axis: -90 degrees to +90 degrees

Rotation around 3rd geometry axis: -180 degrees to +180 degrees

When angle values are passed to a standard cycle, please note that they must be rounded down to zero if they are lower than the calculation resolution of the NCU. The calculation resolution for angle positions on the NCU is specified in machine data 10210 \$MN\_INT\_INCR\_PER\_DEG.

Example of the parameter \_OVR[21] of measuring cycle CYCLE998 (measure angle):

```
_OVR[21]=-0.000345 ;calculation resolution MD $MN_INT_INCR_PER_DEG=1000
```

```
IF ((ABS(_OVR[21] * $MN_INT_INCR_PER_DEG)) < 1)
```

```
_OVR[21]=0
```

```
ENDIF
```

Explanation:

If the value of parameter \_OVR[21] is less than the programmed calculation resolution, it is rounded down to zero.

- If rotary axis values are entered in the active ZO (basic rotation of the workpiece), they will be taken into account when swiveling.

During swiveling to the basic machine setter setting of the machine kinematics (...0,0,0,...) [rotation=0], CYCLE800 arranges the WCS accordingly. This results in rotations in the \$P\_WPFRAME swivel frame (workpiece reference).

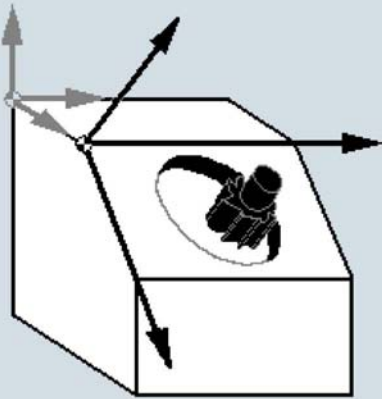

## 3.16.2.4 Examples of input screen form

## Example 1 Making the basic swivel plane setting

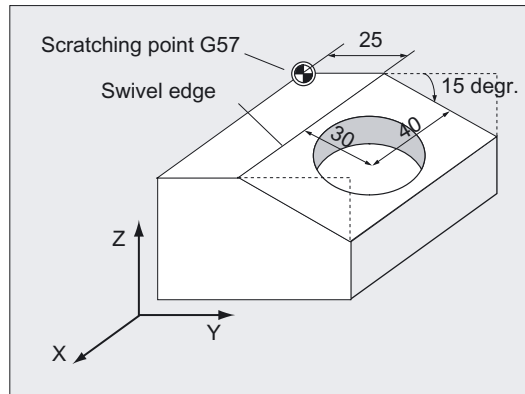
```

%_N_SWIVEL_0_SPF
;$PATH=/_N_WCS_DIR/_N_HAA_SWIVEL_WPD
G54
CYCLE800(1,"",0,57,0,0,0,0,0,0,0,0,-1)
M2

```

Swivel cycle/CYCLE800		Name of swivel data record	
	Name:	MIXED_BC	
	Retract:		Z
	Swivel:		Yes
	Swivel plane:		New
	Ref. point:	X0	0.000
		Y0	0.000
		Z0	0.000
	Swivel mode:		Axis by axis
	Rot. around	X (A)	0.000
	Rot. around	Y (B)	0.000
Rot. around	Z (C)	0.000	
Zero point:	X1	0.000	
	Y1	0.000	
	Z1	0.000	
Direction:		Minus	

### Example 2 Face milling and milling a circular pocket on a machining plane swiveled through 15 degrees



```

%_N_SWIVEL_CIRCULARPOCKET_SPF
; $PATH=/_N_WCS_DIR/_N_HAA_SWIVEL_WPD
N12 T="MILL_26mm"
N14 M6
N16 G57
N18 CYCLE800(1,"",0,57,0,0,0,0,0,0,0,0,0,1)
N20 M3 S5000
N22 CYCLE71(50,2,2,0,0,0,80,60,0,4,15,5,0,2000,31,5) ;Face milling
N24 CYCLE800(1,"",0,57,0,25,0,-15,0,0,0,0,0,-1)

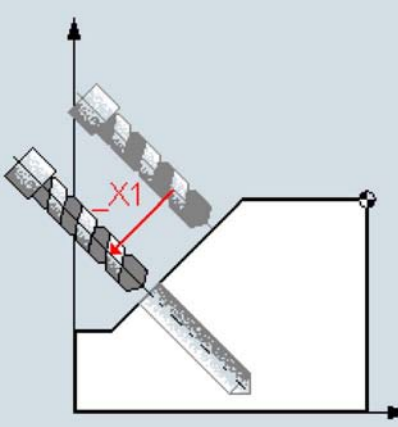

```

Swivel cycle/CYCLE800		Angle of 1st rotation	
		Name:	MIXED_BC
		Retract:	Z
		Swivel:	Yes
		Swivel plane:	New
		Ref. point:	X0 0.000
			Y0 25.000
			Z0 0.000
		Swivel mode:	Axis by axis
		Rot. around	X (A) -15.000
		Rot. around	Y (B) 0.000
Rot. around	Z (C) 0.000		
Zero point:	X1 0.000		
	Y1 0.000		
	Z1 0.000		
Direction:	Minus		

```

N26 CYCLE71(50,12,2,0,0,0,80,60,0,4,10,5,0,2000,31,5) ;Face milling
N28 CYCLE800(1,"",1,57,0,0,0,0,0,0,0,40,30,0,1)

```

Swivel cycle/CYCLE800		Zero point of X axis swivel plane	
		Name:	MIXED_BC
		Retract:	Z
		Swivel:	Yes
		Swivel plane:	Additive
		Ref. point:	X0 0.000
			Y0 25.000
			Z0 0.000
		Swivel mode:	Axis by axis
		Rot. around	X (A) 0.000
		Rot. around	Y (B) 0.000
Rot. around	Z (C) 0.000		
Zero point:	X1 40.000		
	Y1 30.000		
	Z1 0.000		
Direction:	Minus		

```

N30 T="MILL_10mm"
N32 M6
N34 M3 S5000
N36 POCKET4(50,0,1,-15,20,0,0,4,0.5,0.5,1000,1000,0,11,,,,) ;Circular pocket
N38 POCKET4(50,0,1,-15,20,0,0,4,0,0,1000,1000,0,12,,,,)
N40 M2

```

### 3.16.3 Programming using parameters

#### Programming

```
CYCLE800(_FR, _TC, _ST, _MODE, _X0, _Y0, _Z0, _A, _B, _C, _X1, _Y1,
_Z1, _DIR, _FR _I)
```

#### Parameter

Parameter	Data type	Meaning
_FR	integer	Retract
		Values: 0: No retraction 1: Retract Z axis (default) <sup>1)</sup> 2: Retract Z, X, Y axis <sup>1)</sup> 4: Maximum retraction in tool direction <sup>2)</sup> 5: Incremental retraction in tool direction <sup>2)</sup>
_TC	String[20]	Name of swivel data record "0" Deselection of data record
_ST	integer	Swivel plane
		Values: UNITS DIGIT: 0: New 1: Additive
		TENS DIGIT <sup>2)</sup> : 0x: Do not correct tool tip 1x: Correct tool tip Requirement: Correction must be set up by the machine manufacturer (see manufacturing cycle TOOLCARRIER.SPF)
		HUNDREDS DIGIT: 1xx: Setting tool 2xx: Align tool in turning area 3xx: Align tool in milling area
		THOUSANDS DIGIT Internal use
		TEN THOUSANDS DIGIT 1xxxx: Swivel no direction "minus" 2xxxx: Swivel no direction "Plus"
		HUNDRED THOUSANDS DIGIT 0xxxxx: Behavior in earlier version with selection of rotary axis direction "Minus" or "Plus" 1xxxxx: Direction selection "Minus" optimized <sup>1)</sup> 2xxxxx: Direction selection "Plus" optimized <sup>1)</sup> 1) The "Minus" or "Plus" optimized selection for the rotary axis direction functions in conjunction with parameter _DIR and must be enabled in the start-up menu for CYCLE800 by the machine manufacturer.

Parameter	Data type	Meaning
_MODE	integer	Swivel mode Evaluation of angle: Decimal parameter mode, binary coding.
		<p>7 6 5 4 3 2 1 0 Bit</p> <p>01: Rotation around 1st axis 10: Rotation around 2nd axis 11: Rotation around 3rd axis</p> <p>Angle of rotation 1</p> <p>01: Rotation around 1st axis 10: Rotation around 2nd axis 11: Rotation around 3rd axis</p> <p>Angle of rotation 2</p> <p>01: Rotation around 1st axis 10: Rotation around 2nd axis 11: Rotation around 3rd axis</p> <p>Angle of rotation 3</p> <p>00: Axis-by-axis swivel angle (A, B, C) 01: Solid angle (A, B) 10: Projection angle (A, B, C) 11: Swivel mode rotary axes direct</p>
		Note: Bits 0 to 5 (irrelevant for solid angle)
		Coding example: → Axis-by-axis rotation zyx Binary: 00011011 Decimal: 27
_X0, _Y0, _Z0	real	Reference point prior to rotation
_A	real	1. Axis angle (axial, projection angle swivel modes) 2. Angle of rotation about the Z axis in XY plane (solid angle swivel mode)
_B	real	1. Axis angle (axial, projection angle swivel modes) 2. Angle of rotation in space about the Y axis (swivel mode 'solid angle')
_C	real	Axis angle (axial, projection angle swivel modes)
_X1, _Y1, _Z1	real	Zero point after rotation
_DIR	integer	Direction If the NC calculates two solutions when the swivel cycle is called, the operator can select a preferred direction. The machine manufacturer specifies the axis, to which the preferred direction refers.
		Values: -1: (minus): Lower rotary axis value (default) +1: (plus): Higher rotary axis value 0: No traversing of rotary axes (calculation only), see also parameter _ST TEN THOUSANDS DIGIT
_FR_I	real	Value of incremental retraction in tool direction

**Note on 1):**

- If declared in swiveling startup.
- Can be adjusted in the TOOLCARR user cycle.

**Note the machine manufacturer's instructions.**

**Note on 2):**

**Retraction in tool direction**

With Cycles SW 6.5 and higher, existing retraction modes are expanded, as described below:

- **"Maximum retraction in tool direction"**

The tool axis is retracted to the software end position prior to swiveling.

- **"Incremental retraction in tool direction"**

The tool axis is retracted by the amount of the incremental input value (\_FR\_I).

Maximum and incremental retraction in tool direction always occurs in the positive tool direction (away from workpiece).

**Retraction in tool direction is always executed prior to swiveling the rotary axes with the current tool orientation (rotation).**

If the swivel plane is undefined at the start of a program, the axis should first be traversed to a safe position or be safely prepositioned using retraction Z.

Maximum and incremental retraction in tool direction is particularly suitable for machines with swivel heads. In the case of a machine with a swivel head and an active swivel frame (rotations in workpiece reference are not 0), several machine axes can be traversed when traversing in the tool direction.

**Example:**

In the case of a machine with a swivel head, the head (and the WCS) in the G17 plane is swiveled 90 degrees around the X axis. When traversing the Z axis in the program, therefore, the Y axis traverses. With "maximum retraction in tool direction", the Y machine axis traverses to the positive software end position.

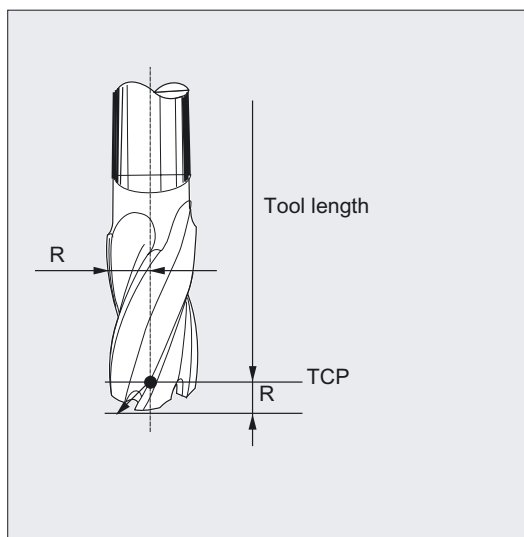
Retraction mode options (can be set in CYCLE800 installation and setup menu)		
Do no retract prior to swiveling	Retract:	no
Retract prior to swiveling	Retract:	Z
Retract machining axes prior to swiveling	Retract:	Z,XY
Maximum retraction in tool direction	Retract:	Tool max.
Incremental retraction in tool direction	Retract:	Tool inc., e.g. 100.00

### 3.16.4 Setting of tools - CYCLE800

#### Function

After "Swiveling plane", the tool orientation is always vertical on the machining plane. When **milling with radial cutters**, it can make technological sense to set the tool at an **angle** to the normal surface vector.

In the swivel cycle, the setting angle is generated by an axis rotation (max. +/- 90 degrees) to the active swivel plane. When setting, the swivel plane is always "additive". With "Setting tool", only rotations are displayed on the swivel cycle input screen form. The user can select the rotation sequence.



**The TCP (Tool Center Point) must be entered as the tool length of the radial cutter.**

If a program with several swivel cycles is to be milled with a set tool, the "new" swivel plane must always be set in the subsequent swivel call.

It is also possible to set tools on machines without the TRAORI option (see LEAD advance angle and TILD sideways angle functions).

#### Screen form

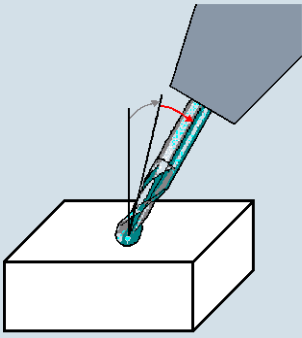
Entry to programs/milling area

Soft key



If the function "B axis kinematics" is set up in the start-up menu for CYCLE800, it is accessed with soft key "Swivel tool" and in the second plane with soft key "Setting tool".



Approach TL/CYCLE800		Name of swivel data record	
		Name:	TABLE
		Retract:	Z
		Swivel plane:	Additive
		Swivel mode:	Axis by axis
		Rot. around	X (A) 8.000
		Rot. around	Y (B) -8.000

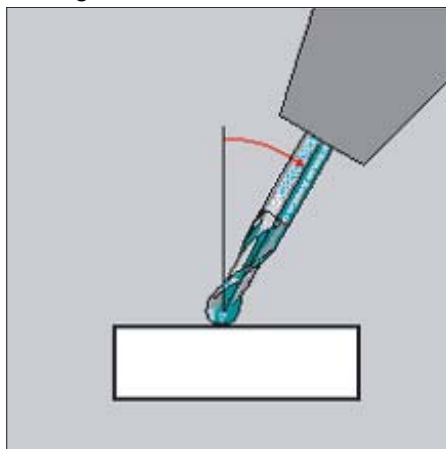
The "Setting tool" input screen form corresponds to the minimized CYCLE800 input screen form.

On the input screen form, the following applies:

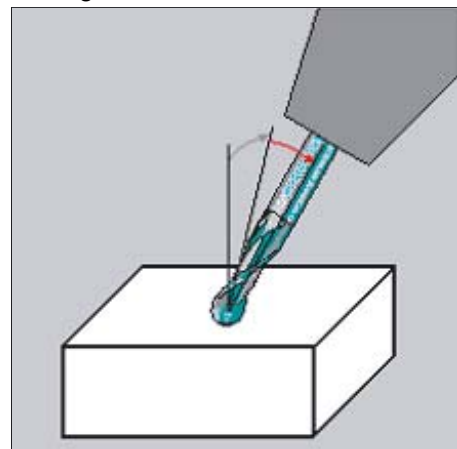
<b>Name _TC:</b>	Current swivel data record
<b>Retraction _FR:</b>	As for CYCLE800
<b>Swivel plane _ST:</b>	Additive (display only)
<b>Swivel mode _MODE:</b>	Axially (display only)
<b>Rotation around:</b>	Rotation around maximum of 2 axes in the WCS XY, YA, XZ
<b>Correct tool:</b>	As for CYCLE800 (for display variant, see CYCLE800 startup)

## Setting tool help displays

Setting tool: 1st rotation



Setting tool: 2nd rotation



### 3.16.5 Alignment of tools - CYCLE800

#### Function

The purpose of the "Align tool" function is to support turning machines with a swivel-mounted B axis. This functionality is designed for use with a specific configuration of turning or milling machines on which the tool orientation is implemented by a swivel axis B (around Y) with associated milling spindle (C1). It is designed to be compatible with both turning and milling tools.

In contrast to "Swivel plane", no rotation is operative in the active frame chain (WCS) in the case of "Swivel tool" or "Align tool". Only the offsets calculated by the NC and the corresponding tool orientation are operative.

The maximum angular range for "Align tool" is + 360 degrees or it is limited by the traversing range of the participating rotary axes. Technological limits are also placed on the angular range depending on the tool used.

In the case of "Align tool" in the turning area, activation of NC function CUTMOD ensures that the correct tools for tool point direction or tool clearance angle of a turning tool are made available.

#### Screen form

Entry to programs area/

Soft key



Entry to programs/milling area

Soft key



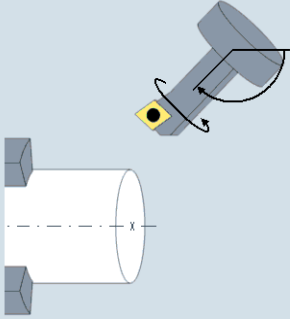
#### Coding of B axis kinematics parameter CYCLE800

- `_MODE = 57` (axis-by-axis rotation XYZ)
- `_ST 100s digit = 2xx`

### Access to B axis kinematics in turning area:

Program / Turning / VK7 "Align tool"

Input screen form "Align tool" under Program in turning area

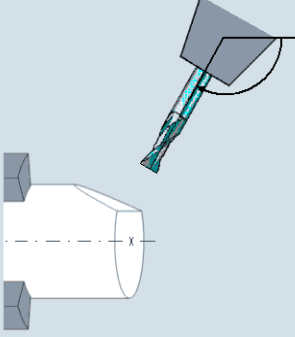

Align tool / CYCLE800		Name of swivel data record
	Name:	TURN_BC
	Retract:	<input type="text" value="No"/>
	Swivel plane:	<input type="text" value="New"/>
	Swivel mode:	<input type="text" value="Axis by axis"/>
	Rot. around Y (B)	<input type="text" value="60.000"/>
	Rot. around Z (C)	<input type="text" value="5.000"/>

On the input screen form, the following applies:

<b>Name _TC:</b>	Select or deselect swivel data records with B axis kinematics
<b>Retraction _FR:</b>	As for CYCLE800
<b>Swivel plane _ST:</b>	New (display only)
<b>Swivel mode _MODE:</b>	Axially (display only)
<b>Rotation about:</b>	The input "Rotate about Y(B)" or "Rotate about Z(C)" for the WCS are used to calculate the tool orientation and the corresponding WCS offsets which result from the machine kinematics.
<b>Correct tool:</b>	As for CYCLE800 (for display variant, see CYCLE800 startup)

**Access to "Align tool" in milling area:**

Program / Milling / &gt;&gt; / Swivel tool / Align tool

Align tool / CYCLE800		Name of swivel data record	
	Name:	TURN_BC	
	Retract:	Z	
	Swivel plane:	New	
	Swivel mode:	Axis by axis	
	Rot. around	Y (B)	25.000

On the input screen form, the following applies:

Parameters, see input screen form "Align tool" in turning area

---

**Note****Options for combining Swivel plane, Align tool and Set tool in the milling area**

- Swivel plane: Swivel plane new or additional.
  - Align tool: Swivel plane new, this deletes any previously programmed swivel plane.
  - Setting tool: Swivel plane is added to a swivel plane defined with the Swivel plane function.
- 

The "Align tool" function must be enabled as "B-axis kinematics" in the start-up menu for CYCLE800.

**Note the machine manufacturer's instructions.**

## 3.16.6 Setting up workpieces with swiveled machining planes

### 3.16.6.1 General information

#### Swiveling in JOG mode

The "Swiveling in JOG" function is used to set up workpieces with swiveled (oblique) machining planes or any clamped workpieces and represents the prerequisite for further setup operations (scratching/measuring edge/corner).

Swiveling in JOG mode is supported by its own input screen form.

Once the input values are applied and the NC is started, the rotary axes (swivel axes) are positioned (rotated with manual rotary axes) and a swivel frame is calculated for this plane. The swivel frame ensures that the tool orientation is aligned perpendicular to the swivel plane.

The following functions can be carried out with "Swiveling in JOG":

- Compensation of basic rotation of the workpiece on machines with swivel tables.
- New swiveling or additive swiveling to any plane if permitted by the machine kinematics.
- Swiveled plane can be set to the new zero plane.

The zero plane corresponds to the tool plane (G17, G18, G19) including the active zero offset (G500, G54, etc.). Rotations of the active ZO and offsets of the rotary axes are taken into account when swiveling in JOG. The "Swiveling in JOG" function writes only rotations either in the workpiece reference (\$\_WPFRAME) or to the active ZO.

Operators/machine setters must be aware of the machine position that the unswiveled plane, the basic setting of the machine kinematics (kinematic chain) or the tool orientation occupies in this basic setting. The machine manufacturer has defined the basic machine kinematics setting when setting up the "Swiveling" or "5-axis transformation (TRAORI)" function.

#### **Note the machine manufacturer's instructions!**

The operator/machine setter can scratch a workpiece or use a probe to measure a workpiece on any swiveled plane. In this case, the determined values of the zero offset refer to the unswiveled plane or the basic setting of the machine kinematics. The operators/machine setter can also specify the plane to be defined as the new zero plane. In this case, the determined values of the zero offset refer to the new basic setting (machine setter).

## 3.16.6.2 Parameters of input screen form

## Screen form

- In JOG mode, the horizontal soft key 8 is the entry soft key for "Swiveling in JOG".
- Horizontal soft key 3 is the entry soft key for "Measuring in JOG".

Anzeige der Nr. der aktiven NV und Summe aller Drehungen der Framekette, einschließlich des Schwenkframes.

Anzeige der Drehungen der aktiven NV

Maschine: TRSL MESS JOG WKS.DIR\HAA\_MESS.WPD  
HAA\_MESS.MPF  
Kanal RESET Programm abgebrochen  
ROV

MKS	Position	Repos-Versch.
X1	0.000 mm	0.000
Y1	0.000 mm	0.000
Z1	0.000 mm	0.000
A1	0.000 grd	0.000
B1	0.000 grd	0.000

Transformation + G-Funktionen  
TCARR = 1  
01:G01 04:STARTFIFO  
06:G17 08:G54  
10:G60 12:G601

Schwenken  
Name des Schwenkdatensatzes  
G54  
Name: MIXED\_AB  
Freifahren: Z  
Schwenkebene: neu  
Schwenkmodus: achsweise  
Drehung um X (A) 45.000  
Drehung um Y (B) 0.000  
Drehung um Z (C) 0.000  
Richtung: Minus

Start Verfahrbewegung Schwenkachsen -> NC-Start betätigen

VSK 1: Alternativ  
VSK 3: Grundstellung  
VSK 4: Nullebene setzen  
VSK 5: Nullebene löschen

- The input screen form for the "Swiveling in JOG" function corresponds to a shortened version of the CYCLE800 Swiveling input screen form.
- The "Retraction", "Direction" and "Correct tool" input fields can be set using display variants in the "Swiveling" startup screen.

## Explanation of input screen form parameters:

## Name\_TC or TRAORI

- Name of swivel data record
- Deselection

An active swivel data record is deselected. Swivel frames are deleted.

In the case of active 5-axis transformation and deselected swivel data record (TCARR=0), TRAORI is displayed. In this case, the swivel plan is approached by means of the 5-axis transformation.

**Retract**

Corresponds to the CYCLE800 parameter \_FR (retraction).

**Swivel plane**

- New swivel plane
- Additive swivel plane

**Swivel mode**

- Axis-by-axis

Rotation around X,Y,Z (selective), same as CYCLE800

- Directly

This allows rotary axis positions to be entered directly. After rotary axes have been positioned (or manual rotary axes have been rotated), a swivel frame is calculated for these positions. In "direct" swivel mode, input fields with the names of the rotary axes of the active swivel data record are displayed instead of "Rotation about" input fields. Vertical soft key 6 provides "Teach-in", in order to transfer the actual values of the automatic rotary axes.

<b>"Direct" swivel mode (example)</b>		
Swivel mode	Directly	Directly
Rotary axis 1	C	-15.00
Rotary axis 2	B	0.000

**Direction (for axis-by-axis swivel mode only)**

Corresponds to the CYCLE800 parameter \_DIR (direction).

**Correct tool**

Corresponds to the CYCLE800 parameter \_ST=1x (correct tool tip).

**Note**

Note the machine manufacturer's instructions.

**Explanation of vertical soft keys of input screen form:****Basic setting (VSK 3)**

Approaching the basic machine setter setting. If the frame chain does not contain any rotations, this corresponds to the machine kinematics basic setting in consideration of the translational offsets of the rotary axes (basic rotation).

Input screen form defaults with vertical soft key basic setting:

- Name: Active swivel data record
- Retract: No
- Swivel mode: Axis-by-axis
- Rotation about X 0.0
- Rotation about Y 0.0
- Rotation about Z 0.0

### Set zero plane (VSK 4)

The approached swivel plane is to be set to the new zero plane.

The rotations of the swivel frame (\$P\_WPFRAME) are transferred to the rotations of the active ZO. When VSK4 "Set zero plane" is pressed, the machine axes do not move. The complete coordinate system (rotations of the frame chain) remain unchanged with "Set zero plane".

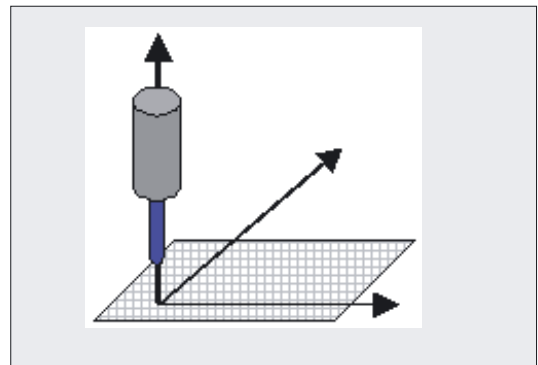
If the rotations of the swivel frame are other than 0, a new basic "machine setter" setting is defined by "Set zero plane", which deviates from the basic machine kinematics setting.

This basic machine setter setting applies only to the active ZO for setting up the workpiece.

All subsequent actions such as scratching, measuring with probe, and swiveling with NC workpiece programs refer to this zero plane or to the basic machine setter setting.

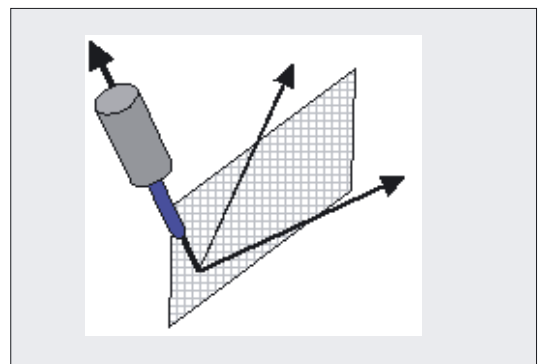
#### Basic machine kinematics setting

- The tool is located vertically on the plane.
- The position of the workpiece coordinate system (WCS) on the plane is "still" undefined.



#### Setup basic setting

- The active ZO contains rotations or an offset in one or both rotary axes.
- The tool is located vertically on the swiveled plane.
- The position of the workpiece coordinate system (WCS) on the plane is "still" undefined.





**Clear zero plane (VSK 5)**

The rotations of the active ZO are transferred to the rotations of the swivel frame (\$P\_WPFRAME). When VSK5 "Clear zero plane" is pressed, the machine axes do not move. The complete coordinate system (rotations of the frame chain) remain unchanged with "Clear zero plane".

---

**Note****Approaching the basic machine kinematics setting**

1. Press VSK 5 "Clear zero plane"
  2. Press VSK 3 "Initial setting"
  3. NC Start
- 

**Example:**

The basic machine kinematics setting of the tool is vertical on the rotary axis C (vertical tool orientation) on a machine with a rotary table around the Z tool axis (rotary axis C) and a swivel head around the Y machine axis (rotary axis B).

If a workpiece is to be machined exclusively with horizontal tool orientation, the tool must first be positioned horizontally using a rotation around Y of 90 degrees (possibly in conjunction with a rotation around Z of 180 degrees) using "Swiveling in JOG".

The ZO is then defined using Scratching or Measuring in JOG, with horizontal tool orientation. The translational values of the ZO refer to the basic machine kinematics setting. If the NC machining program is to start in horizontal tool orientation with 0 swivel, a new "basic machine setter setting" (horizontal) can be defined with VSK4 "Set zero plane".

**Soft key << (VSK 8)**

Input screen form "Swiveling" is exited.

**Note on the active 5-axis transformation (TRAORI)**

With active 5-axis transformation, the "Swiveling in JOG" input screen form fields below are displayed:

- TRAORI (can be selected with deselection or set-up swivel data records)
- New/additive swivel plane → New zero plane
- Swivel mode: Axis-by-axis (cannot be selected)
- Rotation about X
- Rotation about Y
- Rotation about Z

**Note on retracting the tool axis and traversing the rotary axes**

The tool axis (e.g. with G17 = Z) can be retracted prior to swiveling the rotary axes.

Conventional rapid traverse for JOG is used for positioning (MD 32010 \$MA\_JOG\_VELO\_RAPID[AX]).

If the rotary axes are positioned, they are also traversed with G0 and conventional rapid traverse for JOG (MD 32010 \$MA\_JOG\_VELO\_RAPID[AX]). Only the TOOLCARR.SPF cycle is used for traversing, which can be adjusted by the machine manufacturer.

If a swivel plane with active TRAORI is traversed, the TOOLCARR.SPF cycle is also used for this, whereby the corresponding direction vector is used to set the tool orientation to vertical on the swivel plane (e.g. for G17 with C3=1).

**Note on offsets in the zero offset of rotary axes of the swivel data record**

All rotary axes that are declared in the swivel data record can always have a value in the offset of the zero offset. Offsets in the zero offset of the rotary axis that rotate around the tool axis do not cause rotation of the WCS (basic rotation).

In all other cases, offsets in one rotary axis or in both rotary axes result in rotation of the WCS and, thus, in a **change in the basic machine kinematics setting**. This rotation is taken into account in the workpiece reference (WPFRAME) when CYCLE800 is called and must therefore be noted in the subsequent program steps.

This does not apply to manual or semi-automatic rotary axes of the swivel data record as these are unknown to the NC and thus have no zero offset.

**3.16.6.3 Data transfer of swivel data in "Swiveling in JOG"**

Only rotations are written to the active frame chain with the "Swiveling in JOG" function. All translational offsets in the active frame chain are retained.

Depending on the possible combinations of rotations or setting/deleting of the zero plane, the following apply to the transfer of swivel data to the current frame chain:

**Swivel plane: New**

- Deletes and links the rotations (of geometry axes XYZ) of the swivel frame workpiece reference (\$P\_WPFRAME)
- Rotations of the active ZO are retained

**Swivel plane: Additive**

- Links the rotations of the input values to the rotations of the swivel frame workpiece reference (\$P\_WPFRAME)
- Rotations of the active ZO<sup>1)</sup> are retained

### **Set zero plane**

Transfers the rotations of the swivel frame workpiece reference (\$P\_WPFRAME) to the rotations of the active ZO<sup>1)</sup>

### **Delete zero plane**

Transfers the rotations of the active ZO to the rotations of the swivel frame workpiece reference (\$P\_WPFRAME)

---

#### **Note**

##### **1) ZO – current zero offset**

If the basic ZO is active (G500) and the “Set zero point” system frame (\$P\_SETFRAME) is set up, the swivel data are written in the SETFRAME.

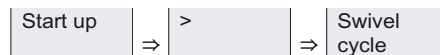
If the basic ZO is active (G500) and the “Set zero point” system frame (\$P\_SETFRAME) is not set up, the swivel data are written in the last active basic frame.

---

## **3.16.7 Startup - CYCLE800**

### **3.16.7.1 General information**

#### **Entry soft key**



When CYCLE800 is started up, the data (swivel data record) are set in the tool data \$TC\_CARR1.... These are grouped to form Swivel startup menus.

#### **References:**

For current information, see:

- "siemensd.txt" file in the supplied software (standard cycles) or
- In the case of HMI Advanced, F:\dh\cst.dir\HLP.dir\siemensd.txt.
- Description of Functions 840D/840Di/810D  
  /W1/ Tool Offset (Inclined Machining with 3 + 2 Axes)
- ISO 841-2001 or DIN 66217

The following cycles must be loaded:

- CYCLE800.SPF, CYCPE\_SC.SPF (standard cycles)
- TOOLCARR.SPF (standard cycle)
- PROG\_EVENT.SPF (standard cycle)

The PROG\_EVENT.SPF cycle is supplied as a standard cycle and is used for prepositioning of rotary axes (swivel axes) after a block search.

The machine manufacturer can expand the functionality of the PROG\_Event by adding his own manufacturer cycles. CYCPE1MA.SPF or CYCPE\_MA.SPF must be used as the name of the manufacturer cycles.

Corresponding exit markers are prepared at the start and end of PROG\_EVENT.SPF for the CYCPE1MA.SPF and CYCPE\_MA.SPF manufacturer cycles.

If the CYCPE1MA.SPF and CYCPE\_MA.SPF manufacturer cycles are loaded in the NC, the program branches from PROG\_EVENT to the manufacturer cycles as appropriate.

- The \_TC\_FR to \_TC\_NUM (GUD7) GUD variables must be activated.

### Activate CYCLE800 installation and startup menu

The screen form in the startup operating area is activated as follows:

- HMI Embedded

The "Swivel cycle" soft key in the COMMON.COM file must be activated. Open the COMMON.COM file and then delete the ";" in front of SC616.

The file is located in the "User Cycles" directory. Then perform an NC reset.

- HMI Advanced:

Delete the ";" in front of the following lines in the STARTUP.COM (standard cycles) file:

```
;HS14=( $82084 , ac7 )
;PRESS (HS14)
;LM( "SWIVEL1a" , "SWIVEL1.com" )
;END_PRESS
```

Following these changes, restart HMI Advanced.

### Activate "Swiveling in JOG"

Horizontal soft key 8 is the "Swiveling" entry soft key in machine/JOG mode.

Soft key 8 is activated in the MA\_JOG.COM file, which is located in the "Standard cycles" directory.

Delete the ";" in front of the following lines in the MA\_JOG.COM file:

```
HS8=( $82119 , , sel )
;PRESS (HS8)
;LM( "SWIVEL3" , "jog_c800.com" )
;END_PRESS
```

### Machine and setting data

To use the swiveling cycle, the machine data below must be set as follows (minimum requirement):

- **Machine data with precise value assignment (P):**  
These are machine data that cannot be modified.
- **Machine data with variable value assignment (V):**

These are machine data, whose default values can be increased or decreased.

MD number	MD identifier	Value	Comment	Value assignment
10602	\$MN_FRAME_GEOAX_CHANGE_MODE	1	<sup>2)</sup>	V
11450	\$MN_SEARCH_RUN_MODE	Bit 1=1	Activate PROG_EVENT after block search	G
11602	\$MN_ASUP_START_MASK	Bit 0=1	Swiveling in JOG	V
11604	\$MN_ASUP_START_PRIO_LEVEL	100	Swiveling in JOG	V
<b>18088</b>	\$MN_MM_NUM_TOOL_CARRIER	n>0	n = Number of swivel data records <sup>1)3)</sup>	V
18114	\$MM_ENABLE_TOOL_ORIENT	2	Basic tool orientation <sup>3)</sup>	V
20108	\$MC_PROG_EVENT_MASK	0	Systemasub PROG_EVENT after block search	V
20110	\$MC_RESET_MODE_MASK	'H4041'	Bit 14=1	G
20112	\$MC_START_MODE_MASK	'H400'	-	G
20126	\$MC_TOOL_CARRIER_RESET_VALUE	0...n	Described in CYCLE800	V
20150	\$MC_GCODE_RESET_VALUES[41]	1	TCOABS <sup>5)</sup>	G
20150	\$MC_GCODE_RESET_VALUES[51]	2	PAROT <sup>5)</sup>	G
20150	\$MC_GCODE_RESET_VALUES[52]	1	TOROTOF <sup>5)</sup> (for kinematics types T and M only)	V
20152	\$MC_GCODE_RESET_MODE[41]	0	TCOABS	G
20152	\$MC_GCODE_RESET_MODE[51]	0	TOROTOF	V
20152	\$MC_GCODE_RESET_MODE[52]	0	PAROT	V
20180	\$MC_TOCARR_ROT_ANGLE_INCR[0]	0	<sup>8)</sup>	G
20180	\$MC_TOCARR_ROT_ANGLE_INCR[1]	0	<sup>8)</sup>	G
20182	\$MC_TOCARR_ROT_ANGLE_OFFSET[0]	0	<sup>8)</sup>	G
20182	\$MC_TOCARR_ROT_ANGLE_OFFSET[1]	0	<sup>8)</sup>	G
20184	\$MC_TOCARR_BASE_FRAME_NUMBER	-1	<sup>8)</sup>	G
20360	\$MC_TOOL_PARAMETER_DEF_MASK	Bit 10=1	For T, M kinematics, if orientation vector is to be retained even for T0 and D0 (no tool).	V
21100	\$MC_ORIENTATION_IS_EULER	0	Angles of rotations are interpreted as RPY	V
21104	\$MC_ORI_IPO_WITH_G_CODE	1	Relevant for swiveling + correct tool with TRAORI (see TOOLCARR.SPF)	V
22530	\$MC_TOCARR_CHANGE_M_CODE	0	<sup>6)</sup>	V
24006	\$MC_CHSFRAME_RESET_MASK	Bit 4=1	If system frame \$P_WPFRAME is to remain active after a reset <sup>9)</sup>	V
24007	\$MC_CHSFRAME_RESET_CLEAR_MASK	Bit 2,3=0 Bit 4=1	- If system frame \$P_WPFRAME is to be deleted on reset <sup>9)</sup>	V

MD number	MD identifier	Value	Comment	Value assignment
24008	\$MC_CHSFRAME_POWERON_MASK	Bit 4, 3, 2=1	If system frames \$P_WPFRAME, \$P_TOOLFRAME and \$P_PARTFRAME are to be deleted on power on	V
28082	\$MC_MM_SYSTEM_FRAME_MASK	Bit 4, 3, 2=1	Setup of system frames <sup>1)</sup> \$P_WPFRAME, \$P_TOOLFRAME and \$P_PARTFRAME	G
28083	\$MC_MM_SYSTEM_DATAFRAME_MASK	Bit 4, 3, 2=1	Standard="H7f" <sup>1)</sup>	G
30455	MISC_FUNCTION_MASK	Bit 2, 0=1	For rotary axes declared as modulo axes <sup>7)</sup>	V
32010	\$MA_JOG_VELO_RAPID[AX]		Rapid traverse in JOG mode for swiveling in JOG  MD is relevant for rotary axes and machine axes which need to be recovered before swiveling commences.	V

SD no.	SD identifier	Value	Comment	Modifiable
42974	\$SC_TOCARR_FINE_CORRECTION	0	<b>Fine offsets of offset vectors</b>	V
42980	\$SC_TOFRAME_MODE	2000	<sup>5)</sup>	V

### Notes on machine and setting data

1. Making changes to the required machine data results in reorganization of the buffered memory (data loss).

Series startup must be performed after setting the MD and before NCK reset.

#### References:

/IAM/ Commissioning Manual HMI

2. **MD 10602**

If the ZO is not to be reprogrammed following TRAORI, e.g., for tool correction.

3. **MD 18088**

If several channels are defined in the NCU, the number of swivel data records is divided, taking MD 28085: MM\_LINK\_TOA\_UNIT into account.

#### – Example:

MD 18088 MM\_NUM\_TOOL\_CARRIER =4

Number of channels=2.

Two swivel data records are provided for each channel.

**4. MD 18114****Support of angular tools with tool orientation (SW 06.05.17.00 and higher)**

Angular tools are created and managed in the HMI or NCU with tool type 130. The tool lengths are entered in the tool of tool type 130. If an angular tool is also to be used on a swiveled machining plane (CYCLE800), the basic tool orientation must be disconnected with MD18114 MM\_ENABLE\_TOOL\_ORIENT=2. This enables a direction vector to be transferred to the tool in parameters \$TC\_DPV3[aa,bb] to \$TC\_DPV5[aa,bb].

The TOROT (G17), TOROTY (G18), and TOROTX (G19) commands must be programmed for the tool change so that the basic tool orientation is updated.

**5. MD 20150/MD 20152 and SD 42980**

Note on kinematics (swivel head/mixed kinematics) with Hirth tooth system:

Depending on the active plane (G17,G18,G19), the TOROT command (or TOROTX, TOROTY) is programmed (G-group 53) to calculate (in the NCU) the compensating frame for the Hirth tooth system in CYCLE800.

If the Hirth tooth system causes the programmed rotation to deviate from the rotary axes' possible positions, a \$P\_TOOLFRAME compensating frame is created for swivel head and mixed kinematics (see HMI active ZO/details tool reference).

The calculation depends on setting data SD 42980: \$SC\_TOFRAME\_MODE.

For kinematics (swivel head/mixed kinematics) with Hirth tooth setting data SD 42980: Set \$SC\_TOFRAME\_MODE=2000!

If the compensation frame (TOOLFRAME) is to be retained after Reset or end of program, the following value must be entered in MD 20150: \$MC\_GCODE\_RESET\_VALUES[52]:

- For G17 (TOROT) ⇒ MD 20150=2
- For G18 (TOROTY) ⇒ MD 20150=3
- For G19 (TOROTX) ⇒ MD 20150=4

**6. MD 22530**

If several swivel data records are declared per channel, and if machine functions need to be activated on changeover between swivel heads or tables, an M command can be issued in the PLC program on switchover to another swivel data record.

– **Example:**

Number of swivel data records in channel 1 = 2

MD 22530: TOCARR\_CHANGE\_M\_CODE = **-800**

Programming of swivel data record 1 (TCARR=1) ⇒ M801

Programming of swivel data record 2 (TCARR=2) ⇒ M802

The PLC can limit or invert the spindle speed, for example, by outputting M commands.

**7. MD 30455**

This allows the rotary axis to traverse for G90 with DC (shortest path); see TOOLCARR.SPF user cycle.

## 8. MD 20180/MD 20182

For rotary axes with the Hirth tooth system, the relevant values are entered in the CYCLE800 startup menu.

## 9. MD 24006/MD 24007

For Swiveling in JOG:

- MD24006 Bit 4=1
- MD24007 Bit 4=0

The workpiece reference (WPFRAME) must be active on Reset, in order to be able to scratch or measure (cascaded measuring) with the active ZO after swiveling in JOG.

