

Programming Guide 10/2004 Edition

sinumerik

SINUMERIK 840D/840Di/810D  
Cycles

**SIEMENS**



# SIEMENS

## SINUMERIK 840D/840Di/810D

### Cycles

#### Programming Guide

General	1
Drilling Cycles and Drilling Patterns	2
Milling Cycles	3
Turning Cycles	4
Error Messages and Error Handling	5
Appendix	A

#### Valid for

##### *Control*

SINUMERIK 840D powerline  
SINUMERIK 840DE powerline (export version)  
SINUMERIK 840Di  
SINUMERIK 840DiE (export version)  
SINUMERIK 810D powerline  
SINUMERIK 810DE powerline (export version)

##### *Drive*

SIMODRIVE 611 digital

##### *Software*

Cycles

##### *Version*

6

## SINUMERIK® Documentation

### Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is indicated by the code in the "Remarks" column.

*Status code in the "Remarks" column:*

- A ....** New documentation
- B ....** Unrevised reprint with new order number
- C ....** Revised edition with new status

If factual changes have been made on the page since the last edition, this is indicated by a new edition coding in the header on that page.

<b>Edition</b>	<b>Order no.</b>	<b>Remarks</b>
02.95	6FC5298-2AB40-0BP0	A
04.95	6FC5298-2AB40-0BP1	C
03.96	6FC5298-3AB40-0BP0	C
08.97	6FC5298-4AB40-0BP0	C
12.97	6FC5298-4AB40-0BP1	C
12.98	6FC5298-5AB40-0BP0	C
08.99	6FC5298-5AB40-0BP1	C
04.00	6FC5298-5AB40-0BP2	C
10.00	6FC5298-6AB40-0BP0	C
09.01	6FC5298-6AB40-0BP1	C
11.02	6FC5298-6AB40-0BP2	C
03.04	6FC5298-7AB40-0BP0	C
10.04	6FC5298-7AB40-0BP1	C

### Trademarks

SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIROTEC®, SINUMERIK® and SIMODRIVE® are Siemens trademarks. Other product names used in this documentation may be trademarks, which, if used by third parties, could infringe the rights of their owners.

More information is available on the Internet at:  
<http://www.siemens.com>

This document has been created with WinWord V 9.0  
and Designer V 7.0.

© Siemens AG, 1995-2004. All rights reserved

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

We have checked the contents of this publication for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. The information contained in this document is, however, reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

Technical data subject to change

## Preface

### Structure of the documentation

The SINUMERIK documentation is organized in 3 parts:

- General documentation
- User documentation
- Manufacturer/service documentation

### Audience

This Manual is intended for machine-tool users. The document describes in detail all the technical facts an operator needs to understand how to operate the SINUMERIK 810D and 840D control systems.

### Standard scope

This Programming Guide describes the functionality afforded by standard functions (cycles, SW 6.5). Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

More detailed information about other publications concerning SINUMERIK 810D and 840D and publications that apply to all SINUMERIK controls (e.g., Universal Interface, Measuring Cycles, etc.) can be obtained from your local Siemens branch office.

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

### Validity

This Programming Guide is valid for Cycles Version 6.5.

## SINUMERIK 840D powerline

Since September 2001, improved performance versions of

- SINUMERIK 840D powerline and
- SINUMERIK 840DE powerline

have been available. For a list of the available powerline modules, please refer to Section 1.1 of the Hardware Description /PHD/.

## SINUMERIK 810D powerline

Since December 2001, improved performance versions of

- SINUMERIK 810D powerline and
- SINUMERIK 810DE powerline

have been available. For a list of the available powerline modules, please refer to Section 1.1 of the Hardware Description /PHC/.

## 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 at hand.

## 1. A quick overview

If you need to look up a cycle that is rarely used or the meaning of a parameter, you can see at a glance how the function is programmed together with an explanation of the cycles and parameters.

This information is always displayed at the top of the page.

Note:

Due to lack of space, it is not possible to specify all modes of display offered by the programming language for individual cycles and parameters. For this reason, the method of programming shown is the one most commonly used in workshops.