## 3.16.7.2 Startup of kinematic chain

The CYCLE800 startup process is supported by the following menu displays:

- Startup of kinematic chain
- Startup of rotary axis parameter
- Startup of fine kinematics

## Startup of kinematic chain menu display

Kinematic channel1			
Kinematics	Swivel table	Name:	TABLE No.: 2
Retract:	X	Y	Z
			Max. + inc. tool direction
Retract position			200.000
Offset vector I2	260.000000	200.000000	0.000000
Rotary axis vector V1	-1.000000	0.000000	0.000000
Offset vector I3	0.000000	0.020000	20.400000
Rotary axis vector V2	0.000000	0.000000	-1.000000
Offset vector I4	-260.000000	-200.020000	-20.400000
Swivel mode:	axial +projection angle		
Swivel mode direct	Yes		
Direction:	Rot. axis 2 optimized		
Tracking TL	No		
B axis kinematics	No		

You must set up a swivel data record for every swivel head, swivel table or combination of both.

Swivel data records can be declared in several channels (see machine data).

A swivel data record consists of 40, or on NCU Version 7.1, of 65 parameters that are assigned to the tool data \$TC\_CARR1[n] to \$TC\_CARR40[n] or \$TC\_CARR65[n]. Only those parameters that are relevant for the kinematic type are displayed in the Startup of kinematic chain menu display. The parameter values are taken over immediately after entry.



Data record  
Save

#### Explanation of vertical soft keys:

Current swivel data record is saved as the part program.

The part program name corresponds to the name of the swivel data record.

Data record  
Delete

All parameters of the swivel data record are deleted.

#### Explanation of parameters:

The meaning of the parameters in the "Kinematics" screen form is as follows:

#### Name: Swivel data record \$TC\_CARR34[n]

n ⇒ Swivel data record number

If several swivel data records are declared in each NC channel, then a name is assigned to each swivel data record. No name needs to be specified if the swivel-mounted toolholder is not exchangeable (i.e. one swivel data record per channel).

The program is advanced to the next swivel data record and the next channel via soft keys (channel +/- swivel data record +/-).

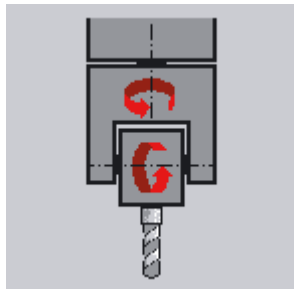
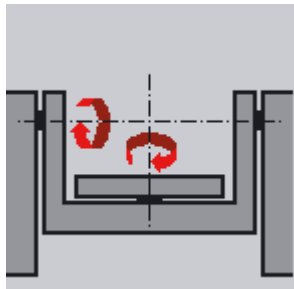
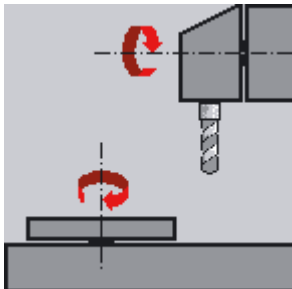
#### Note

Swivel data record names may only contain permitted NC programming characters (A...Z, 0..9 and \_)!

#### Kinematic type \$TC\_CARR23[n]

Selection:

- Swivel head (type T)
- Swivel table (type P)
- Swivel head + swivel table (type M)

Swivel head (type T)	Swivel table (type P)	Swivel head + swivel table (type M)
		
Offset vector I1	Offset vector I2	Offset vector I1

Swivel head (type T)	Swivel table (type P)	Swivel head + swivel table (type M)
Rotary axis vector V1	Rotary axis vector V1	Rotary axis vector V1
Offset vector I2	Offset vector I3	Offset vector I2
Rotary axis vector V2	Rotary axis vector V2	Offset vector I3
Offset vector I3	Offset vector I4	Rotary axis vector V2
		Offset vector I4

The vectors always refer to the **basic setting** of the **machine kinematics**.

#### Swivel head

- I3 distance from the tool adapter to the pivot point/intersection of the 2nd rotary axis
- I2 distance from the pivot point/intersection of the 2nd rotary axis to the pivot point/intersection of the 1st rotary axis
- I1 closure of the  $I1 = -(I2 + I3)$  vector chain, if the swivel head cannot be changed

#### Swivel table

- I2 distance from the machine reference point to the pivot point/intersection of the 1st rotary axis
- I3 distance from the pivot point/intersection of the 1st rotary axis to the pivot point/intersection of the 2nd rotary axis (or to the reference point of the tool adapter)
- I4 closure of the  $I4 = -(I2 + I3)$  vector chain, if the swivel table cannot be changed

#### Swivel head/swivel table (mixed kinematics)

- I2 distance from the tool adapter to the pivot point/intersection of the 1st rotary axis
- I1 closure of the  $I1 = -I2$  vector chain, if the swivel head cannot be changed
- I3 distance from the machine reference point to the pivot point/intersection of the 2nd rotary axis (or to the reference point of the tool adapter)
- I4 closure of the  $I4 = -I3$  vector chain, if the swivel table cannot be changed


The offset vectors do not have to point to the pivot point of the rotary axes. The important thing is that they point to a point on the direction of rotation (intersection).


The sign of the offset vectors (I1 to I4) and the rotary axis vectors (V1, V2) result from the specifications of the axis directions according to ISO 841-2001 or DIN 66217 (right-hand rule). In the case of kinematics that move the workpiece (rotary table), the motion of the workpiece relative to the tool must be taken into account.


#### Interrelation of TOOLCARRIER data ↔ 5-axis transformation (transformer type 24, 40, 56):


##### Note

With NCU SW 7.1 and higher, transformer type 72 (\$MC\_TRAFO\_TYPE\_1) can be used for a 5-axis transformation. The toolholder (TOOLCARRIER) vectors are used with transformer type 72.

Swivel head (swivel-mounted tool)		
\$TC_CARR23[1]="T"		\$MC_TRAFO_TYPE_1=24
I1 \$TC_CARR1...3[n]		\$MC_TRAFO5_PART_OFFSET_1[0...2]
I2 \$TC_CARR4...6[n]		\$MC_TRAFO5_JOINT_OFFSET_1[0...2]
I3 \$TC_CARR15...17[n]		\$MC_TRAFO5_BASE_TOOL_1 [0...2]
Closure of the I1=-(I2+I3) vector chain; for fixed-mounted machine kinematics		

Swivel table (swivel-mounted workpiece)		
\$TC_CARR23[1]="P"		\$MC_TRAFO_TYPE_1=40
I2 \$TC_CARR4...6[n]		\$MC_TRAFO5_BASE_TOOL_1 [0..2]
I3 \$TC_CARR15...17[n]		\$MC_TRAFO5_JOINT_OFFSET_1 [0...2]
I4 \$TC_CARR18...20[n]		\$MC_TRAFO5_PART_OFFSET_1[0...2]
Closure of the I4=-(I2+I3) vector chain; for fixed-mounted machine kinematics		

Swivel head + swivel table (swivel-mounted tool + workpiece)		
\$TC_CARR23[1]="M"		\$MC_TRAFO_TYPE_1=56
I1 \$TC_CARR1...3[n]		\$MC_TRAFO5_JOINT_OFFSET_1 [0...2]
I2 \$TC_CARR4...6[n]		\$MC_TRAFO5_BASE_TOOL_1 [0...2]
I3 \$TC_CARR15...17[n]		\$MC_TRAFO5_JOINT_OFFSET_PART_1[0...2]
I4 \$TC_CARR18...20[n]		\$MC_TRAFO5_PART_OFFSET_1[0...2]
Closure of the I1=-I2 I4=-I3 vector chain; for fixed-mounted machine kinematics		

Rotary axis vectors V1, V2		
Rotary axis vector V1		
\$TC_CARR7..9[n]		\$MC_TRAFO5_AXIS1_1[0..2]
Rotary axis vector V2		
\$TC_CARR10..12[n]		\$TC_CARR10..12[n]

**Retraction/Retraction position****\$TC\_CARR38[n] X; \$TC\_CARR39[n] Y; \$TC\_CARR40[n] Z**

n ⇒ Swivel data record number

The startup engineer determines whether the 'Retract Z axis' and 'Retract Z,X,Y axes' or 'Retract in tool direction' may be included in the input menu for the swivel cycle (see display variants below).

The mode of retraction can be modified in the TOOLCARR.SPF user cycle (see User Cycle TOOLCARR.SPF), if necessary. If the TOOLCARR.SPF user cycle is not modified, the retraction position is approached as an absolute machine position.

**NOTICE****Please note the following when traversing the tool axes:**

Retract the tool axis in such a way that the tool and workpiece cannot collide when swiveled.

**Offset vectors I1 to I4 for rotary axis vectors V1, V2****\$TC\_CARR1[n] ... \$TC\_CARR20[n]**

→ see also kinematics (\$TC\_CARR23)

The vectors always contain three components, which represent the reference to the X, Y, Z machine axes (see also Kinematics: Type above).

The positions in the kinematic chain are measured by the machine manufacturer; they are always relevant with respect to a swivel head/swivel table (swivel data record).

Offset vectors I1 to I4 refer to the non-swiveled state of the rotary axes (machine kinematics basic setting).

The machine kinematics used do not need to be fully implemented. However, be aware that the traversing range in the swivel planes may be restricted. If machine kinematics are to be implemented with just one rotary axis, this must always be declared as the 1st rotary axis.

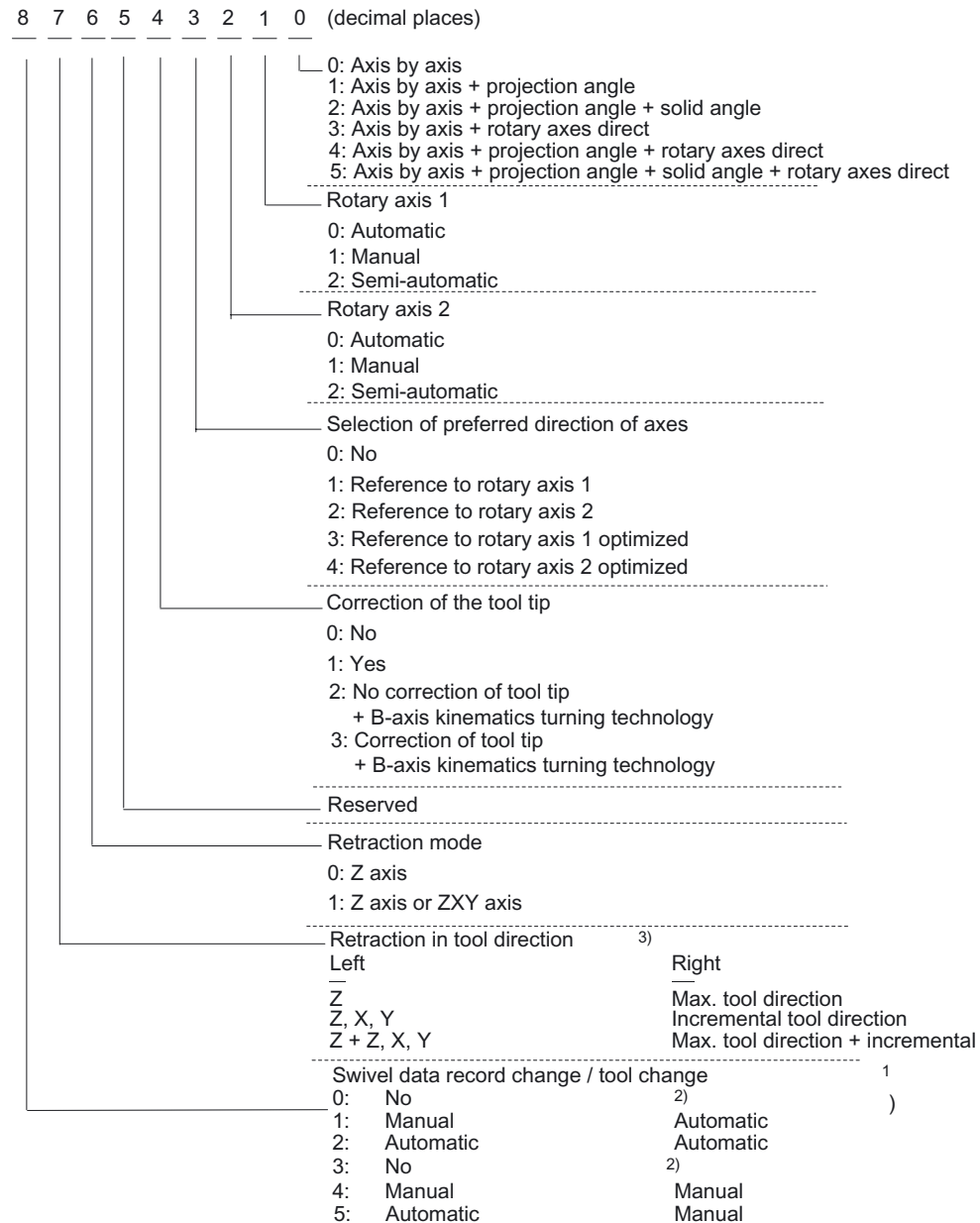
Manually adjustable rotary axes (manual mode) are possible with and without measuring systems and can be used with "plain machines".

## Display variants of input screen forms CYCLE800

\$TC\_CARR37[n]

(n ⇒ swivel data record)

If the following display variants are not set, the value will not be displayed in the input screen form (see Section "Programming via Screen Form").



1) Only relevant for ShopMill/ShopTurn

2) If no swivel data record change is declared, the setting automatic/manual tool change setting is not relevant

3) Coding of retraction modes, see following table

Table 3-1 Coding of retraction modes Z, ZXY, maximum, incremental tool direction

Z	Z, X, Y	Maximum tool direction	Incremental tool direction	Coding of \$TC_CARR37 xXXXXXXXX
1	0	0	0	00
1	1	0	0	01
0	1	0	0	02
0	0	0	0	03
0	0	1	0	04
1	0	1	0	05
0	1	1	0	06
1	1	1	0	07
0	0	0	1	08
1	0	0	1	09
0	1	0	1	10
1	1	0	1	11
0	0	1	1	12
1	0	1	1	13
0	1	1	1	14
1	1	1	1	15

The following display variants affect the input screen form for the swivel cycle:

- **Swivel mode**
  - ⇒ **Axis-by-axis**
  - ⇒ **Axis-by-axis and projection angle**
  - ⇒ **Axis-by-axis and projection angle and solid angle**
  - ⇒ **Swivel mode direct: yes/no** ⇒ see rotary axis direct

Example:

Selection of swivel mode during startup: Axis-by-axis, projection angle  
 In this case, only 'axis by axis' or 'projection angle' is included in the input menu.  
 It is neither desirable nor possible to program a solid angle in this example.

- **Direction**
  - ⇒ **Rotary axis 1**
  - ⇒ **Rotary axis 2**
  - ⇒ **No**
  - ⇒ **Rotary axis 1 optimized, rotary axis 2 optimized**

Reference to **rotary axis 1** or **2** for the travel direction selection in the input screen form for the swivel cycle.

The NCU calculates two possible solutions using the angle range of the **rotary axes** of the machine kinematics. Usually, one of these solutions is technologically suitable.

The **rotary axis** to which the two solutions are to refer is selected in the startup menu (see note for DIR under "Parameters of Input Screen Form").

The solution to be applied is selected in the input screen form for the swivel cycle.

If "no" is set, the Direction parameter is not displayed in the input screen form and parameter DIR is always set to the default -1 (minus).

**Example of effect of "Direction" parameter using mixed kinematics:**

1<sup>st</sup> rotary axis

Swivel head A rotates around machine axis X

2<sup>nd</sup> rotary axis

Rotary axis C rotates around machine axis Z

Angle range of 1st rotary axis                      -30 to +90 degrees

Angle range of 2nd rotary axis 0 to +360 degrees (modulo axis)

Direction parameter in startup screen form      Swivel on rotary axis 1

Program 1: Swivel CYCLE800 (... rotate by X = 20 degrees, ... minus direction)  
Axis A rotates to - 20 degrees and axis C to 180 degrees

Program 2: Swivel CYCLE800 (... rotate by X = 20 degrees, ... plus direction)  
Axis A rotates to 20 degrees and axis C to 0 degrees

Program 3: Swivel CYCLE800 (... rotate by X = 45 degrees, ... minus direction)  
Axis A rotates to 45 degrees and axis C to 0 degrees

Program 4: Swivel CYCLE800 (... rotate by X = 45 degrees, ... plus direction)  
Axis A rotates to 45 degrees and axis C to 0 degrees

The Direction parameter has no effect in programs 3 and 4 because two solutions are not produced for the kinematics starting at 30 degrees around X or Y due to the limited angle range of the A axis (-30 to +90) and therefore the first solution is always applied.

## Note

**Regarding the direction settings "Rotary axis 1 optimized" and "Rotary axis 2 optimized"**

If option "Direction rotary axis 1 or 2 optimized" is selected in the start-up menu for CYCLE800, two different direction settings "Plus" and "Minus" are provided in the input screen form for CYCLE800 in the basic machine kinematics setting and the rotary axis is positioned accordingly. This functionality did not exist in earlier versions of CYCLE800 (version 06.05.xx or earlier).

The functions "Rotary axis 1/2 optimized" are coded in transfer parameters \_ST (100000s digit) and DIR of CYCLE800 → see parameter description for CYCLE800

- **Correct tool**

⇒ **No**

⇒ **Yes**

"Correct Tool" display in the input screen form for the swivel cycle.

The correct tool function requires the 5-axis transformation (TRAORI) option.

- **B-axis kinematics**

⇒ **No**

⇒ **Yes**


See Align tool!

## See also

Manufacturer Cycle TOOLCARR.SPF - CYCLE800 (Page 265)

### 3.16.7.3 Starting up kinematic rotary axes

#### Menu screen for start-up of rotary axes

Rotary axis channel1					
Kinematics		Inclinable head	Name:	HEAD_B	No.: 2
Rot. axis 1	Identifier	C	Mode	Manual	
	Angle area	0.000	-	360.000	
	OffsetKinematics	0.000	Angle offset	0.000	
	Hirth teeth	Yes	Angle grid	1.500	
Rot. axis 2	Identifier	B	Mode	Manual	
	Angle area	-90.000	-	90.000	
	OffsetKinematics	0.000	Angle offset	0.000	
	Hirth teeth	Yes	Angle grid	1.500	
Swivel data record change		No			

#### Explanation of parameters:

#### Rotary axis 1, rotary axis 2

\$TC\_CARR35[n]; \$TC\_CARR36[n]

n = number of swivel data record

The following identifiers should be chosen, where possible:



- Axis rotates around machine axis X  $\Rightarrow$  A
- Axis rotates around machine axis Y  $\Rightarrow$  B
- Axis rotates around machine axis Z  $\Rightarrow$  C

If the NCU axes are known, the same axis identifiers of the relevant NC rotary axes must be selected (see automatic mode).

If the NCU axes are unknown, you can use any axis identifier (up to 6 letters or numbers).

## Mode

### \$TC\_CARR37[n] (see Display Variants)

- **Automatic**

NC rotary axes are automatically traversed to the appropriate swivel angle.

- **Manual**

**Rotary axes** are moved manually by the operator to the appropriate position, e.g., "very simple machines" with manually adjustable **rotary axes** (measuring system: steel gauge).

Messages 62180/62181 display the swivel angle to be set.

- **Semi-automatic**

In "semi automatic" mode, messages 62180/62181 (display of the angle values to be set) are not displayed.

Application:

Manual rotary axes that are positioned at the calculated angle by an appropriate mechanism. For this purpose, the machine manufacturer can modify the TOOLCARR user cycle to pass the swivel angle (local variable \_A1, \_A2) to a subroutine (markers \_M21... for manual rotary axes).

Both mixed machine kinematics (e.g., 1st rotary axis automatic, 2nd rotary axis manual) and "incomplete" machine kinematics with only one rotary axis are permitted.

If machine kinematics are to be implemented with just one rotary axis, this must always be declared as the 1st rotary axis.

## Angle range

### \$TC\_CARR30[n] .. \$TC\_CARR33[n]

A valid angle range (e.g., -90 to +90 degrees) must be assigned to each **rotary axis**. This does not have to be the software end position range of the rotary axis in question.

With modulo axes, you must enter a traversing range of between **0 and 360 degrees**.

## Offset kinematiks

### \$TC\_CARR24[n]; \$TC\_CARR25[n]

In the "Offset Kinematics" input field, it is possible to enter the offset of rotary axis 1 (\$TC\_CARR24[n]) or rotary axis 2 (\$TC\_CARR25[n]) if the position of the rotary axes is not equal to 0 in the basic setting.

#### Hirth tooth system

\$TC\_CARR26[n]... \$TC\_CARR29[n]

Selection:

- No

Subsequent fields are concealed.

- Yes

⇒ **Angular offset** of Hirth tooth system at start of gearing.

⇒ **Angular grid** of Hirth tooth system

⇒ **Automatic correction Yes/No** (omitted starting in Cycle SW version 6.3)

#### Swivel data record change (relevant only to ShopMill/ShopTurn)

\$TC\_CARR37[n] (see Display Variants)

Selection:

- No
- Automatic
- Manual

#### Tool change (relevant only to ShopMill/ShopTurn)

\$TC\_CARR37[n] (see Display Variants)

Selection:

- Automatic
- Manual

"Tool change" display for kinematic type T and M only

### 3.16.7.4 Startup of fine kinematics

#### General information

##### Input screen form for fine offsets of offset vectors for swiveling (NCU 7.1 and higher):

In the "Fine kinematics" startup screen, fine offsets can be entered for the vectors of machine kinematics I1 to I4 or the offset values of the rotary axes. The display and entry of fine offsets of offset vectors takes place in a separate display in the "Start up swivel" area. You use vertical soft key Kinematics, Rotary axes, Fine kinematics to enter this screen. The fine offsets are activated by the following setting data: SD 42974: \$SC\_TOCARR\_FINE\_CORRECTION = 1.

The fine offsets act in addition to the corresponding base vectors when the function "swiveling" CYCLE800 or the NC function TCARR=n is called.

Assignment of base vectors to the fine offset vectors:

- I1 \$TC\_CARR1..3[n] → \$TC\_CARR41..43[n]
- I2 \$TC\_CARR4..6[n] → \$TC\_CARR44..46[n]

- I3 \$TC\_CARR15..17[n] → \$TC\_CARR55..57[n]
- I4 \$TC\_CARR18..20[n] → \$TC\_CARR58..60[n]

Offset vectors of rotary axes

- \$TC\_CARR24..25[n] → \$TC\_CARR64..65[n]

n...number of swivel data record

## Application

The fine offsets can be applied for temperature compensation of machine kinematics. For this purpose, the fine offsets can be written, activated, and deactivated in a corresponding manufacturer cycle.

### 3.16.7.5 Startup examples for machine kinematics

#### Example 1: Swivel head 1 "HEAD\_1"

- Rotary axis 1(C) (manual)  
about Z
- Rotary axis 2(A) (manual)  
about X
- Manually adjustable changeable  
swivel head (manual)

Vectors relate to the kinematics  
basic setting (drawing not true-to-  
scale)

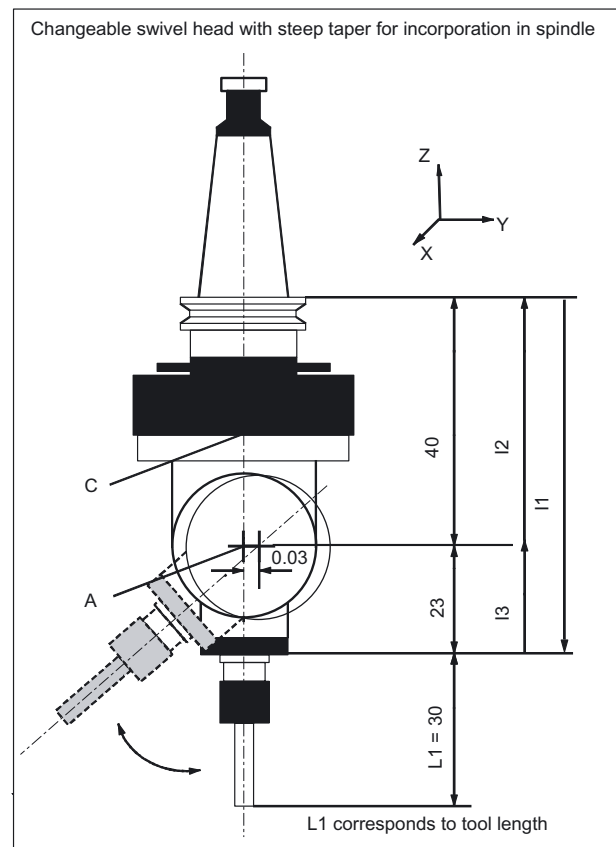


Table 3-2 Startup SK Swivel, Kinematics (Example 1)

Kinematics	Swivel head		HEAD_1
Retract	Z		
	X	Y	Z
			200.000
Offset vector I1	0.000	0.030	-63.000
Rotary axis vector V1	0.000	0.000	1.000
Offset vector I2	0.000	0.000	40.000
Rotary axis vector V2	1.000	0.000	0.000
Offset vector I3	0.000	-0.030	23.000
Display options			
Swivel mode	Axis by axis		
Direction	Rotary axis 2		
Correct tool	No		
<b>Rotary axes</b>			
Rotary axis 1	C	Mode	Manual
Angular range	0.000		360.000
Rotary axis 2	A	Mode	Manual
Angular range	-15.000		100.000

## Example 2: Swivel head 2 "HEAD\_2"

- Rotary axis vector V1: Rotary axis B rotates about Y  
 Rotary axis vector V2: Rotary axis C rotates around Y **and** around Z  
 Offset vector I1: Closure of vector chain with fixed-mounted swivel head  
 $I1 = -(I2 + I3)$   
 Offset vector I2: Distance between pivot point of **rotary axis 1** and pivot point of **rotary axis 2**  
 Offset vector I3: Distance between reference point of tool and pivot point of **rotary axis 2**

Cardanic swivel head (manually adjustable) with Hirth tooth system

**Vectors relate to the basic setting of the kinematics.**

If the swivel head is fixed-mounted, the vector chain is closed (see I1)

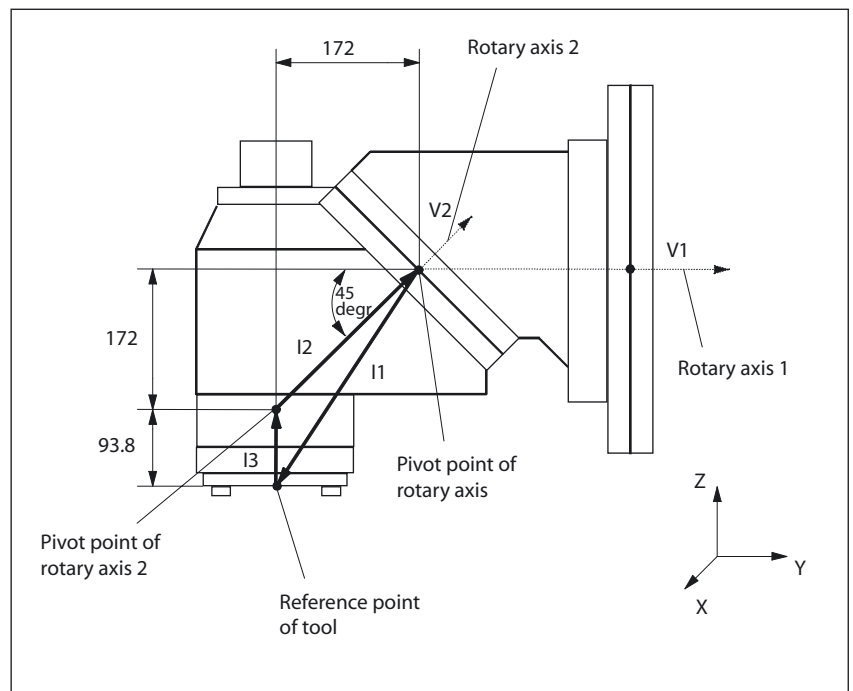


Table 3-3 Startup SK Swivel, Kinematics (Example 2)

Kinematics	Swivel head		HEAD_2
Retract	Z	Tool direction	Max.+inc.
	X	Y	Z
			200.000
Offset vector I1	0.000	-172.000	-265.800
Rotary axis vector V1	0.000	1.000	0.000
Offset vector I2	0.000	172.000	172.000
Rotary axis vector V2	0.000	1.000 <sup>1)</sup>	1.000 <sup>1)</sup>
Offset vector I3	0.000	0.000	93.800

Kinematics	Swivel head		HEAD_2
Display options			
Swivel mode	Axis by axis		
Direction	Rotary axis 2		
<b>Rotary axes</b>			
Rotary axis 1	B	Mode	Manual
Angular range	0.000		360.000
Kinematics offset	0.000		
Hirth tooth system	Yes	Angular grid	1.000
Rotary axis 2	C	Mode	Manual
Angular range	0.000		180.000
Kinematics offset	0.000		

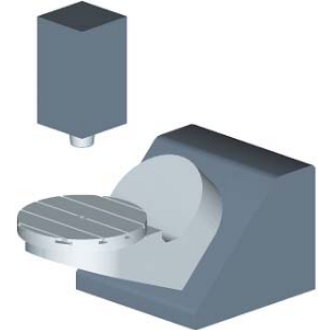
The reference point of the pivot point of rotary axes 1 and 2 can be offset on the line of rotation and does not have to coincide with the mechanical pivot point.

- 1) Calculation of rotary axis vector V2: 45 degree angle  
 $V2Y = \sin(45) = 0.7071$   
 $V2z = \cos(45) = 0.7071$   
V2Y and V2z can be normalized to 1.

### Example 3: Cardanic table "TABLE\_45"

Vectors relate to the basic setting of the kinematics.

- Rotary axis vector V1: Rotary axis B rotates around Y and around Z.  
 Rotary axis vector V2: Rotary axis C rotates about Z.  
 Offset vector I2: Distance from the reference point of the machine to the pivot point/intersection of **rotary axis 1**  
 Offset vector I3: Distance between pivot point/intersection of **rotary axis 1** and pivot point/intersection of **rotary axis 2**  
 Offset vector I4: Closure of vector chain  $I4 = -(I2 + I3)$



#### Side view of the machine

Spindle (tool adapter) is positioned on a block dimension above the top edge of the table (rotary axis C) or the center of the table.

A measuring rod in the spindle is used to determine the turning center of rotary axis C.

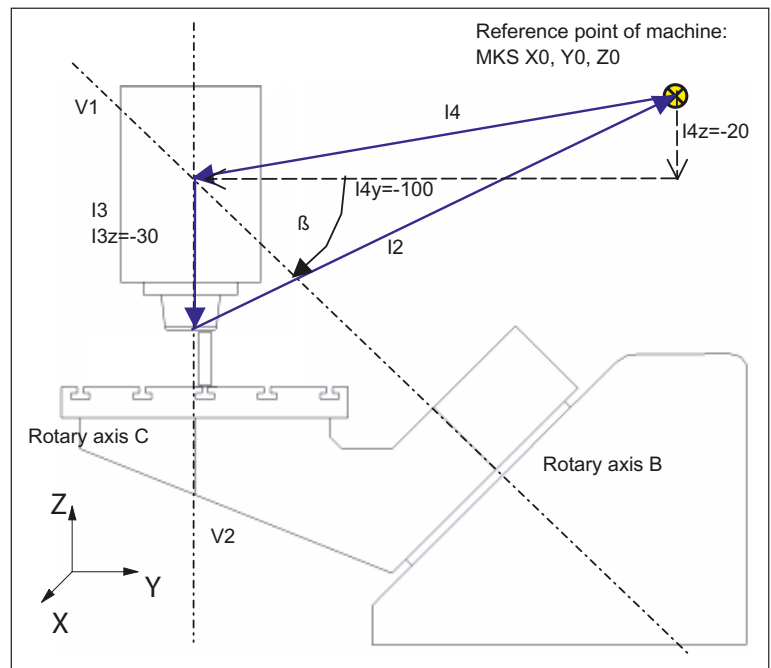


Table 3-4 Startup SK Swivel, Kinematics (Example 3)

Kinematics	Swivel head		TABLE_45
	X	Y	Z
Offset vector I2	0.000	100.000	50.000
Rotary axis vector V1	0.000	-1.000 <sup>1)</sup>	1.000 <sup>1)</sup>
Offset vector I3	0.000	0.000	-30.000
Rotary axis vector V2	0.000	0.000	-1.000
Offset vector I4	0.000	-100.000	-20.000

Kinematics	Swivel head		TABLE_45
Display options			
Swivel mode	Axis by axis		
Direction	Rotary axis 2		
Correct tool	nein		
<b>Rotary axes</b>			
Rotary axis 1	B	Mode	Auto
Angular range	0.000		180.000
Rotary axis 2	C	Mode	Auto
Angular range	0.000		360.000

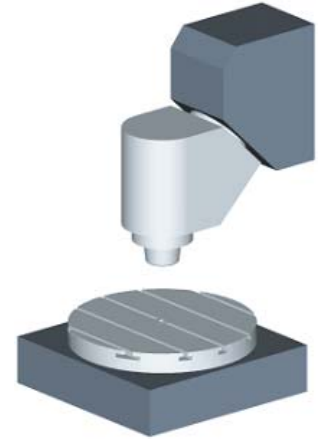
- 1) Calculation of rotary axis vector V1:  $\beta = -45$  degrees  
 $V1Y = \sin(-45) = -0.7071$   
 $V1z = \cos(-45) = 0.7071$   
 $V1Y$  and  $V1z$  can be normalized to -1 and 1.



#### Example 4: Swivel head/rotary table "MIXED\_45"

Vectors relate to the basic setting of the kinematics.

Rotary axis vector V1:	Rotary axis B rotates around Y and around Z.
Rotary axis vector V2:	Rotary axis C rotates about Z.
Offset vector I2:	Distance from the reference point of the tool adapter to the pivot point/intersection of <b>rotary axis 1</b>
Offset vector I1:	Closure of vector chain I1=-I2
Offset vector I3:	Distance from the reference point of the machine to the pivot point/intersection of <b>rotary axis 2</b>
Offset vector I4:	Closure of vector chain I4=-I3



Side view of the machine:  
Spindle (tool adapter) is positioned on a block dimension above the top edge of the table (rotary axis C) or the center of the table.  
A measuring rod in the spindle is used to determine the turning center of rotary axis C.

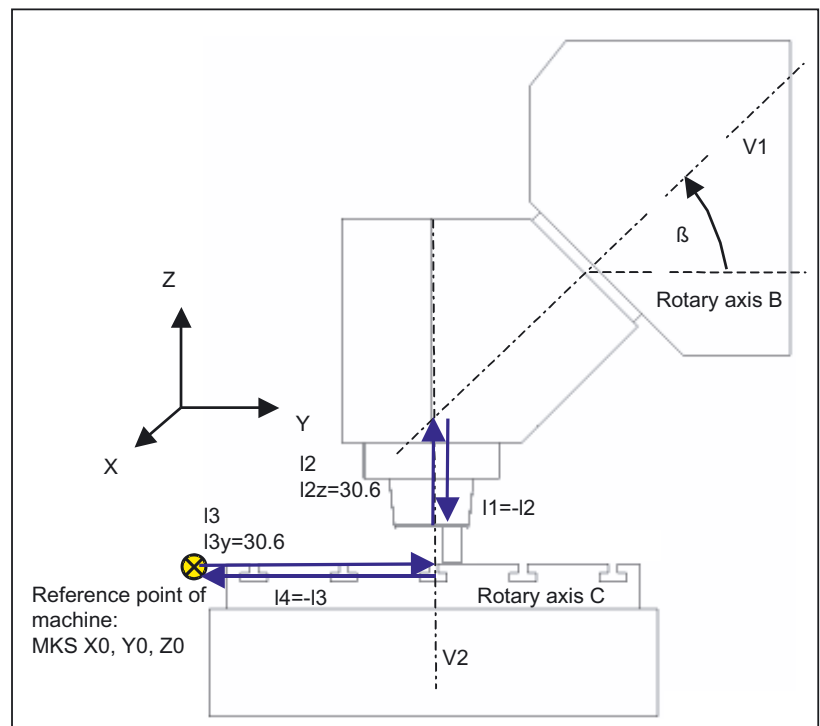


Table 3-5 Startup SK Swivel, Kinematics (Example 4)

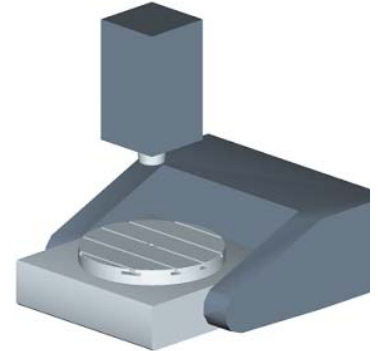
Kinematics	Swivel head		MIXED_45
	X	Y	Z
Offset vector I1	0.000	0.000	-30.600
Rotary axis vector V1	0.000	1.000 <sup>1)</sup>	1.000 <sup>1)</sup>
Offset vector I2	0.000	0.000	30.600
Offset vector I3	300.000	150.000	0.000
Rotary axis vector V2	0.000	0.000	-1.000
Offset vector I4	-300.000	-150.000	0.000
Display options			
Swivel mode	Axis by axis		
Direction	Rotary axis 1		
Correct tool	ja		
<b>Rotary axes</b>			
Rotary axis 1	B	Mode	Auto
Angular range	0.000		180.000
Rotary axis 2	C	Mode	Auto
Angular range	0.000		360.000

- 1) Calculation of rotary axis vector V1:  $\beta = 45$  degrees  
 $V1Y = \sin(45) = -0.7071$   
 $V1z = \cos(45) = 0.7071$   
 $V1Y$  and  $V1z$  can be normalized to 1.

### Example 5: Swivel table "TABLE\_5"

Vectors relate to the basic setting of the kinematics.

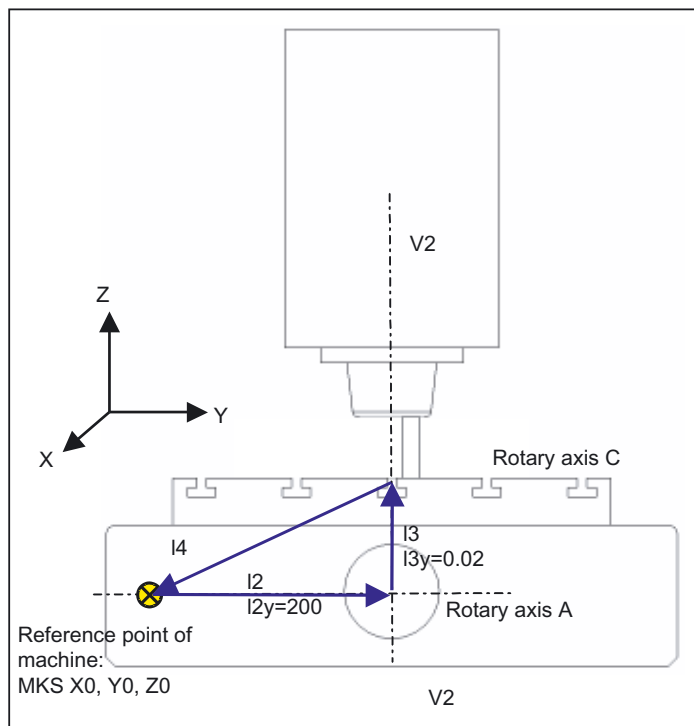
Rotary axis vector V1:	Rotary axis A rotates about X.
Rotary axis vector V2:	Rotary axis C rotates about Z.
Offset vector I2:	Distance from the reference point of the machine to the pivot point/intersection of <b>rotary axis 1</b>
Offset vector I3:	Distance from the pivot point of rotary axis 1 to the pivot point/intersection of <b>rotary axis 2</b>
Offset vector I4:	Closure of vector chain $I4 = -(I2 + I3)$



#### Side view of the machine from the X direction

Spindle (tool adapter) is positioned on a block dimension above the top edge of the table (rotary axis C) or the center of the table.

A measuring rod in the spindle is used to determine the turning center of rotary axis C.



Side view of the machine from the Y direction

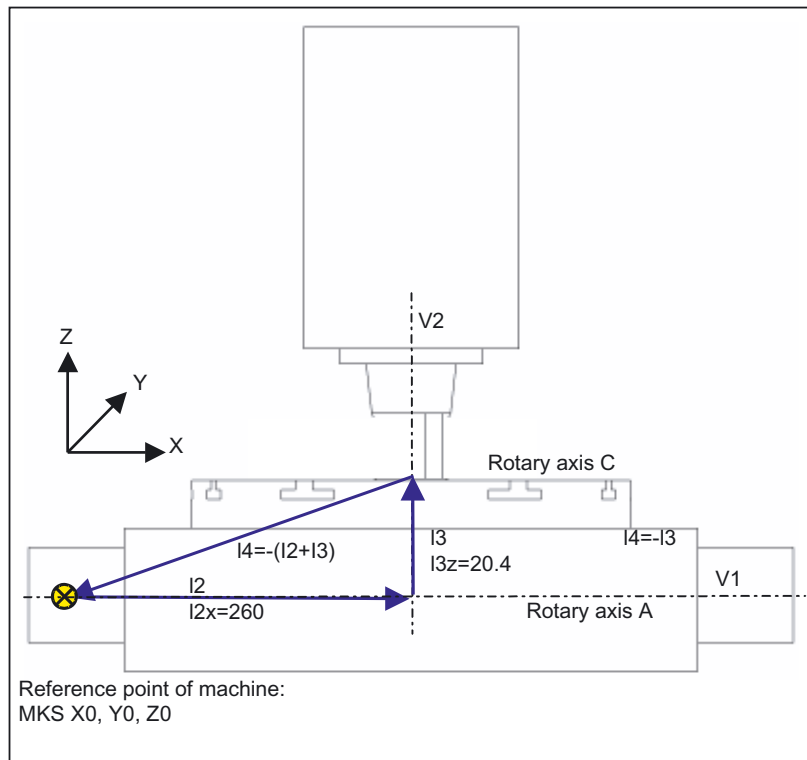


Table 3-6 Startup SK Swivel, Kinematics (Example 5)

Kinematics	Swivel head		TABLE_5
	X	Y	Z
Offset vector I2	260.000	200.000	0.000
Rotary axis vector V1	-1.000	0.000	0.000
Offset vector I3	0.000	0.020	20.400
Rotary axis vector V2	0.000	0.000	-1.000
Offset vector I4	-260.000	-200.020	-20.400
Display options			
Swivel mode	Axis by axis		
Direction	Rotary axis 1		
Correct tool	No		
<b>Rotary axes</b>			
Rotary axis 1	A	Mode	Auto
Angular range	-90.000		90.000
Rotary axis 2	C	Mode	Auto
Angular range	0.000		360.000

### 3.16.8 Manufacturer Cycle TOOLCARR.SPF - CYCLE800

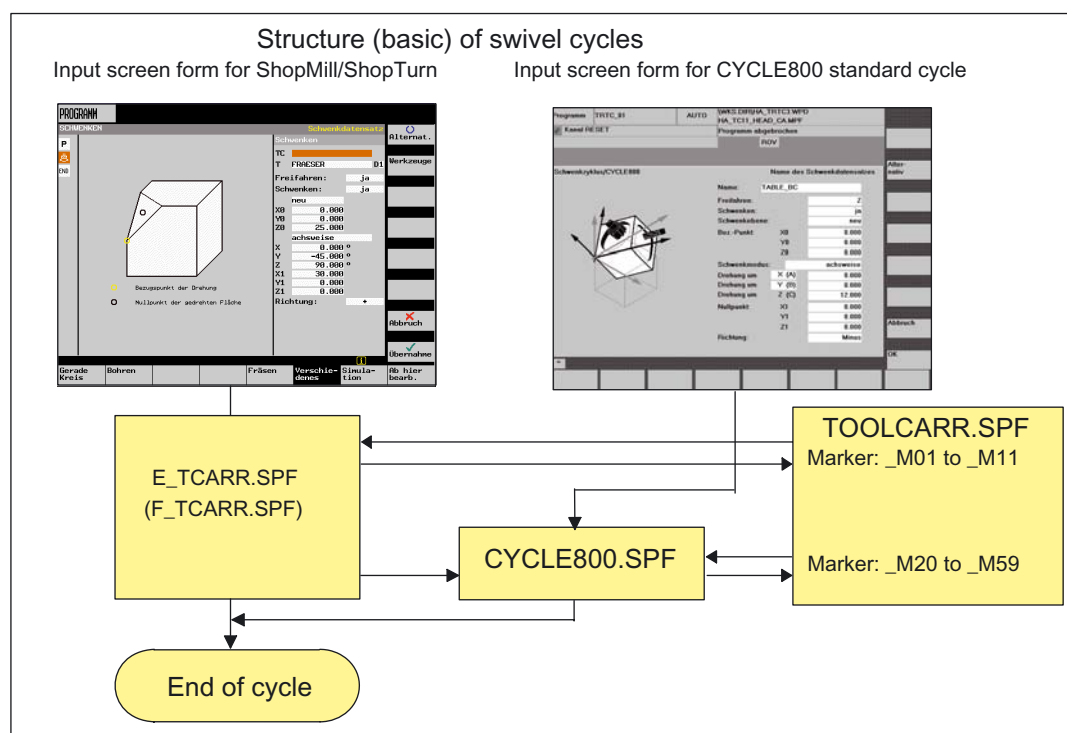
#### Function

##### Customization by the machine manufacturer

During swiveling, all axis positions are traversed using the TOOLCARR.SPF cycle. This is always called from the CYCLE800 swivel cycle or E\_TCARR (ShopMill) or F\_TCARR (ShopTurn). The cycle can be modified by the machine manufacturer during startup, in order to cater to the properties of specific machines.