2
Drilling cycles and Drilling Parameters  
2.1 – Drilling cycles
2

**2.1.2 Drilling, centering – CYCLE81**

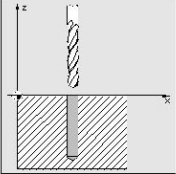
**Programming**

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

RFP	real	Retraction plane (absolute)
RZF	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)

**Function**

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



**Sequence**

Position reached before the beginning of the cycle:  
The drilling position is the position in the two axes of the selected plane.

The cycle implements the following motion sequence:

Approach of the reference plane brought forward by the safety distance with G0.

- Traverse to final drilling depth with the feedrate (G1) programmed in the calling program
- Retraction to retraction plane with G0.

© Siemens AG 2004. All rights reserved.
9/38

## 2. Detailed explanations

You will find detailed answers to the following questions in the theory section:



What is the cycle used for?



What does the cycle do?



How is it programmed and executed?

What do the parameters do?

What else do I need to know?

The theoretical sections are primarily intended as learning material for the NC entry-level user. You should work through the manual at least once to get an idea of the functional scope and capability of your SINUMERIK control.



## 3. From theory to practice

The programming example shows you how to include the cycles in an operating sequence.

An application example of almost all the cycles is provided after the theoretical section.

2
Drilling cycles
2

**?** Explanation of parameters

**RFP and RTP (reference plane and retraction plane)**  
The reference plane (RFP) and retraction plane are generally set to different values. The cycle assumes that the retraction plane is in front of the reference plane. The distance between the retraction plane and the final drilling depth is therefore greater than the distance between the reference plane and the final drilling depth.

**SDIS (safety distance)**  
The safety distance (SDIS) is effective with regard to the reference plane which is brought forward by the safety distance.  
The direction in which the safety distance is active is automatically determined by the cycle.

**DP and DPR (final drilling depth)**  
The final drilling 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.

**Other information**

If a value is entered both for the DP and the DPR, the final drilling depth is derived from the DPR. If the DP deviates from the absolute depth programmed, the message "Depth: According to value for relative depth" is displayed in the dialog line.

If the values for the reference plane and the retraction plane are identical, a relative depth must not be programmed. The following alarm is output: "61101 Reference plane incorrectly defined" and the cycle is not executed.  
This error message is also output if the retraction plane lies behind the reference plane, i.e. the distance to the final drilling depth is smaller.

**?** Programming example

**Drilling\_centering**  
You can use this program to make three holes using the drilling cycle CYCLE81, whereby this cycle is called with different parameter settings. The drilling axis is always the Z axis.

```

N10 G0 F100 Z100 M3           ;Specification of technology values
N20 X1 Y1 Z110              ;Traverse to retraction plane
N30 M8
N40 X40 Y110                ;Traverse to first drilling position
N50 CYCLE81 (L10, L00, Z, 35) ;Cycle call with absolute final drilling
                             ;depth, safety distance, and incomplete
                             ;parameter list
N60 Z10                     ;Traverse to next drilling position
N70 CYCLE81 (L10, L01, , 35) ;Cycle call without safety distance
N80 G0 F100 Z100 M3        ;Specification of technology values
N90 X80                      ;Approach next position
N100 CYCLE81 (L10, L00, Z, , 65) ;Cycle call with relative final drilling depth
                             ;and safety distance
N110 M30                    ;End of program

```

2
Drilling cycles
2

**?** Programming example

**Drilling\_centering**  
You can use this program to make three holes using the drilling cycle CYCLE81, whereby this cycle is called with different parameter settings. The drilling axis is always the Z axis.

```

N10 G0 F100 Z100 M3           ;Specification of technology values
N20 X1 Y1 Z110              ;Traverse to retraction plane
N30 M8
N40 X40 Y110                ;Traverse to first drilling position
N50 CYCLE81 (L10, L00, Z, 35) ;Cycle call with absolute final drilling
                             ;depth, safety distance, and incomplete
                             ;parameter list
N60 Z10                     ;Traverse to next drilling position
N70 CYCLE81 (L10, L01, , 35) ;Cycle call without safety distance
N80 G0 F100 Z100 M3        ;Specification of technology values
N90 X80                      ;Approach next position
N100 CYCLE81 (L10, L00, Z, , 65) ;Cycle call with relative final drilling depth
                             ;and safety distance
N110 M30                    ;End of program

```



**Explanation of symbols**



**Sequence**



**Explanation**



**Function**



**Parameters**



**Programming example**



**Programming**



**Other Information**



Cross-references to other documentation or sections



Danger notes and sources of error



Additional notes or background information



## Warnings

The following warnings, with varying degrees of severity, appear in this document.



### Danger

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



### Warning

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



### Caution

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

### Caution

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

### Notice

Used without the safety alert symbol indicates a potential situation which, if not avoided, **may** result in an undesirable result or state.

### Principle

Your SIEMENS 810D and 840D have been designed and constructed to the latest standards of technology and recognized safety rules, standards and regulations.

### 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.

### Personnel

Only **properly trained, authorized, reliable personnel** must be allowed to use this equipment. No-one without the necessary training may be allowed to operate the control, even temporarily.

The **areas of responsibility** assigned to personnel involved in setting up, operating and maintaining the equipment must be clearly **specified** and their compliance **verified**.

### Procedure

**Before** the control is started up, personnel responsible for its operation must have read and understood the Operator's Guides. The company using this equipment is also obliged to carry out **continuous monitoring** of the overall technical condition of the control (with a view to identifying externally visible defects and damage as well as changes in the operating behavior of the control).

### Servicing

Repairs to equipment may only be carried out by **personnel specially trained and qualified** for the application in question, in accordance with the provisions specified in the maintenance and servicing guides. All relevant safety regulations must be followed.

### Hotline

If you have any questions, please get in touch with our hotline:

A&D Technical Support Phone: +49 (0) 180 5050 - 222

Fax: +49 (0) 180 5050 - 223

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

Please send any queries about the documentation (suggestions or corrections) to the following fax number or e-mail address:

Fax: +49 (0) 0131 98 - 2176

E-mail: [motioncontrol.docu@erlf.siemens.de](mailto:motioncontrol.docu@erlf.siemens.de)

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

### Internet address

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

**Note**

The following are deemed as **improper usage** and **exclude the manufacturer from all liability**:

**Any** usage or application incompatible with or beyond the scope of the items specified above.

Cases where the control is operated **in a technically imperfect condition**, without due provision for safety considerations and/or hazards, or in contravention of any of the instructions in the Operator's Guide.

Cases where faults, which could affect safety, are not remedied **before** starting up the control.

Any **modification, bypassing or decommissioning** of equipment on the control, whose intended purpose is to ensure proper functioning, unrestricted use of equipment and/or active and passive safety.



**Unforeseen danger** can arise with reference to:

- Life and limb of personnel,
- The control, machine or other assets of the owner and the user.

## Notes

## Contents

<b>General</b>	<b>1-17</b>
1.1	General ..... 1-18
1.2	Overview of cycles ..... 1-18
1.2.1	Drilling cycles, drill pattern cycles, milling cycles and turning cycles ..... 1-19
1.2.2	Cycle auxiliary subroutines ..... 1-20
1.3	Programming the cycles ..... 1-21
1.3.1	Call and return conditions ..... 1-21
1.3.2	Messages during execution of a cycle ..... 1-22
1.3.3	Cycle call and parameter list ..... 1-23
1.3.4	Simulation of cycles ..... 1-26
1.4	Cycle support in program editor ..... 1-27
1.4.1	Menus, cycle selection ..... 1-27
1.4.2	New input screen form functions ..... 1-28
1.5	Cycle support for user cycles ..... 1-35
1.5.1	Overview of required files ..... 1-35
1.5.2	Entry to cycle support ..... 1-35
1.5.3	Configuring cycle support ..... 1-36
1.5.4	Bitmap size and screen resolution ..... 1-37
1.5.5	Storing bitmaps in data management for HMI Advanced ..... 1-38
1.5.6	Handling bitmaps for HMI Embedded ..... 1-38
1.6	Installing and starting up the cycles ..... 1-40
1.6.1	Machine data ..... 1-40
1.6.2	Definition files for cycles GUD7.DEF and SMAC.DEF ..... 1-41
1.6.3	New form of delivery for cycles in HMI Advanced ..... 1-43
1.6.4	Upgrading the cycles SW 6.4 and higher in HMI Advanced SW 6.3 and higher ..... 1-43
1.7	Special functions for cycles ..... 1-44
<b>Drilling Cycles and Drilling Patterns</b>	<b>2-47</b>
2.1	Drilling cycles ..... 2-48
2.1.1	Prerequisites ..... 2-50
2.1.2	Drilling, centering – CYCLE81 ..... 2-51
2.1.3	Drilling, counterboring – CYCLE82 ..... 2-54
2.1.4	Deep-hole drilling – CYCLE83 ..... 2-56
2.1.5	Rigid tapping – CYCLE84 ..... 2-63
2.1.6	Tapping with compensating chuck – CYCLE840 ..... 2-70
2.1.7	Boring 1 – CYCLE85 ..... 2-78
2.1.8	Boring 2 – CYCLE86 ..... 2-81
2.1.9	Boring 3 – CYCLE87 ..... 2-85
2.1.10	Boring 4 – CYCLE88 ..... 2-87
2.1.11	Boring 5 – CYCLE89 ..... 2-89
2.2	Modal call of drilling cycles ..... 2-91