If the cycle is not modified, the Z axis (marker \_M41) or the Z axis followed by the X, Y axes (marker \_M42) is first moved when retracting prior to swiveling. Retraction can also take place in the tool direction with swiveled tool axes (see marker \_M44, \_M45).

The positions correspond to the CYCLE800 "Kinematics" ⇒ retraction positions startup menu.

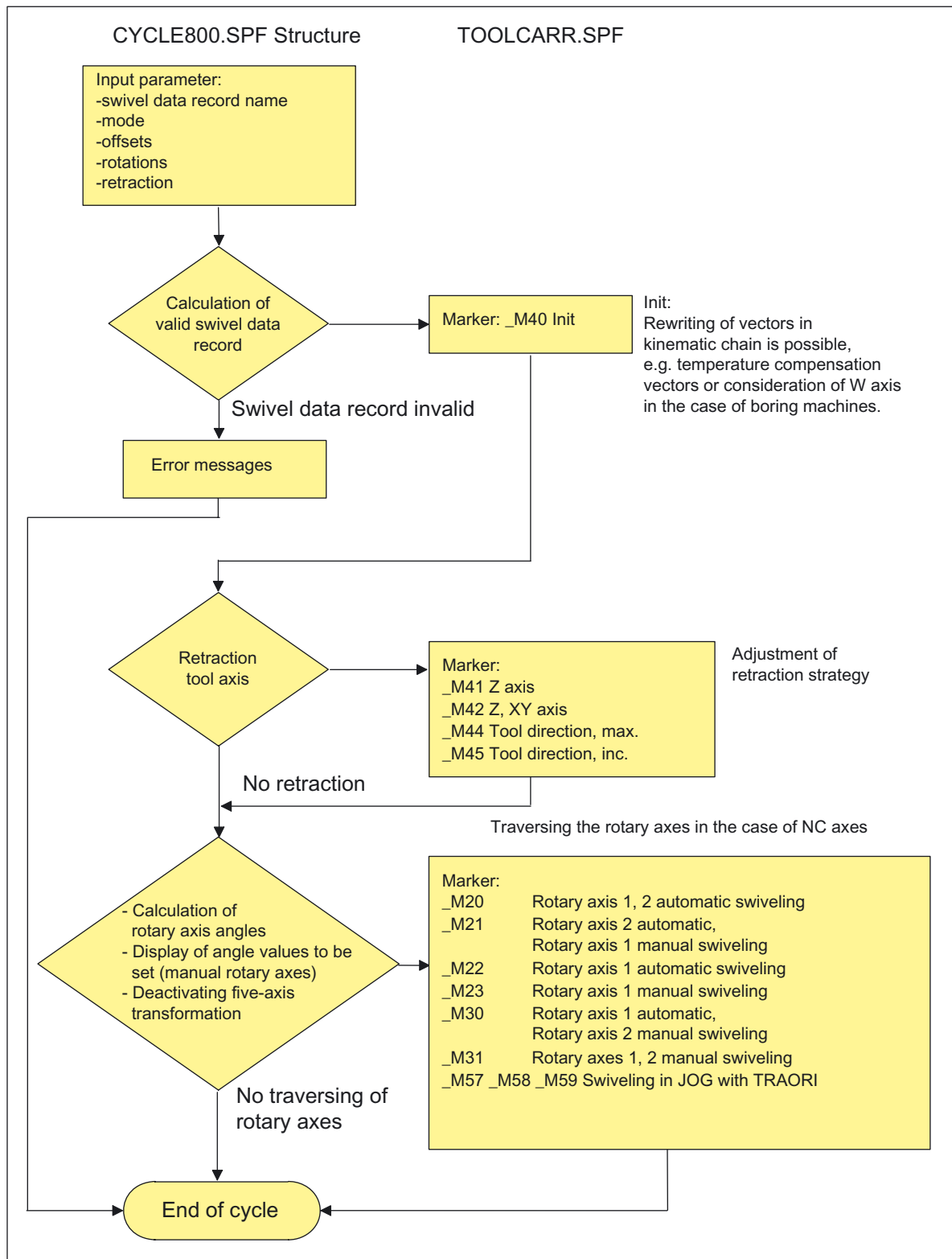


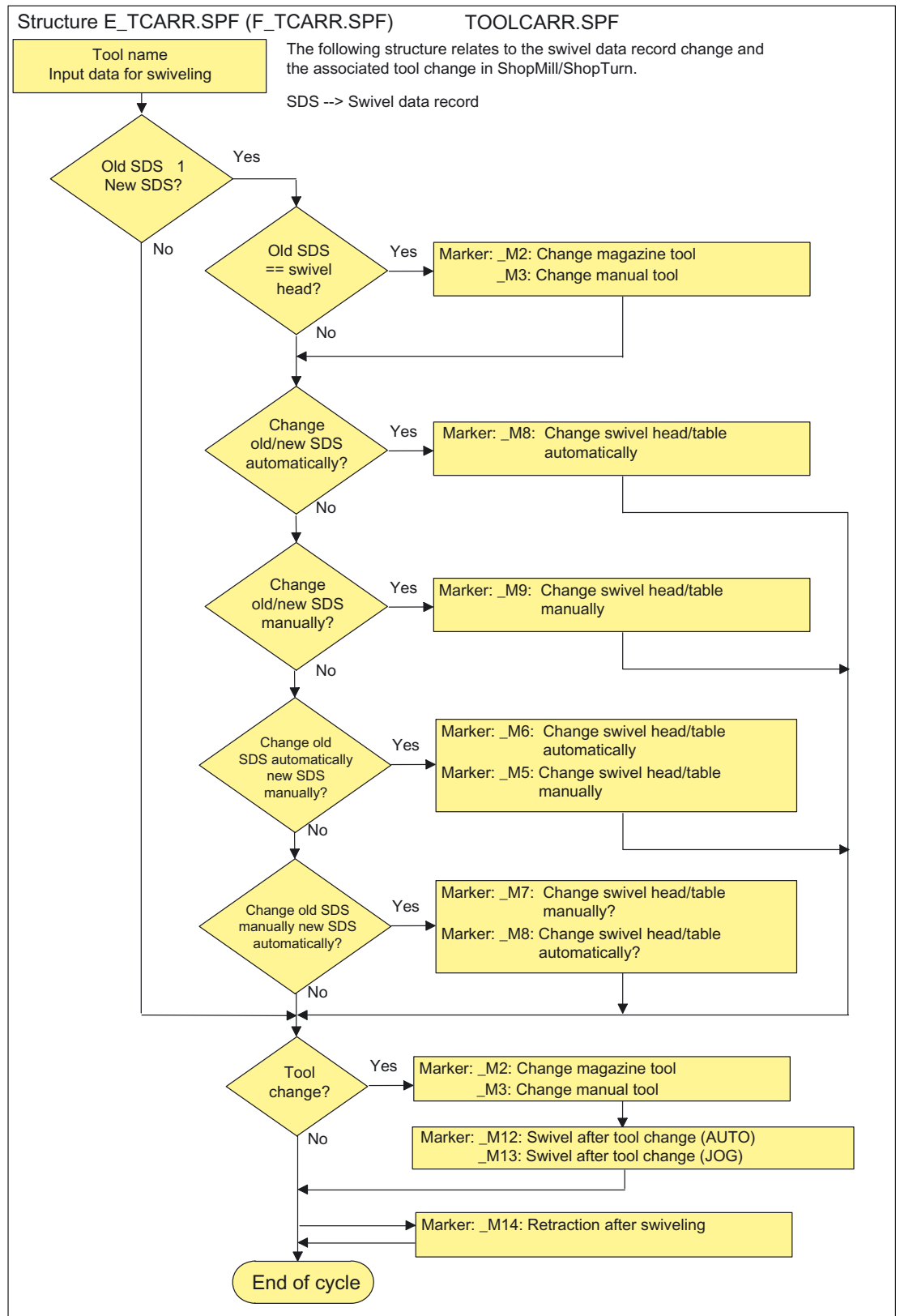
#### Note

The parameters are only valid in conjunction with the relevant jump label in TOOLCARR ⇒ see the program structure below.

If INCH is the basic system of the machine, the TOOLCARR cycle must be modified accordingly.

Note the machine manufacturer's instructions.





**Notes**

- **For Marker \_M20 to \_M31**

Markers \_M20 to \_M31 are distinguished by kinematics with two rotary axes or one rotary axis. A distinction is also made between automatic rotary axes (known to the NCU) and manual rotary axes.

There is only ever one valid marker for the active swivel data record. Checks are carried out via the \_TC\_ST parameter/GUD7 variable.

- **For ShopMill/ShopTurn markers \_M2 to \_M13**

When the swivel data record or tool is changed, the linear axes are recovered using the last retraction mode (modal). If this behavior is not desired in conjunction with ShopMill/ShopTurn, the corresponding calls must be converted to comment with a semicolon (;).

The E\_SWIV\_H or F\_SWIV\_H cycle is called in ShopMill/ShopTurn (see markers \_M2 to \_M9) in the TOOLCARR.SPF manufacturer cycle.

Parameter E\_SWIV\_H (Par1, Par2, Par3)

- Par1: Number of swivel data record (\_TC1)
- Par2: Angle of 1st rotary axis
- Par3: Angle of 2nd rotary axis

Modification examples:

If the rotary axes (swivel head/table) are **not to be positioned** during swivel data change/tool change, the call of the E\_SWIV\_H cycle can be commented out at the relevant markers.

If the rotary axes are to move to a certain position, an angle value can be transferred to parameters Par 2, Par 3.

- **For ShopMill/ShopTurn markers \_M14, \_M15**

Depending on the values of the retraction plane and the programmed swivel plane, it is possible that the linear axes now also travel the swiveled retraction plane while running up from the current position to the software limit switches after a block search.

To avoid this problem, marker \_M14 in the TOOLCARR.SPF is called after swiveling. The E\_SP\_RP(30) cycle preset there runs up to the ShopMill retraction plane, whereby travel may be along the software limit switches. An appropriate retraction after block search can be set at marker \_M15.

- **For ShopMill/ShopTurn marker \_M46**

Retraction before swiveling after a block search can be set at marker \_M46. Variable \_E\_VER is 1, if it is a ShopMill program.

- **For "Correct tool"**

"Correct tool" requires that a 5-axis transformation is set up, which is equivalent to the corresponding swivel data record.

The programming section is integrated into marker \_M20 in the TOOLCARR.SPF manufacturer cycle.

The relevant program lines can be activated by commenting them out.



- **For tool change + swivel**

In general, the swivel (CYCLE800) and tool change functions for a machine are independent of each other. Thus, the swiveled work plane can be retained in a technological sequence with multiple tools (e.g. centering, drilling, tapping).

If the rotary axes of the active swivel data record are involved in the mechanical sequence of the tool change or these rotary axes have to be retracted, the tool change sequence program must ensure that the rotary axis positions before the tool change are approached again after the tool change. If linear axes (geometry axes) are also involved in the tool change, the rotations in the NC (swivel frame) must not be deleted. Rather, the linear axes can be positioned as machine axes during the tool change using the G153 or SUPA commands.

- **For swiveling without active tool compensation**

If the user wants to prevent swiveling or rotary axes without active tool cutting edge (D0), the machine manufacturer can adapt this in the TOOLCARR.SPF cycle.

Example with syntax as message:

```
_M40:
IF ((NOT $P_TOOL) AND _TC1)
LOOP
MSG ("no tool cutting edge active")
M0
STOPRE
ENDLOOP
ENDIF
GOTOF _MEND
```

## Parameter

TOOLCARR(\_MODE, \_TC1, \_A1, \_A2, \_TC2)

Parameter	Data type	Meaning
_MODE		Corresponds to the markers of the described structure
_TC1		Number of the swivel head/table
_A1		Angle of the 1st rotary axis
_A2		Angle of the 2nd rotary axis (if present)
_TC2		Feedrate reduction when Swiveling in JOG (with Cycles SW 6.5 and higher)

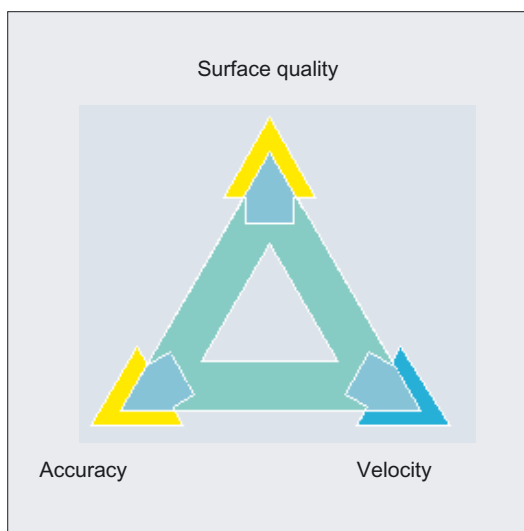
## 3.17 High speed settings - CYCLE832

### 3.17.1 General information

The standard high-speed settings cycle CYCLE832 is available for the HMI with SW 6.3 and higher and NCU SW 6.3 (CCU SW 4.3).

#### Application of CYCLE832:

- For technological support when machining free-form contours (surfaces) in the 3 or 5axis high-speed machining range (High Speed Cutting - HSC).
- Predominantly used in the HSC milling range (possibility of use for turning or grinding)
- Groups together the important G codes and machine and setting data that are needed for HSC machining
- Separation of technology/geometry using a relevant NC program architecture



When executing CAM programs in the HSC range, the control has to process high feedrates with the shortest NC blocks. The user expects a good **surface finish quality** with high **accuracy** in the  $\mu\text{m}$  range and with extremely high **machining feedrates**  $>10$  m/min. By applying different machining strategies, the user can fine tune the program with the aid of CYCLE832.

When **roughing**, the emphasis is on **speed** due to the smoothing of the contour.

When **finishing**, the emphasis is on **accuracy**.

In both cases, specifying a tolerance ensures that the machining contour is observed in order to achieve the desired surface finish.

When defining the tolerance values for smoothing the contour, the operator must have precise knowledge of the subsequent CAM program. The CYCLE832 cycle supports machine types where a maximum of three linear and two rotary axes are involved in machining.

## Function

The following functions can be defined or enabled/disabled by CYCLE832:

- 4 types of machining: Finishing, rough-finishing, roughing, deselection (default setting)
- Tolerance band of the contour to be machined
- Smoothing (G64, G641, G642)
- NC block compressor (COMPCAD, COMPCURV, COMPOF)<sup>1)</sup>
- Feedforward control (FFWON, FFWOF)
- Jerk limitation (SOFT, BRISK)
- 5-axis transformation (TRAORI, TRAFOOF)<sup>1)</sup>
- B SPLINE<sup>1)</sup>

1) Only if the relevant option is set.

The CYCLE832 cycle groups together the important G codes and the machine or setting data required for HSC machining.

## Machining modes

In CYCLE832, a distinction is made between four technological operations:

- "Finishing"
- "Rough-finishing"
- "Roughing"
- "Deselection" (default setting)

In CAM programs in the HSC range, the four machining types directly relate to the accuracy and speed of the path contour (see help display). The operator/programmer uses the **tolerance value** to give a corresponding weighting.

Different tolerances and settings (customizing technology) can be assigned to the four machining types.

## Screen form

The relevant G codes are preset (customizing technology) in the input screen form, in order to ensure a smoothing path contour or that the CAM program is executed at optimum speed.

The cycle precedes the CAM program in the main menu (see typical CYCLE832 call).

Different interpretations of the tolerance values are taken into account. For example, for G641, the tolerance value is transferred as ADIS= and for G642, the axis-specific MD 33100 COMPRESS\_POS\_TOL[AX] is updated.

When activating the "Customize technology" input field, you can enable or disable (keyswitch position  $\geq 2$ ):

- Compression (COMPCAD, COMPCURV, COMPOF, B-SPLINE),
- Continuous-path mode (G64, G641, G642) or
- Velocity control (FFWON, FFWOF, SOFT, BRISK).

If 5-axis transformation (TRAORI) is set up, this can be enabled/disabled in the transformation input field.

<b>NOTICE</b>
Never ignore information provided by the machine manufacturer!

## 3.17.2 Programming via input screen form

### 3.17.2.1 General information

#### Call of CYCLE 832 in HMI menu tree

Entry to Programs/**Milling** area

Soft key



is displayed.

#### CYCLE832 input screen form on standard interface

High Speed Settings		Machining selection	
	Machining	Finishing	
	Tolerance	_TOL	0.005
	Transformation		TRAORI
	Adjustment		Yes
	Compression		COMPCAD
	Path control		G642
	Feedforward control		FFWON SOFT

### 3.17.2.2 Parameters of input screen form

#### Machining (\_TOLM)

- Finishing (default)
- Rough-finishing
- Roughing
- Deselection

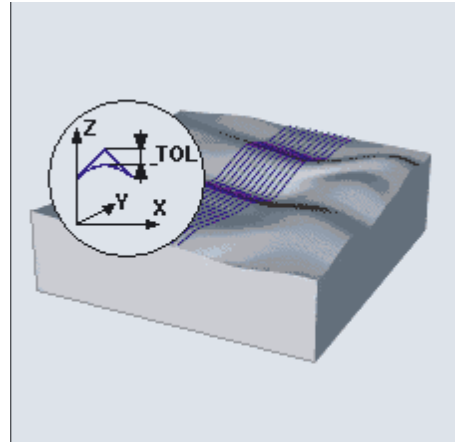
Variable \_TOLM Coding see Parameter Description for CYCLE832.

## Tolerance (\_TOL)

This refers to the tolerance of axes involved in machining. The tolerance value is written to the relevant machine or setting data depending on the G codes (G642, COMPCAD, COMPCURV, etc.).

If the machining axis is a rotary axis, the tolerance value is written to the relevant machine or setting data of the rotary axis with a factor (default factor = 8).

With G641, the tolerance value corresponds to the ADIS value.



On initial entry, the following values are set for the tolerance:

- **Finishing:** 0.01 (linear axes) 0.08 deg (rotary axes)
- **Rough-finishing:** 0.05 (linear axes) 0.4 deg (rotary axes)
- **Roughing:** 0.1 (linear axes) 0.8 deg (rotary axes)
- **Deselection:** 0.01 (linear axes) 0.1 deg (rotary axes)

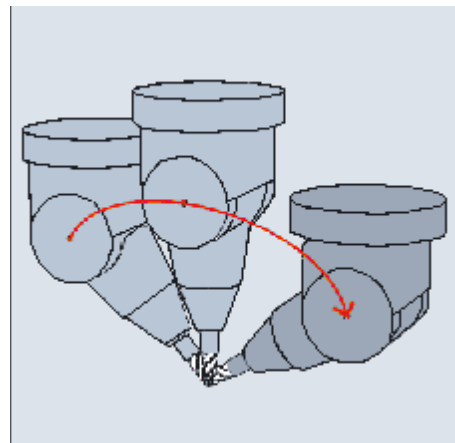
The mm/inch system of units is taken into account.

If the tolerance value is also to be effective for rotary axes, 5-axis transformation must be set up by the machine manufacturer but not activated in every case, e.g., machining of CAM programs with resolved rotary axes.

## Transformation (\_TOLM)

The transformation input field is only displayed when the NC option is set (5 axes machining package set).

- **No**
- **TRAORI** → Enable first 5-axis transformation
- **TRAORI(2)** → Enable second 5-axis transformation



Selecting the transformation number or the manufacturer cycle to call the 5-axis transformation.

- It is possible to store the name of a manufacturer cycle, which calls the manufacturer cycle of the transformation, in the \_TOLT2 GUD7 variables. If \_TOLT2 is empty ("" default), selecting transformation 1,2... calls the 5-axis transformation with TRAORI(1) or TRAORI(2).

- To start a 5-axis transformation program to a swiveled plane (see CYCLE800), the toolholder is deleted and the WPFRAME swivel frame (workpiece reference) is used by TRAOR after activation.

### Customization, Customizing technology

→ see Section "Customizing Technology"

- **Yes**
- **No**

The subsequent input parameters can only be changed if customization is set to "Yes".

The "Customization" input field and the subsequent "Compression", "Path control", and "Feedforward control" input fields are not shown if the keyswitch is in position 0 or 1.

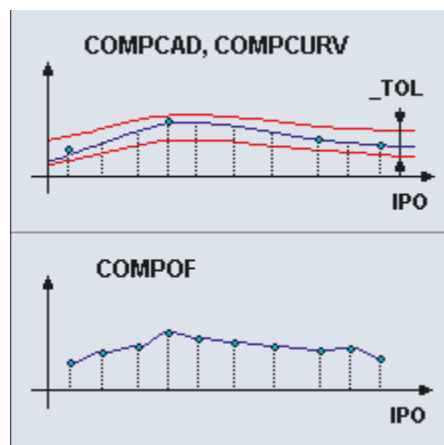
### Compression, NC block compressor (\_TOLM)

- **COMPOF** (default)
- **COMPCAD**
- **COMPCURV**
- **B SPLINE**

The input field is only displayed if the compressor function option is set.

B spline is only selected if the spline interpolation option is set.

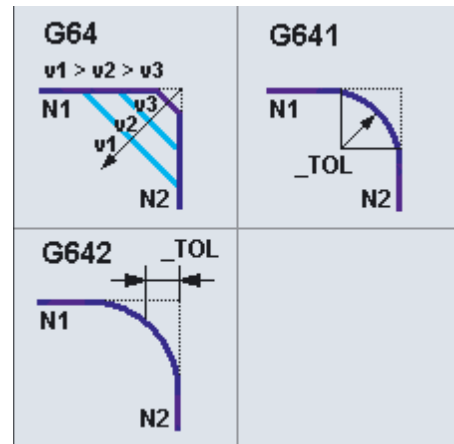
Option → A, B and C splines/compressor function



## Path control (\_TOLM)

- **G642** (default)
- **G641**
- **G64**

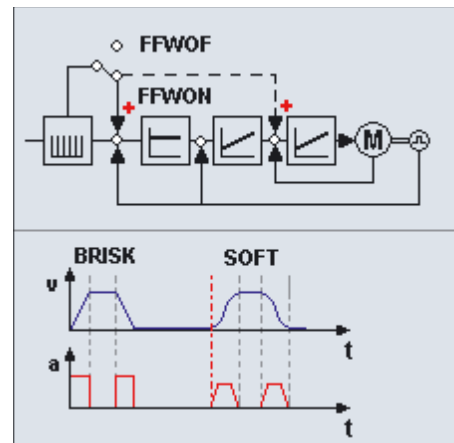
With an NC block compressor with COMPCAD, COMPCURV, G642 is permanently selected.



## Feedforward control, velocity control (\_TOLM)

- **FFWOF SOFT** (default)
- **FFWON SOFT**
- **FFWOF BRISK**

The selection of feedforward control (FFWON) and jerk limitation (SOFT) requires that the machine manufacturer has optimized the control or the machining axes.





### 3.17.3 Programming via parameters

#### Programming

CYCLE832(\_TOL, \_TOLM)

#### Note

CYCLE832 does not absolve the machine manufacturer from performing the required optimization tasks when starting up the machine. This involves optimization of the axes involved in machining and the NCU settings (feedforward control, jerk limitation, etc.).

#### Parameter

Parameter	Data type	Meaning
_TOL	real	Tolerance of machining axes ⇒ unit mm/inch; degrees
_TOLM	integer	<p>Tolerance mode</p> <p>7 6 5 4 3 2 1 0 (decimal places)</p> <p>0: Deselection 1: Finishing (default) <sup>1)</sup> 2: Rough-finishing 3: Roughing</p> <hr/> <p>0: 1: 0: TRAFOOF (default) <sup>1)</sup> 1: TRAORI(1) 2: TRAORI(2)</p> <hr/> <p>0: G64 1: G641 2: G642 (default) <sup>1)</sup></p> <hr/> <p>0: FFWOF SOFT (default) <sup>1)</sup> 1: FFWON SOFT 2: FFWOF BRISK</p> <hr/> <p>0: COMPOF (default) <sup>1)</sup> 1: COMPCAD 2: COMPCURV 3: B spline</p> <hr/> <p>Reserved</p> <hr/> <p>Reserved</p> <p><sup>1)</sup> Setting can be changed by the machine manufacturer, see Section "Customizing Technology"</p>

**Example of CYCLE832 call**

```

T1 D1
G54
M3 S12000
CYCLE832(0.20,1003)                ;Roughing
EXTCALL "CAM_Form_Schrupp"
CYCLE832(0.01,102001)              ;Finishing
EXTCALL "CAM_Form_Schlicht"
CYCLE832(0.1,0)                    ;Deselection (default setting)
M02

```

**Reduced program call**

The following options to call CYCLE832 with a reduced parameter transfer are available:

- **CYCLE832 ( )**  
Corresponds to the selection of the "Machining" "Deselection" input screen form.  
The G codes used in CYCLE832 (see CYCLE832 "Interfaces") are set to the value set in MD 20150: GCODE\_RESET\_VALUE.
- **CYCLE832 (0.01)**  
Entering the tolerance value.  
The active G commands are not changed in the cycle.

**3.17.4 Customizing technology****3.17.4.1 General information**

With the "Customizing technology" input field set to "Yes", both the machine manufacturer and the machine setter/programmer can customize the technology for HSC machining.

Always take note of the technology of the subsequent CAM program.

**3.17.4.2 Customizing of machine setter/programmer****Conditions**

- **The manufacturer password is deleted**
- Keyswitch position 2 or 3 or protection level 5, 4, 3, 2
- "Customizing technology" input variable → "Yes"

### Customizing technology

In order to customize the technology, the machine setter/programmer must have precise knowledge of the subsequent CAM machining program.

The modified data are used to generate CYCLE832 and applies to the current call of CYCLE832.

The machine manufacturer's default settings are not changed.

#### 3.17.4.3 Customization by the machine manufacturer

##### Conditions

- The manufacturer password is set
- "Customizing technology" input field → "Yes"

##### Customizing technology

When the CYCLE832 input screen form is opened, the parameters are preassigned with the values of the \_TOLV[n], \_TOLT[n] GUD7 variables.

n → machining: finishing, rough-finishing, roughing, deselection

When the parameters are changed, the values are written directly to the \_TOLV[n] or \_TOLT[n] GUD7 variables.

This gives the machine manufacturer the opportunity to customize the default settings to his machining tasks.

##### Example:

Calling cycle CYCL832 roughing with three axes, machining axes tolerance 0.1mm with G642 (Siemens default values).

The machine manufacturer can modify the roughing technology with setting:

machining axes tolerance 0.3 mm, TRAORI, G641.

Each time the tolerance cycle is called, this setting is then displayed and takes effect during the execution.

##### Notes for the machine manufacturer

- In order to optimize the path control response when traversing with G64, the velocity jumps overload factor is recalculated in CYCLE832, in accordance with the following table:

Calculation of the velocity jumps overload factor for all machining axes

IPO [msec]	Overload factor
≥ 12	1.2
9	1.3
6	1.4
4	1.6
3	1.8

IPO: MD 10071: \$MN\_IPO\_CYCLE\_TIME

Overload factor MD 32310: \$MA\_MAX\_ACCEL\_OVL\_FACTOR[AX]

Computation of the overload factor by CYCLE832 can be disabled by setting the local variable **\_OVL\_on=0** in CYC\_832T.

- If the NC block compressor (COMPCAP) or smoothing (G642) is active, the tolerance is written in CYCLE832 to the MD 33100: \$MA\_COMPRESS\_POS\_TOL[AX] (linear machining axes). If rotary axes are involved in machining (TRAORI), this tolerance is written with factor 8 to MD 33100: \$MA\_COMPRESS\_POS\_TOL[AX] of the rotary axes. If a different factor is to be used, the CYC\_832T FACTOR local variable can be preassigned with the relevant value.

#### 3.17.4.4 Customizing additional program parameters CYC\_832T

##### General information

If the machine manufacturer requires additional responses from CYCLE832 beyond those, which can be achieved by customizing the technology, it is possible to make the necessary changes in the CYC\_832T cycle. For this purpose, CYC\_832T should be copied into the CMA directory (HMI manufacturer) and loaded into the NCU.

The CYC\_832T functions as a frame program. The machine manufacturer must document any changes.

If it is loaded into the NCU, the CYC\_832T cycle is automatically called by CYCLE832.

Customizing CYC\_832T does not mean that changes to CYCLE832 are required.

##### Parameter

CYC\_832T(\_ASVS, \_FACTOR, \_OVL\_on)

Parameter	Data type	Meaning
_ASVS		Machine-specific customization is possible at markers _M0 to _M4.
		Markers: _M4 = Init CYCLE832 call made before the actual program sequence of CYCLE832
		_M0 = Deselection of CYCLE832
		_M1 = Finishing machining
		_M2 = Rough-finishing machining
		_M3 = Roughing machining
_FACTOR <sup>1)</sup>		Rotary axis tolerance factor
_OVL_on <sup>1)</sup>		0 = Do not customize overload factor MD \$MA_MAX_ACCEL_OVL_FACTOR
1) The _FACTOR and _OVL_on parameters are only effective at marker _M4(INIT).		

### Example of customizing

The machine manufacturer wants to perform the following customization:

1. The rotary axis tolerance is to be 12 times higher than the tolerances of the linear axes.
2. In the "Finishing", "Rough-finishing" and "Roughing" machining modes, the path jerk (MD \$MC\_MAX\_PATH\_JERK) is defined as value 15 and the axis jerk (MD \$MA\_MAX\_AX\_JERK[AX]) as value 150.
3. On deselection of CYCLE832, the changed machine data must be reset to the default settings (1000.15).

### Example of customizing technology

The machine manufacturer wants to perform the following customization:

1. The rotary axis tolerance is to be 12 times higher than the tolerances of the linear axes.  
The rotary axis factor is to refer always to the metric system.
2. In the "Finishing", "Rough-finishing" and "Roughing" machining modes, the path jerk (MD \$MC\_MAX\_PATH\_JERK) is defined as value 15 and the axis jerk (MD \$MA\_MAX\_AX\_JERK[AX]) as value 150.
3. On deselection of CYCLE832, the changed machine data must be reset to the default settings (1000.15).

The G-groups (G-group 59) technology is used. The following machine data were set up by the machine manufacturer for this purpose:

MD20600: MAX\_PATH\_JERK[0]=1000 Deselection DYNNORM

MD20600: MAX\_PATH\_JERK[2]=15 Roughing DYNROUG

MD20600: MAX\_PATH\_JERK[3]=15 Rough-Finishing DYNSEMIFIN

MD20600: MAX\_PATH\_JERK[4]=15 Finishing DYNFINISH

MD32431: MAX\_AX\_JERK[0,n]=15 n=AX1 .. AX5

MD32431: MAX\_AX\_JERK[2,n]=150

MD32431: MAX\_AX\_JERK[3,n]=150

MD32431: MAX\_AX\_JERK[4,n]=150

The technology-dependent changeover occurs in CYCLE832T

```
%_N_CYC_832T_SPF
; $PATH=/_N_CST_DIR
PROC CYC_832T(INT _ASVS,VAR REAL _FACTOR,VAR INT _OVL_on) SBLOF DISPLOF
...
_M4
_FACTOR=12
IF NOT($MN_SCALING_SYSTEM_IS_METRIC)
_FACTOR=_FACTOR*$MN_SCALING_VALUE_INCH
ENDIF
GOTOF _MEND
```

```
_M0: ;* Deselection
IF ISVAR("DYNNORM")
DYNNORM
ENDIF
GOTOF _MEND

_M1: ;* Finishing
IF ISVAR("DYNNORM")
DYNFINISH
ENDIF
GOTOF _MEND

_M2: ;* Rough-finishing
IF ISVAR("DYNSEMIFIN")
DYNSEMIFIN
ENDIF
GOTOF _MEND

_M3: ;* Roughing
IF ISVAR("DYNROUGH")
DYNROUGH
ENDIF
GOTOF _MEND

_MEND:
RET
```

To permit the machine data in CYC\_832T to be overwritten at all protection levels during the program run, they must be redefined with REDEF.

Example:

```
%_N_MGUD_DEF
;$PATH=/_N_DEF_DIR

REDEF $MC_MAX_PATH_JERK APR 7 APW 7
REDEF $MA_MAX_AX_JERK APR 7 APW 7

M30
```

### 3.17.5 Ports

#### G-codes

List of the G commands programmed in CYCLE832:

- G64, G641, G642
- G601
- FFWON, FFWOF
- SOFT, BRISK
- COMPCAD, COMPCURV, COMPOF, B-SPLINE
- TRAORI, TRAORI(2), TRAOF OF
- UPATH

---

#### Note

The G commands should not be generated in the subsequent CAM program.

Separation of technology and geometry.

---

#### Machine data

The following machine data are evaluated in CYCLE832 in order to describe the tolerance values accordingly:

MD number	MD identifier	Comment
10071	\$MN_IPO_CYCLE_TIME	IPO cycle
20480	\$MC_SMOOTHING_MODE	
20482	\$MC_COMPRESSOR_MODE.	
24100 to 24462	5-axis transformation machine data	

The following machine data is rewritten in CYCLE832:

MD number	MD identifier	Comment
20490	\$MC_IGNORE_OVL_FACTOR_FOR_ADIS	
33100	\$MA_COMPRESS_POS_TOL[AX]	Geometry axes 1...3
33100	\$MA_COMPRESS_POS_TOL[AX]	Rotary axes 1 and 2 <sup>1)</sup>
32310	\$MA_MAX_ACCEL_OVL_FACTOR[AX]	Geometry axes 1...3
32310	\$MA_MAX_ACCEL_OVL_FACTOR[AX]	Rotary axes 1 and 2 <sup>1)</sup>
1) acc. to 5-axis transformation machine data		

## Setting data

List of setting data rewritten in CYCLE832:

SD no.	SD identifier	Comment
42450	\$SC_CONTPREC	For CPRECON and G64
42465	\$SC_SMOOTH_CONTUR_TOL <sup>1)</sup>	Corresponds to the tolerance of the linear axes
42466	\$SC_SMOOTH_ORI_TOL <sup>1)</sup>	Corresponds to the tolerance of the rotary axes
42475	\$SC_COMPRESS_CONTUR_TOL <sup>2)</sup>	For COMPCURV only
42476	\$SC_COMPRESS_ORI_TOL <sup>2)</sup>	For COMPCURV only
1) The effectiveness of the \$SC_SMOOTH_CONTUR_TOL and \$SC_SMOOTH_ORI_TOL setting data depends on MD20480: \$MC_SMOOTHING_MODE. 2) The effectiveness of the \$SC_COMPRESS_CONTUR_TOL and \$SC_COMPRESS_ORI_TOL setting data depends on MD20482: \$MC_COMPRESSOR_MODE		

## Channel-specific variable GUD7

The following channel-specific variables must be activated for CYCLE832 functionality (machine manufacturer).

The definitions make up part of the GUD7 definitions in the SIEMENS standard cycle package.

Parameter	Format	Assignment	Comment
_TOLT2 [ 2 ]	STRING[32]	"" (Default)	Program name of subroutine for calling the 5-axis transformation
_TOLT [ 4 ]	integer	<b>Field (4):</b> 0: Deselection 1: Finishing 2: Rough-finishing 3: Roughing	Field for saving the machine manufacturer's technological data settings Coding corresponds to the _TOLM variable (see parameters)
_TOLV [ 4 ]	real	<b>Field (4):</b> 0: Deselection 1: Finishing 2: Rough-finishing 3: Roughing	Field for saving the tolerance values of the machining axes using the machine manufacturer settings (see "Customizing the technology").
			Default values: (GUD7.def)
			0.01 deselection
			0.01 finishing
			0.05 rough-finishing
			0.1 roughing



## 3.18 Engraving cycle CYCLE60

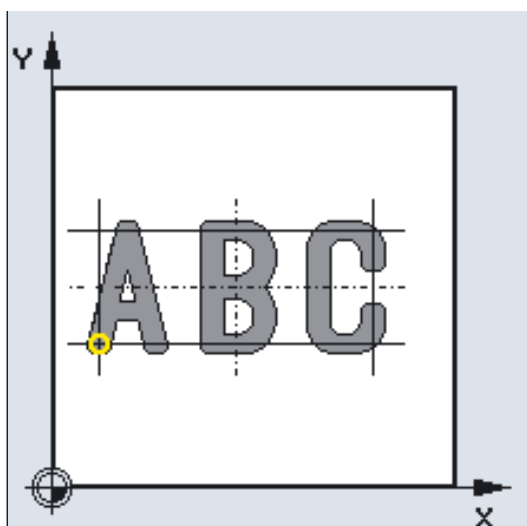
### Function

With engraving cycle CYCLE60, you can mill text positioned on a line or circle. The text may be positioned on an upward or downward circle.

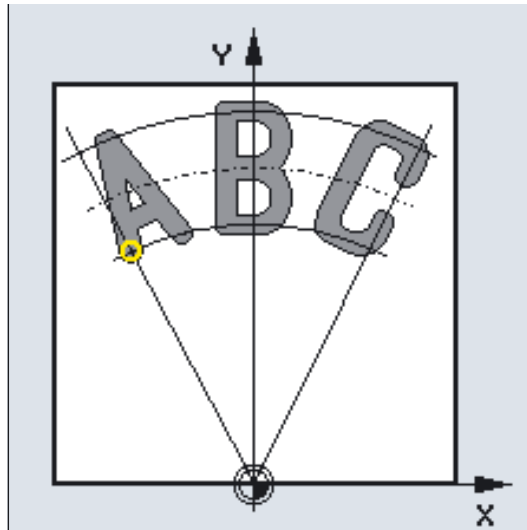
The character height and overall text width and the orientation of the text can be varied using various parameters. Character spacing or the arc angle can also be changed in the case of circular text.

The form of the characters cannot be changed. The cycle uses a proportional font, i.e., the individual characters are of different widths. The line width of the characters corresponds to the tool diameter.

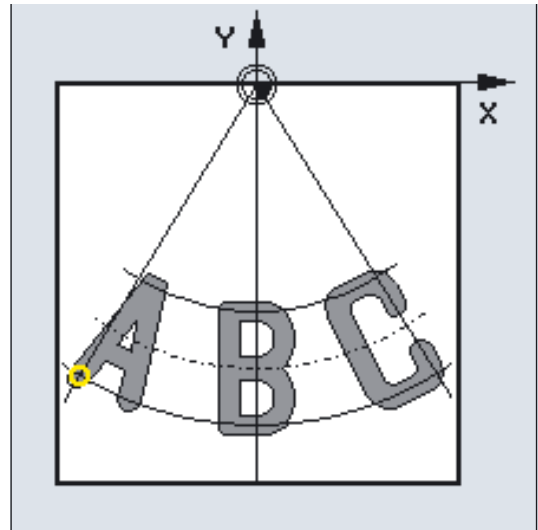
#### Linear text



Upward circular text



Downward circular text



## Programming

CYCLE60 (\_TEXT, \_RTP, \_RFP, \_SDIS, \_DP, \_DPR, \_PA, \_PO, \_STA, \_CP1, \_CP2, \_WID, \_DF, \_FFD, \_FFP1, \_VARI, \_CODEP)

## Parameter

Parameter	Data type	Meaning
_TEXT	STRING	Text to be engraved (up to 91 characters)
_RTP	Real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety distance (to be added to _RFP, enter without sign)
_DP	real	Depth (absolute)
_DPR	real	Depth relative to the reference plane (enter without sign)
_PA	real	Reference point for text orientation (absolute) <ul style="list-style-type: none"> <li>• Position of 1st axis (for _VARI = rectangular) or</li> <li>• Radius of the arc (for _VARI = polar)</li> </ul>
_PO	real	Reference point for text orientation (absolute) <ul style="list-style-type: none"> <li>• Position of 2nd axis (for _VARI = rectangular) or</li> <li>• Angle with 1st axis (for _VARI = polar)</li> </ul>
_STA	real	Angle with 1st axis (for _VARI = linear only)
_CP1	real	Center point of the circle (absolute), (for circular arrangement only) <ul style="list-style-type: none"> <li>• Position of 1st axis (for _VARI = rectangular) or</li> <li>• Radius of the arc (for _VARI = polar), with reference to the circle center</li> </ul>

Parameter	Data type	Meaning
_CP2	real	Center point of the circle (absolute), (for circular arrangement only) <ul style="list-style-type: none"> <li>Position of 2nd axis (_VARI = rectangular) or</li> <li>Angle with 1st axis (for _VARI = polar)</li> </ul>
_WID	real	Character height (enter without sign)
_DF	real	Specification of the character width (corresponds to _VARI hundred thousandths place) <ul style="list-style-type: none"> <li>Incremental character spacing in mm/inch or</li> <li>Incremental overall text width in mm/inch or</li> <li>Arc angle in degrees</li> </ul>
_FFD	real	Feedrate for depth infeed
_FFP1	real	Feedrate for surface machining
_VARI	integer	Machining type: (enter without sign)
		Values: UNITS DIGIT: Reference point 0: Right-angled (cartesian) 1: Polar
		TENS DIGIT: Text alignment 0: Linear text 1: Upward circular text 2: Downward circular text
		HUNDREDS DIGIT: Reserved
		THOUSANDS DIGIT: Reference point of the text horizontal 0: Left 1: Middle 2: Right
		TEN THOUSANDS DIGIT: Reference point of the text vertical 0: Bottom 1: Middle 2: Top
		HUNDRED THOUSANDS DIGIT: Text width 0: Character spacing 1: Overall text width (linear text only) 2: Opening angle (only for circular text)
		7th digit from right (million): Circle center 0: Right-angled (cartesian) 1: Polar
_CODEP	integer	Code page number for character set entered 1252...Code page for Central European languages (value 0 is internally treated as equivalent to 1252)

**Example 1: Engraving linear text**

This program engraves the text "SINUMERIK" on a line.