2.3	Drill pattern cycles .....	2-94
2.3.1	Prerequisites .....	2-94
2.3.2	Row of holes – HOLES1 .....	2-95
2.3.3	Hole circle – HOLES2 .....	2-99
2.3.4	Dot matrix – CYCLE801 .....	2-102
<b>Milling Cycles</b>		<b>3-105</b>
3.1	General.....	3-107
3.2	Prerequisites .....	3-108
3.3	Thread milling - CYCLE90 .....	3-110
3.4	Elongated holes on a circle - LONGHOLE.....	3-117
3.5	Slots on a circle - SLOT1 .....	3-122
3.6	Circumferential slot - SLOT2.....	3-130
3.7	Milling rectangular pockets - POCKET1 .....	3-136
3.8	Milling circular pockets - POCKET2.....	3-140
3.9	Milling rectangular pockets - POCKET3 .....	3-144
3.10	Milling circular pockets - POCKET4.....	3-153
3.11	Face milling - CYCLE71 .....	3-158
3.12	Path milling - CYCLE72 .....	3-164
3.13	Milling rectangular spigots - CYCLE76 .....	3-174
3.14	Milling circular spigots - CYCLE77.....	3-179
3.15	Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75.....	3-183
3.15.1	Transfer pocket edge contour - CYCLE74.....	3-184
3.15.2	Transfer island contour - CYCLE75.....	3-186
3.15.3	Contour programming .....	3-187
3.15.4	Pocket milling with islands - CYCLE73.....	3-189
3.16	Swiveling - CYCLE800.....	3-211
3.16.1	Operation, parameter assignment, input screen form .....	3-213
3.16.2	Operating instructions, programming instructions .....	3-217
3.16.3	Parameters.....	3-218
3.16.4	Setting tools (with Cycles SW 6.5 and higher).....	3-222
3.16.5	Setting up workpieces with swiveled machining planes (Cycles SW 6.5 and higher) .....	3-223
3.16.6	Operation, parameter assignment “Swiveling in JOG” (with Cycles SW 6.5 and higher) .....	3-227
3.16.7	Startup of swivel cycle CYCLE800.....	3-230
3.16.8	TOOLCARR.SPF user cycle.....	3-251
3.16.9	Error messages.....	3-255
3.17	High-speed settings - CYCLE832 (with Cycles SW 6.3 and higher) .....	3-257
3.17.1	Call CYCLE 832 in HMI menu tree .....	3-260

3.17.2	Parameters.....	3-263
3.17.3	Customizing technology.....	3-264
3.17.4	Customizing additional program parameters CYC_832T.....	3-266
3.17.5	Interfaces .....	3-268
3.17.6	Error messages.....	3-269
3.18	Engraving cycle CYCLE60 (with Cycles SW 6.4 and higher).....	3-270
<b>Turning Cycles</b>		<b>4-281</b>
4.1	General .....	4-282
4.2	Prerequisites .....	4-283
4.3	Grooving cycle – CYCLE93 .....	4-286
4.4	Undercut cycle – CYCLE94 .....	4-295
4.5	Stock removal cycle – CYCLE95.....	4-299
4.6	Thread undercut – CYCLE96 .....	4-312
4.7	Thread cutting – CYCLE97 .....	4-316
4.8	Thread chaining – CYCLE98 .....	4-323
4.9	Thread recutting.....	4-329
4.10	Extended stock removal cycle - CYCLE950.....	4-331
<b>Error Messages and Error Handling</b>		<b>5-353</b>
5.1	General .....	5-354
5.2	Error handling in the cycles .....	5-354
5.3	Overview of cycle alarms .....	5-355
5.4	Cycle messages.....	5-361
<b>Appendix</b>		<b>A-363</b>
A	Abbreviations .....	A-364
B	Definition of Terms.....	A-373
C	References.....	A-393
D	Identifiers.....	A-405
E	Index .....	I-409
F	Commands, identifiers .....	I-411

## Notes



## General

1.1	General .....	1-18
1.2	Overview of cycles .....	1-18
1.2.1	Drilling cycles, drill pattern cycles, milling cycles and turning cycles.....	1-19
1.2.2	Cycle auxiliary subroutines .....	1-20
1.3	Programming the cycles .....	1-21
1.3.1	Call and return conditions .....	1-21
1.3.2	Messages during execution of a cycle.....	1-22
1.3.3	Cycle call and parameter list.....	1-23
1.3.4	Simulation of cycles .....	1-26
1.4	Cycle support in program editor.....	1-27
1.4.1	Menus, cycle selection.....	1-27
1.4.2	New input screen form functions.....	1-28
1.5	Cycle support for user cycles.....	1-35
1.5.1	Overview of required files .....	1-35
1.5.2	Entry to cycle support .....	1-35
1.5.3	Configuring cycle support .....	1-36
1.5.4	Bitmap size and screen resolution .....	1-37
1.5.5	Storing bitmaps in data management for HMI Advanced .....	1-38
1.5.6	Handling bitmaps for HMI Embedded.....	1-38
1.6	Installing and starting up the cycles .....	1-40
1.6.1	Machine data.....	1-40
1.6.2	Definition files for cycles GUD7.DEF and SMAC.DEF .....	1-41
1.6.3	New form of delivery for cycles in HMI Advanced .....	1-43
1.6.4	Upgrading the cycles SW 6.4 and higher in HMI Advanced SW 6.3 and higher.....	1-43
1.7	Special functions for cycles.....	1-44

## 1.1 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.

## 1.2 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 defining parameters.

The system provides you with the technologies

- Drilling
- Milling
- Turning

in various standard cycles.

### 1.2.1 Drilling cycles, drill 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

#### Drill pattern cycles

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

#### Milling cycles

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)
CYCLE90	Thread milling
POCKET3	Rectangular pocket milling (with any milling tool)
POCKET4	Circular pocket milling (with any milling tool)
CYCLE71	Face milling
CYCLE72	Contour milling
CYCLE73	Pocket milling with islands
CYCLE74	Transfer of pocket edge contour
CYCLE75	Transfer of island contour
CYCLE76	Rectangular spigot milling
CYCLE77	Circular spigot milling

## 1.2 Overview of cycles

### Turning cycles

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

### 1.2.2 Cycle auxiliary subroutines

The following auxiliary routines are part of the cycles package

- PITCH and
- MESSAGE.

These must always be loaded in the control.

### 1.3 Programming the cycles

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



The cycles are supplied on diskette or, in the case of HMI Advanced, with the corresponding software release. They are loaded into the parts program memory of the control via the RS 232C interface (see Operator's Guide).