The reference point is at X10 Y25 lower left. The text is 14 mm high with 5 mm spacing between characters.

**Linear**

```
N10 G17 DIAMOF F2000 S1500 M3
N20 T1 D1
N30 M6
N40 G0 G90 Z100
N50 CYCLE60("SINUMERIK",100,0,1,-1.5,0,10,25,0,,14,5,2500,2000,0,1252)
N60 M30
```

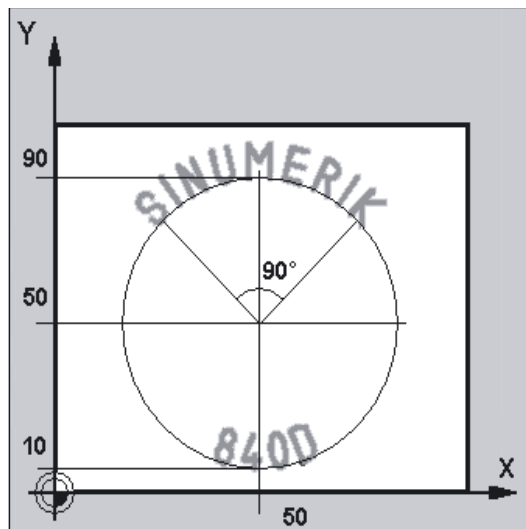
### Example 2: Engraving circular text

This program engraves two texts on a circle: "SINUMERIK" on upward circle and "840D" on a downward circle.

The reference points are in the center and bottom at X50, Y90 and at X50 Y10. The 40 mm radius of the circle results from the distance between the reference points and the circle center point of X50, Y50.

The text is 9 mm high, the character spacing results from the programmed arc angles of 90 degrees and 30 degrees respectively.

#### Circular



```
N10 G17 DIAMOF F2000 S800 M3
N20 T1 D1
N30 M6
N40 G0 G90 Z100
N50 CYCLE60("SINUMERIK",100,0,1,-1.5, -> ;Text on an upward circle
-> 0,50,90,0,50,50,9,90,2500,2000, ->
-> 201010,1252)
N60 CYCLE60("840D",100,0,1,-1.5, -> ;Text on a downward circle
->
0,50,10,0,50,50,9,34,2500,2000,201020,1252)
M30
```

---

#### Note

-> means: it must be programmed in a single block

---

## Sequence

### Position reached prior to cycle start:

The starting position can be any position, from which the starting position of the first character can be approached without collision.

### The cycle creates the following sequence of motions:

- Rapid traverse positioning to the starting position in the machining plane; followed by positioning with G0 to the reference plane brought forward by the safety distance.
- Insertion to the programmed depth at the feedrate for depth infeed \_FFD.
- Machining of each character at feedrate for surface machining \_FFP1.
- After completion of each character, retraction to the safety plane is performed with G0, followed by positioning on the starting point for the next character with G0.
- After execution of all programmed characters, the tool is positioned on the retraction plane with G0.

## Explanation of the parameters

### \_TEXT (text to be engraved)

The text to be engraved is limited to 91 characters. All uppercase and lowercase letters, numerals, and most special characters in code page 1252 are permitted.

See "Character set" on next page for all permissible special characters.

If the text contains impermissible characters, the cycle is aborted with alarm 61179 "Characters do not exist".

Using a soft key labeled "Variable text", you can toggle the engraving string between direct text input and entry of a parameter name in the input screen form of the engraving cycle (CYCLE60).

Program	CHAN1	JOG Ref	MPF0
Channel reset		Program aborted	
		ROV	

Engraving cyc./CYCLE60		Input field for ident., max. of 32 char.	
Identifier	GRAVURTEXT_VARIABLE		

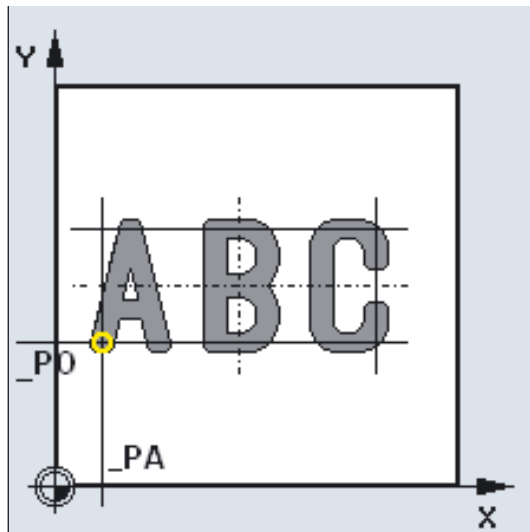
Ref. point	Rectangular	
Alignment	Linear	
Pos. 1st axis	Left	
Pos. 2nd axis	Bottom	
Retract plane	RTP	10.000
Ref. plane	RFP	0.000
Safety dist.	SDIS	2.000
Letter depth	DP	-3.000
Ref. point	PA	12.000
Ref. point	PO	10.000
Angle	STA	0.000

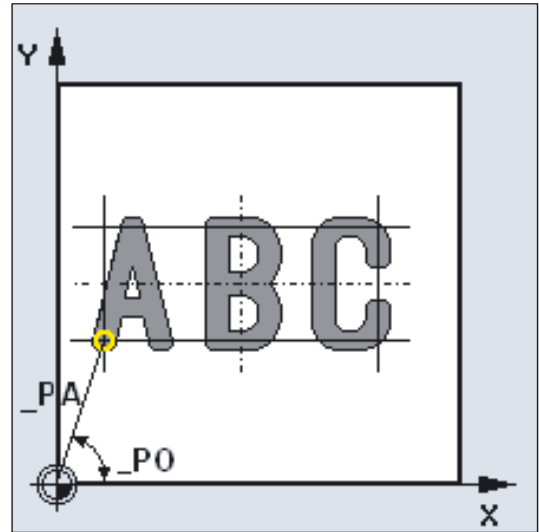
Letter height	WID	8.000
Font width	Total width	

abs

Reference point at right angles



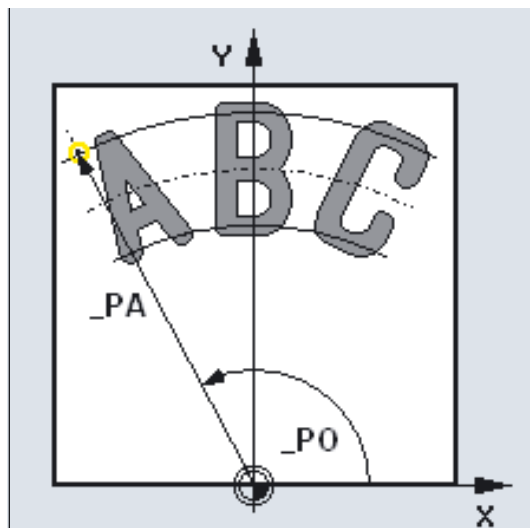
Polar reference point



If the text is of a circular orientation with polar reference point programming, the reference point always refers to the circle center point.

Whether the reference point is rectangular or polar is specified in parameter \_VARI.

Reference point is center of circle





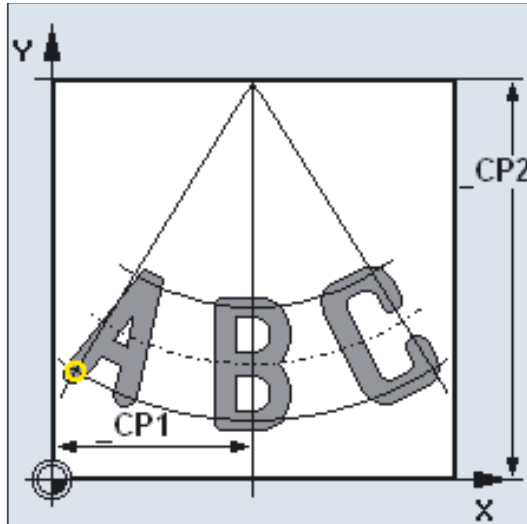
**CP1, \_CP2 (center of circle)**

When positioned on a circle, the circle center point can also be programmed as either rectangular (Cartesian) or polar.

Whether the circle center point is rectangular or polar is specified in parameter VARI.

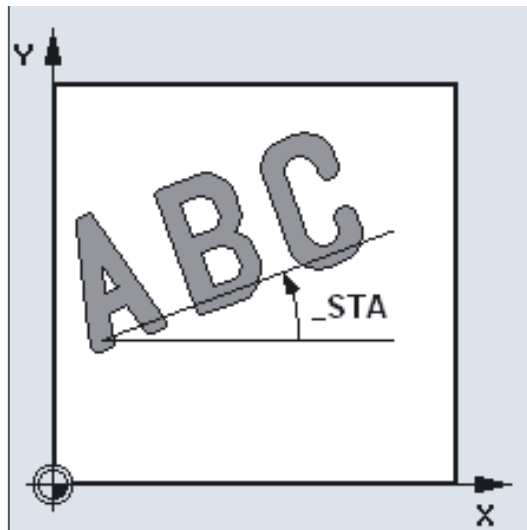
These parameters only apply to circular orientation.

**Center of circle - cartesian**



**STA (text orientation angle)**

STA specifies the angle between the 1st axis of the plane (abscissa) and the longitudinal axis (line, on which the text is written) of the text to be written. This parameter only applies to linear text orientation.

**Angle for text orientation****\_WID (character height)**

The programmed character height corresponds to the height of uppercase letters or numerals minus twice the mill radius.

For special characters, for example (), the dimension  $0.15 * \text{\_WID}$  must be added above and below.

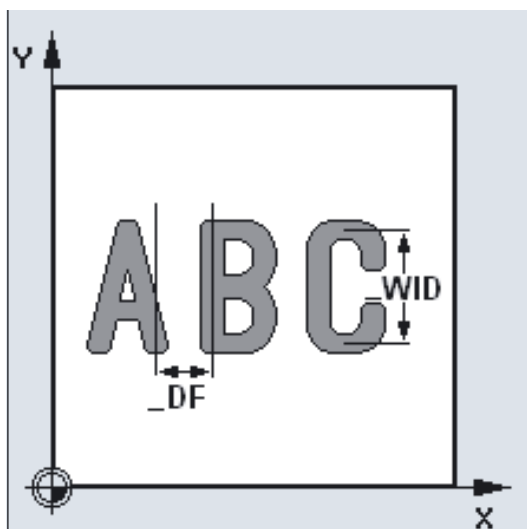
**\_DF (character spacing)**

For linear text, either the character spacing or the overall text width may be specified. These are incremental values.

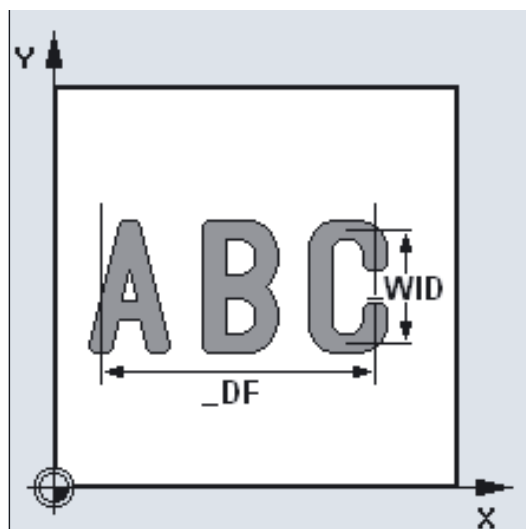
The cycle monitors whether it is possible to implement the programmed overall width, i.e., whether the sum of the character widths exceeds it.

In case of error, the alarm: 61176 "\_DF text length programmed too small" is output.

Linear text character spacing



Overall width character spacing

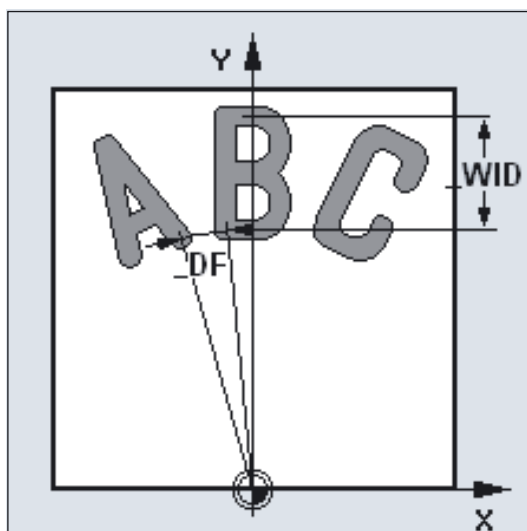


For orientation on a circle, the character spacing or arc angle between the first and the last characters can be specified. The character spacing is not the linear distance between adjacent characters, but is applied to the arc as a radian measure.

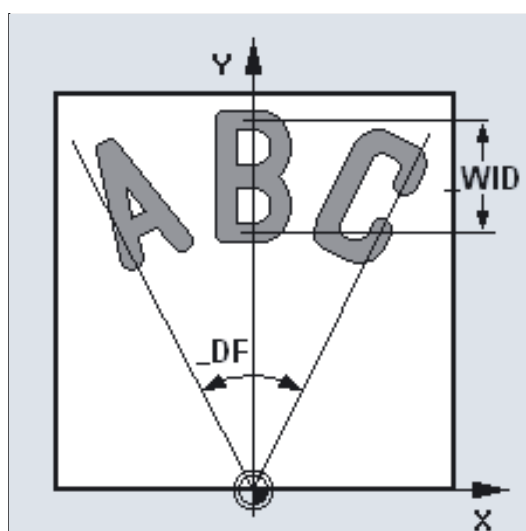
The arc angle is always specified as a positive value. It refers to the angle between the center line of the first character and the center line of the last character.

The character spacing, overall text width or arc angle is specified in parameter \_VARI.

Character spacing with circular text



Opening angle character spacing



To distribute the characters evenly in a circle, simply program `_DF=360`. The cycle then distributes the characters automatically over the complete circle. There is no need to calculate the arc angle between the first and last character.

#### **\_FFD, \_FFP1 (feedrate)**

Feedrate `_FFP1` applies to all movements in the plane (milling characters), feedrate `_FFD` applies to vertical insertion with `G1`.

#### **\_VARI (text orientation setting)**

The `_VARI` parameter defines the text orientation.

See the `_VARI` parameter in this section for values.

#### **\_CODEP (code page number)**

CYCLE60 is currently only capable of engraving characters from code page 1252. Therefore, the parameter always has the value 1252.

If a number is passed that the cycle does not know, alarm

61178 "Channel %1 block %2: Lettering cycle: code page not available" is generated and the cycle is aborted.

#### **Character set**

In addition to letters and numerals, the following special characters (punctuation marks, parentheses, arithmetic symbols, currency symbols, and other special characters) are possible:

.	,	:	;	!	?	"	'	`	´	^	_
(	)	[	]	{	}		#	Ø	ø	°	
+	-	*	/	\	÷	×	<	>	=	~	
€	\$	£	§	&	%	@	©	®			
à	á	â	ã	ä	å	À	Á	Â	Ã	Ä	Å
è	é	ê		ë		È	É	Ê		Ë	
ì	í	î		ï		Ì	Í	Î		Ï	
ò	ó	ô	õ	ö		Ò	Ó	Ô	Õ	Ö	
ù	ú	û		ü		Ù	Ú	Û		Ü	
	ý			ÿ		Ý				Ÿ	
ñ	Ñ	š	Š	ž	Ž	ç	Ç	¢	¥		
ß	μ	đ	Ð	þ	Þ	æ	Æ	œ	Œ		

There is a special rule for programming single quotes and double quotes (speech marks), as these characters already have a function in string processing of the NC language. They must be written within the text string, enclosed in single quotes.

#### **Example:**

To obtain the following sentence:

This is a text with " and ',

program the following in `_TEXT`:

This is a text with ' ' ' and ' ' ' .

---

**Note**

For an explanation of parameters RTP, RFP, SDIS, see Drilling, centering – CYCLE81.

---

**See also**

Drilling, centering - CYCLE81 (Page 44)

## 3.19 Trochoidal milling / plunge cutting - CYCLE899

### 3.19.1 General information

#### Compatibility

The cycle and screen forms described in this document have been designed as machine-specific manufacturer cycles. They can be run on the standard HMI in the "Milling" programming area.

- HMI SW 7.3 or higher and a PCU50 are required.
- NCU SW 6.5 or higher is required.

You can use this cycle to machine open grooves as part of the vortex milling (trochoidal milling) or plunge cutting (plunging) application.

### 3.19.2 Function

#### 3.19.2.1 Vortex milling

##### General information

The vortex milling (trochoidal milling) application involves machining workpieces with open grooves.

Particularly where hardened materials are concerned, this process is used for roughing and contour preparation with coated VHM milling cutters.

Vortex milling is the preferred method of HSC roughing, since it ensures that the tool is never completely inserted, i.e. it is never engaged with its full diameter; it also ensures that the tool paths are followed smoothly and using a circular motion, and that the set degree of overlap is strictly adhered to.

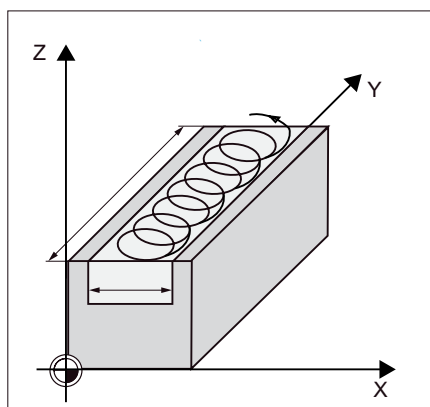


Figure 3-1 Width and length of open groove

Depending on the angle (Parameter\_STA1), the groove can be generated anywhere in the plane.

### Geometric and technological fundamentals

- Groove width at least  $1.15 \cdot \text{milling cutter diameter} + \text{finishing allowance}$
- Groove width  $< 2 \cdot \text{milling cutter diameter} + 2 \cdot \text{finishing allowance}$
- Maximum infeed depth  $\leq$  section height of milling cutter (cannot be tested)
- Maximum radial infeed dependent on milling cutter

### Roughing machining

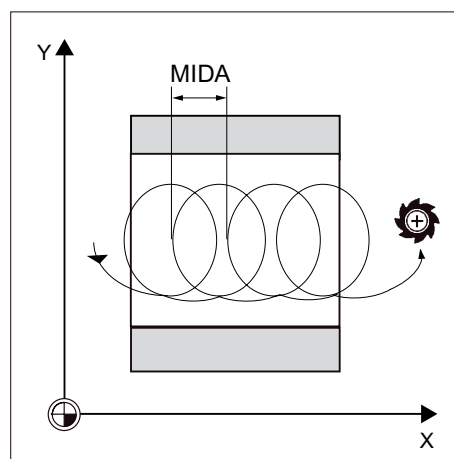
Roughing takes place by directing the milling cutter along a circular path. While in motion, the milling cutter is continuously fed more and more in the mill direction. The milling cutter's path of motion is calculated in the cycle in advance. If the milling cutter has traveled through the entire groove, it is fed back in the direction of the tool and continues to move in a circular fashion as it is returning. By doing this, it removes the next layer in the "Z direction".

This process is repeated until the set groove depth + finishing allowance has been achieved.

In the case of hard materials, a lower infeed should be selected; for soft materials, a higher one. The milling cutter is inserted into the material from the first path.

The milling cutter begins the infeed from outside the workpiece.

At the end of the milling path, the milling cutter will be outside the workpiece.



## Finishing machining

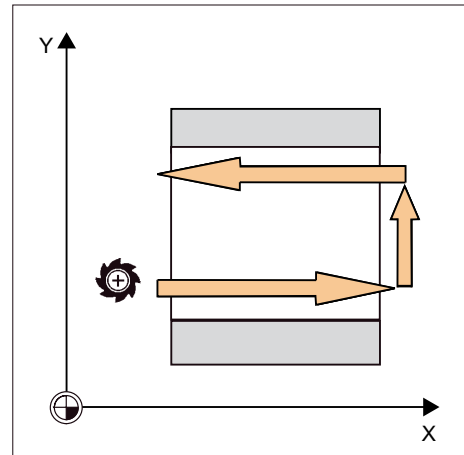
The following types of finishing exist:

- Rough-finishing
- Finishing
- Finishing of edge
- Finishing of base

Rough-finishing removes any residual corners caused by the technological roughing process. The finishing allowance stays the same here.

When finishing walls, the milling cutter travels along the groove walls: to do this, it is again fed in the Z direction, unit by unit. In doing this, the milling cutter travels along the SDIS beyond the beginning and end of the groove, so that it can guarantee an even groove-wall surface across the entire length of the groove.

When finishing the base, the milling cutter executes one motion across the final groove and back again.



### 3.19.2.2 Plunge cutting

#### General information

Plunge cutting is the preferred method of machining pockets and grooves for "unstable" machines and workpiece geometries.

This method generally only exerts forces along the main spindle axis, i.e., perpendicular to the surface of the pocket/groove to be machined (with XY plane, in Z direction). For this reason, the tool is subject to virtually no deformation. Thanks to the axial loading of the tool, there is hardly any danger of vibrations affecting unstable workpieces. The depth of cut can be increased considerably. The plunge cutter, as it is known, ensures a longer service life due to fewer vibrations occurring with long collar lengths.

Application:

- Roughing and opening deep grooves with boring motions in the Z direction
- Roughing deep cavities

#### Geometric and technological fundamentals

- Maximum radial infeed is dependent on the cutting width of the milling cutter (cannot be tested  $\Rightarrow$  must be selected carefully by the operator).
- The lateral increment results from the required groove width, the milling cutter diameter and the finishing allowance.



- Retraction involves the milling cutter being retracted at a  $45^\circ$  angle following insertion, if the wrap angle is less than  $180^\circ$ . Otherwise, retraction is vertical, as is the case with boring.
- Retraction is performed perpendicular to the wrapped surface.
- The milling cutter motion travels beyond the end of the workpiece, along the length of the safety clearance, to avoid rounding of the groove walls at the ends.

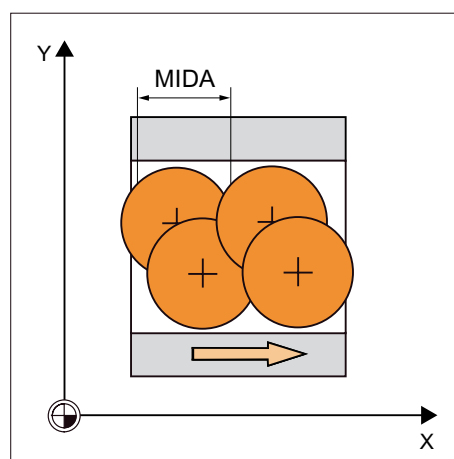
## Roughing machining

Roughing of the groove takes place sequentially along the length of the groove, with the milling cutter performing vertical insertions at the machining feedrate.

The milling cutter is then retracted and repositioned at the next insertion point. The milling cutter moves along the length of the groove, at half the infeed rate, and inserts alternately at the left-hand and right-hand wall.

The first insertion takes place at the groove edge, with the milling cutter inserted at half the infeed, less the safety clearance (if the safety clearance is greater than the infeed, this will be on the outside). For this cycle, the maximum width of the grooves must be less than double the width of the milling cutter, plus the finishing allowance.

Following each insertion, the milling cutter is lifted by the height of the safety clearance at the machining feedrate. As far as possible, this occurs during what is known as the retraction process, i.e., if the milling cutter's wrap angle is less than  $180^\circ$ , it is lifted at a  $45^\circ$  angle from the base in the opposite direction to the bisector of the wrap area. The milling cutter then travels rapidly along the material.



## Finishing machining

Same procedure as vortex milling; refer to this section.

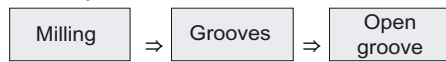
### 3.19.3 Programming via a screen form

#### 3.19.3.1 General information

##### Calling CYCLE899

Entering the programs/milling area

Softkey



The following milling technologies can be selected in the vertical softkey bar displayed:

Elongated  
hole

Groove

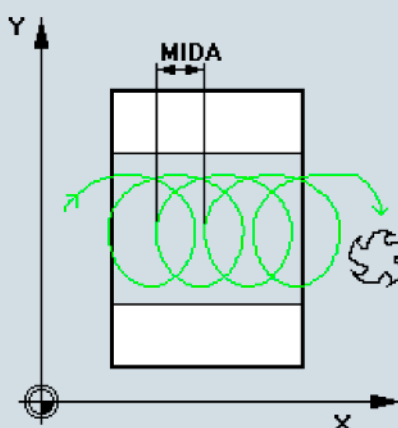
Circumferential  
groove

Open  
groove

## 3.19.3.2 Parameters of the screen form

## Vortex milling screen forms

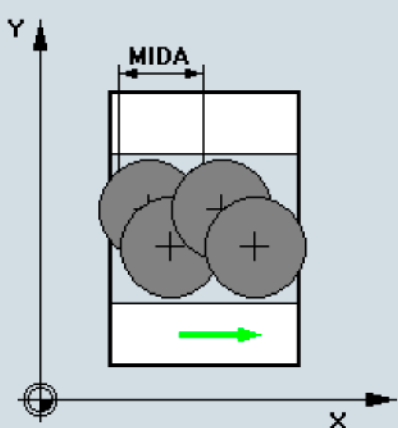
**Open groove/CYCLE899** Select: Vortex milling or plunge cutting



Retract plane	RTP	100.000	
Ref. plane	RFP	0.000	
Safety dist.	SDIS	1.000	
Groove depth	DP	-12.000	abs
Technology		<b>Vortex milling</b>	<input checked="" type="radio"/>
Operation		Roughing	
Dimensions		+X egde	
Start. point	PA	0.000	
Start. point	PO	0.000	
Groove length	LENG	35.000	
Groove width	WID	12.000	
Fin. allow.	FAL	0.200	
Fin. allow.	FALD	0.100	
Mill. direct.		Up-cut	
Start. angle	STA1	0.000	
Feedr. surface	FFP1	3000.000	

## Plunge cutting screen forms

**Open groove/CYCLE899** Select: Vortex milling or plunge cutting



Retract plane	RTP	100.000	
Ref. plane	RFP	0.000	
Safety dist.	SDIS	1.000	
Groove depth	DP	-12.000	abs
Technology		<b>Plunge cutting</b>	<input checked="" type="radio"/>
Operation		Roughing	
Dimensions		+X egde	
Start. point	PA	0.000	
Start. point	PO	0.000	
Groove length	LENG	35.000	
Groove width	WID	12.000	
Fin. allow.	FAL	0.200	
Fin. allow.	FALD	0.100	
Start. angle	STA1	0.000	
Feedr. surface	FFP1	3000.000	

The operator must enter values in the entry fields.

The following selection boxes must be set:

#### Groove depth

- Absolute (ABS)
- Incremental (INC)

#### Machining

- Vortex milling
- Plunge cutting

#### Machining type

- Roughing
- Rough-finishing
- Finishing
- Finishing of edge
- Finishing of base

#### Dimensioning

- Center
- Edge +X
- Edge -X

The position of the reference point is specified by means of the "Dimensioning" selection box. The position of the workpiece is then specified by means of the reference point, starting point, \_PA/\_PO and angle \_STA1. Angle \_STA1 always refers to the reference point.

With "Edge +X", machining on the plane occurs in the negative X direction, and with "Edge -X" and "Center", in the positive X direction.

#### Mill direction CDIR (only circular-milling roughing or finishing)

- Synchronism
- Up-cut
- Alternate (only with circular-milling roughing)

#### Plane infeed per cut (MIDA)

- As a value [in mm/inch]
- As a percentage (%), relative to the tool diameter

### 3.19.4 Programming using parameters

#### Programming

```
CYCLE899(_RTP, _RFP, _SDIS, _DP, _LENG, _WID, _PA, _PO, _STA1,
_MID, _MIDA, _FAL, _FALD, _FFP1, _CDIR, _VARI, _GMODE, _DMODE,
_AMODE)
```

#### Parameters

Parameter	Data type	Description
_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety clearance (incremental)
_DP	real	Groove depth, absolute/incremental
_LENG	real	Groove length, incremental
_WID	real	Groove width, incremental
_PA	real	Starting point (or reference point) in X (absolute)
_PO	real	Starting point (or reference point) in Y (absolute)
_STA1	real	Angle to 1st axis
_MID	real	Infeed depth (only circular-milling roughing or finishing) (incremental)
_MIDA	real	Plane infeed per cut
_FAL	real	Finishing allowance in X, Y at edge
_FALD	real	Finishing allowance in Z at base
_FFP1	real	Surface machining main feedrate
_CDIR	integer	Mill direction (only circular-milling roughing or finishing) Values: 0: Synchronism 1: Up-cut 4: Alternate (only with circular-milling roughing)
_VARI	integer	Machining type Values: UNITS DIGIT: 1: Roughing 2: Finishing 3: Finishing of base 4: Finishing of edge 5: Rough-finishing THOUSANDS DIGIT: 1xxx: Vortex milling 2xxx: Plunge cutting
_GMODE	integer	Evaluation of geometrical data THOUSANDS DIGIT: Dimensioning via center/edge 0xxx: Dimensioning via center 1xxx: Dimensioning via edge, -X 2xxx: Dimensioning via edge, +X
_DMODE	integer	Display mode (internal) Values: 0

Parameter	Data type	Description
_AMODE	integer	Alternate mode
		Values: UNITS DIGIT: Groove depth 0: Absolute 1: Incremental
		TENS DIGIT: Unit for plane infeed (_MIDA) 0x: mm 1x: % of tool diameter

### Explanation of the parameters

#### **\_RFP and \_RTP (reference plane and retraction plane)**

Normally, the reference plane (\_RFP) and retraction plane (\_RTP) have different values. The cycle assumes that the retraction plane precedes the reference plane.

#### **\_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 is active is automatically determined by the cycle.

#### **\_DP (groove depth, absolute/incremental)**

The groove depth can be specified as either an absolute or an incremental value in relation to the reference plane. If the incremental option is selected, the cycle automatically calculates the depth on the basis of the reference and retraction plane positions.

#### **\_LENG (groove length, incremental) and \_WID (groove width, incremental)**

Use the parameters \_LENG and \_WID to define the form of an open groove in the plane.

The groove can be dimensioned from the center or from a corner point.

#### **\_PA, \_PO (starting point/reference point, absolute)**

With the parameters \_PA and \_PO, you define the starting point/reference point of the groove in the X/Y machining plane.

#### **\_STA1 (angle)**

\_STA indicates the angle between the 1st axis of the plane (abscissa) and the longitudinal axis of the groove. The rotation around \_STA1 is always carried out at the reference point.

#### **\_MID (infeed depth, incremental) (only circular-milling roughing or finishing)**

Use this parameter to define the maximum infeed depth. The depth infeed is performed by the cycle in equally-sized infeed steps.

The cycle calculates this infeed automatically on the basis of \_MID and the total depth.

The minimum possible number of infeed steps is used as the basis. \_MID = 0 means that the infeed is carried out in one cut up to the groove depth.

#### **\_MIDA (plane infeed per cut)**

Use this parameter to define the plane infeed per cut when machining in a plane. This value is never exceeded.

**\_FAL (finishing allowance in the X/Y plane at edge)**

The finishing allowance only affects the machining of the groove in the plane on the edge.

If the finishing allowance  $\geq$  tool diameter, the groove will not necessarily be machined completely.

**\_FALD (finishing allowance in Z at the base)**

A separate finishing allowance on the base is taken into account in roughing operations.

**\_FFP1 (main feedrate for surface machining)**

The feedrate \_FFP1 is active for all movements in the plane which are traversed at feedrate when machining.

**\_CDIR (milling direction, only circular-milling roughing or finishing)**

You define the groove machining direction in this parameter.

The milling direction can be programmed as follows using the \_CDIR parameter:

- 0 = Synchronism
- 1 = Up-cut
- 4 = Alternate (only circular-milling roughing)

**\_VARI (machining type)**

Use the parameter \_VARI to define the machining type. Possible values are:

- Units digit (select machining)
  - 1 = Roughing
  - 2 = Finishing
  - 3 = Finishing of edge
  - 4 = Finishing of base
  - 5 = Rough-finishing
- Thousands digit (select method)
  - 1 = Vortex milling
  - 2 = Plunge cutting

**\_GMODE (evaluation of geometrical data)**

You use this parameter to specify the geometrical data.

- Thousands digit (dimensioning)
  - 0 = Dimensioning via center
  - 1 = Dimensioning via edge, -X
  - 2 = Dimensioning via edge, +X

**\_DMODE (display mode)**

Internal = 0

**\_AMODE (alternate mode)**

You use this parameter to specify the plane infeed per cut.

- Units digit (groove depth)
  - 0 = Absolute
  - 1 = Incremental
- Tens digit (unit for plane infeed)
  - 0 = mm/inch
  - 1 = % of tool diameter

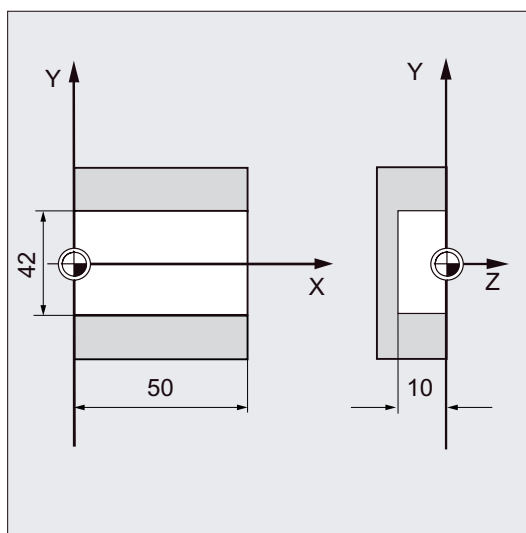


### 3.19.5 Programming example

#### Example: Vortex milling

The example demonstrates roughing of an open groove with vortex milling.

A milling cutter with a diameter of 24 mm is used for machining purposes.



The groove has the following dimensions:

Length (\_LENG): 50 mm

Width (\_WID): 42 mm

Depth (\_DP): 10 mm

Finishing allowance at edge  
(\_FAL): 1 mm

Finishing allowance at base  
(\_FALD): 1 mm

Position of groove: The center of the groove at the left-hand edge is at  $(X,Y) = (0,0)$ . The upper edge of the groove is at  $Z = 0$ .  
The groove is parallel to the X axis.

Roughing of the open groove should take place with the following parameter settings:

Parameter	Description	Value
_SDIS	Safety clearance	1 mm
_CDIR	Milling direction	Synchronism
_FFP1	Feedrate in the plane during roughing	300 mm/min
_MID	Infeed depth	5 mm
_MIDA	Plane infeed per cut	5 mm

```

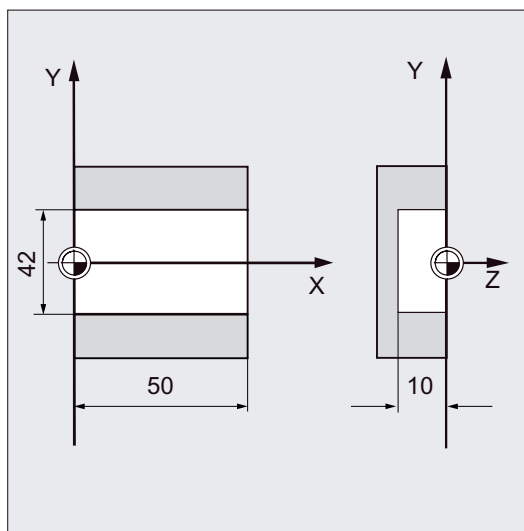
N10 G17 G54 G90 S800 M3                ;Specification of technology values
N20 T1 D1
N30 M6
N40 G0 X0 Y0 Z40                        ;Approach start position
;Cycle call, open groove with circular-milling roughing
N50 CYCLE899(20,0,1,-10,50,42,0,0,0,5,5,1,1,300,0,1001,1000,0,0)
N60 M30                                ;End of program

```

### Example: Plunge cutting

This example demonstrates roughing of an open groove with plunge cutting, followed by rough-finishing and finishing.

A milling cutter with a diameter of 30 mm is used for machining purposes.



The groove has the following dimensions:

Length (_LENG):	60 mm
Width (_WID):	52 mm
Depth (_DP):	28 mm
Finishing allowance at edge (_FAL):	1 mm
Finishing allowance at base (_FALD):	1 mm
Position of groove:	The center of the groove is at (X,Y) = (50,40).

The upper edge of the groove is at Z = 0.

The groove is parallel to the X axis.

Machining of the open groove should take place with the following parameters set:

Parameter	Description	Value
_SDIS	Safety clearance	1 mm
_CDIR	Milling direction	Synchronism
_FFP1	Feedrate for roughing	380 mm/min
	Feedrate for rough-finishing	300 mm/min
	Feedrate for finishing	250 mm/min
_MID	Infeed depth for rough-finishing	6 mm
	Infeed depth for finishing	6 mm
_MIDA	Plane infeed per cut for roughing	12 mm

```

N10 G17 G54 G90 S800 M3                ;Specification of technology values
N20 T1 D1
N30 M6
N40 G0 X0 Y0 Z40                        ;Approach start position
;Cycle call, open groove with plunge-milling roughing
N50 CYCLE899(20,0,1,28,60,52,50,40,0,,12,1,1,380,0,2001,0,0,1)
;Cycle call, open groove rough-finishing
N60 CYCLE899(20,0,1,28,60,52,50,40,0,6,,1,1,300,0,2005,0,0,1)
;Cycle call, open groove finishing
N70 CYCLE899(20,0,1,28,60,52,50,40,0,6,,,250,0,2002,0,0,1)
N80 M30                                ;End of program

```



## Turning cycles

### 4.1 General information

The following sections describe how turning cycles are programmed.

This section is intended to guide you in selecting cycles and assigning parameters to them. In addition to a detailed description of the function of the individual cycles and the corresponding parameters, you will also find a programming example at the end of each section in order for you to familiarize yourself with the use of cycles.

### 4.2 Conditions

#### Data block for turning cycles

Turning cycles require the GUD7.DEF module. This is supplied along with the cycles.

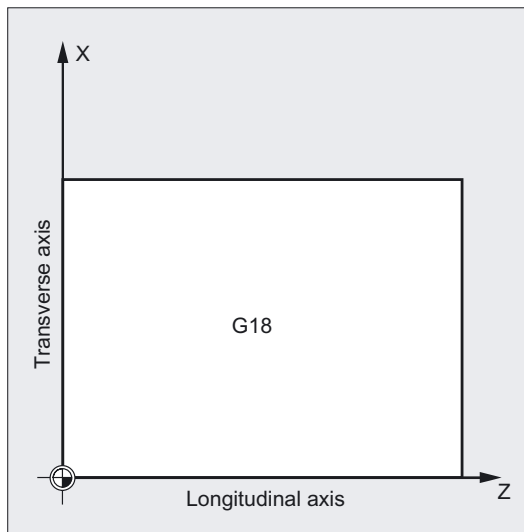
#### Call and return conditions

The G functions active before the cycle is called and the programmable frame 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 are hereinafter referred to as the longitudinal axis (first axis of this plane) and transverse axis (second axis of this plane).

In the turning cycles, the second axis of the plane is always allocated as the transverse axis when diameter programming is active.



**References:** /PG/, Programming Guide: Fundamentals

### Spindle handling

The turning cycles are written in such a way that the spindle commands contained within them always refer to the active master spindle of the control.

If you want to use a cycle on a machine with several spindles, the active spindle must first be defined as the master spindle.

**References:** /PG/, Programming Guide: Fundamentals

### Messages with regard to the machining state

Status messages are displayed on the control monitor while the turning cycles are processed. The following messages are possible:

- "Thread start <No.> - longitudinal thread machining"
- "Thread start <No.> - face thread machining"

In each case, <No.> stands for the number of the thread start in the message text.

These messages do not interrupt the program execution and continue to be displayed until the next message is displayed or the cycle is completed.

### Cycle setting data

Setting data is stored in the GUD7.DEF module for the CYCLE95 stock removal cycle.

Cycle setting data `_ZSD[0]` can be used to vary the calculation of the MID depth infeed in CYCLE95. If it is set to zero, the parameter is calculated as before.

- `_ZSD[0]=1` MID is a radius value
- `_ZSD[0]=2` MID is a diameter value

Setting data is located in the GUD7.DEF module for the CYCLE93 grooving cycle. Cycle setting data `_ZSD[4]` can be used to influence retraction after the first groove.

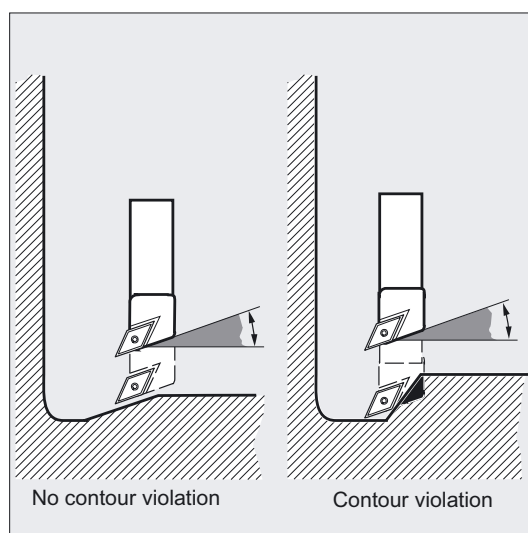
- `_ZSD[4]=1` Retraction with G0
- `_ZSD[4]=0` Retraction with G1 (same as before)

There is an option in `_ZSD[6]` to set mirroring behavior for the CYCLE93 grooving cycle.

- `_ZSD[6]=0` Tool compensations are interchanged in the cycle when mirroring is active (to be used without an orientatable toolholder).
- `_ZSD[6]=1` Tool compensations are not interchanged in the cycle when mirroring is active (to be used with an orientatable toolholder).

### Contour monitoring relative to the clearance angle of the tool

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 parameter P24). A value between 0 and 90 degrees (unsigned) can be entered for the angle.

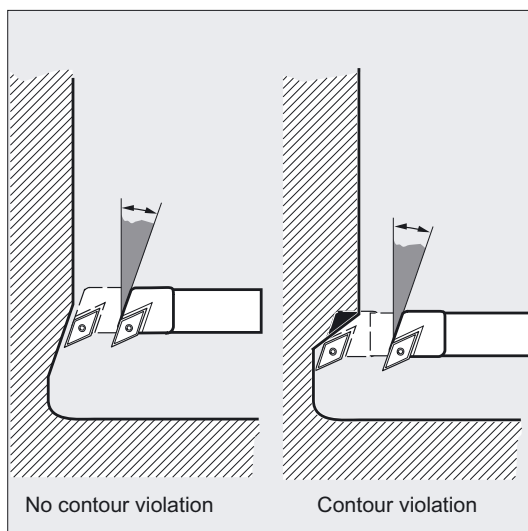


When entering the tool clearance angle, note that this depends on the machining type 'longitudinal' or 'face'. If you want to use one tool for longitudinal and face machining, two tool compensations must be used in the case of different tool clearance angles.

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.



Note that active scale factors or rotations in the current plane modify the relationships at the angles, and that this cannot be allowed for in the contour monitoring that takes place within the cycle.

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.

### Turning cycles with active adapter transformation

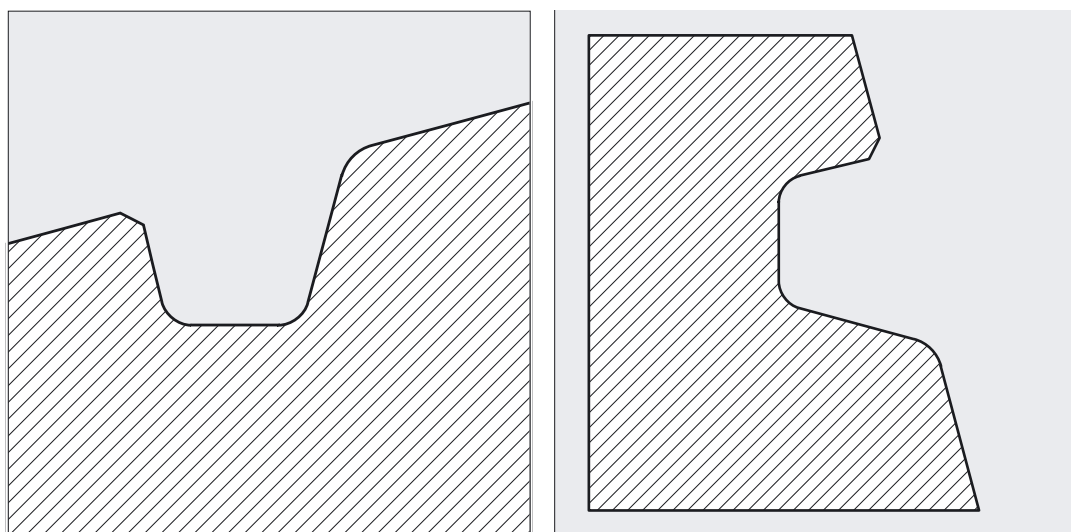
With NCK SW 6.2 and higher, turning cycles can also be executed with an active adapter transformation. The transformed tool offset data for the tool point direction and clearance angle are always read.



## 4.3 Grooving cycle - CYCLE93

### Function

With the grooving cycle, you can make symmetrical and asymmetrical grooves for longitudinal and face machining on straight contour elements. You can machine both external and internal grooves.



### Programming

CYCLE93 (SPD, SPL, WIDG, DIAG, STA1, ANG1, ANG2, RCO1, RCO2, RCI1, RCI2, FAL1, FAL2, IDEP, DTB, VARI, \_VRT, \_DN)

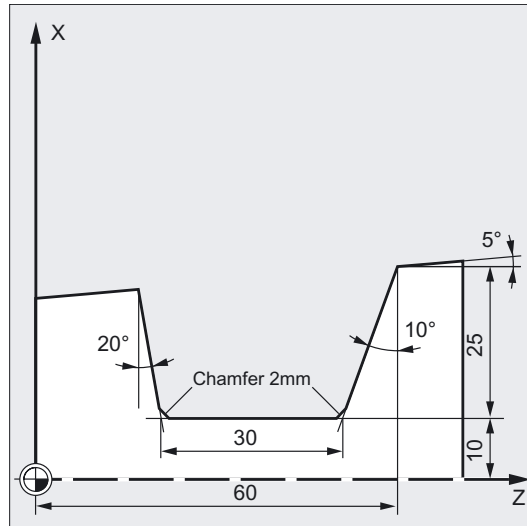
### Parameter

Parameter	Data type	Meaning
SPD	real	Starting point in the facing axis (enter without sign)
SPL	real	Starting point in the longitudinal axis
WIDG	real	Groove width (enter without sign)
DIAG	real	Groove depth (enter without sign)
STA1	real	Angle between contour and longitudinal axis Range of values: $0 \leq STA1 \leq 180$ degrees
ANG1	real	Flank angle 1: on the groove side determined by the starting point (enter without sign) Range of values: $0 \leq ANG1 < 89.999$ degrees
ANG2	real	Flank angle 2: on the other side (enter without sign) Range of values: $0 \leq ANG2 < 89.999$

Parameter	Data type	Meaning
RCO1	real	Radius/chamfer 1, externally: on the side determined by the starting point
RCO2	real	Radius/chamfer 2, externally
RCI1	real	Radius/chamfer 1, internally: on the starting point side
RCI2	real	Radius/chamfer 2, internally
FAL1	real	Finishing allowance at the recess base
FAL2	real	Finishing allowance at the flanks
IDEP	real	Infeed depth (enter without sign)
DTB	real	Dwell time at recess base
VARI	integer	Machining type Range of values: 1...8 and 11...18
_VRT	real	Variable retraction distance from contour, incremental (enter without sign)
_DN	integer	D number for 2nd edge of tool

### Example for plunge-cutting

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 uses tool offsets D1 and D2 of tool T1. The cutting tool must be defined accordingly.



```
DEF REAL SPD=35, SPL=60, WIDG=30, -> ;Definition of parameters with value
-> DIAG=25, STA1=5, ANG1=10, ANG2=20, -> ;assignments
-> RCO1=0, RCI1=-2, RCI2=-2, RCO2=0, ->
-> FAL1=1, FAL2=1, IDEP=10, DTB=1
DEF INT VARI=5
N10 G0 G18 G90 Z65 X50 T1 D1 S400 M3 ;Starting point before the beginning of
;the cycle
N20 G95 F0.2 ;Specification of technology values
N30 CYCLE93 (SPD, SPL, WIDG, DIAG, -> ;Cycle call
-> STA1, ANG1, ANG2, RCO1, RCO2, ->
-> RCI1, RCI2, FAL1, FAL2, IDEP, ->
-> DTB, VARI)
N40 G0 G90 X50 Z65 ;Next position
N50 M02 ;Program end
```

### Note

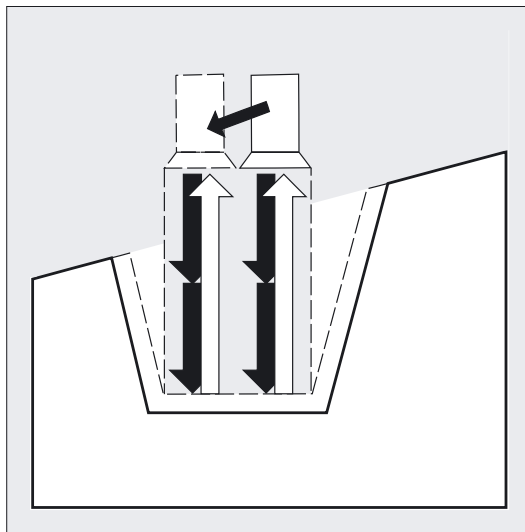
-> means: it must be programmed in a single block.

## Sequence

The depth infeed (towards the base of the groove) and infeed across the width (from groove to groove) are distributed evenly and with the greatest possible value. If the groove is being machined on an inclined surface, travel from one groove to the next follows the shortest path, i.e., parallel to the cone, on which the groove is being machined. During this process, a safety clearance to the contour is calculated internally in the cycle.

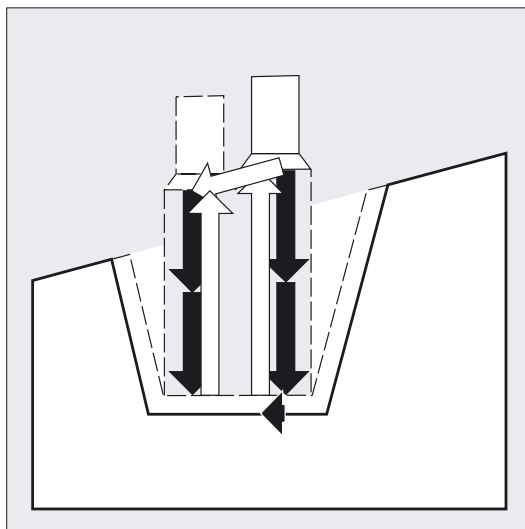
### Step 1

Paraxial roughing to the base of the groove in single infeed steps. After each infeed, the tool is retracted for chip breaking.



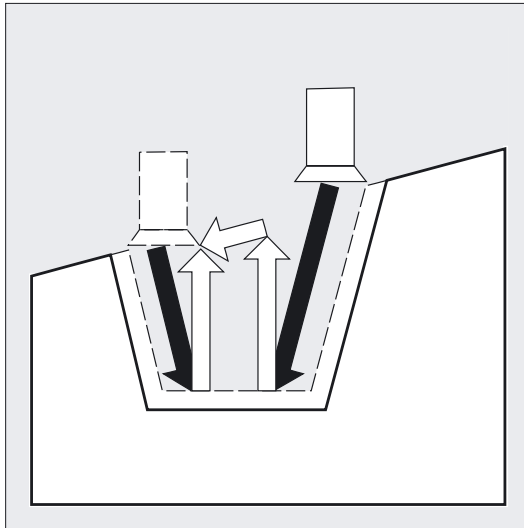
### Step 2

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.



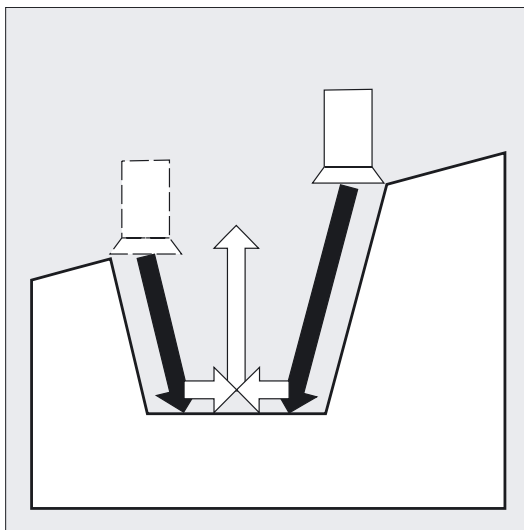
### Step 3

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.



### Step 4

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.



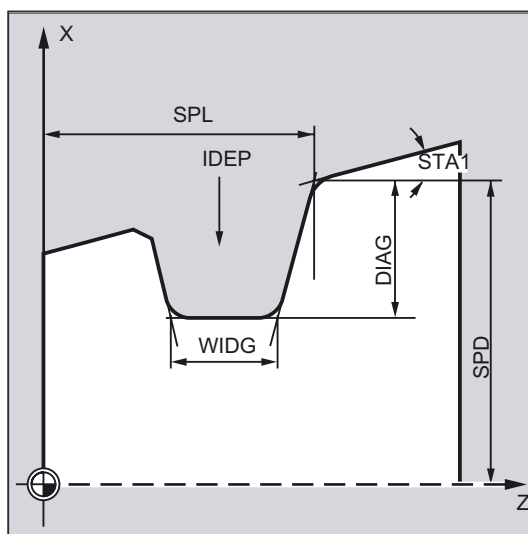
## Explanation of the parameters

### SPD and SPL (starting point)

You define the starting point of the groove, from which the cycle calculates the shape, using these coordinates. The cycle determines its own starting point. For an external groove, movement begins in the direction of the longitudinal axis, for an internal groove in the direction of the facing 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.

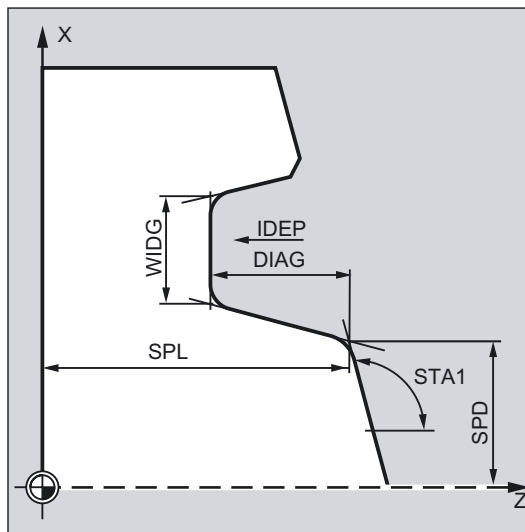
Radii and chamfers at the groove edge make sense with bent contours only if the appropriate edge point is on the straight line specified for the cycle.



### WIDG and DIAG (groove width and groove depth)

The shape of the groove is defined with the groove width (WIDG) and groove depth (DIAG) parameters. 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.

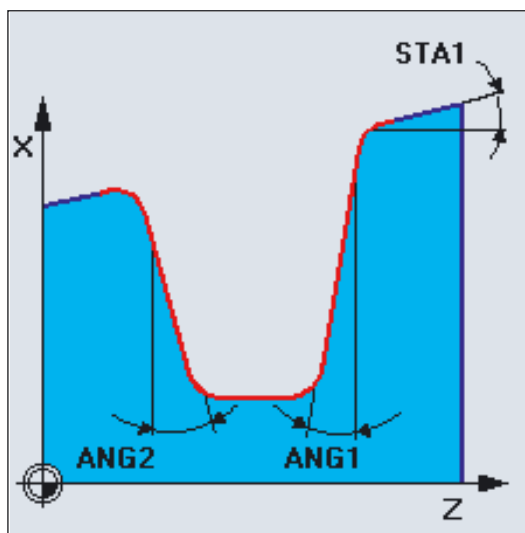


If the programmed groove width is less than the actual tool width, the error message 61602 "Tool width defined incorrectly" appears.

The cycle is not started 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 STA1 parameter 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.



#### Note

For transverse grooving, the angle STA1 is usually 90° (paraxial case).

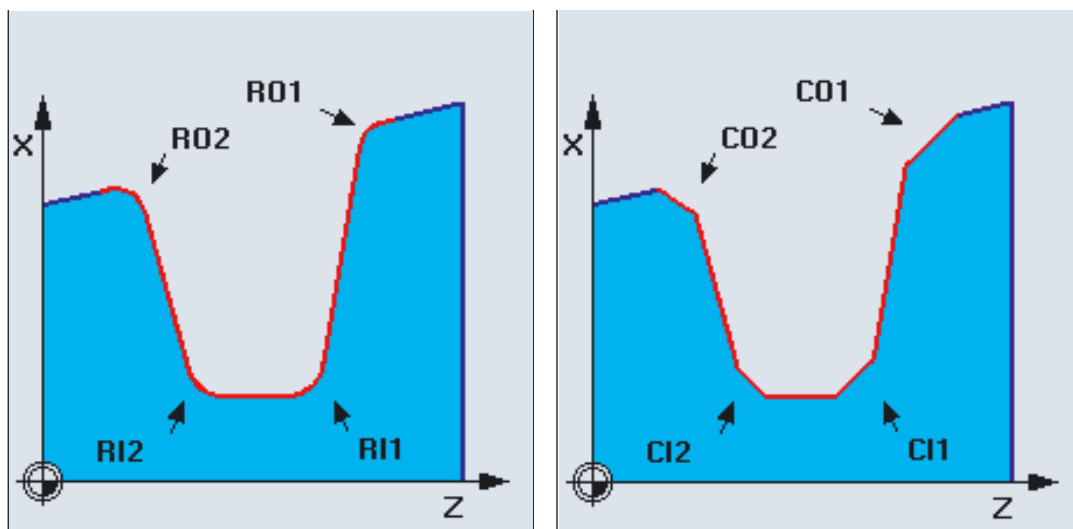
**ANG1 and ANG2 (flank angle)**

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 shape of the groove can be modified by entering radii/chamfers for the edge or base of the groove. The values for the radii must always be positive, the values for the chamfers must always be negative. You can use the tens setting for the VARI parameter to determine the type of calculation for programmed milling.

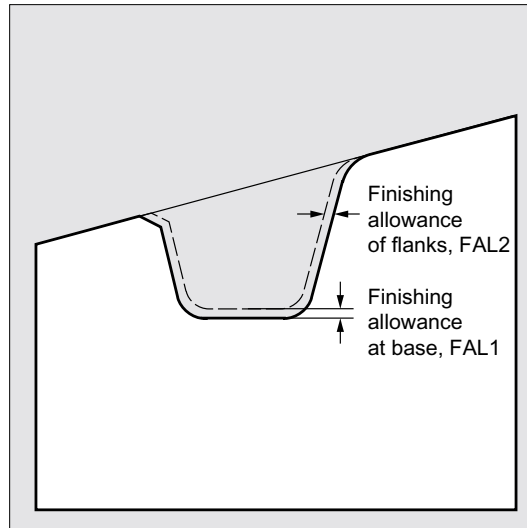
- With VARI < 10 (tens=0), the absolute value of this parameter is regarded as chamfer length (chamfering with CHF programming).
- With VARI > 10, it is regarded as reduced path length (chamfering with CHR programming).





#### FAL1 and FAL2 (finishing allowance)

You can program separate final machining allowances for the groove base and the flanks. Roughing is performed to these final machining allowances. The same tool is then used to machine a contour-parallel cut along the final contour.



#### IDEP (infeed depth)

By programming an infeed depth, you can divide the paraxial grooving action into several depth infeeds. After every infeed, the tool is retracted by 1 mm or by the distance programmed under \_VRT for chip breaking.

The IDEP parameter 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 units digit of the VARI parameter determines the type of processing for the groove. 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: Chamfers are calculated as CHF

VARI 11...18: Chamfers are calculated as CHR






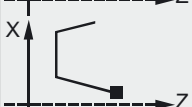
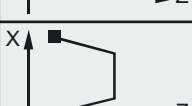

Input of the units digit in cycle support is distributed among three selection fields:

Field 1: longitudinal/face

Field 2: outer/inner

Field 3: starting point left/right (for longitudinal) or top/bottom (for face)

## 4.3 Grooving cycle - CYCLE93

VARI		Selection in cycle support
1/11		longitudinal, outside, left
5/15		longitudinal, outside, right
3/13		longitudinal, inside, left
7/17		longitudinal, inside, right
6/16		face, outside, top
8/18		face, outside, bottom
2/12		face, inside, top
4/14		face, inside, bottom

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".

**\_VRT (variable retraction path)**

The retraction path can be programmed in the \_VRT parameter on the basis of the outside or inside diameter of the groove.

If \_VRT=0 (parameter not programmed), retraction is 1 mm. The retraction path is always measured according to the programmed system of units, inch or metric.

The same retraction path is also used for chip breaking after each depth infeed into the groove.

**\_DN (D number for 2nd edge of tool)**

A new parameter, \_DN, for programming the D number of the 2nd tool edge has been added to CYCLE93. If this parameter is not programmed, the next edge after the D number which is active when the cycle is called is used (as in earlier versions).

**Note**

You must activate a double-edged tool before calling the grooving cycle. You must enter the offset values for the two tool edges in two successive D numbers of the tool, the first of which must be activated before the cycle is called. 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".

---

This cycle setting data `_ZSD[4]` can affect the retraction after the 1st groove:

- `_ZSD[4]=0`: means retraction with G1 as before
- `_ZSD[4]=1`: means retraction with G0.

Cycle data `_ZSD[6]` can be set to define how the tool offset is handled in the cycle:

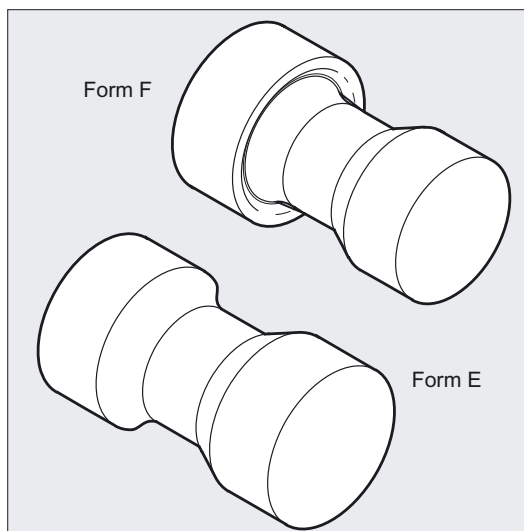
- `_ZSD[6]=0`: TO is interchanged internally in the cycle (without orientatable toolholder)
- `_ZSD[6]=1`: TO is not interchanged internally in the cycle (with orientatable toolholder)

## 4.4 Undercut cycle - CYCLE94

### Function

With this cycle, you can machine undercuts of form E and F in accordance with DIN509 with the usual load on a finished part diameter of >3 mm.

Another cycle, CYCLE96, exists for producing thread undercuts (see Section "Thread Undercut - CYCLE96").



### Programming

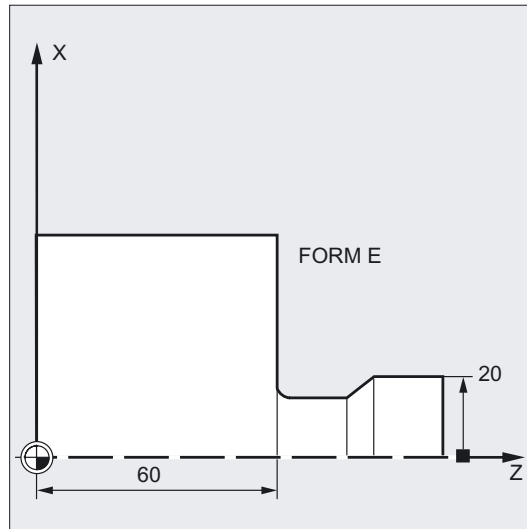
CYCLE94 (SPD, SPL, FORM, \_VARI)

### Parameter

Parameter	Data type	Meaning
SPD	real	Starting point in the facing axis (enter without sign)
SPL	real	Starting point of the contour in the longitudinal axis (enter without sign)
FORM	char	Definition of the form Values: E (for form E), F (for form F)
_VARI	integer	Specification of undercut position Values: 0 (corresponding to the tool point direction of the tool), 1 to 4 (define position)

## Example of Undercut\_Form\_E

You can machine an undercut of form E with this program.



```
N10 T25 D3 S300 M3 G18 G95 F0.3      ;Specification of technology values
N20 G0 G90 Z100 X50                  ;Selection of starting position
N30 CYCLE94(20, 60, "E")              ;Cycle call
N40 G90 G0 Z100 X50                  ;Approach next position
N50 M02                              ;End of program
```

## Sequence

### Position reached prior to cycle start:

The starting position can be any position from which each undercut can be approached without collision.

### The cycle creates the following sequence of motions:

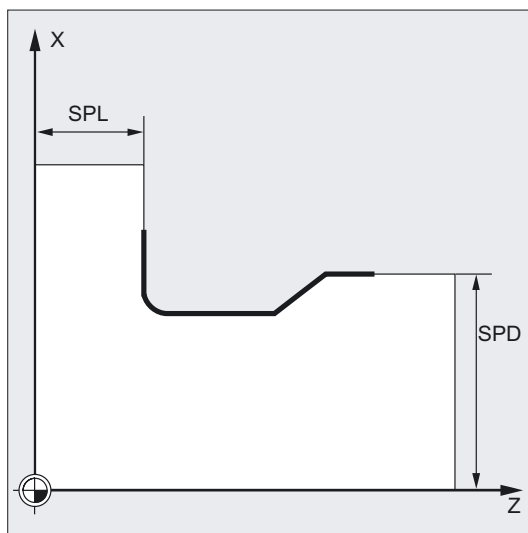
- Approach of the starting point determined in the cycle by using G0
- Selection of the cutter radius compensation according to the active tool point direction and traveling along the undercut contour at the feedrate programmed prior to the cycle call
- Retraction to the starting point with G0 and deselection of the cutter radius compensation with G40

## Explanation of the parameters

### SPD and SPL (starting point)

The finished part diameter for the undercut is entered in the SPD parameter. You define the finished part dimensions in the longitudinal axis using the SPL parameter.

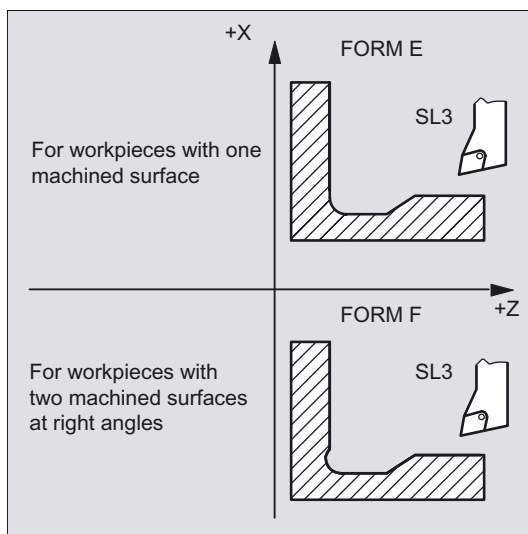
If a final diameter of <3 mm results for the value programmed for SPD, the cycle is canceled, and alarm 61601 "Finished part diameter too small" is issued.



### 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".



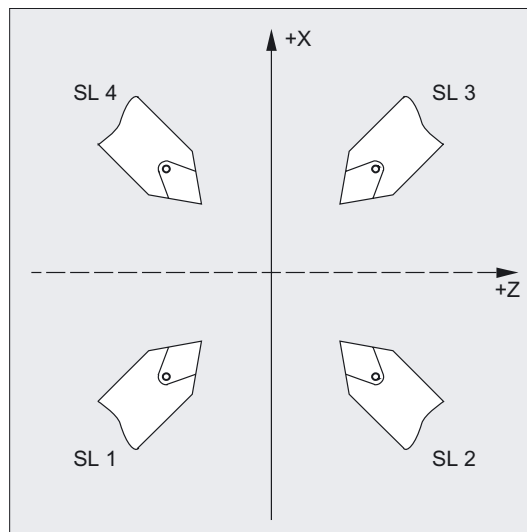
#### **\_VARI (undercut position)**

The position of the undercut can be either specified directly or derived from the tool point direction with the `_VARI` parameter.

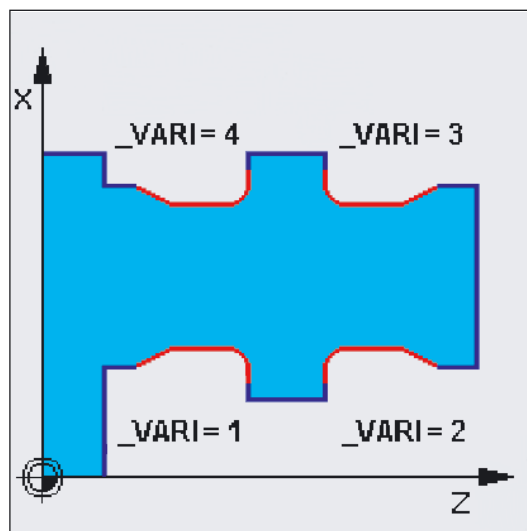
`VARI=0`: According to tool point direction

The tool point direction is determined by the cycle automatically from the active tool compensation. The cycle can then only work with tool point directions 1 to 4.

If the cycle detects a tool point direction between 5 and 9, the alarm 61608 "Wrong tool point direction programmed" is displayed and the cycle is aborted.



#### **\_VARI=1...4: Definition of undercut position**



When `_VARI` <> 0, the following applies:

- The actual tool point direction is not checked, i.e., all directions can be used if technologically suitable;
- No special consideration is given to functions such as adapter transformation or orientable toolholders in the cycle. It is assumed that the user is familiar with the conditions on the machine and has specified the position correctly;
- Similarly, mirroring is not specially treated, the user must be responsible for making the correct settings.

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.

---

**Note**

You must activate a tool offset before you call the cycles. Otherwise, the cycle is aborted after alarm 61000 "No tool compensation active" has been output.

---

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.

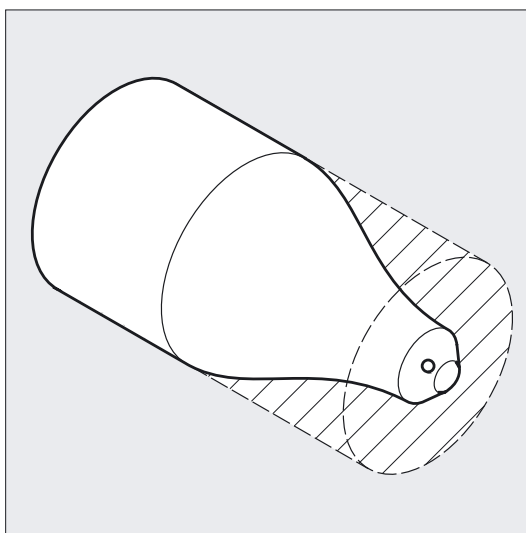


## 4.5 Stock removal cycle - CYCLE95

### Function

With the stock removal cycle, you can machine any user-programmed contour from a blank with 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 performed up to the final machining allowance programmed.

Finishing is performed in the same direction as roughing. The tool radius compensation is selected and deselected by the cycle automatically.



### Programming

CYCLE95 (NPP, MID, FALZ, FALX, FAL, FF1, FF2, FF3, VARI, DT, DAM, \_VRT)

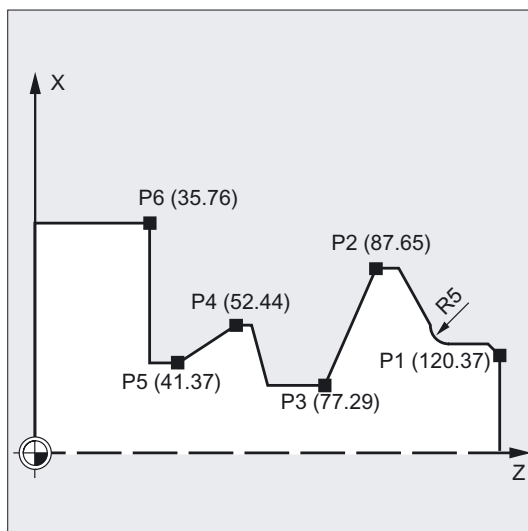
### Parameter

Parameter	Data type	Meaning
NPP	string	Name of contour subroutine
MID	real	Infeed depth (enter without sign)
FALZ	real	Finishing allowance in the longitudinal axis (enter without sign)
FALX	real	Finishing allowance in the transverse axis (enter without sign)
FAL	real	Finishing allowance according to the contour (enter without sign)

Parameter	Data type	Meaning
FF1	real	Feedrate for roughing without undercut
FF2	real	Feedrate for insertion into relief cut elements
FF3	real	Feedrate for finishing
VARI	integer	Machining type Range of values: 1...12, 201...212 HUNDREDS DIGIT: Values: 0: With rounding at the contour No residual corners remain, the contour is rounded with overlapping. This means that rounding is performed across multiple intersections. 2: Without rounding at the contour Contours are always rounded to the previous roughing intersection followed by retraction. Residual corners can remain, depending on the ratio of the tool radius to infeed depth (MID).
DT	real	Dwell time fore chip breaking when roughing
DAM	real	Path length after which each roughing step is interrupted for chip breaking
_VRT	real	Lift-off distance from contour when roughing, incremental (to be entered without sign)

### Stock removal cycle example 1

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.



DEF STRING[8] UPNAME	;Definition of a variable for the contour
	;name
N10 T1 D1 G0 G18 G95 S500 M3 Z125 X81	;Approach position before cycle call
UPNAME="CONTOUR_1"	;Assignment of subroutine name
N20 CYCLE95 (UPNAME, 5, 1.2, 0.6, , ->	;Cycle call
-> 0.2, 0.1, 0.2, 9, , , 0.5)	
N30 G0 G90 X81	;Reapproach starting position
N40 Z125	;Traverse axis by axis
N50 M30	;Program end
%_N_ KONTUR_1_SPF	;Beginning of contour subroutine
;\$PATH=/_N_SPF_DIR	
N100 G1 Z120 X37	;Traverse axis by axis
N110 Z117 X40	
N120 Z112	;Rounding with radius 5
N130 G1 Z95 X65 RND=5	;Traverse axis by axis
N140 Z87	
N150 Z77 X29	
N160 Z62	
N170 Z58 X44	
N180 Z52	
N190 Z41 X37	
N200 Z35	
N210 G1 X76	
N220 M17	;End of subroutine

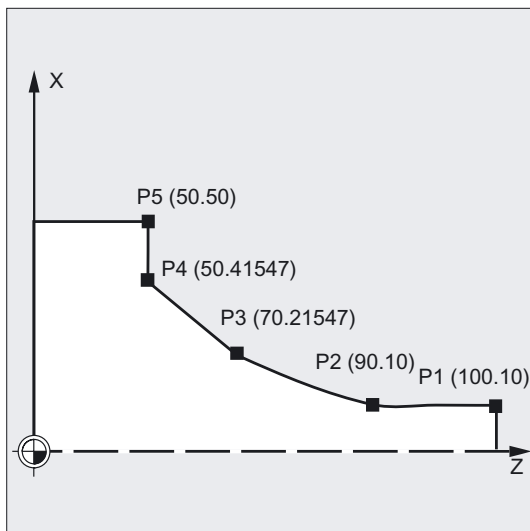
---

**Note**-> means: it must be programmed in a single block.

---

**Stock removal cycle example 2**

The stock removal contour is defined in the calling program. The program is completed after the stock removal cycle.



```

N110 G18 DIAMOF G90 G96 F0.8
N120 S500 M3
N130 T11 D1
N140 G0 X70
N150 Z60
N160 CYCLE95 ("ANFANG:ENDE",2.5,0.8, -> ;Cycle call
-> 0.8,0,0.8,0.75,0.6,1)
N170 M02
START:
N180 G1 X10 Z100 F0.6
N190 Z90
N200 Z=AC(70) ANG=150
N210 Z=AC(50) ANG=135
N220 Z=AC(50) X=AC(50)
END:
N230 M02

```

---

**Note**

-> means: it must be programmed in a single block.

---

## Sequence

### Position reached prior to cycle start:

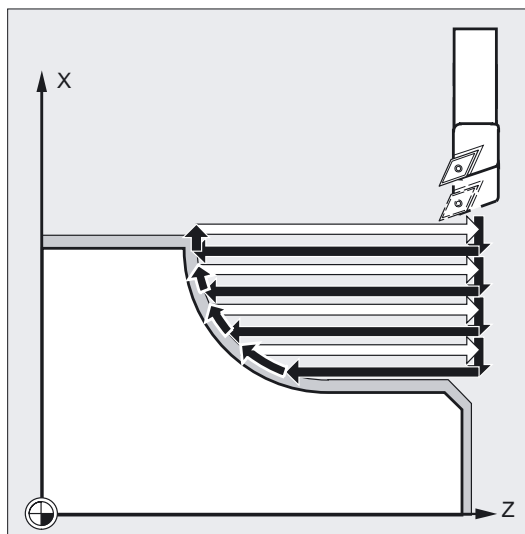
The starting position is any position from which the contour starting point can be approached without collision.

### 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.

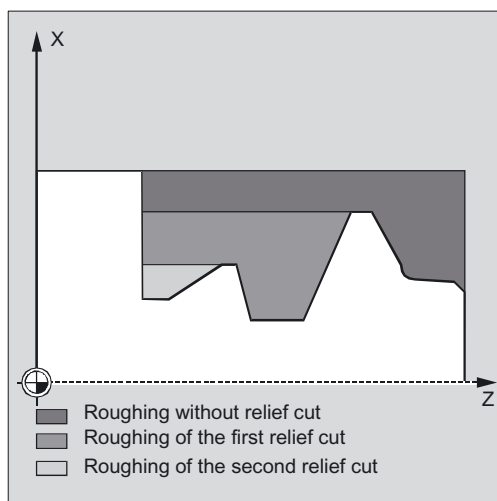
### Roughing without relief cut elements:

- 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.
- Round parallel to the contour at contour + final machining allowance to the last roughing intersection point with G1/G2/G3 and FF1.
- Lift-off 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.
- Round to the last roughing intersection. Lift off and retract as in the first machining section.
- If there are further relief cut elements, this sequence is repeated for each relief cut.

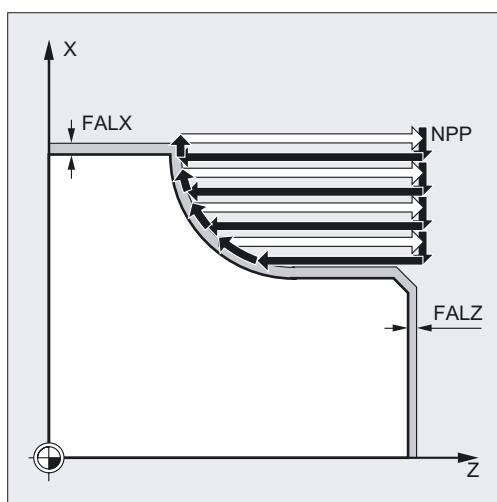
**Finishing:**

- The calculated cycle starting point is approached in both axes simultaneously with G0 and tool nose radius compensation is selected.
- Motion continues with both axes simultaneously and with G0 minus an amount equivalent to the final machining allowance + tool nose radius + 1 mm safety clearance ahead of the contour starting point, and from there with G1 to the contour starting point.
- Finish cutting along the contour with G1/G2/G3 and FF3.
- Retraction to starting point with both axes and G0.

**Explanation of the parameters****NPP (name)**

In this parameter, you enter either the name of the contour subroutine or the block number or labels for a program section. This contour subroutine must not be a subroutine with a parameter list.

The name of the contour subroutine is subject to all name conventions described in the Programming Guide.



**NOTICE**

The program from which CYCLE95 is called must not have the same name as the contour definition program.

**References:** /PG/ Programming Guide

The machining contour can also be a section of the calling routine or from any other program. The section is identified by start or end labels or by block numbers. Program name and labels/block number are indicated with ":".

**Examples:**

NPP="KONTUR_1"	;The stock removal contour is the ;complete "Contour_1" program.
NPP="ANFANG:ENDE"	;The stock removal contour is ;defined as a section in the calling ;program, which starts from the ;block containing label ANFANG to ;the block containing label ENDE.
NPP="/_N_SPF_DIR/_N_KONTUR_1_SPF:N130:N210"	;The stock removal contour is ;defined in blocks ;N130 to N210 in the ;CONTOUR_1 program. ;The program name must be ;entered complete with path and ;extension, see description of ;CALL in References: /PGA/ ;Programming Guide ;Advanced.

**Note**

If the section is defined by block numbers, it must be noted that, for the NPP section, these block numbers must be adjusted if the program is modified and subsequently renumbered.

**MID (infeed depth)**

You define the maximum possible infeed depth for the roughing operation in the MID parameter. With SW 4 and higher, the interpretation of this parameter depends on the cycle setting data \_ZSD[0] (see Section "Requirements").

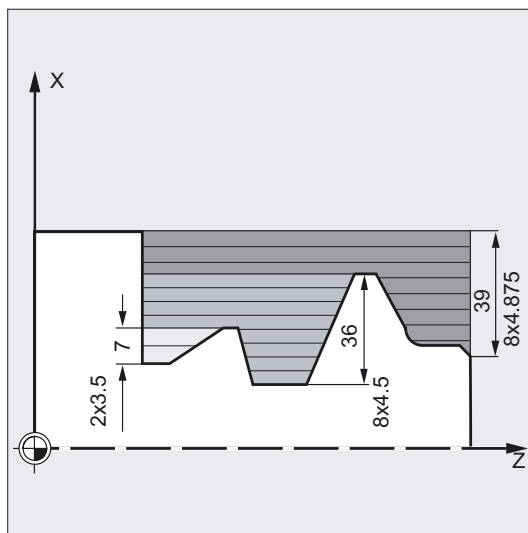
The cycle automatically calculates the current infeed depth for roughing. With contours containing relief cut elements, the roughing process is divided by the cycle into individual roughing sections. The cycle calculates a new current infeed depth for each roughing section. This infeed depth is always 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. The machining steps shown in the figure above are used for roughing this contour.

Example of calculating the current infeed depth:

Machining section 1 has a total depth of 39 mm. If the maximum infeed depth is 5 mm, eight roughing cuts are required. 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.

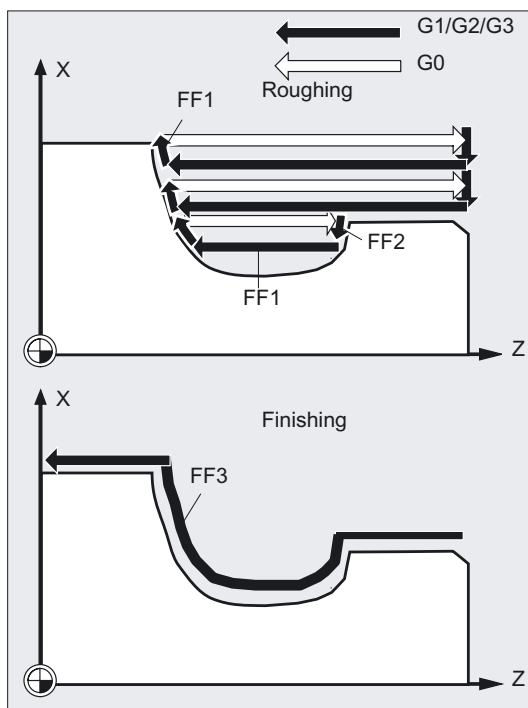
Roughing is always carried out up to these finishing allowances. After each paraxial roughing operation, the resulting residual corners are immediately cut parallel to the contour, so that these do not have to be removed after the roughing operation is completed. If no finishing allowances are programmed, stock is removed when roughing up to the final contour.

The finishing machining mode does not consider programmed values for the final machining allowance. Turning always takes place to the finishing dimension.



### FF1, FF2 and FF3 (feedrate)

You can define different feedrates for the different machining steps, as shown in the figure on the right.