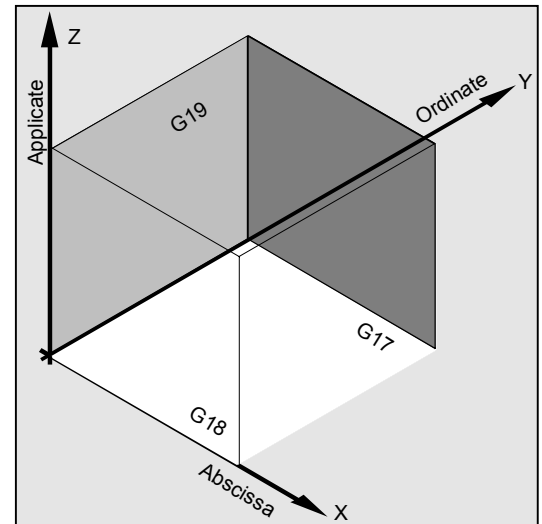
#### 1.3.1 Call and return conditions

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

Define the machining plane (G17, G18, G19) before calling the cycle. A cycle operates in the current plane with

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

In drilling cycles, the hole is machined in the axis that corresponds to the applicate of the current plane. The depth infeed is performed in this axis with milling applications.



#### Plane and axis assignments

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

### 1.3.2 Messages during execution of a cycle

For some cycles, messages that refer to the state of machining are displayed on the control screen during execution.

These messages do not interrupt program processing and continue to be displayed on the screen until the next message appears.

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



You will find a summary of all the relevant 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.

### 1.3.3 Cycle call and parameter list

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



*Cycle calls must always be programmed in a separate block.*

#### **Basic instructions regarding assignment of standard cycle parameters**

The Programming Guide describes the parameter list of each cycle together with the

- Sequence and
- Type.

The sequence of the defining parameters must be observed.

Each defining parameter of a cycle is of a specific data type. The parameter type in use must be observed when the cycle is called. The parameter list is used to pass

- Variables or
- Constants.

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

- With an incomplete parameter list or
- By leaving out parameters.

If you want to exclude the last transfer parameters that have to be written in a call, you can prematurely terminate the parameter list with ")". To leave out parameters in the middle of the sequence, use a comma, "... , ...", as a placeholder.

## 1.3 Programming the cycles



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.

If the cycle is called with a parameter list containing more items than are defined as parameters in the cycle, the general NC alarm 12340 "Too many parameters" is output and the cycle is not executed.

### 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	Definition of the machined shape Values: E and F
MID	Infeed depth (enter without sign)
FFR	Feed
VARI	Machining type Values: 0, 1 or 2
FAL	Final machining allowance

The cycle is called with the command  
Cycle100 (FORM, MID, FFR, VARI, FAL).

#### 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 arithmetic variables that you define and assign values to before you call the cycle.



**Example**

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

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

For defining cycles with parameters you may use variables such as R parameters (R variables).

**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.



More detailed information about data types and type conversion and compatibility is given in the Programming Guide. If the types are incompatible, the following alarm is output  
12330 "Parameter type ... incorrect".

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

If a defining parameter is not required for a cycle call or it is to be assigned the value zero, it can be omitted from the parameter list. A comma, "... , ... ", must be written in its place to ensure correct assignment of the subsequent parameters or the parameter list must be concluded prematurely with ")".

## 1.3 Programming the cycles

### Example

CYCLE100 ("F", 3, 0.3, , 1)	;Cycle call, ;omit 4th parameter (i.e., zero setting)
CYCLE100 ("F", 3, 0.3)	;Cycle call ;the value zero is assigned to the last two ;parameters (i.e., they have been left out)

### 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, feedrate 300

### 1.3.4 Simulation of cycles

Programs with cycle calls can be tested initially by the simulation function.



#### 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 MMC.

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.

## 1.4 Cycle support in program editor

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



### Function

The cycle support offers the following functions:

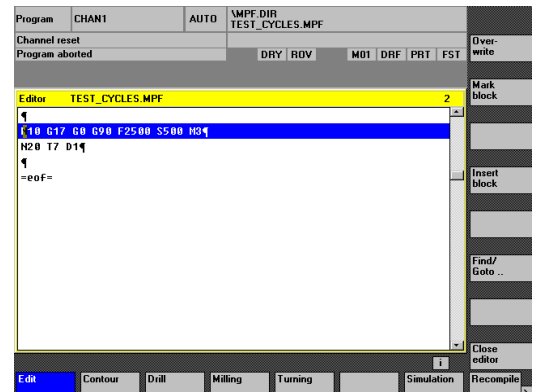
- Cycle selection via soft keys
- Input screen forms for parameter assignment with help displays
- Online help for each parameter (HMI Advanced only)
- Support of contour input

Retranslatable program code is generated from the individual screen forms.