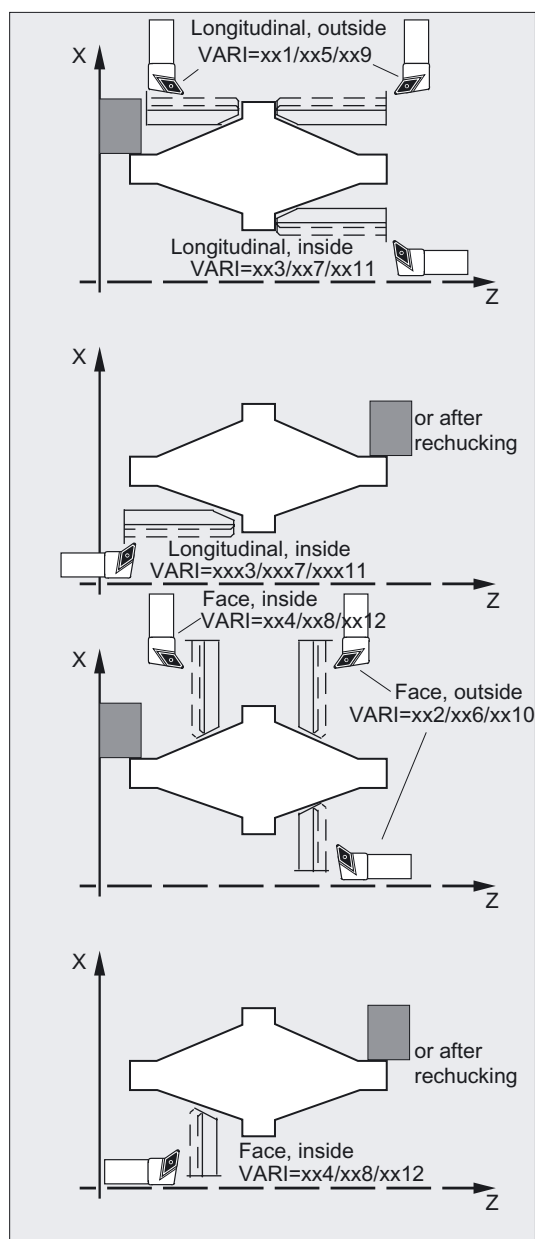
### VARI (machining type)

You can call the machining type in the cycle support as follows:

Machining	<input type="text" value="Roughing"/>	(roughing/finishing/complete machining)
Selection	<input type="text" value="Longitudinal"/>	(longitudinal/face)
Selection	<input type="text" value="Outside"/>	(outside/inside)

The machining type can be found in the table below.

Value	Processing	Selection	Selection
1/201	Roughing	Longitudinal	External
2/202	Roughing	Transverse	External
3/203	Roughing	Longitudinal	Internal
4/204	Roughing	Transverse	Internal
5/205	Finishing	Longitudinal	External
6/206	Finishing	Transverse	External
7/207	Finishing	Longitudinal	Internal
8/208	Finishing	Transverse	Internal
9/209	Complete machining	Longitudinal	External
10/210	Complete machining	Transverse	External
11/211	Complete machining	Longitudinal	Internal
12/212	Complete machining	Transverse	Internal

**Note**

In longitudinal machining, the infeed is always carried out along the transversal axis, and in face machining - along the longitudinal axis.

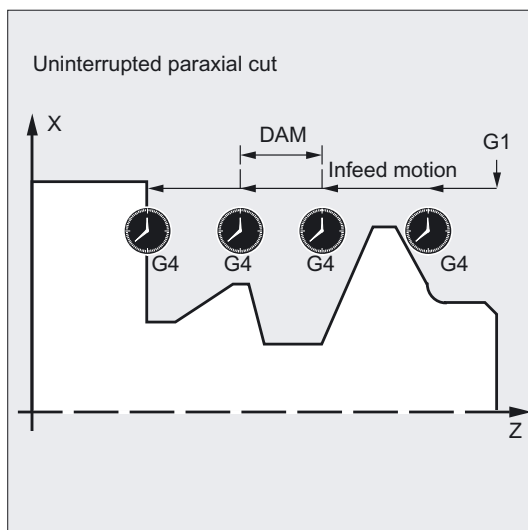
External machining means that the infeed is carried out in the direction of the negative axis. With internal machining, the infeed is carried out in the direction of the positive axis.

When roughing in CYCLE95, you can choose between "with rounding" or "without rounding" at the contour. In order to do so, the HUNDREDS DIGIT is incorporated in the VARI parameter.

The VARI parameter is subjected to a plausibility check. If you select an invalid value, the cycle is aborted and alarm 61002 "Incorrect machining type defined" is output.

**DT and DAM (dwell time and path length)**

With these two parameters, you can program an interruption of the individual roughing cuts after a defined displacement for the purposes of 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, you can also program a dwell time to be executed at each of the interruption points. If no distance is specified for the cut interruption (DAM=0), uninterrupted roughing steps without dwell times are created.

**\_VRT (lift-off distance)**

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), liftoff is 1 mm. The lift-off distance is always measured in inches or in metric units depending on the programmed measuring system, i.e., \_VRT=1 for inch -> programming produces a lift-off of 1 inch.

The maximum lift-off clearance from the contour in the cycle is the setting for \_VRT + finishing allowance. This is particularly important with respect to internal machining.

---

**Note****Contour definition**

You program the contour in the form of NC blocks. The contour must contain at least 3 blocks with motions in the two axes of the machining plane.

The machining plane (G17, G18, G19) is set in the main program before the cycle is called or applied according to the basic setting of this G-group on the machine. It cannot be altered in the contour.

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 may contain motions for other axes; however, their travel distances do not take effect during execution of the 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 a traversing motion in the current machining plane must contain a motion command G0, G1, G2 or G3; otherwise, the cycle is canceled, and alarm 15800 "Incorrect prerequisites for CONTPRON" is issued. This alarm is also activated when G41/G42 is active.

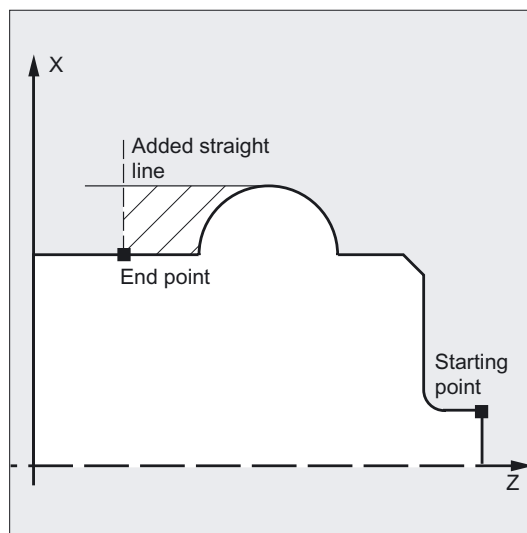
The starting point of the contour is the first position in the machining plane programmed in the contour.

The maximum number of blocks with movements in the plane permissible in the contour depends on the type of contour. In principle, there is no limit to the possible number of relief cuts.

If a contour contains more contour elements than the cycle-internal memory can hold, the cycle is aborted with alarm 10934 "Contour table overrun".

Machining must then be divided into several machining sections, each of which is represented by its own contour subroutine and each cycle is called separately.

If the maximum diameter in one contour is not located within the programmed end point or starting point of the contour, the cycle automatically extends a paraxial straight line extending up to the maximum point of the contour at the end of the machining operation and this part of the contour is then removed as a relief cut.



### 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.

If only finishing is selected, the contour is always traversed in the programmed direction.

When deciding on the machining direction, the first and the last programmed contour points are taken into account. For this reason, both coordinates must always be programmed in the first block of the contour subroutine.

### Contour monitoring

The cycle performs contour monitoring of the following:

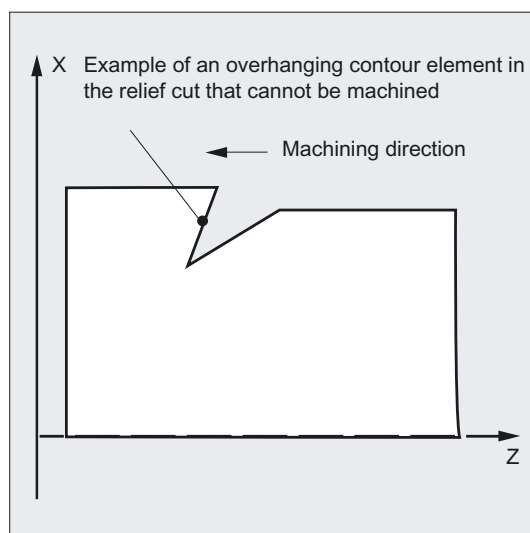
- Clearance angle of the active tool
- Circular programming of arcs with an arc angle > 180 degrees

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.

Overhanging contours cannot be machined by CYCLE95. Contours of this type are not monitored by the cycle and consequently there is no alarm.



### Start point

The cycle determines the starting point for the machining operation automatically. The starting point is located in the axis in which the depth infeed is carried out, shifted from the contour by the amount of the finishing allowance + lift-off distance (parameter `_VRT`). 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.

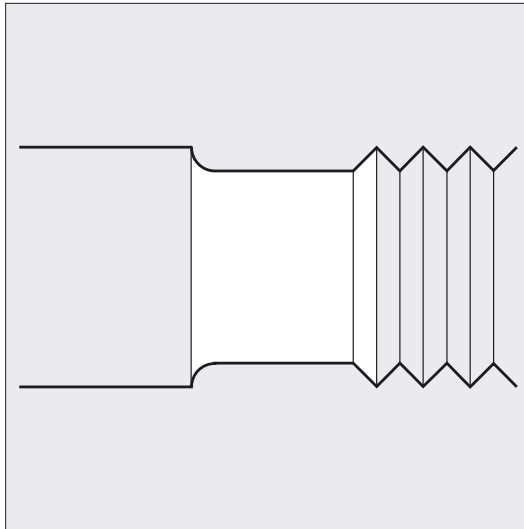
### Cycle approach strategy

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.

## 4.6 Thread undercut - CYCLE96

### Function

This cycle is for machining thread undercuts in accordance with DIN 76 on parts with a metric ISO thread.



### Programming

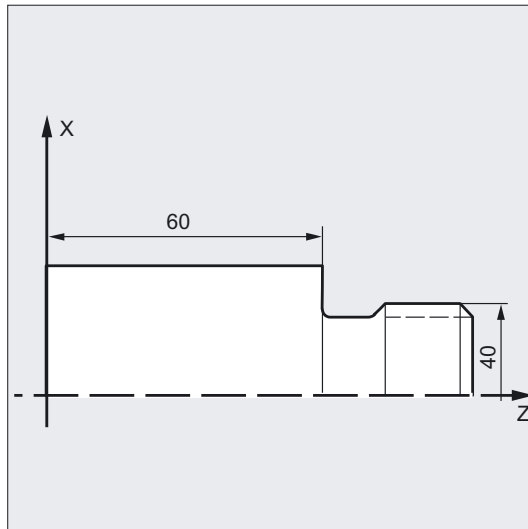
CYCLE96 (DIATH, SPL, FORM, \_VARI)

### Parameter

Parameter	Data type	Meaning
DIATH	real	Nominal diameter of the thread
SPL	real	Starting point on the contour of the longitudinal axis
FORM	char	Definition of the form Values: A (for form A) B (for form B) C (for form C) D (for form D)
_VARI	integer	Specification of undercut position Values: 0: Corresponding to tool point direction 1 to 4: Define position

**Example of Thread Undercut\_Form\_A**

This program can be used to program a thread undercut of form A.



N10 D3 T1 S300 M3 G95 F0.3	;Specification of technology values
N20 G0 G18 G90 Z100 X50	;Selection of starting position
N30 CYCLE96 (10, 60, "A")	;Cycle call
N40 G90 G0 X30 Z100	;Approach next position
N50 M30	;End of program

**Flow****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 the 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 tool radius compensation with G40.

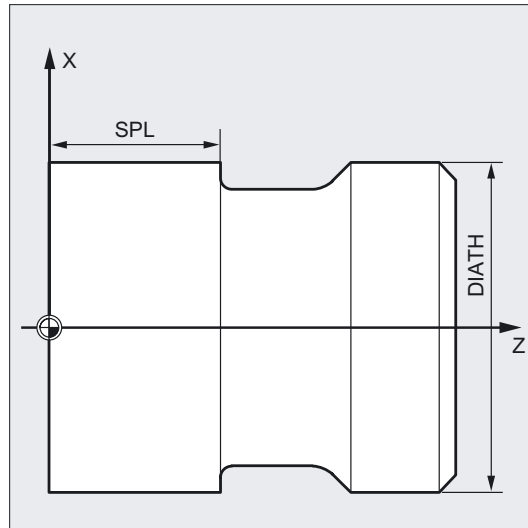
**Explanation of the parameters****DIATH (nominal diameter)**

With this cycle, you can machine thread undercuts for metric ISO threads from M3 to M68.



If a final diameter of  $<3$  mm results for the value programmed for DIATH, the cycle is canceled, and alarm 61601 "Finished part diameter too small" is issued.

If the parameter has a value other than that specified in DIN76 Part 1, the cycle is also canceled, generating alarm 61001 "Thread pitch incorrectly defined".

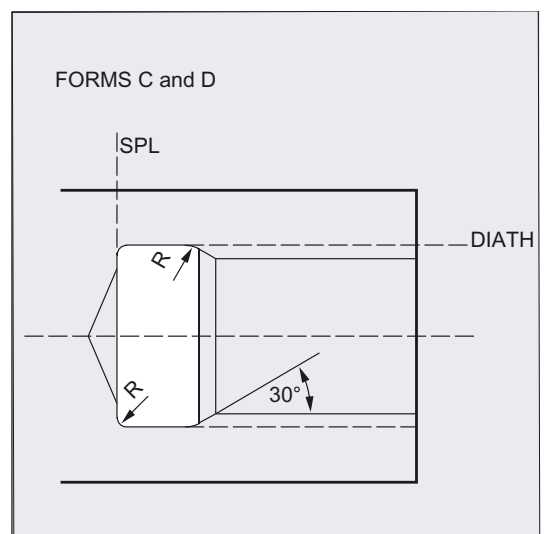
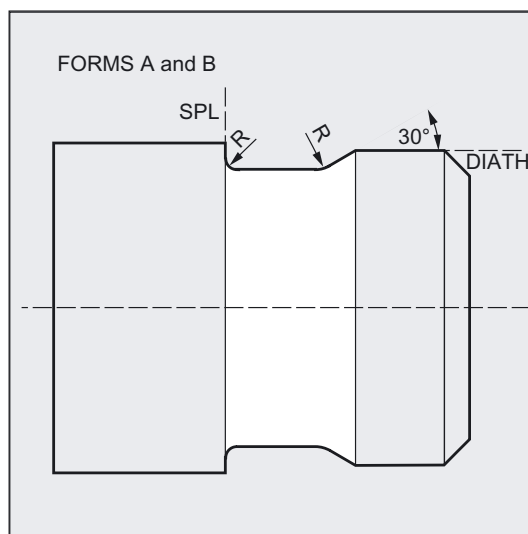


#### SPL (starting point)

The finished dimension in the longitudinal axis is defined using the parameter SPL.

#### 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.

**\_VARI (undercut position)**

The position of the undercut can be either specified directly or derived from the tool point direction with the \_VARI parameter. See \_VARI for CYCLE94.

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.

---

**Note**

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.

---

## 4.7 Thread cutting - CYCLE97

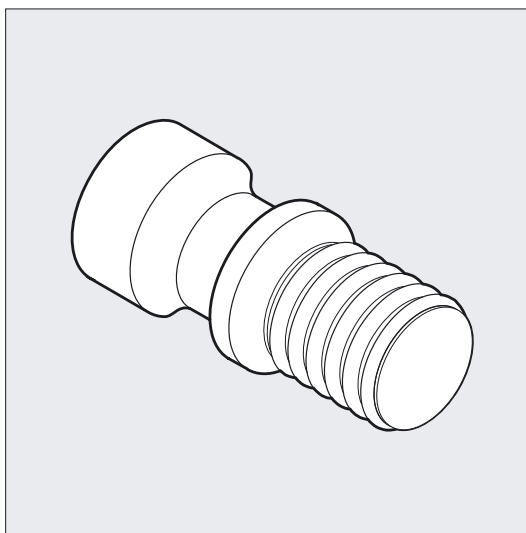
### Function

With this thread cutting cycle, you can machine cylindrical and tapered outside and inside threads with constant pitch in longitudinal or face machining. The thread can be single or multiple. With multiple threads, the individual thread turns are machined one after the other.

Infeed is automatic. You can select either constant infeed per cut or constant cross-section of cut.

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.

Neither feedrate nor spindle override have any effect in thread travel blocks. The spindle override must not be changed during thread machining.



**References:** /PG/, Programming Manual: Fundamentals, Section "Thread cutting with constant lead, G33"

---

### Note

To be able to use this cycle, a speed-controlled spindle with position measuring system is required.

---

### Programming

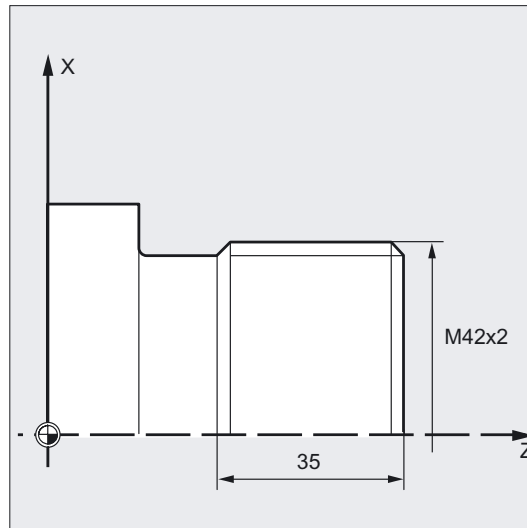
CYCLE97 (PIT, MPIT, SPL, FPL, DM1, DM2, APP, ROP, TDEP, FAL, IANG, NSP, NRC, NID, VARI, NUMT, \_VRT)

**Parameter**

Parameter	Data type	Meaning
PIT	real	Thread pitch as a value (enter without sign)
MPIT	real	Thread pitch as thread size Range of values: 3 (for M3) ... 60 (for M60)
SPL	real	Thread starting point in the longitudinal axis
FPL	real	Thread end point in the longitudinal axis
DM1	real	Thread diameter at the starting point
DM2	real	Thread diameter at the end point
APP	real	Run-in path (enter without sign)
ROP	real	Run-out path (enter without sign)
TDEP	real	Thread depth (enter without sign)
FAL	real	Finishing allowance (enter without sign)
IANG	real	Infeed angle Range of values: "+" (for flank infeed at the flank) "-" (for alternating flank infeed)
NSP	integer	Starting point offset for the first thread turn (enter without sign)
NRC	integer	Number of roughing cuts (enter without sign)
NID	integer	Number of idle passes (enter without sign)
VARI	integer	Definition of the machining type for the thread Range of values: 1 ... 4
NUMT	integer	Number of thread turns (enter without sign)
_VRT	real	Variable retraction distance based on initial diameter, incremental (enter without sign)

### Example of thread cutting

With this program, you can cut an M42x2 metric outside thread 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.



```
DEF REAL MPIT=42, SPL=0, FPL=-35,           ;Definition of parameters with
DM1=42, DM2=42, APP=10, ROP=3,              ;value assignments
TDEP=1.23, FAL=0, IANG=30, NSP=0
DEF INT NRC=5, NID=2, VARI=3, NUMT=1
N10 G0 G18 G90 Z100 X60                     ;Selection of starting position
N20 G95 D1 T1 S1000 M4                       ;Specification of technology values
N30 CYCLE97 ( , MPIT, SPL, FPL, DM1, ->      ;Cycle call
-> DM2, APP, ROP, TDEP, FAL, IANG, ->
-> NSP, NRC, NID, VARI, NUMT)
N40 G90 G0 X100 Z100                         ;Approach next position
N50 M30                                     ;End of program
```

### Note

-> means: it must be programmed in a single block.

## 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.

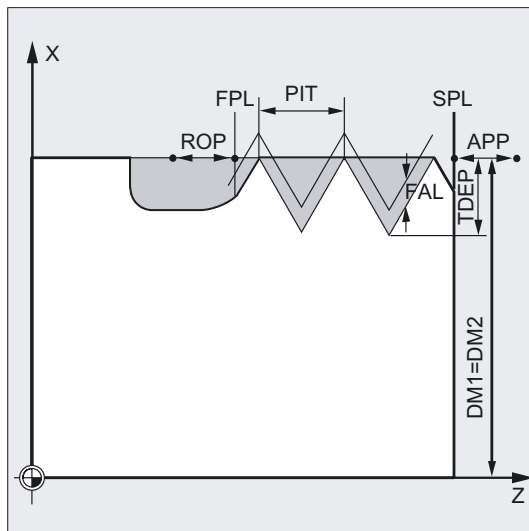
### The cycle creates the following sequence of motions:

- 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)

This parameter is used to set the thread diameter of the start and end points 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 lift-off plane is generated automatically within the control system.

#### 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 VARI parameter.

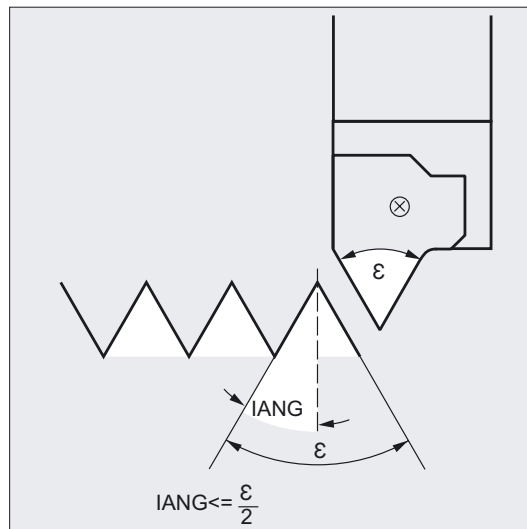
When the thread depth is divided into infeeds with constant cutting cross-section, the cutting force will remain constant over all roughing cuts. In this case, the infeed will be performed using different values for the infeed depth.

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.

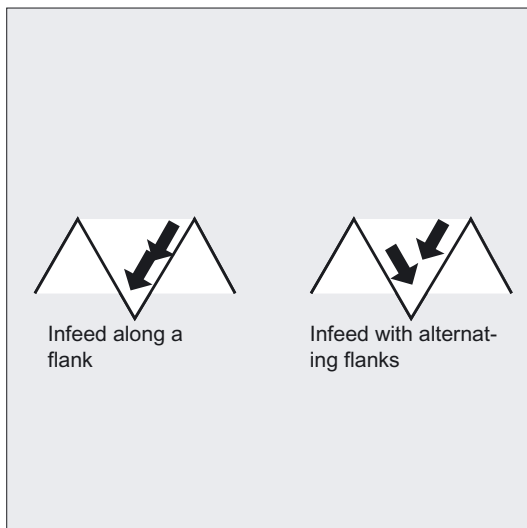
The finishing allowance FAL is removed after roughing in one step. Then the idle passes programmed under parameter NID are executed.

#### IANG (infeed angle)

You define the infeed angle with the IANG parameter. 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 can 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.

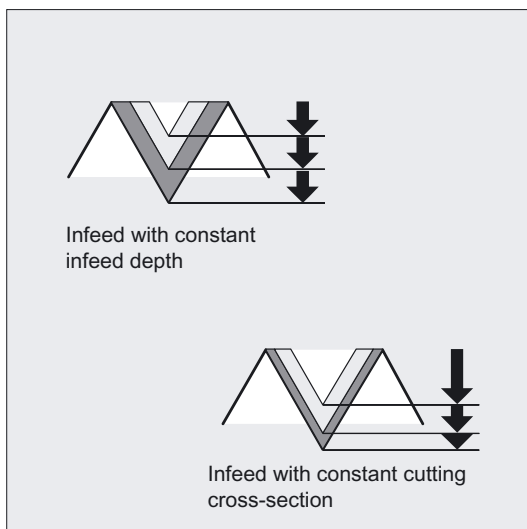


#### NSP (starting point offset)

With this parameter, you can program the angular value that defines the point of the first cut of the first thread start on 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.

#### VARI (machining type)

With the VARI parameter, you define if machining is to be internal or external and, with which technology the infeed will be machined during roughing. The VARI parameter can assume values between 1 and 4 with the following meaning:





Value	external/internal	Const. infeed/const. cutting cross-section
1	External	Constant infeed
2	Internal	Constant infeed
3	External	Constant cutting cross-section
4	Internal	Constant cutting cross-section

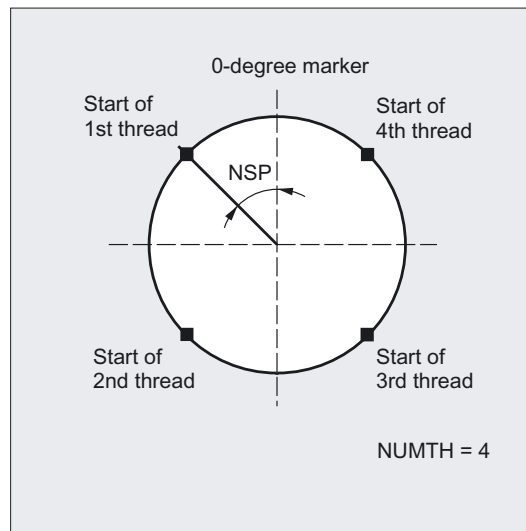
If a different value is programmed for the VARI parameter, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

#### NUMT (number of thread turns)

You specify the number of thread starts for a multiple thread with the NUMT parameter. 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 NSP parameter.

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.



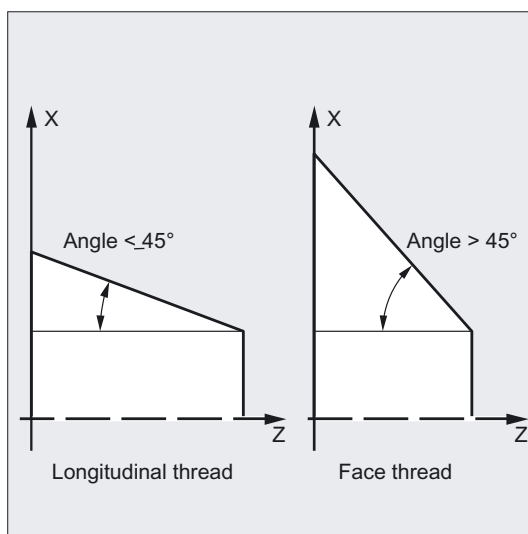
#### \_VRT (variable retraction path)

The retraction path can be programmed on the basis of the initial thread diameter in the \_VRT parameter. When \_VRT = 0 (parameter not programmed), the retraction path is 1 mm. The retraction path is always measured according to the programmed system of units, inch or metric.

**Note****Differentiation between longitudinal and face thread**

The decision whether a longitudinal or face 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  $\leq 45$  degrees, the longitudinal axis thread is machined, otherwise it will be the face thread.

---



## 4.8 Thread chaining - CYCLE98

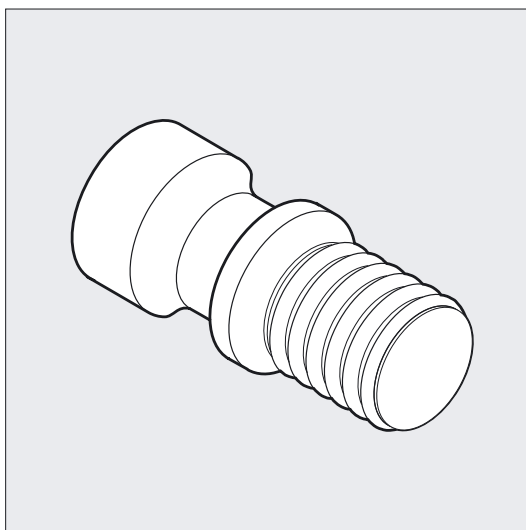
### Function

With this cycle, you can produce several concatenated cylindrical or tapered threads with a constant pitch in longitudinal or face machining, all of which can have different thread pitches.

The thread can be single or multiple. With multiple threads, the individual thread turns are machined one after the other.

Infeed is automatic. You can select either constant infeed per cut or constant cross-section of cut. 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.

Neither feedrate nor spindle override have any effect in thread travel blocks. The spindle override must not be changed during thread machining.



**References:** /PG/, Programming Manual: Fundamentals, Section “Thread cutting with constant lead, G33”

### Programming

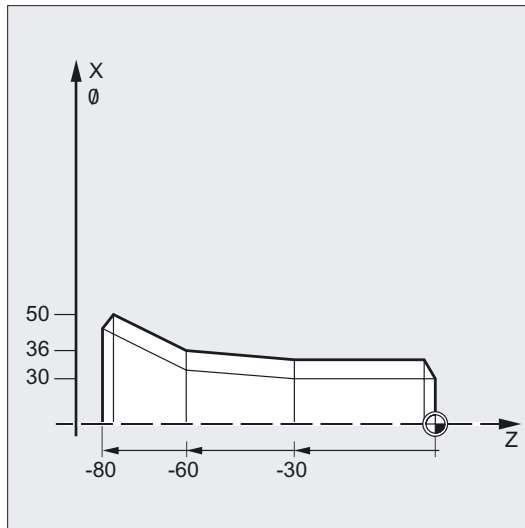
CYCLE98 (PO1, DM1, PO2, DM2, PO3, DM3, PO4, DM4, APP, ROP, TDEP, FAL, IANG, NSP, NRC, NID, PP1, PP2, PP3, VARI, NUMT, \_VRT)

**Parameter**

Parameter	Data type	Meaning
PO1	real	Thread starting point in the longitudinal axis
DM1	real	Thread diameter at the starting point
PO2	real	First intermediate point in the longitudinal axis
DM2	real	Diameter at the first intermediate point
PO3	real	Second intermediate point
DM3	real	Diameter at the second intermediate point
PO4	real	Thread end point in the longitudinal axis
DM4	real	Diameter at the end point
APP	real	Run-in path (enter without sign)
ROP	real	Run-out path (enter without sign)
TDEP	real	Thread depth (enter without sign)
FAL	real	Finishing allowance (enter without sign)
IANG	real	Infeed angle Range of values: "+" (for flank infeed at the flank) "-" (for alternating flank infeed)
NSP	real	Starting point offset for the first thread turn (enter without sign)
NRC	integer	Number of roughing cuts (enter without sign)
NID	integer	Number of idle passes (enter without sign)
PP1	real	Thread pitch 1 as a value (enter without sign)
PP2	real	Thread pitch 2 as a value (enter without sign)
PP3	real	Thread pitch 3 as a value (enter without sign)
VARI	integer	Definition of the machining type for the thread Range of values: 1 ... 4
NUMT	integer	Number of thread turns (enter without sign)
_VRT	real	Variable retraction distance based on initial diameter, incremental (enter without sign)

### Example of thread chain

With this program, you can produce a chain of threads, starting with a cylindrical thread. The infeed is performed vertically to the thread; neither finishing allowance, nor starting point offset are programmed. Five roughing cuts and one noncut are performed. The machining type is defined as longitudinal, external, with constant cross-section of cut.



```
N10 G18 G95 T5 D1 S1000 M4 ;Specification of technology values
N20 G0 X40 Z10 ;Approach starting position
N30 CYCLE98 (0, 30, -30, 30, -60, -> ;Cycle call
-> 36, -80, 50, 10, 10, 0.92, , , ->
-> 5, 1, 1.5, 2, 2, 3, 1)
N40 G0 X55 ;Traverse axis by axis
N50 Z10
N60 X40
N70 M30 ;End of program
```

### Note

-> means: it must be programmed in a single block.

**Flow****Position reached prior to cycle start:**

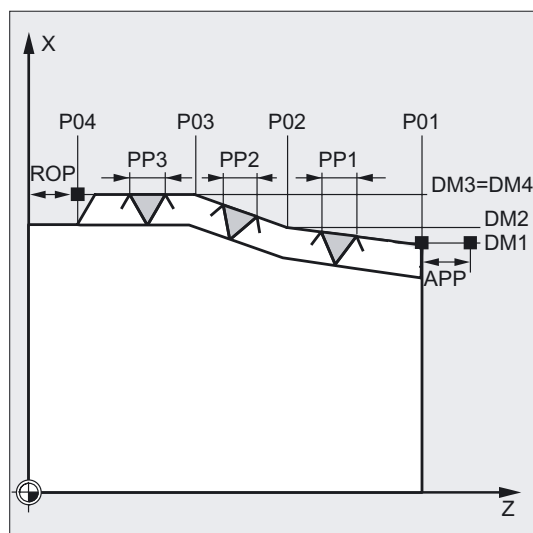
Starting position is any position from which the programmed thread starting point + run-in path can be approached without collision.

**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****PO1 and DM1 (starting point and diameter)**

You define the original starting point of the thread chain using these parameters. 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)**

You define two intermediate points in the thread using these parameters.

**PO4 and DM4 (end point and diameter)**

The original end point of the thread is programmed under parameters PO4 and DM4.

---

**Note**

With an inside thread, DM1 to DM4 corresponds to 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 lift-off plane is generated automatically within the control system.

#### Interrelation between TDEP, FAL, NRC and NID (thread depth, finishing allowance, number of roughing and idle passes)

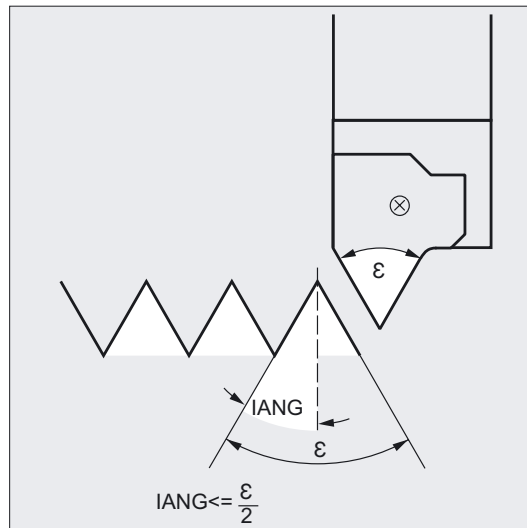
The programmed finishing allowance acts paraxially and is subtracted from the specified thread depth TDEP; the remainder is divided into roughing cuts. The cycle will calculate the individual infeed depth automatically, depending on the VARI parameter. The thread depth to be machined is divided into infeeds with the same cross-section of cut so that the cutting pressure remains constant for all rough 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.

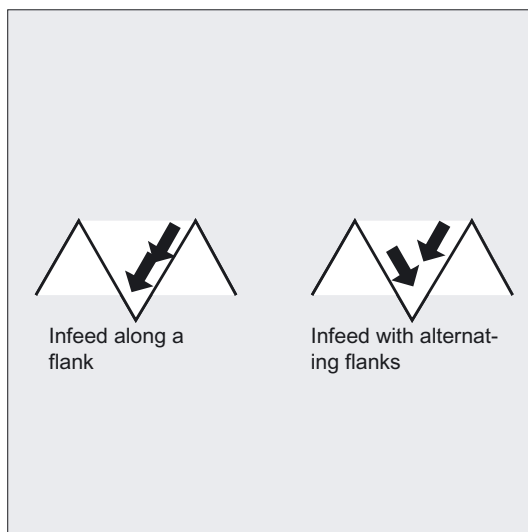
The finishing allowance FAL is removed after roughing in one step. Then the idle passes programmed under parameter NID are executed.

#### IANG (infeed angle)

You define the infeed angle with the IANG parameter. 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 can 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)**

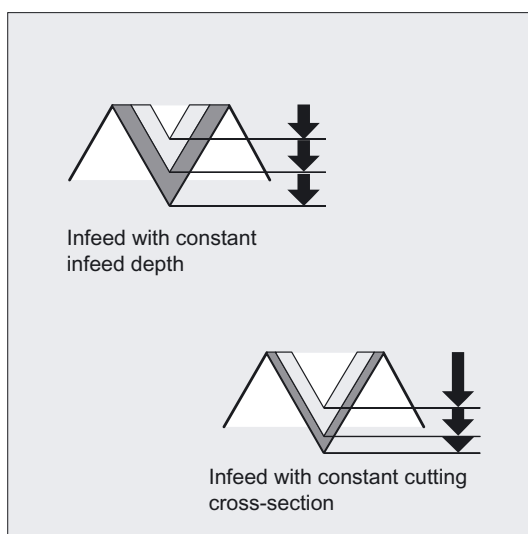
With this parameter, you can program the angular value that defines the point of the first cut of the first thread start on the circumference of the turned part. This is a starting point offset. The parameter can assume values between 0.0001 and +359.9999 degrees. If no starting point offset is specified 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)**

With these parameters, you determine the thread pitch from the three sections of the thread chain. The pitch value must be entered as a paraxial value without sign.

**VARI (machining type)**

With the VARI parameter, you define if machining is to be internal or external and, with which technology the infeed will be machined during roughing. The VARI parameter can assume values between 1 and 4 with the following meaning:





Value	external/internal	Const. infeed/const. cutting cross-section
1	External	Constant infeed
2	Internal	Constant infeed
3	External	Constant cutting cross-section
4	Internal	Constant cutting cross-section

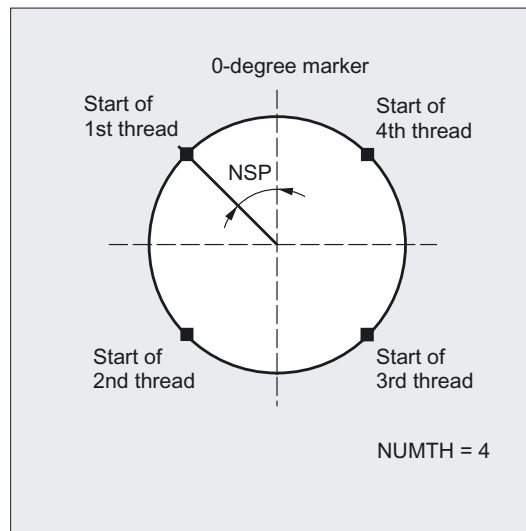
If another value is assigned to the VARI parameter, the cycle is aborted and the following alarm: 61002 "Machining type incorrectly programmed".

#### NUMT (number of thread turns)

You specify the number of thread starts for a multiple thread with the NUMT parameter. 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 NSP parameter.

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.



#### \_VRT (variable retraction path)

The retraction path can be programmed on the basis of the initial thread diameter in the \_VRT parameter. When \_VRT = 0 (parameter not programmed), the retraction path is 1 mm. The retraction path is always measured according to the programmed system of units, inch or metric.

## 4.9 Thread recutting

### Function

The angular offset of a thread start resulting from tool breakage or remeasurement is taken into account and compensated for by the "Thread recut" function. This function can be executed in JOG mode in the Machine operating area.

The cycles use the data stored in the thread start during synchronization to calculate an additional offset angle for each thread, which is applied in addition to the programmed starting point offset.

### Note

Thread cutting cycles CYCLE97 and CYCLE98 can be used for thread recutting.

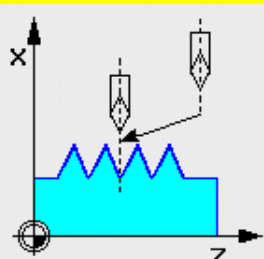
### Conditions

The channel in which the thread recutting program is to be executed is selected; the relevant axes must already be referenced. The channel is in the Reset state, the spindle is stationary.

### Flow

- Select JOG in "Machine" operating area.
- Press "Recut thread" soft key -> open screen form for this function.

**Finish thread**
**Select plane G17, G18, G19**



**Select plane**

G18

<b>Spindle position</b>	<b>C</b>	<b>0.000</b>	<b>grd</b>
<b>Position</b>	<b>Z</b>	<b>0.000</b>	<b>mm</b>
<b>Position</b>	<b>X</b>	<b>0.000</b>	<b>mm</b>

- Thread into thread start using the threading tool.
- When the cutting tool is positioned exactly in the thread start, press the "Sync Point" soft key.
- Press the "Cancel" soft key to return to the upper level soft key menu without activating the function (no data are then stored in the NC).
- Select the "OK" soft key to transfer all values to the GUD in the NC.
- Then retract the tool and move it to its starting position.
- Select "Automatic" and, using block search, position the program pointer in front of the thread cycle call.
- Start the program with NC start.

## Miscellaneous functions

You can delete values stored earlier by selecting another soft key labeled "Delete".

If several spindles are operating in the channel, another selection box is displayed in the screen form, in which you can select a spindle to machine the thread.

## Start up

Thread recutting requires operation in the JOG basic display. This is activated with the "Finish thread" HS8 soft key in file MA\_JOG.COM.

- Startup of thread recutting for HMI Advanced:

First open the MA\_JOG.COM file and then delete the semicolon preceding the following lines:

- ;HS8=(\$80720,,se1)
- ;PRESS(HS8)
- ;LM("GENS","drehen2.com")
- ;END\_PRESS

The file is located in the "Standard Cycles" directory. The soft key is then active. The HMI must then be restarted.

- Startup of thread recutting for HMI Embedded:

First open the COMMON.COM file and then delete ";" in front of SC108. The file is located in the "User Cycles" directory. The HMI must then be restarted.

## 4.10 Extended stock removal cycle - CYCLE950

### Function

With the CYCLE950 extended stock removal cycle, you can machine a contour programmed with paraxial or parallel-contour stock removal. Any blank can be defined and is considered during stock removal. The finished part contour must be continuous and may contain any number of relief cut elements. You can specify a blank as a contour or by means of axial values.

Contours can be machined longitudinally or facing with this cycle. You can freely select a technology (roughing, finishing, complete machining, machining and infeed directions). It is possible to update a blank.

For roughing, the programmed infeed depth is observed precisely; the last two roughing steps are divided equally. Roughing is performed up to the final machining allowance programmed. Finishing is performed in the same direction as roughing. The tool radius compensation is selected and deselected by the cycle automatically.

---

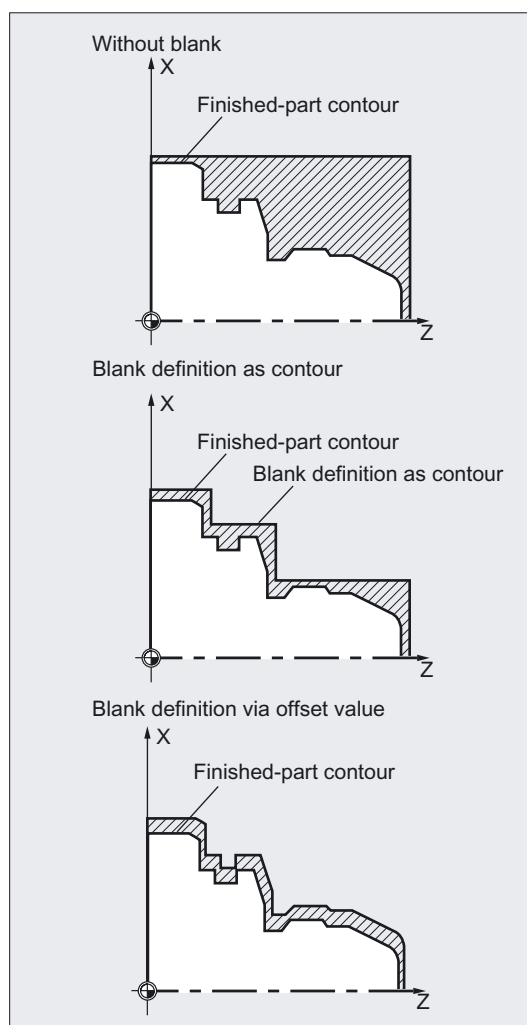
#### Note

The extended stock removal cycle is an option. It requires SW 6 in both the NCK and HMI Advanced.

---

**New functions compared to CYCLE95:**

- You can define a blank either by programming a contour, specifying an allowance on the finished part contour or entering a blank cylinder (or hollow cylinder in the case of internal machining), from which stock must be removed.
- It is possible to detect residual material that cannot be machined with the active tool. The cycle can generate an updated blank contour, which is stored as a program in the parts program memory.
- The contours for stock removal can be specified in any of the following ways:
  - In a separate program
  - In the calling main program
  - As a section of any program
- For roughing, it is possible to choose between paraxial and contour-parallel machining.
- During roughing, you have the option of machining along the contour so that no corners are left over, or removing stock immediately at the roughing intersection.
- The angle for stock removal at the contour during roughing can be programmed.
- Optionally, relief cuts can be machined or skipped during roughing.



## Programming

CYCLE950 (\_NP1, \_NP2, \_NP3, \_NP4, \_VARI, \_MID, \_FALZ, \_FALX, \_FF1, \_FF2, \_FF3, \_FF4, \_VRT, \_ANGB, \_SDIS, \_NP5, \_NP6, \_NP7, \_NP8, \_APZ, \_APZA, \_APX, \_APXA, \_TOL1)

## Parameter

Parameter	Data type	Meaning
_NP1	string	Name of the contour subroutine for the finished part contour
_NP2	string	Label/block number start of finished part contour, optional (this can be used to define contour sections)
_NP3	string	Label/block number end of finished part contour, optional (this can be used to define contour sections)
_NP4	string	Name of the stock removal program to be generated
_VARI	integer	Machining type: (enter without sign)

Parameter	Data type	Meaning
		<p>Values:</p> <p>UNITS DIGIT: Stock removal type 1: longitudinal 2: transversal 3: parallel to the contour</p> <p>TENS DIGIT: Infeed direction 1: programmed infeed direction X- 2: programmed infeed direction X+ 3: programmed infeed direction Z- 4: programmed infeed direction Z+</p> <p>HUNDREDS DIGIT: Machining process 1: Roughing 2: Finishing 3: Complete</p> <p>THOUSANDS DIGIT: Rounding 1: With rounding 2: Without rounding (lift-off)</p> <p>TEN THOUSANDS DIGIT: Relief cuts 1: Machining of relief cuts 2: No machining of relief cuts</p> <p>HUNDRED THOUSANDS DIGIT: Machining direction 1: programmed machining direction X- 2: programmed machining direction X+ 3: programmed machining direction Z- 4: programmed machining direction Z+</p>
_MID	real	Infeed depth (enter without sign)
_FALZ	real	Finishing allowance in the longitudinal axis (enter without sign)
_FALX	real	Finishing allowance in the transverse axis (enter without sign)
_FF1	real	Feedrate for longitudinal roughing
_FF2	real	Feedrate for face roughing
_FF3	real	Feedrate for finishing
_FF4	real	Feedrate at contour transition elements (radius, chamfer)
_VRT	real	Lift-off distance for roughing, incremental (enter without sign)
_ANGB	real	Lift-off angle for roughing
_SDIS	real	Safety clearance for avoiding obstacles, incremental
_NP5	string	Name of contour program for blank contour
_NP6	string	Label/block number start of blank contour, optional (this can be used to define contour sections)
_NP7	string	Label/block number end of blank contour, optional (this can be used to define contour sections)
_NP8	string	Name of contour program for updated blank contour
_APZ	real	Axial value for defining blank for longitudinal axis
_APZA	integer	Evaluation of parameter _APZ absolute or incremental 90 = absolute, 91 = incremental
_APX	real	Axial value for defining blank for facing axis
_APXA	integer	Evaluation of parameter _APX absolute or incremental 90 = absolute, 91 = incremental
_TOL1	real	Blank tolerance

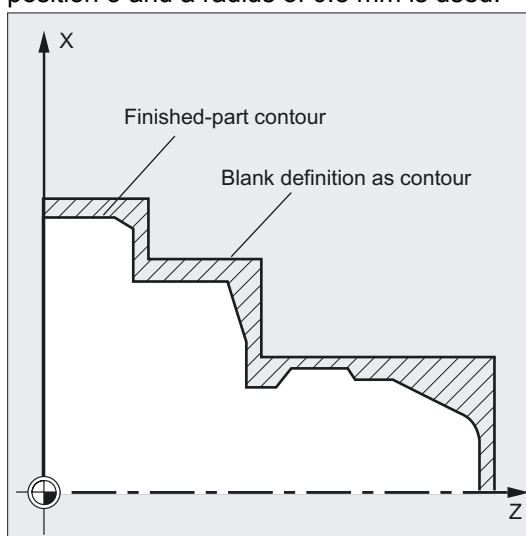
**Example 1**

The contour saved in the PART1.MPF program is to be machined from a preformed blank.

The type of machining for the stock removal process is

- Roughing only,
- Longitudinal,
- External,
- With rounding (so that no corners are left over),
- Relief cuts are to be machined.

The blank contour is specified in the BLANK1.MPF program. A turning tool with tool point position 3 and a radius of 0.8 mm is used.

**Machining program:**

```
%_N_EXAMPLE_1_MPF
; $PATH=/_N_WKS_DIR/_N_STOCK_REMOVAL_NEW_WPD
; Example 1: Stock removal with blank
; Sca, 01.04.99
;
; Tool offset data
N10 $TC_DP1[3,1]=500 $TC_DP2[3,1]=3 $TC_DP6[3,1]=0.8
$TC_DP24[3,1]=60
N15 G18 G0 G90 DIAMON
N20 T3 D1
N25 X300
N30 Z150
N35 G96 S500 M3 F2
N45 CYCLE950("Part1",,, "Machined_Part1", ->
-> 311111,1.25,1,1,0.8,0.7,0.6,0.3,0.5,45,2,"Blank1",,,,,,1)
N45 G0 X300
N50 Z150
N60 M2
```

**Finished part contour:**

```

%_N_PART1_MPF
; $PATH=/_N_WKS_DIR/_N_STOCK_REMOVAL_NEW_WPD
; Finished part contour Example 1
;
N100 G18 DIAMON F1000
N110 G1 X0 Z90
N120 X20 RND=4
N130 X30 Z80
N140 Z72
N150 X34
N160 Z58
N170 X28 Z55 F300
N180 Z50 F1000
N190 X40
N200 X60 Z46
N210 Z30
N220 X76 CHF=3
N230 Z0
N240 M17

```

**Blank contour:**

```

%_N_BLANK1_MPF
; $PATH=/_N_WKS_DIR/_N_STOCK_REMOVAL_NEW_WPD
; Blank contour Example 1
;
N100 G18 DIAMON F1000
N110 G0 X0 Z93
N120 G1 X37
N130 Z55
N140 X66
N150 Z35
N160 X80
N170 Z0
N180 X0
N190 Z93
;End point=starting point
;Blank contour must be closed
N200 M17

```

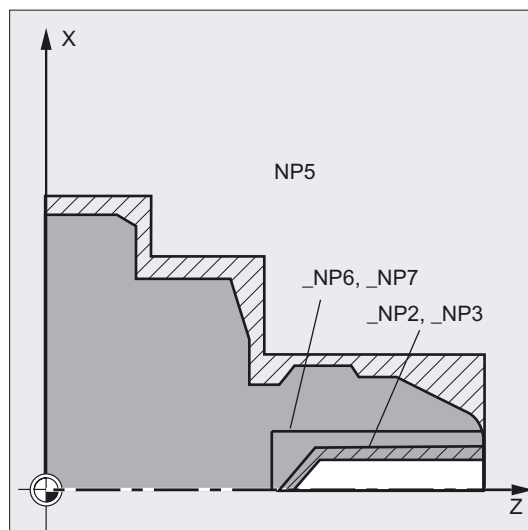
After machining, the STOCK\_REMOVAL\_NEW.WPD workpiece contains a new program called MACHINED\_PART1.MPF. This program is created during the first program call and contains the traversing motions for machining the contour in accordance with the blank.



## Example 2

A simple inside contour is to be machined on the same part as in example 1. A center hole is first rough-drilled with a 10 diameter drill. The inside contour is then rough-machined in parallel with the contour as the hole corresponds approximately to the finished contour. This is achieved by again defining a blank contour for inside machining.

The stock removal contour is located in the same program as the cycle call in the blocks N400 to N420, the blank contour in blocks N430 to N490.



### Machining program:

```
%_N_EXAMPLE_2_MPF
; $PATH=/_N_WKS_DIR/_N_STOCK_REMOVAL_NEW_WPD
; Example 2: Internal stock removal, parallel
to contour
; Sca, 01.04.99
;
; Tool offset data for turning tool, inside
N100 $TC_DP1[2,1]=500 $TC_DP2[2,1]=6
$TC_DP6[2,1]=0.5 $TC_DP24[2,1]=60
N105 $TC_DP1[1,1]=200 $TC_DP3[1,1]=100
$TC_DP6[1,1]=5
N110 G18 G0 G90 DIAMON
N120 X300
N130 Z150
N140 T1 D1 M6 ;Change drill with diameter 10
N150 X0 ;Center drilling in three steps
N160 Z100
N170 F500 S400 M3
N175 G1 Z75
N180 Z76
```

```

N190 Z60
N200 Z61
N210 Z45
N220 G0 Z100
N230 X300 ;Approach tool change point
N240 Z150
N250 T2 D1 M6 ;Insert turning tool for inside
;machining

N260 G96 F0.5 S500 M3
N275 CYCLE950("", "N400", "N420",
"Machined_Part1_inside", 311123, 1.25, 0, 0,
0.8, 0.5, 0.4, 0.3, 0.5, 45, 1, "", "N430", "N490", , , , ,
, 1)
N280 G0 X300
N290 Z150
N300 GOTOF _END ;Skip contour definition
N400 G0 X14 Z90 ;N400 to N420 finished part contour
N410 G1 Z52
N420 X0 Z45
N430 G0 X10 Z90 ;N430 to N490 blank contour
N440 X16
N450 Z40
N460 X0
N470 Z47
N480 X10 Z59
N490 Z90
N500 _END:M2

```

### Example 3

The same part from example 1 is now to be machined in two steps.

In the first machining step (N45), roughing is carried out using a tool with tool point position 9 and a large radius with deep infeed depth and no blank specified. The result to be generated is an updated blank called BLANK3.MPF.

The type of machining for this step is:

- Roughing only,
- Longitudinal,
- External,
- With rounding,
- Relief cuts are not to be machined.

In the second machining step (N75), the residual material on this blank is machined with a different tool and then finished.

The type of machining for this step is:

- Complete machining (roughing and finishing)
- Longitudinal,
- External,
- With rounding (so that no corners are left over),
- Relief cuts are to be machined.

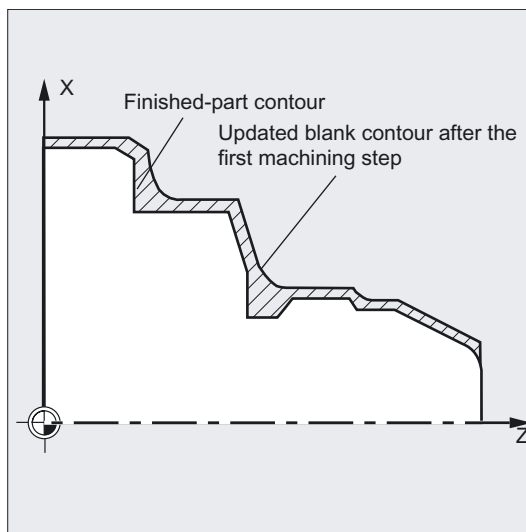
**Machining program:**

```

%_N_EXAMPLE_3_MPF
; $PATH=/_N_WKS_DIR/_N_STOCK_REMOVAL_NEW_W
PD
; Example 3: Stock removal in two steps
with blank updating
; Sca, 09.04.99
;
; Tool offset data
; T3: Roughing steel for rough machining, tool point position 9, radius 5
N05 $TC_DP1[3,1]=500 $TC_DP2[3,1]=9
$TC_DP6[3,1]=5 $TC_DP24[3,1]=80
; T4: Turning tool for residual material
and finishing
; Tool point position 3, radius 0.4
N10 $TC_DP1[4,1]=500 $TC_DP2[4,1]=3
$TC_DP6[4,1]=0.4 $TC_DP24[4,1]=80
N15 G18 G0 G90 DIAMON
N20 T3 D1 ;Tool for roughing
N25 X300
N30 Z150
N35 G96 S500 M3 F2
N45 CYCLE950("Part1",,, "Machined_Part3",
321111,8,1,1,0.8,0.7,0.6,0.5,1,45,6,
"DEFAULT",,, "Blank3",0,91,0,91,1)
N50 G0 X300
N55 Z150
N60 T4 D1 ;Tool for roughing residual material and
;finishing
N65 G96 S500 M3 F2
N75
CYCLE950("Part1",,, "Finishing_Part3",311311,0.5,0.25,0.25,0.8,0.7,0.6,0.5,1,45,6,
"Blank3",,,,,,1)
N160 M2

```

**Finished part contour:** as in Example 1



## Flow

### Position reached prior to cycle start:

The starting position can be any position, from which the blank contour can be approached without collision. The cycle calculates collision-free approach movements to the starting point for machining, but does not consider the toolholder data.

### Movement for paraxial roughing:

- The starting point for roughing is calculated internally in the cycle and approached with G0.
- The infeed to the next depth, calculated in accordance with the specifications in the `_MID` parameter, is carried out with G0, and paraxial roughing is then performed with G1. The feedrate during roughing is calculated internally in the cycle according to the path; it is the feedrate that results from the values specified for longitudinal and face feed (`_FF1` and `_FF2`).
- For "Rounding along contour", the previous intersection is approached parallel to the contour.
- When the previous intersection is reached or for "Without rounding along contour" machining, the tool is lifted off at the angle programmed in `_ANGB` and then retracted to the starting point for the next infeed with G0. If the angle is 45 degrees, the programmed lift-off distance `_VRT` is also followed precisely; it is not exceeded for other angles.
- This procedure is repeated until the total depth of the machining section has been reached.

### Sequence of motions for roughing in parallel with contour:

- The starting point for roughing and the individual infeed depths are calculated as for paraxial roughing and approached with G0 or G1.
- Roughing is carried out in contour-parallel paths.
- Lift-off and retraction are carried out as paraxial roughing.

## Explanation of the parameters

### **\_NP1, \_NP2, \_NP3 (contour programming finished part)**

The finished part contour can be programmed optionally in a separate program or in the current main program that calls the routine. The data is transferred to the cycle via parameters \_NP1 – Name of the program or \_NP2, \_NP3 – ID of program section from ... to using block numbers or labels.

So there are three options for contour programming:

- The contour is defined in a separate program, in which case only \_NP1 needs to be programmed (see example 1)
- The contour is defined in the calling program, in which case only \_NP2 and \_NP3 have to be programmed (see example 2)
- The stock removal contour is part of a program but not part of the program that calls the cycle, in which case all three parameters must be programmed.

When the contour is programmed as a program section, the last contour element (block with label or block number end of blank contour) must not contain a radius or chamfer. Write the program name in \_NP1 with path name and program type.

Example:

\_NP1="/\_N\_SPF\_DIR/\_N\_PART1\_SPF"

### **\_NP4 (name of the stock removal program)**

The stock removal cycle generates a program for the travel blocks that are required for stock removal between the blank and the finished part. If no other path is specified when this program is generated, it is stored in the parts program memory, in the same directory as the calling program. If a path is entered, it is stored accordingly in the file system. If no other type is specified, the program is a main program (type MPF).

The \_NP4 parameter defines the name of this program.

### **\_VARI (machining type)**

The \_VARI parameter defines the type of machining.

Possible settings are:

- Units digit: (stock removal type)
  - 1 = Longitudinal
  - 2 = Face
  - 3 = Parallel to contour
- Tens digit: (infeed direction)
  - 1 = Programmed infeed direction X-
  - 2 = Programmed infeed direction X+
  - 3 = Programmed infeed direction Z-
  - 4 = Programmed infeed direction Z+
- Hundreds digit: (machining process)
  - 1 = Roughing
  - 2 = Finishing
  - 3 = Complete

- Thousands digit: (machining direction)
  - 1 = With rounding
  - 2 = Without rounding (lift-off)
- Selecting with or without rounding on the contour specifies whether liftoff should be executed immediately at the roughing intersection or if rounding should be executed along the contour up to the previous intersection, such that no residual corners remain.
- Ten thousands digit: (relief cuts)
  - 1 = Machine relief cuts
  - 2 = No machining of relief cuts
- Hundred thousands digit: (machining direction)
  - 1 = Programmed machining direction X-
  - 2 = Programmed machining direction X+
  - 3 = Programmed machining direction Z-
  - 4 = Programmed machining direction Z+

**Example:**

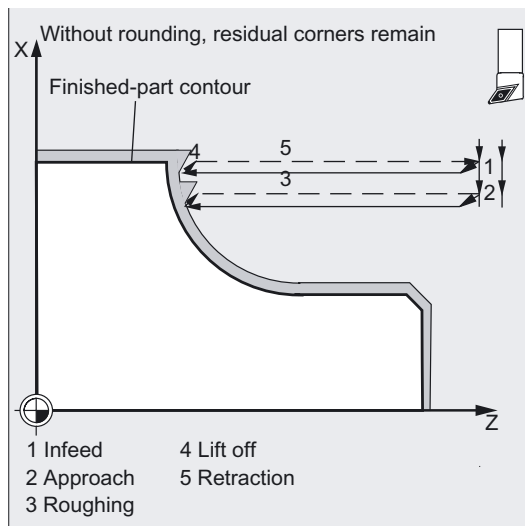
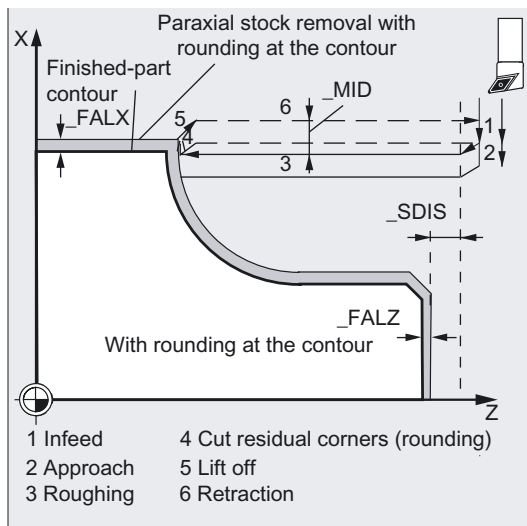
\_VARI=312311 means machining:

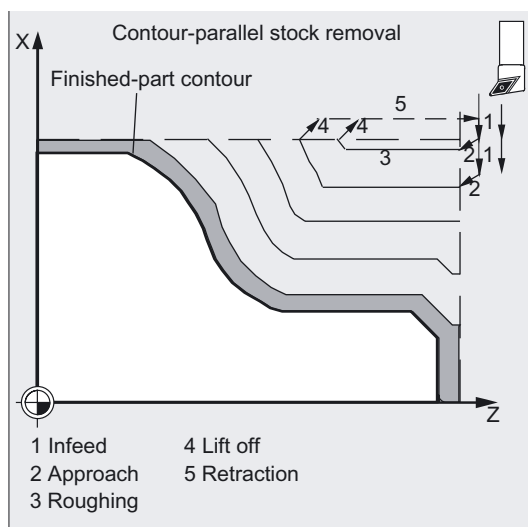
longitudinally,

infeed direction X- (thus, outside),

complete;

no rounding along the contour, relief cuts are machined, machining direction Z-.





### **\_MID (infeed depth for roughing)**

The infeed depth for roughing is programmed with the `_MID` parameter. Roughing steps are generated with this infeed until the remaining depth is less than twice the infeed depth. Two steps are then performed, each at half of the remaining depth. If the facing axis is involved in the infeed for roughing, `_MID` is interpreted as a radius or diameter depending on the value of cycle setting data `_ZSD[0]`.

Cycle setting data	Description
<code>_ZSD[0]=0</code>	<code>_MID</code> is evaluated in accordance with the G group for radius/diameter programming; in the case of DIAMOF it is a radius, otherwise it is a diameter.
<code>_ZSD[0]=1</code>	<code>_MID</code> is a radius value
<code>_ZSD[0]=2</code>	<code>_MID</code> is a diameter value

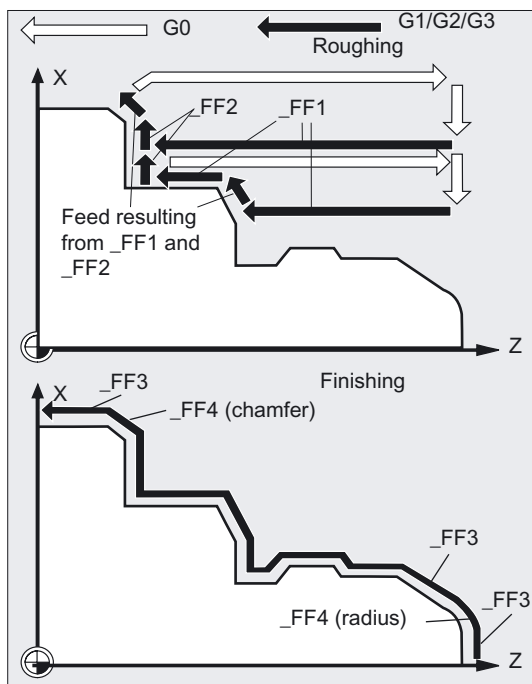
When roughing parallel to contour, the infeed depth does not act in relation to the specified infeed axis, but vertically to the contour. This always results in more cuts than in paraxial roughing with the same infeed depth value.

### **\_FALZ, \_FALX (final machining allowance)**

The default for a final machining allowance for roughing is provided by the `FALZ` (for Z axis) and `FALX` (for X axis) parameters. Roughing is always carried out up to these finishing allowances. If no finishing allowances are programmed, stock is removed when roughing up to the final contour.

### **\_FF1, \_FF2, \_FF3 and FF4 (feedrate)**

Separate feedrates can be specified for roughing and finishing, as shown in the figure.



Separate feedrates apply for longitudinal (\_FF1) and face (\_FF2) during roughing. If inclined or circular path sections are traversed when machining the contour, the appropriate feedrate is calculated automatically inside the cycle.

The feedrates programmed at the contour are active during finishing. If none are programmed there, the finishing feedrate in \_FF3 and the feedrates at radii and chamfers in \_FF4 apply to these contour transition elements. (See example 1 for programming of the part in the above figure)

#### **\_VRT (lift-off distance) and \_ANGB (lift-off angle)**

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.

It is also possible to program the angle, at which the axis is retracted from the contour in the \_ANGB parameter. If nothing is programmed, the angle is 45°.

#### **\_SDIS (safety clearance)**

The \_SDIS parameter determines the amount of clearance for obstructions. This clearance is active for retraction from a relief cut and approach to the next relief cut, for example. If no value is programmed, the clearance is 1 mm.

#### **\_NP5, \_NP6, \_NP7 (contour programming blank)**

If a blank is programmed as a contour, it can be programmed as a program name using the \_NP5 parameter or as a program section with the \_NP6 and \_NP7 parameters. Otherwise, programming is carried out as for finished parts (see \_NP1, \_NP2, \_NP3).



**\_NP8 (name of contour program for updated blank contour)**

The CYCLE950 cycle can detect residual material that cannot be removed with the active tool. To continue this machining with a different tool, it is possible to generate an updated blank contour automatically. This is stored as a program in the parts program memory. You can specify the program name in the \_NP8 parameter with or without path details (see example 3).

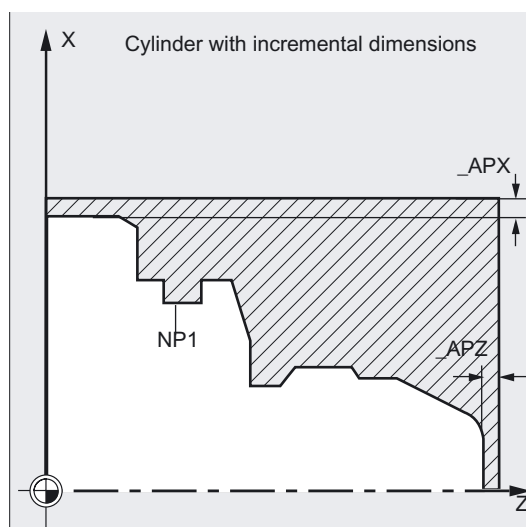
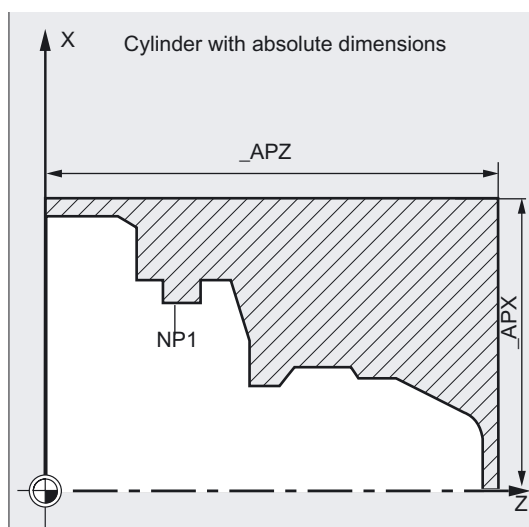
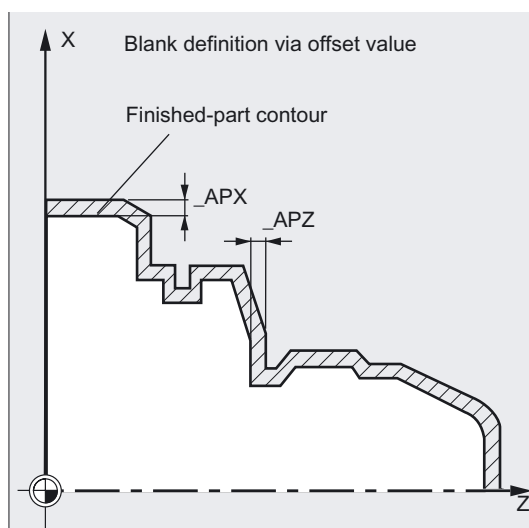
An updated blank contour is always generated when a travel program is generated.

**\_APZ, \_APZA, \_APX, \_APXA (blank definition)**

You can also define a blank by entering the dimensions of a blank cylinder (or hollow cylinder) or as an allowance on the finished part contour in the \_APZ and \_APX parameters.

You can enter the cylinder dimensions as either absolute or incremental values, although an allowance on the finished part contour is always interpreted incrementally.

Absolute or incremental values are selected via the \_APZA and \_APXA parameters (\_APZA, \_APXA: 90 - absolute, 91 - incremental).

**\_TOL1 (blank tolerance)**

Since a blank does not always correspond exactly to the blank definition, when it is cast or forged, for example, it makes sense not to travel to the blank contour with G0 for roughing and for the infeed, but to activate G1 shortly beforehand, in order to compensate for any tolerances. The `_TOL1` parameter defines the distance from the blank, at which G1 becomes active.

Traversing is started with G1 at this incremental amount before the blank. If the parameter is not programmed, it has the value 1 mm.

---

#### Note

##### Contour definition

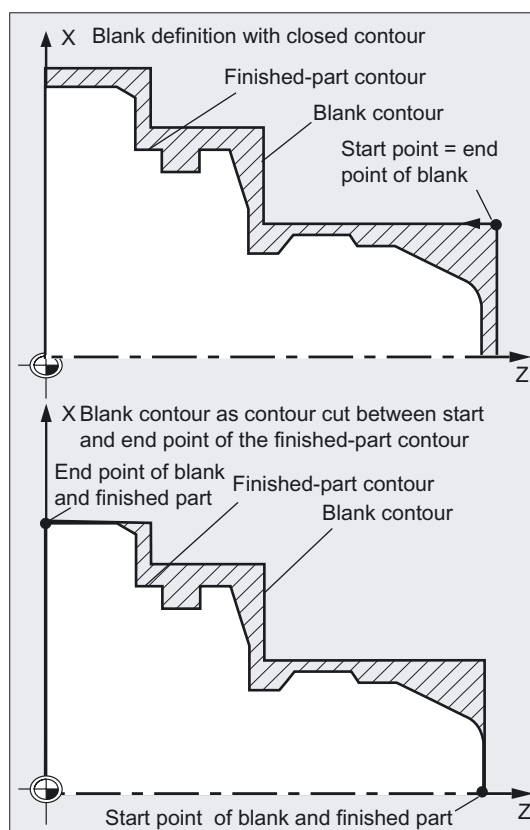
Unlike CYCLE95, one block that contains a link to the current plane is sufficient for contour programming.

For further details of contour definition, see CYCLE95.

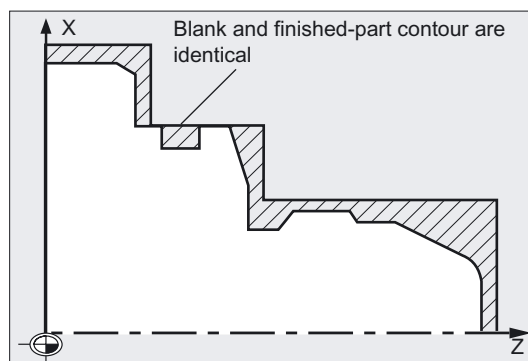
---

##### Blank contour definition

A blank contour must either be a closed contour (starting point = end point), which encompasses the finished part contour either partially or fully, or a contour section between the starting and end points of the finished part contour. The programmed direction is irrelevant.



Blank contours must always be described in such a way that they are not partially identical to the finished part contours, i.e., the machined materials are not combined.



### Explanation of the cycle structure

CYCLE950 is used to solve very complex problems during stock removal, which require high processing power in the control. For best timing, the calculation is carried out in the HMI. The calculation is started in the cycle and a program with traversing blocks for stock removal is generated in its result and stored in the control's file system, where it is called and executed immediately. This structure means that it is only necessary to perform the calculation the first time a program is executed with the CYCLE950 call. When called a second time, the traversing program is available and can be called by the cycle.

Recalculation is performed when:

- A finished contour has been modified,
- Transfer parameters of the cycle have changed,
- A tool with different tool offset data has been activated before the cycle call.

### Program storage in the file system

If the contours for CYCLE950 are programmed outside the program that makes the call, the following applies to searching in the file system of the control:

- If the calling program is stored in a workpiece directory, then the programs, which contain the finished part contour or blank contour, must also be stored in the same workpiece directory, or at least programmed with path information.
- If the invoked program is located in the "Parts program" directory (MPF.DIR), the programs are also searched for there, if no path is specified.

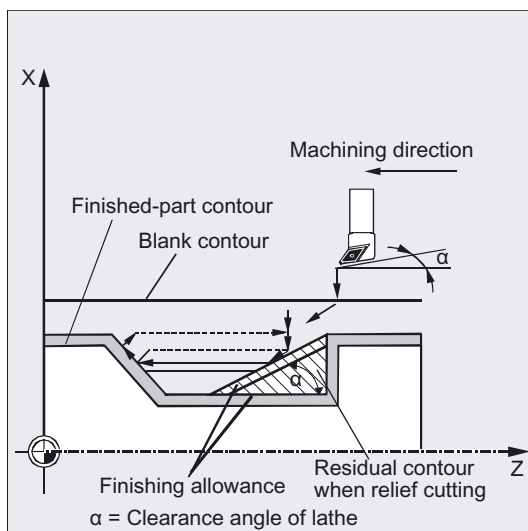
The cycle creates a program that contains the traversing blocks for stock removal and, optionally, an updated blank contour. These are either stored in the same directory as the cycle-calling program or in accordance with the specified path.

### Simulation CYCLE950

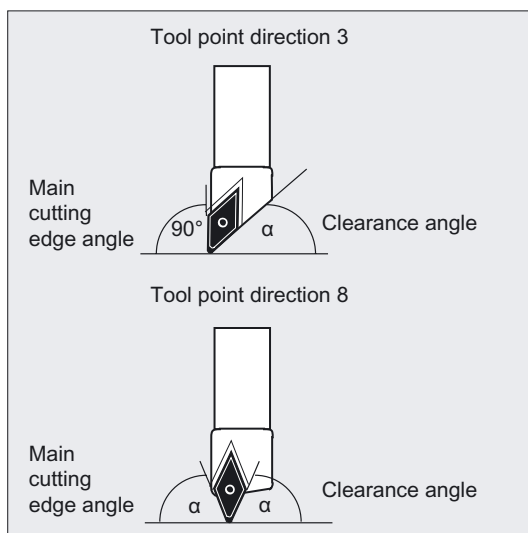
In the simulation of the CYCLE950 extended stock removal cycle, the generated programs are saved to the NCU file system. Therefore, only the "NC Active Data" setting is practical, since tool offset data are included in the program calculation.

## Blank updating

The CYCLE950 extended stock removal cycle detects residual material during roughing and is able to generate an updated blank contour outside the machining process, which can be used in a further machining step.



To do this, the cycle internally considers the angle at the tool point. The tool clearance angle must be entered in the tool offset data (parameter 24). The cycle automatically defines the main cutting angle, according to the tool point position. For tool point positions 1 to 4, the blank update is calculated with a main cutting angle of  $90^\circ$ . For tool point positions 5 to 9, the main cutting angle is assumed to be identical to the tool clearance angle.



If CYCLE950 is called with blank update in the same program more than once, different names must be assigned to the generated blank contours; it is not permissible to use the program name (parameter \_NP8) more than once.

---

**Note**

Extended stock removal cannot be performed in m:n configurations.

---

---

**Note****Special information for HMI-Advanced OEM users:**

The "Extended Stock Removal Cycle" package occupies Task 26 in the HMI for communication between the cycle and the HMI.

For this purpose, a file is created in the \ADD\_ON directory when the package is installed:

- REGIE.INI

As a result, this task is not available to other OEM applications.

---

**Description of alarm source CYCLE950**

see References: /DA/ SINUMERIK 840D sl/840D/840Di/810D Diagnostics Manual



## Error messages and Error handling

### 5.1 General information

If error conditions are detected in the cycles, an alarm is generated and the execution of the cycle is aborted.

The cycles continue to output messages in the dialog line of the control. These message will not interrupt the program execution.

---

#### Note

For more information on errors and required responses, as well as messages output in the control's dialog line, please refer to the diagnostics guide.

**References:** see /DA/ Diagnostics Guide SINUMERIK 840D/840Di/810D

---

### 5.2 Error handling in the cycles

If error conditions are detected in the cycles, an alarm is generated and the machining is aborted. Alarms with numbers between 61000 and 62999 generated in the cycles. This range of numbers, in turn, is divided again with regard to alarm responses and cancel criteria. The error text that is displayed together with the alarm number gives you more detailed information on the error cause.

Alarm number	Clearing criterion	Alarm Response
61000 ... 61999	NC_RESET	Block preparation in the NC is aborted
62000 ... 62999	Clear key	Program execution is not interrupted; display only

#### Overview of cycle alarms

With a small number of exceptions, the error numbers are subject to the classifications below:

6	–	X	–	–
---	---	---	---	---

- X = 0 General cycle alarms
- X = 1 Alarms generated by the drilling, drilling pattern and milling cycles
- X = 6 Alarms generated by turning cycles

The table in the diagnostics guide includes a list of all errors occurring in the cycles with their location of occurrence and appropriate instructions for fault correction.

**see References:** Diagnostics Guide SINUMERIK sl/840D/840Di/810D

## 5.3 Messages in the cycles

The cycles output messages in the dialog line of the control. These message will not interrupt the program execution.

Messages provide information with regard to a certain behavior of the cycles and with regard to the progress of machining and are usually kept beyond a machining step or until the end of the cycle. The following messages are possible:

Message text	Source
"Depth: according to the value for the relative depth"	CYCLE81...CYCLE89, CYCLE840
"Machining elongated hole"	LONGHOLE
"Slot is being processed"	SLOT1
"Circumferential slot is being processed"	SLOT2
"Incorrect mill direction: G3 will be generated"	SLOT1, SLOT2, POCKET1, POCKET2, CYCLE90
"Changed form of the undercut"	CYCLE94, CYCLE96
"1 <sup>st</sup> drilling depth: according to the value for the relative depth"	CYCLE83
"Caution: Final machining allowance >= tool diameter!"	POCKET1, POCKET2
"Thread start: - machining as a longitudinal thread"	CYCLE97, CYCLE98
"Thread start: - machining as a face thread"	CYCLE97, CYCLE98
"Simulation active, no tool programmed, final contour being traversed"	POCKET1...POCKET4, SLOT1, SLOT2, CYCLE93, CYCLE72
"Simulation active, no tool programmed"	CYCLE71, CYCLE90, CYCLE94, CYCLE96
"Waiting for spindle reversal"	CYCLE840



# A

## List of abbreviations

A	Output
ASCII	American Standard Code for Information Interchange American coding standard for the exchange of information
ASIC	Application Specific Integrated Circuit: User switching circuit
ASUB	Asynchronous subroutine
AuxF	Auxiliary function
AV	Job planning
BB	Ready to run
BCD	Binary Coded Decimals: Decimal numbers encoded In binary code
BCS	Basic Coordinate System
BIN	Binary files ( <b>B</b> inary <b>F</b> iles)
BIOS	Basic Input Output System
BO	Operating mode
BOT	Boot files: Boot files for SIMODRIVE 611D
BP	Basic program
C1 .. C4	Channel 1 to channel 4
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
C-bus	Communication bus
CNC	Computerized Numerical Control Computerized numerical control
COM	Communication
COR	Coordinate rotation
CP	Communications Processor
CPU	Central Processing Unit Central processing unit
CR	Carriage Return
CRC	Cutter radius compensation
CRT	Cathode Ray Tube picture tube
CSB	Central Service Board: PLC module
CTS	Clear To Send: Signal from serial data interfaces
CUTCOM	Cutter radius compensation: Tool radius compensation
DAC	Digital-to-Analog Converter
DB	Data Block in the PLC
DBB	Data Block Byte in the PLC
DBW	Data Block Word in the PLC
DBX	Data block bit in the PLC
DC	Direct Control: Movement of the rotary axis via the shortest path to the absolute position within one revolution

DCD	Data Carrier Detect
DDE	Dynamic Data Exchange
DIN	Deutsche Industrie Norm (German Industry Standard)
DIO	Data Input/Output: Data transfer display
DIR	DIRectory Directory
DLL	Dynamic Link Library
DOS	Disk Operating System
DPM	Dual-Port Memory
DPR	Dual-Port RAM
DRAM	Dynamic Random Access Memory
DRF	Differential Resolver Function: (Handwheel)
DRY	DRY run: Dry run feedrate
DSB	Decoding Single Block: Decoding single block
DTE	Data Terminal Equipment
DW	Data word
EIA code	Special punched tape code, number of holes per character always odd
ENC	Encoder: Actual value encoder
EPROM	Erasable Programmable Read Only Memory
ERROR	Error from printer
FB	Function block
FBD	Function plan (PLC programming method)
FBS	Slimline screen
FC	Function Call: Function block in the PLC
FDB	Product database
FDD	Floppy Disk Drive
FDD	Feed Drive
FEPROM	Flash-EPROM: Read and write memory
FIFO	First In First Out: memory, which works without address specification and whose data are read in the order in which they were stored.
FIPO	Fine Interpolator
FM	Function Module
FM-NC	Function module – numerical control
FPU	Floating Point Unit Floating Point Unit
FRA	Frame block
FRAME	Data block (FRAME)
FST	Feed Stop: Feed stop
GUD	Global User Data Global user data
HD	Hard Disk Hard disk
HEX	Abbreviation for hexadecimal number
HHU	HHU
HMI	Human Machine Interface: Operator functionality of SINUMERIK for operation, programming and simulation.
HMS	High-resolution Measuring System
HW	Hardware

I	Input
I/O	Input/output
I/R	Infeed/regenerative feedback unit (power supply) of the SIMODRIVE 611(D)
IBN	Start up
ICA	Interpolatory Compensation Interpolatory compensation
IF	Drive module pulse enable
IK (GD)	Implicit communication (global data)
IM	Interface Module Interface module
IMR	Interface Module Receive: Interconnection module for receiving data
IMS	Interface Module Send: Interconnection module for sending data
INC	Increment Increment
INI	INItializing data Initializing data
IPO	Interpolator
IS	Interface signal
ISA	Industry Standard Architecture
ISO	International Standard Organization
ISO code	Special punched tape code, number of holes per character always even
JOG	JOGging: Setup mode
$K_{\dot{U}}$	Speed ratio
$K_v$	Servo gain factor
LAD	Ladder diagram (PLC programming method)
LC	Line Feed
LCD	Liquid crystal display
LEC	Leadscrew error compensation
LED	Light-emitting diode
LR	Position controller
LUD	Local User Data
MB	Megabyte
MC	Measuring circuit
MCP	Machine control panel
MCS	Machine coordinate system
MD	Machine data
MDA	Manual Data Input Manual input
MLFB	Machine-readable product designation
Mode group	Mode group
MPF	Main Program File: NC part program (main program)
MPI	Multi-Point Interface Multiport Interface
MS	Microsoft (software manufacturer)
MSD	Main Spindle Drive
NC	Numerical Control: Numerical Control
NCK	Numerical Control Kernel: NC kernel with block preparation, traversing range, etc.
NCU	Numerical Control Unit: NCK hardware unit

NRK	Name for the operating system of the NCK
NURBS	Non-Uniform Rational B-Spline
OB	Organization block in the PLC
OEM	Original Equipment Manufacturer
OP	Operator Panel: Operating setup
OPI	Operator Panel Interface: Interface for connection to the operator panel
OPT	Options: Options
OSI	Open Systems Interconnection: Standard for computer communications
P-bus	I/O bus
PC/	Personal computer
PCIN	Name of the SW for data exchange with the control
PCMCIA	Personal Computer Memory Card International Association plug-in memory normalization board
PCU	PC Unit: PC box (computer unit)
PG	Programming device
PLC	Automation system
PLC	Programmable Logic Control Controller
PMS	Position Measuring System
POS	Positioning drive
RAM	Random Access Memory: Program memory that can be read and written to
REF	Reference point approach function
REPOS	Reposition function
RISC	Reduced Instruction Set Computer: Type of processor with small instruction set and ability to process instructions at high speed
ROV	Rapid Override Input correction
RPA	R-Parameter Active: Memory area on the NCK for R parameter numbers
RPY	Roll Pitch Yaw: Rotation type of a coordinate system
RTS	Request To Send: RTS, control signal of serial data interfaces
SBL	Single Block Single block
SD	Setting Data
SDB	System Data Block
SEA	Setting Data Active: Identifier (file type) for setting data
SFB	System Function Block
SFC	System Function Call
SK	Soft key
SKP	SKiP: Skip block
SM	Stepper Motor
Software	Software
SPF	Sub Program File: Subprogram
SPS	Programmable Logic Controller
SR	Subprogram
SRAM	Static RAM (non-volatile)
SSI	Serial Synchronous Interface: Synchronous serial interface

STL	Statement list
SYF	System Files System files
T	Tool
TC	Tool change
TEA	Testing Data Active: Identifier for machine data
TLC	Tool length offset
TNRC	Tool nose radius compensation
TO	Tool offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
TRANSMIT	Transform Milling Into Turning: Coordinate conversion on turning machine for milling operations
TRC	Tool Radius Compensation
UFR	User Frame: Work offset
UI	Operator interface
V.24	Serial interface (definition of the exchange lines between DTE and DCE)
WCS	Workpiece coordinate system
WO	Work offset
WOP	Workshop-oriented Programming
WPD	Workpiece Directory: Workpiece directory
ZOA	Zero Offset Active: Identifier (file type) for work offset data
μC	Micro Controller



## References

### Overview of publications

An overview of publications that is updated monthly is provided in a number of languages in the Internet at:

<http://www.siemens.com/motioncontrol>

Select the menu items: "Support" > "Technical documentation" > "Summary of publications" or "DOConWEB".