### 1.4.1 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

Drill

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 one screen form, the user may switch between cycles via soft key, e.g., with tapping or undercut.

## 1.4 Cycle support in program editor

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.

### 1.4.2 New input screen form functions



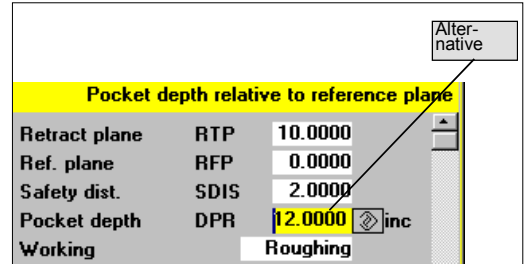
#### Function


- Many cycles allow you to influence the processing type via the VARI parameter. It contains several settings composing one code. Each individual setting has its own input field in the new cycle support screen forms. You can switch between input fields with 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. In the example, this is the case with the parameter for the finishing feedrate.
- One input may, therefore, automatically assign several dependent parameters. This is the case with threading, which currently supports metric thread tables. With the threading cycle CYCLE97, for example, entering 12 in the thread size input field (MPIT parameter) automatically assigns 1.75 to the thread pitch and 1.137 to the thread depth (TDEP parameter). 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.

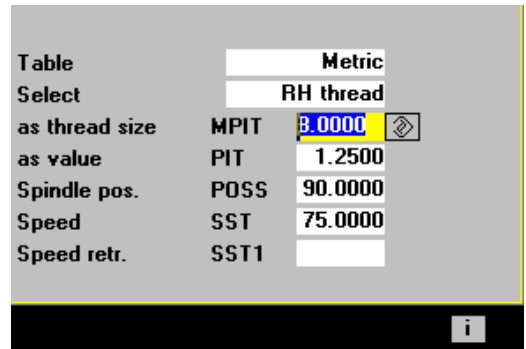
When cycles are called up several times in a row in the same program (e.g., pocket milling when roughing and finishing), few parameters then have to be changed.

Machining: Complete/roughing/finishing	
NPP	Welle1
Working	Complete
Select	Longitudinal
Select	Outside
Infeed depth	MID 2.0000

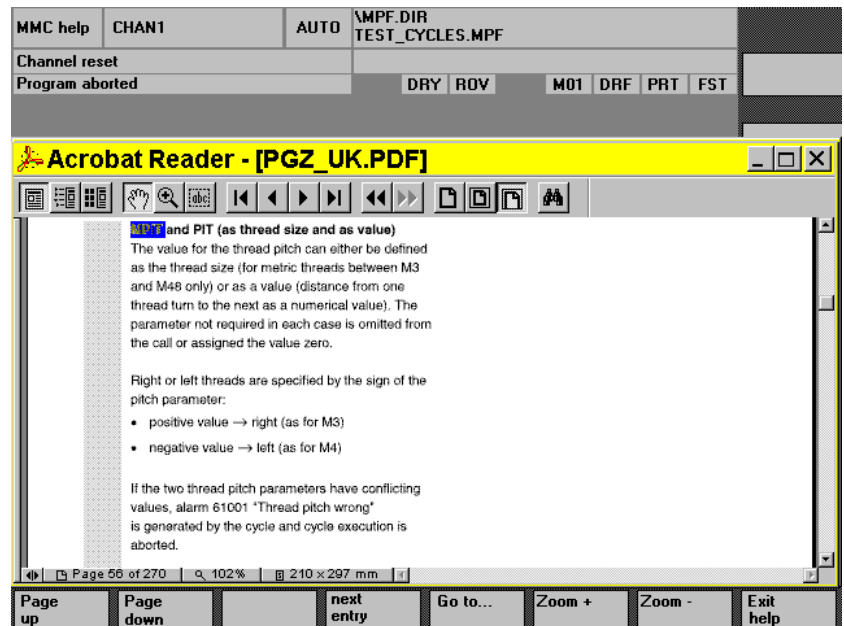
- In screen forms for drilling and milling cycles, certain parameters may be input as absolute or incremental values. The abbreviation ABS for absolute and INC for incremental input is displayed behind the input field. To switch between them, use the "Alternative" soft key. This setting will remain the next time these screen forms are called.