## List of parameters

List of input/output variables for cycles

Name	Meaning in English
AD	Allowance depth
AFSL	Angle for slot length
ANG1, ANG2	Flank angle
ANGB	Liftoff angle for roughing
AP1	Unfinished dimension in plane
AP2	Unfinished dimension in plane
APP	Approach path
APX	Axial value for defining blank for facing axis
APXA	Absolute or incremental evaluation of parameter APX
APZ	Axial value for defining blank for longitudinal axis
APZA	Absolute or incremental evaluation of parameter APZ
AS1, AS2	Direction of approach/approach travel
AXN	Tool axis
BNAME	Name for program of drill positions
CDIR	Circle direction
CPA	Center point, abscissa
CPO	Center point, ordinate
CRAD	Corner radius
DAM	Degression value, Path for roughing interrupt
DBH	Distance between holes
DIAG	Groove depth
DIATH	Diameter of thread
DIS1	Distance
DIS1	Distance between columns
DIS2	Number of lines, Distance between rows
DM1 ... DM4	Diameter
DN	Tool offset number (D number)
DP	Depth
DP1	First depth
DPR	Depth, relative

Name	Meaning in English
DT	Dwell time
DTB	Dwell time at bottom
DTD	Dwell time at depth
DTS	Dwell time at starting point
ENC	Tapping with/without encoder
FAL	Finish allowance
FAL1	Finish allowance on groove base
FAL2	Finish allowance on flanks
FALD	Finish allowance depth
FALZ	Finish allowance, z axis
FALX	Finish allowance, x axis
FDEP	First depth
FDIS	First distance
FDP1	Overrun path in direction to plane
FDPR	First depth, relative
FF1	Feedrate for roughing
FF2	Feedrate for insertion
FF3	Feedrate for finishing
FF4	Feedrate for contour transition elements
FFCP (SW 6.3 and higher)	Feedrate for circular positioning
FFD	Feedrate for depth
FFP1	Feedrate surface
FFP2	Feedrate for finishing
FFR	Feedrate
FORM	Definition of form
FPL	Final point along longitudinal axis
FRF	Feedrate factor
IANG	Infeed angle
INDA	Incremental angle
IDEP	Infeed depth
KNAME	Name of the contour subroutine
LSANF	Block number/label identifying start of contour definition
LSEND	Block number/label identifying end of contour definition
LP1	Length of approach travel, radius
LP2	Length of return travel, radius
KDIAM	Internal diameter of thread
LENG	Elongated hole length, pocket length
MDEP	Minimum depth

Name	Meaning in English
MID	Maximum infeed depth
MIDA	Maximum infeed width
MIDF	Maximum infeed depth for finishing
MPIT	Thread lead as thread size
NID	Number of noncuts
NP1 ... NP8	Name/Label ...
NPP	Name of parts program
NRC	Number of roughing cuts
NSP	Start point offset first thread
NUM	Number of holes
NUM1	Number of columns
NUM2	Number of lines
NUMT	Number of threads
PA	Reference point, abscissa
PO	Reference point, ordinate
PO1 ... PO4	Point in longitudinal axis
PIT	Thread lead
PNAME	Name for pocket milling machining program
POSS	Position for oriented spindle stop
PP1 ... PP3	Thread pitch 1...3 as value
PRAD	Pocket radius
RAD	Radius
RAD1	Radius
RCO1, RCO2	Radius/chamfer outside
RCI1, RCI2	Radius/chamfer inside
RFF	Retract feed
RFP	Reference plane
ROP	Run out path
RPA	Retract path, abscissa
RPAP	Retract path, applicate
RPO	Retract path, ordinate
RL	Bypass contour
RTP	Retract plane
SDAC	Spindle direction after cycle
SDIR	Spindle direction
SDIS	Safety clearance
SDR	Spindle direction for retraction
SPCA	Reference point, abscissa
SPCO	Reference point, ordinate

Name	Meaning in English
SPD	Starting point in the facing axis
SPL	Starting point along longitudinal axis
SSF	Speed for finishing
SST	Speed for tapping
SST1	Speed for retraction
STA, STA1	Angle
STA2	Insertion angle
TDEP	Thread depth
TN	Name of stock removal tool
TOL1	Blank tolerance
TYPTH	Type of thread
VARI	Working variant
VRT	Variable return path
WID	(Pocket) width
WIDG	Groove width

# Glossary

## Access authorization

The CNC program blocks and data are protected by a 7-level system of access restrictions:

- Three password levels for system manufacturers, machine manufacturers and users and
- Four keylock switch settings, which can be evaluated via the PLC.

## Approach machine (fixed-point)

Approach motion towards one of the predefined -> fixed machine points.

## Auxiliary functions

Auxiliary functions can be used to transfer -> parameters to the -> PLC in -> parts programs, where they trigger reactions which are defined by the machine manufacturer.

## Backlash compensation

Compensation for a mechanical machine backlash, e.g., backlash on reversal for ball screws. Backlash compensation can be entered separately for each axis.

## Backup battery

The backup battery ensures that the -> user program is reliably backed up in the -> CPU against mains failure and that fixed data areas and markers, times and counters are kept in non-volatile memory.

## Blank

Workpiece as it is before the part is machined.

## Block

"Block" is the term given to any files required for creating and processing programs.

## **Block search**

The block search function allows any point in the part program to be selected, at which machining must start or be continued. The function is provided for the purpose of testing part programs or continuing machining after a program abort.

## **Booting**

Loading the system program after power on.

## **Channel**

A channel is characterized by its ability to execute a -> parts program independently of other channels. A channel exclusively controls the axes and spindles assigned to it. Parts programs run on various channels can be coordinated by -> synchronization.

## **Channel structure**

The channel structure enables the -> programs of the individual channels to be executed simultaneously and asynchronously.

## **Circular interpolation**

The -> tool is required to travel in a circle between defined points on the contour at a specified feedrate while machining the workpiece.

## **Clear all**

The following -> CPU memories are erased by a general reset operation:

- the -> working memory
- the read/write area of the -> load memory
- the -> system memory
- the -> backup memory

## **CNC**

-> NC

## **CNC programming language**

The CNC programming language is based on DIN 66025 with high-level language expansions. The -> high-level CNC language and programming allow, among other things, macros to be defined (groupings of individual instructions).

**COM**

Component of the NC control for the implementation and coordination of communication.

**Compensation axis**

Axis with a setpoint or actual value modified by the compensation value

**Compensation table**

Table containing interpolation points. It provides the compensation values of the compensation axis for selected positions on the basic axis.

**Compensation value**

Difference between the axis position measured by the encoder and the desired, programmed axis position.

**Connecting cable**

Connecting cables are pre-assembled or user-assembled 2-wire cables with a connector at each end. These interconnecting cables connect the -> CPU via the -> multipoint interface (MPI) with a -> programming device or other CPUs.

**Contour**

Outline of a -> workpiece.

**Contour monitoring**

The following error is monitored within a definable tolerance band as a measure of contour accuracy. An impermissibly high following error might be caused by a drive overload, for example. In such cases, an alarm is output and the axes are stopped.

**Coordinate system**

See -> machine coordinate system, -> workpiece coordinate system.

**CPU**

Central Processor Unit, -> PLC

**Cycle**

Protected subroutine for executing a repetitive machining process on the -> workpiece.

### **Cycle support**

The available cycles are listed in the "Cycle support" menu in the "Program" operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plain text.

### **Data Block**

1. Data unit of the -> PLC, which the -> HIGHSTEP programs can access.
2. Data unit of the -> NC: Data modules contain data definitions for global user data. These data can be initialized directly when they are defined.

### **Data transfer program PCIN**

PCIN is an auxiliary program for sending and receiving CNC user data (e.g., part programs, tool offsets, etc.) via a serial interface. The PCIN program can run in MS-DOS on standard industrial PCs.

### **Data word**

A data unit, two bytes in size, within a -> data block.

### **Diagnostics**

1. Operating area of the control.
2. The control has both a self-diagnostics program as well as test functions for servicing purposes: status, alarm and service displays.

### **Digital input/output module**

Digital modules are signal formers for binary process signals.

### **Dimensions specification, metric and inches**

Position and pitch values can be programmed in inches in the machining program. The control is set to a basic system regardless of the programmable dimensional specification (G70/G71).

### **DRF**

Differential Resolver Function: NC function which generates an incremental work offset in Automatic mode in conjunction with an electronic handwheel.



**Drift compensation**

When the CNC axes are in the constant motion phase, automatic drift compensation is implemented in the analog speed control (SINUMERIK FM-NC).

**Dynamic feedforward control**

Inaccuracies in the contour caused by following errors can be almost entirely eliminated with the aid of dynamic, acceleration-dependent feedforward control. The result is extraordinary machining precision even at high tool path feed-rates. The feedforward control can be individually selected and deselected for each axis in the parts program.

**Editor**

The editor makes it possible to create, edit, extend, join, and import programs/texts/program blocks.

**Electronic handwheel**

Electronic handwheels can be used to traverse the selected axes simultaneously in manual mode. The handwheel clicks are analyzed by the increment analyzer.

**Exact stop**

With a programmed exact stop instruction, the position stated in a block is approached precisely and very slowly, if necessary. In order to reduce the approach time, -> exact stop limits are defined for rapid traverse and feed.

**Exact stop limit**

When all path axes reach their exact stop limits, the control responds as if it had reached its precise destination point. The -> part program continues execution at the next block.

**External zero offset**

Zero offset specified by the -> PLC.

**Fast lift-off from contour**

When an interrupt is received, it is possible to initiate a motion via the CNC machining program, which allows the tool to be lifted quickly from the workpiece contour currently being machined. The retraction angle and the distance retracted can also be parameterized. An interrupt routine can also be executed following the fast lift-off of the tool. (SINUMERIK FM-NC, 840D).

### **Feedrate override**

The current feedrate setting entered via the control panel or by the PLC is overlaid on the programmed feedrate (0-200 %). The feedrate can also be corrected by a programmable percentage factor (1-200%) in the machining program.

### **Finished-part contour**

Contour of the finished workpiece. See also -> Blank.

### **Fixed machine point**

A point defined uniquely by the machine tool, e.g., the reference point.

### **Fixed-point approach**

Machine tools can approach fixed points such as a tool change point, loading point, pallet change point, etc. in a defined way. The coordinates of these points are stored in the control. Where possible, the control moves these axes in -> rapid traverse.

### **Frame**

A frame is an arithmetic rule that transforms one Cartesian coordinate system into another Cartesian coordinate system. A frame contains the components -> zero offset, -> rotation, -> scaling, -> mirroring.

### **Geometry**

Description of a -> workpiece in the -> workpiece coordinate system.

### **Geometry axis**

Geometry axes are used to describe a 2- or 3-dimensional range in the workpiece coordinate system.

### **Global main program/subprogram**

Each global main program/subprogram may appear only once under its name in the directory. It is not possible to use the same program name in different directories with different contents as a global program.

### **Ground**

Ground is taken as the total of all linked inactive parts of a device which will not become live with a dangerous contact voltage even in the event of a malfunction.

**Helical interpolation**

The helical interpolation function is ideal for machining internal and external threads using form milling cutters and for milling lubrication grooves. The helix comprises two movements:

1. Circular movement in one plane
2. Linear movement perpendicular to this plane.

**High-level CNC language**

The high-level language offers: -> user variable, -> pre-defined user variable, -> system variable, -> indirect programming, -> mathematical and trigonometric functions, -> comparison operations and logic operations, -> program jumps and program branching, -> program coordination (SINUMERIK 840D), -> macro technology.

**High-speed digital inputs/outputs**

The digital inputs can be used for example to start fast CNC program routines (interrupt routines). High-speed, program-driven switching functions can be initiated via the digital CNC outputs (SINUMERIK 840D).

**HIGHSTEP**

Summary of the programming options for the -> PLC in the AS300/AS400 system.

**I/O module**

I/O modules represent the link between the CPU and the process. I/O modules are:

- ->Digital input/output modules
- ->Analog input/output modules
- ->Simulator modules

**Identifier**

In accordance with DIN 66025, words are supplemented using identifiers (names) for variables (arithmetic variables, system variables, user variables), subroutines, key words, and words with multiple address letters. These supplements have the same meaning as the words with respect to block format. Identifiers must be unique. It is not permissible to use the same identifier for different objects.

**Inch measuring system**

Measuring system, which defines distances in inches and fractions of inches.

### **Inclined surface machining**

Drilling and milling operations on workpiece surfaces that do not lie in the coordinate planes of the machine can be performed easily using the function "inclined-surface machining".

### **Increment**

Travel path length specification based on number of increments. The number of increments can be stored as a -> setting data or selected with keys labeled with 10, 100, 1000, 10,000.

### **Incremental dimension**

A destination for axis traversal is defined by a distance to be covered and a direction referenced to a point already reached. See also -> absolute dimension.

### **Initialization block**

Initialization blocks are special -> program blocks. They contain value assignments that are performed before program execution.

The primary purpose of initialization blocks is to initialize predefined data or global user data.

### **Initialization files**

It is possible to create an initialization file for each -> workpiece. Various variable assignments which are intended to apply specifically to one workpiece can be stored in this file.

### **Intermediate blocks**

Traversing movements with tool offset selected (G41/G42) can be interrupted by a limited number of intermediate blocks (block without axis movements in the compensation plane) whereby the tool offset can still be correctly calculated. The permissible number of intermediate blocks which the control reads ahead can be set in system parameters.

### **Interpolator**

Logical unit of the -> NCK which determines intermediate values for the movements to be traversed on the individual axes on the basis of destination positions specified in the parts program.

### **Interpolatory compensation**

Interpolatory compensation can be used to compensate for leadscrew errors and measuring system errors (LEC, MSEC) caused during production.

## Interrupt routine

Interrupt routines are special -> subroutines which can be started on the basis of events (external signals) in the machining process. A parts program block which is currently being worked through is interrupted and the position of the axes at the point of interruption is automatically saved.

## Interrupts

All alarms and -> messages are output on the operator panel in plain text with the date and time and a symbol indicating the cancel criterion. Alarms and messages are displayed separately.

### 1. Alarms and messages in the part program:

Alarms and messages can be displayed in plain text directly from the part program.

### 2. Alarms and messages from PLC

Alarms and messages for the machine can be displayed in plain text from the PLC program. No additional function block packages are required for this purpose.

## Inverse time feedrate

On SINUMERIK FM-NC and 840D controls, it is possible to program the time required to traverse the path of a block instead of the feedrate speed for the axis movement (G93).

## JOG

Control operating mode (setup operation): In JOG mode, the machine can be set up. Individual axes and spindles can be traversed in JOG mode by means of the direction keys. Other JOG mode functions are -> reference point approach, -> Repos and -> Preset (set actual value).

## Key switch

- S7-300: The key switch is the operating mode switch of the -> CPU. The keylock switch is operated by a removable key.
- 840D/FM-NC: The keyswitch on the -> machine control panel has 4 settings, to which functions are assigned by the operating system of the control. Further, the keylock switch has three differently colored keys, which can be removed in the specified positions.

## Keywords

Words with a specific notation, which have a defined meaning in the programming language for -> parts programs.

## KUE

Speed ratio

## **Kv**

Servo gain factor, a control variable in a control loop.

## **Languages**

The operator-prompt display texts, system messages and system alarms are available (on diskette) in five system languages: German, English, French, Italian, and Spanish. The user can select two of the listed languages at a time in the control.

## **Leadscrew error compensation**

Compensation for the mechanical inaccuracies of a leadscrew participating in the feed. The control uses stored deviation values for the compensation.

## **Limit speed**

Maximum/minimum (spindle) speed: The maximum speed of a spindle may be limited by values defined in the machine data, the -> PLC or -> setting data.

## **Linear axis**

The linear axis is an axis, which, in contrast to a rotary axis, describes a straight line.

## **Linear interpolation**

The tool travels along a straight line to the destination point while machining the workpiece.

## **Load memory**

For the CPU 314 of the -> PLC, the load memory is equal to the -> Work memory .

## **Look ahead**

With the look ahead function, a configurable number of traversing blocks is read in advance in order to calculate the optimum machining velocity.

## **Look-ahead detection of contour violations**

The control can recognize and signal the following types of collision:

1. The path distance is shorter than the tool radius.
2. The width of the inner corner is smaller than the tool diameter.

## **Machine**

Operating area of the control.

**Machine axes**

Physically existent axes on the machine tool.

**Machine control panel**

An operator panel on a machine tool with operating elements such as keys, rotary switches, etc., and simple indicators such as LEDs. It is used to directly influence the machine tool via the PLC.

**Machine coordinate system**

A coordinate system, which is related to the axes of the machine tool.

**Machine zero**

Fixed point of the machine tool to which all (derived) measuring systems can be traced back.

**Macro techniques**

Grouping of a set of instructions under a single identifier. The identifier represents the set of consolidated instructions in the program.

**Main block**

A block prefixed by ":" containing all the parameters required to start execution of a -> parts program.

**Main program**

Parts program identified by a number or identifier in which further main programs, subroutines or -> cycles may be called.

**MDA**

Control operating mode: Manual Data Input: In the MDI mode, individual program blocks or block sequences with no reference to a main program or subprogram can be input and executed immediately afterwards through actuation of the NC start key.

**Measuring circuit**

- SINUMERIK FM-NC: The requisite control circuits for axes and spindles are integrated in the control module as standard. A maximum total of 4 axes and spindles can be implemented, with no more than 2 spindles.
- SINUMERIK 840D The signals from the sensors are analyzed in the SIMODRIVE 611D drive modules. The maximum configuration is 8 axes and spindles in total, but including no more than 5 spindles.

## Messages

All messages programmed in the parts program and -> alarms recognized by the system are output on the operator panel in plain text with the date and time and a symbol indicating the cancel criterion. Alarms and messages are displayed separately.

## Metric system

Standardized system of units: For length, e.g., mm (millimeters), m (meters).

## Mirroring

Mirroring reverses the signs of the coordinate values of a contour, with respect to an axis. It is possible to mirror with respect to more than one axis at a time.

## Multipoint interface

The multipoint interface (MPI) is a 9-pin sub-D port. A configurable number of devices can be connected to a multipoint interface and then communicate with each other.

- Programming devices
- Operator control and monitoring equipment
- Further automation systems

The parameter block "Multipoint Interface MPI" of the CPU contains the -> parameters which define the properties of the multipoint interface.

## NC

Numerical Control: Numerical control (NC) includes all components of machine tool control: -> NCK, -> PLC, -> MMC, -> COM.

---

### Note

CNC (computerized numerical control) would be more appropriate for the SINUMERIK 840D or FM-NC controls: MARS and Merkur controls.

---

## NCK

Numerical Control Kernel: Component of the NC control which executes -> parts programs and essentially coordinates the movements on the machine tool.

## Network

The term "network" describes the connection of several S7-300 and other terminal devices, e.g. a programming device, via -> interconnecting cables. A data exchange takes place over the network between the connected devices.



**Node number**

The node number represents the "contact address" of a -> CPU or the -> programming device or any other intelligent periphery module if these are communicating via a -> network with each other. The node number is assigned to the CPU or the programming device with the S7 tool -> "S7 configuration".

**NRK**

Numeric Robotic Kernel (operating system of the -> NCK)

**NURBS**

Internal motion control and path interpolation are performed using NURBS (non-uniform rational B-splines). A standard procedure is thus available (SINUMERIK 840D) as an internal control function for all modes of interpolation.

**OEM**

The scope for implementing individual solutions (OEM applications) for the SINUMERIK 840D has been provided for machine manufacturers, who wish to create their own operator interface or integrate process-oriented functions in the control.

**Offset memory**

Data range in the control, in which the tool offset data are stored.

**Oriented spindle stop**

Stops the workpiece spindle in a specified angular position, e.g., in order to perform additional machining at a particular location.

**Oriented tool retraction**

RETOOL: If machining is interrupted (e.g., when a tool breaks), a program command can be used to retract the tool in a user-specified orientation by a defined distance.

**Override**

Manual or programmable control feature, which enables the user to override programmed feedrates or speeds in order to adapt them to a specific workpiece or material.

## Parameter

- **S7-300:** a distinction is made between 2 types of parameters:
  - Parameters of a STEP 7 instruction  
A parameter of a STEP 7 instruction is the address of the operand to be processed or a constant.
  - Parameters of a -> parameter block  
A parameter of a parameter block determines the behavior of a module.
- **840D:**
  - Operating area of the control.
  - Arithmetic parameter for which the programmer of the parts program can assign or request values as required in the program.

## Part program management

Part programs can be organized according to -> workpieces. The number of programs and data to be managed determine the size of the user memory. Each file (programs and data) can be given a name consisting of a maximum of 24 alphanumeric characters.

## Parts program

A sequence of instructions to the NC control which combine to produce a specific -> workpiece. Likewise, performing a certain machining operation on a specific -> blank.

## PG

Programming device

## PLC

Programmable Logic Control -> Programmable Logic Controller. Component of the -> NC control: Programmable controller for processing the control logic of the machine tool.

## PLC program memory

- **SINUMERIK FM-NC:** The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory on the CPU 314. The S7-CPU314 has a 24 Kbyte user memory for this purpose.
- **SINUMERIK 840D** The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory. The PLC user memory can be expanded up to 96 KB with memory expansions.

**PLC Programming**

The PLC is programmed with STEP 7 software. STEP 7 programming software is based on the standard WINDOWS operating system and incorporates the functionality of STEP5 programming with innovative expansions and developments.

**Polar coordinates**

A coordinate system, which defines the position of a point on a plane in terms of its distance from the origin and the angle formed by the radius vector with a defined axis.

**Polynomial interpolation**

Polynomial interpolation provides a means of generating a very wide range of curves, including straight-line, parabolic and exponential functions (SINUMERIK 840D).

**Positioning axis**

Axis that performs an auxiliary movement on a machine tool (e.g., tool magazine, pallet transport). Positioning axes are axes that do not interpolate with -> path axes.

**Power on**

Control is switched off and then switched on again.

**Preset**

The control zero point can be redefined in the machine coordinate system by means of the Preset function. Preset does not cause the axes to move; instead, a new position value is entered for the current axis positions.

**Program**

1. Operating area of the control.
2. Sequence of instructions to the control.

**Program block**

Program blocks contain the main and subroutines of the -> part programs.

**Programmable frames**

Programmable -> frames can be used to define new coordinate system starting points dynamically while the parts program is running. A distinction is made between absolute definition using a new frame and additive definition with reference to an existing starting point.

### **Programmable Logic Controller**

Programmable logic controllers (PLC) are electronic controls, the function of which is stored as a program in the control unit. This means that the layout and wiring of the device do not depend on the function of the control. The programmable logic controller has the same structure as a computer; it consists of a CPU (central module) with memory, input/output modules and an internal bus system. The peripherals and the programming language are matched to the requirements of the control technology.

### **Programmable working area limitation**

Limitation of the motion space of the tool to a space defined by programmed limitations.

### **Programming key**

Characters and character sequences, which have a defined meaning in the programming language for -> parts programs (see programming guide).

### **Protection zone**

Three-dimensional space within the -> working area which the tool tip is not permitted to enter.

### **Quadrant error compensation**

Contour errors at quadrant transitions, which arise as a result of changing friction conditions on the guideways, can be virtually entirely eliminated with the quadrant error compensation. Parameterization of the quadrant error compensation is performed by means of a circuit test.

### **R parameters**

Arithmetic parameter for which the programmer of the -> parts program can freely assign or request values in the program.

### **Rapid traverse**

The highest traverse rate of an axis. It is used for example to move the tool from rest position to the -> workpiece contour or retract the tool from the contour.

### **Reference point**

Point on the machine tool used to reference the measuring system of the -> machine axes.

### **Reference point approach**

If the system of position measuring employed is not based on absolute-value encoders, the control must perform a reference point approach in order to ensure that the actual values supplied by the measuring system agree with the machine coordinate values.

**REPOS**

1. Repositioning on the contour using operator input

The REPOS function can use the direction keys to reposition at the point of interruption.

2. Repositioning on the contour by program

The program commands provide various approach strategies: Approach point of interruption, approach start of block, approach end of block, approach a point on the path between start of block and interruption.

**Retentive memory**

Data areas in data blocks and times, counters and markers are retentive (non-volatile) if their contents are not lost when the system is restarted or the mains supply is disconnected.

**Rigid tapping**

This function allows threads to be tapped without a compensating chuck. By using the method whereby the spindle, as a rotary axis, and the drilling axis interpolate, threads can be cut to a precise final drilling depth (e.g. for blind hole threads) (requirement: spindles in axis operation).

**Rotary axis**

Rotary axes apply a workpiece or tool rotation to a defined angular position.

**Rotary axis, turning continuously**

Depending on the application, the travel range of a rotary axis can be limited to less than 360 degrees or the axis can be continuously turned in both directions. Continuously turning rotary axes are used, for example, for eccentric machining, grinding and winding.

**Rotation**

Component of a -> frame which defines a rotation of the coordinate system through a specific angle.

**Rounding axis**

Rounding axes rotate a workpiece or tool to an angular position corresponding to an indexing grid. When a grid index is reached, the rounding axis is "in position".

**S7-300 bus**

The S7-300 bus is a serial data bus by means of which the modules communicate with each other and are supplied with the required voltage. The connections between the modules are made with the -> bus connectors.

## S7-Configuration

S7 configuration is a tool with the aid of which modules can be parameterized. With S7 configuration, various -> parameter blocks of the -> CPU and the I/O modules are set on the -> PG. These parameters are transmitted to the CPU.

## Scaling

Component of a -> frame, which causes axis-specific scale modifications.

## Security functions

The controls contain watchdog monitors which are always active. These monitors detect problems in the CNC, PLC or machine in time to prevent damage to workpiece, tool or machine as far as possible. In the event of a fault, the machining operation is interrupted and the drives stopped. The cause of the malfunction is logged and output as an alarm. At the same time, the PLC is notified that a CNC alarm has been triggered.

## Serial V.24 interface

For data input/output there is

- The MMC100 MMC module includes a serial V.24 interface (RS232)
- The MMC101 and MMC102 MMC modules have two RS232C interfaces

are available. Machining programs and manufacturer and user data can be loaded and saved via these interfaces.

## Services

Operating area of the control.

## Set

A section of a -> parts program terminated with a line feed. A distinction is made between -> main blocks and -> subblocks.

## Setting data

Data, which communicates the properties of the machine tool to the NC control, as defined by the system software.

## Simulator module

A simulator module is a module

- on which digital input variables can be simulated via control elements and
- digital output quantities can be displayed

**Soft key**

A key, whose name appears on an area of the screen. The choice of soft keys displayed is dynamically adapted to the operating situation. The freely assignable function keys (soft keys) are assigned defined functions in the software.

**Software limit switch**

Software limit switches define the limits of the travel range of an axis and prevent the slide contacting the hardware limit switches. 2 pairs of values can be assigned on each axis, which can then be activated separately via the -> PLC.

**Spindles**

The spindle functionality is a two-level construct:

1. Spindles: speed or position-controlled spindle drives, analog digital (SINUMERIK 840D)
2. Auxiliary spindles: speed-controlled spindle drives, "auxiliary spindle" function package, e.g., for driven tools.

**Spline interpolation**

Using the spline interpolation function, the control is able to generate a smooth curve from just a small number of specified interpolation points along a setpoint contour.

**SPS**

-> Programmable Logic Controller

**Standard cycles**

Standard cycles are provided for machining operations, which are frequently repeated:

- Cycles for drilling/milling applications
- Cycles for turning applications (SINUMERIK FM-NC)

The available cycles are listed in the "Cycle support" menu in the "Program" operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plain text.

**Standard rail**

A mounting rail is used to attach the modules of an S7-300.

**Subblock**

Block preceded by "N" containing information for a sequence, e.g., positional data.

## Subprogram

A sequence of instructions in a -> parts program, which can be called repeatedly with various defining parameters. The subroutine is called from a main program. Every subprogram can be protected against unauthorized read-out and display. -> Cycles are a form of subroutine.

## Synchronization

Instructions in -> parts programs for coordination of sequences in different -> channels at specific machining points.

## Synchronized actions

### 1. Auxiliary function output

During the workpiece machining, technology functions (-> auxiliary functions) can be issued from the CNC program to the PLC. These auxiliary functions are used for example to control additional equipment for the machine tool, such as quills, grabbers, clamping chucks etc.

### 2. Fast auxiliary function output

For time-critical switching functions, the confirmation times for the -> auxiliary functions are minimized, and unnecessary stopping points in the machining process can be avoided.

## Synchronized axes

Synchronized axes take the same time to traverse their path as the geometry axes take for their path.

## System memory

The system memory is a memory in the CPU in which the following data are stored:

- Data required by the operating system
- The operands times, counters, markers

## System variables

A variable, which exists although it has not been programmed by the -> parts program programmer. It is defined by a data type and the variable name preceded by the character \$. See also -> User-defined variable.

## Teach In

Teach-in is a means of creating or correcting parts programs. The individual program blocks can be input via the keyboard and executed immediately. Positions approached via the direction keys or handwheel can also be stored. Additional information such as G functions, feedrates or M functions can be entered in the same block.



**Text editor**

-> Editor

**Tool**

A tool is a software tool for inputting and changing the -> parameters of a parameter block. Tools include:

- -> S7 configuration
- S7-TOP
- S7-Info

A part used on the machine tool for machining. Examples of tools include lathe tools, milling cutters, drills, laser beams, etc.

**Tool nose radius compensation**

Contour programming assumes that the tool is pointed. Since this is not actually the case in practice, the curvature radius of the tool used must be communicated to the control, which then takes it into account. The curvature center is maintained equidistantly around the contour, offset by the curvature radius.

**Tool offset**

By programming a T function (5 decades, integer) in the block, you can select the tool. Every T number can be assigned up to nine cutting edges (D addresses). The number of tools to be managed in the control is set at the configuration stage.

**Tool radius compensation**

In order to program a desired -> workpiece contour directly, the control must traverse a path equidistant to the programmed contour, taking into account the radius of the tool used (G41/G42).

**Transformation**

Programming in a Cartesian coordinate system, execution in a non-Cartesian coordinate system (e.g., with machine axes as rotary axes).

**Traversing range**

The maximum permissible travel range for linear axes is  $\pm 9$  decades. The absolute value depends on the selected input and position control resolution and the unit of measurement (inch or metric).

### User-defined variable

The user can declare user-defined variables for any use in the -> parts program or data block (global user data). A definition contains a data type specification and the variable name. See also -> System variable.

### Variable definition

A variable definition includes the specification of a data type and a variable name. The variable names can be used to access the value of the variables.

### Velocity control

In order to be able to achieve an acceptable traversing velocity on very short traverse movements within a single block, predictive velocity control can be set over several blocks (-> look ahead).

### Work offset

Specification of a new reference point for a coordinate system through reference to an existing zero point and a -> frame.

#### 1. settable

SINUMERIK FM-NC: Four independent zero offsets can be selected for each CNC axis.

SINUMERIK 840D A configurable number of settable work offsets are available for each CNC axis. The offsets - which are selected by means of G functions - take effect alternately.

#### 2. external

In addition to all offsets that define the position of the workpiece zero, it is possible to superimpose an external zero offset via one of the following:

- Handwheel (DRF offset)
- PLC

#### 3. Programmable

Zero offsets are programmable for all path and positioning axes with the TRANS command.

### Workpiece

Part to be made/machined by the machine tool.

### Workpiece contour

Setpoint contour of the -> workpiece to be made/machined.

### **Workpiece coordinate system**

The starting position of the workpiece coordinate system is the ->workpiece zero. In machining operations programmed in the workpiece coordinate system, the dimensions and directions refer to this system.

### **Workpiece zero**

The workpiece zero is the starting point for the -> workpiece coordinate system. It is defined by the distance from the machine zero.



# Index

## A

Axis assignment, 15

## B

Behavior when quantity parameter is zero, 91  
Blank, 368  
Boring, 41  
Boring 1, 72  
Boring 2, 75  
Boring 3, 79  
Boring 4, 82  
Boring 5, 85

## C

Call, 14  
Call conditions, 14  
Centering, 44  
Circle of holes, 95  
Circular spigot milling - CYCLE77, 179  
Circumferential slot - SLOT2, 125  
Contour definition, 344, 382  
Contour monitoring, 315, 346  
Contour programming, 377  
CONTPRON, 345  
Cycle alarms, 387  
Cycle call, 15  
Cycle setting data, milling, 102  
Cycle setting data, turning, 314  
Cycle support for user cycles, 30  
Cycle support in the program editor, 20

CYCLE60, 285  
CYCLE72, 164  
CYCLE73, 190  
CYCLE75, 186  
CYCLE76, 173  
CYCLE77, 179  
CYCLE800, 211  
CYCLE801, 98  
CYCLE81, 44  
CYCLE82, 48  
CYCLE83, 50  
CYCLE832, 271  
CYCLE84, 58  
CYCLE840, 65  
CYCLE85, 72  
CYCLE86, 75  
CYCLE87, 79  
CYCLE88, 82  
CYCLE89, 85  
CYCLE899  
    Compatibility, 298  
CYCLE90, 104  
CYCLE93, 317  
CYCLE94, 328  
CYCLE95, 333  
CYCLE950, 367  
CYCLE96, 347  
CYCLE97, 351  
CYCLE98, 359

## D

Deep-hole drilling, 50  
Dot matrix, 98  
Drilling, 44  
Drilling cycles, 11  
Drilling pattern cycles, 91  
Drilling, counterboring, 48

## E

Engraving cycle – CYCLE60, 285  
Extended stock removal cycle - CYCLE950, 367  
External thread, 106

## F

Face thread, 358  
FGROUP, 104  
Final drilling depth, 47, 115, 122, 136, 160, 207  
Free contour programming, 24

## G

Geometrical parameters, 41  
Groove depth, 306  
Grooving cycle - CYCLE93, 317

## H

High speed settings – CYCLE832, 271  
Hole pattern cycles, 12  
Hole pattern cycles without drilling cycle call, 91  
HOLES1, 92  
HOLES2, 95

## L

Long holes located on a circle - LONGHOLE, 112  
LONGHOLE, 112  
Longitudinal thread, 358

## M

Machining parameters, 41  
Machining plane, 14  
MCALL, 88  
Messages, 15, 388  
Milling a circular pocket - POCKET4, 151  
Milling a rectangular pocket - POCKET3, 142  
Milling circular pockets - POCKET2, 137  
Milling cycles, 12  
Milling rectangular pockets - POCKET1, 132  
Modal call, 88

## O

Overview of cycle alarms, 387  
Overview of cycles, 11

## P

Parallel-contour, 367  
Parameter list, 15  
Path milling, 164  
Plane definition, 14  
Plausibility checks, 91  
Pocket milling with islands - CYCLE73, 190  
POCKET1, 132  
POCKET2, 137  
POCKET3, 142  
POCKET4, 151

## R

Rectangular spigot milling - CYCLE76, 173  
Reference plane, 207, 306  
Relative drilling depth, 47, 115, 122, 136, 160, 207  
Residual material, 368  
Retraction plane, 46, 207, 306  
Return conditions, 14  
Rigid tapping, 58  
Row of holes, 92

## S

Safety clearance, 47, 207, 306  
SETMS, 102  
Simulation of cycles, 19  
Simulation without tool, 19  
SLOT1, 118  
SLOT2, 125  
Slots on a circle - SLOT1, 118  
Spindle handling, 314  
SPOS, 61, 62  
Start, 42  
Start point, 346  
Stock removal cycle - CYCLE95, 333  
Swiveling – CYCLE800, 211  
Swiveling in JOG mode, 233

**T**

Tapping with compensating chuck, 65  
Tapping with compensating chuck with encoder, 68  
Tapping with compensating chuck without encoder, 68  
Thread chaining - CYCLE98, 359  
Thread cutting - CYCLE97, 351  
Thread milling, 104  
Thread recutting, 366  
Thread undercut - CYCLE96, 347  
Transfer island contour - CYCLE75, 186  
Turning cycles, 12

**U**

Undercut angle, 315  
Undercut cycle - CYCLE94, 328





To  
SIEMENS AG

A&D MC MS1  
Postfach 3180  
D-91050 Erlangen

Tel.: +49 (0) 180 5050 – 222 [Hotline]  
Fax: +49 (0) 9131 98 – 63315 [Documentation]  
<mailto:docu.motioncontrol@siemens.com>

**From**

Name

Company/Dept.

Street:

Zip code:                      Town:

Phone:                                      /

Fax:    /

Suggestions

Corrections

For Publication/Manual:

SINUMERIK 840D sl/840Di sl/  
840D/840Di/810D

Cycles

User Documentation

Programming Manual

Order No.:            6FC5398-3BP20-1BA0

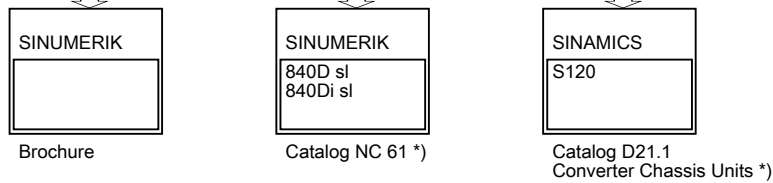
Edition 01/2008

Should you come across any printing errors when  
reading this publication, please notify us on this sheet.  
Suggestions for improvements are also welcome.

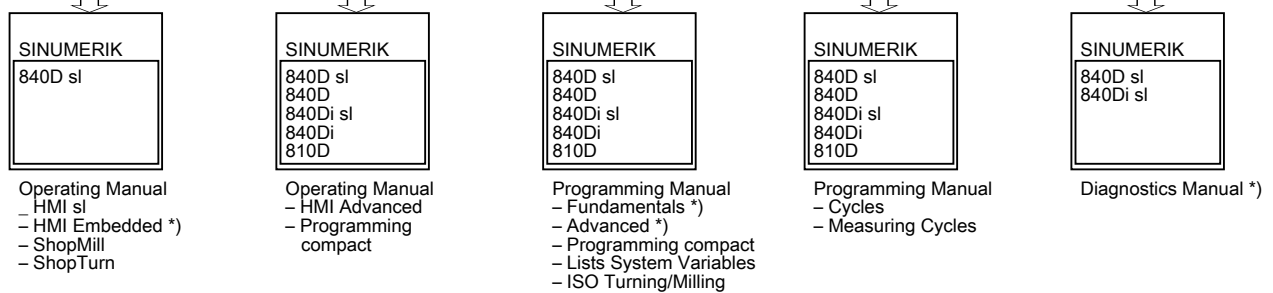
**Suggestions and/or corrections**

## Overview of SINUMERIK 840D sl / 840Di sl Documentation

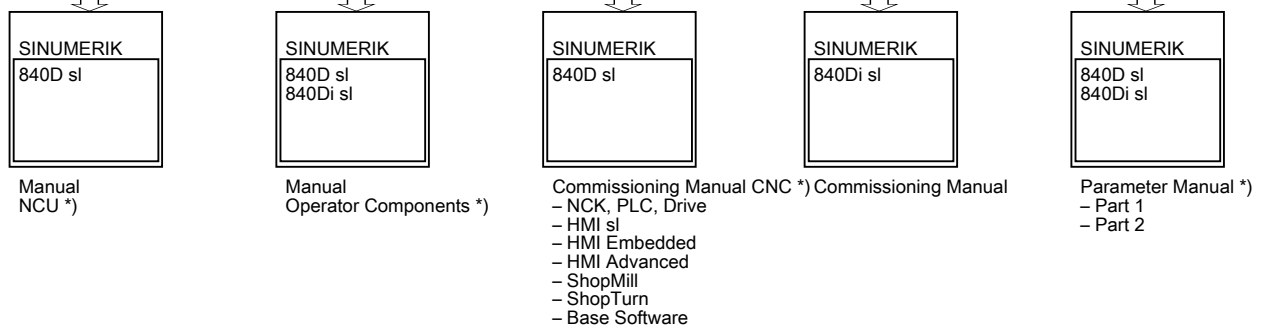
### General Documentation



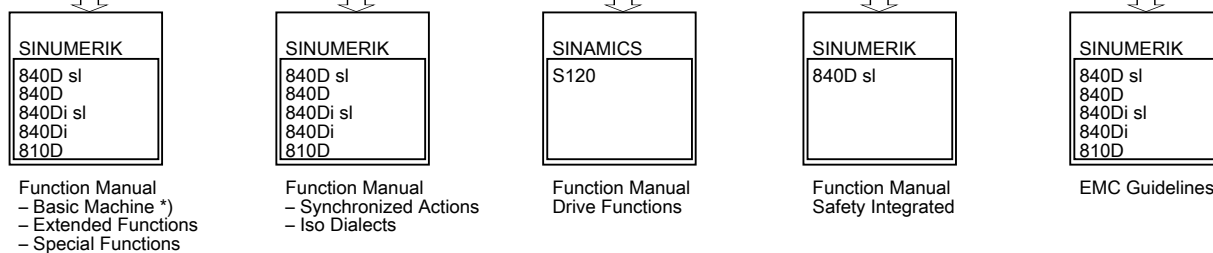
### User Documentation



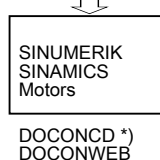
### Manufacturer/Service Documentation



### Manufacturer/Service Documentation



### Electronic Documentation



\*) These documents are a minimum requirement