- HMI Advanced allows you to view additional information on each cycle parameter in the online help. Simply place the cursor on a parameter and the help icon  will appear on the bottom right of the screen. Now you can activate the help function.



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





## Operator commands in 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

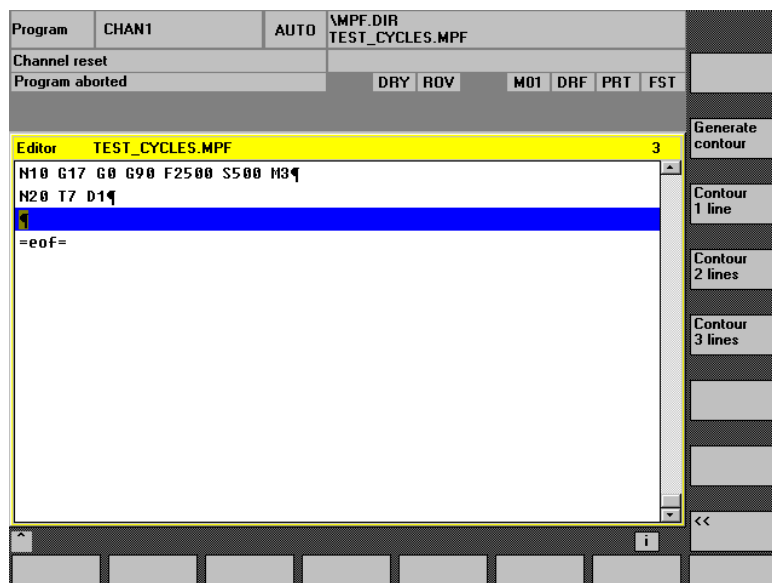
### Free contour programming

Generate contour	Starts free contour programming for entering assembled contour sections (see References: /BA/, Chapter 6)
------------------	---

### Blueprint programming

Contour 1 line	These soft keys support the possible blueprints.
Contour 2 lines	
Contour 3 lines	

These consist of one or several straight lines with intermediate 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 input screen form for a 2-straight-line contour definition creates the following DIN code:

Program	CHAN1	AUTO	\MPF.DIR TEST_CYCLES.MPF	
Channel reset	Program aborted		DRY	ROV
			M01	DRF
			PRT	FST
Contour 2 lines			Add free NC code to NC block	
Select plane			G17	
End point 2				
X	20.0000		abs	
Y			abs	
Angle	ANG1	87.3000		
Transit.	RD	2.5000		
Free NC program input			F2500 S500 M3	
End point 3				
X	10.0000		inc	
Y	-20.0000		inc	
Angle	ANG2			
Free NC program input				

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.

Drilling centering

Deep hole drilling

Bore

Tapping

Selection of drilling patterns

Hole pattern pos.

Deselect modal

Program	CHAN1	AUTO	\MPF.DIR TEST_CYCLES.MPF	
Channel reset	Program aborted		DRY	ROV
			M01	DRF
			PRT	FST
Editor TEST_CYCLES.MPF			1	
N10 G17 G0 G90 F2500 S500 M3				
N20 T7 D1				
=eof=				



Cycles CYCLE81, CYCLE87 and CYCLE89 cannot be parameterized with this support.

The CYCLE81 function is covered by CYCLE82 (soft key "Drilling centering"), as is the function of CYCLE89.

The CYCLE87 function is covered by the function of CYCLE88 (soft keys "Drilling centering" → "Drilling with stop").

## 1.4 Cycle support in program editor

Drilling patterns may be repeated if, for example, drilling and tapping are to be executed in succession. A name is simply assigned to the drilling pattern and entered later in screen form "Repeat position".



### Programming example generated by cycle support

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	;Modal call of tapping cycle
CYCLE84 (10,0,2,0,30,,3,5,0.8,180,300,500)	
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			
Select plane G17			
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 five positions may be programmed in the plane; all values are either absolute or incremental (use the "Alternative" soft key to toggle between them). The "Delete all" soft key creates an empty screen form.



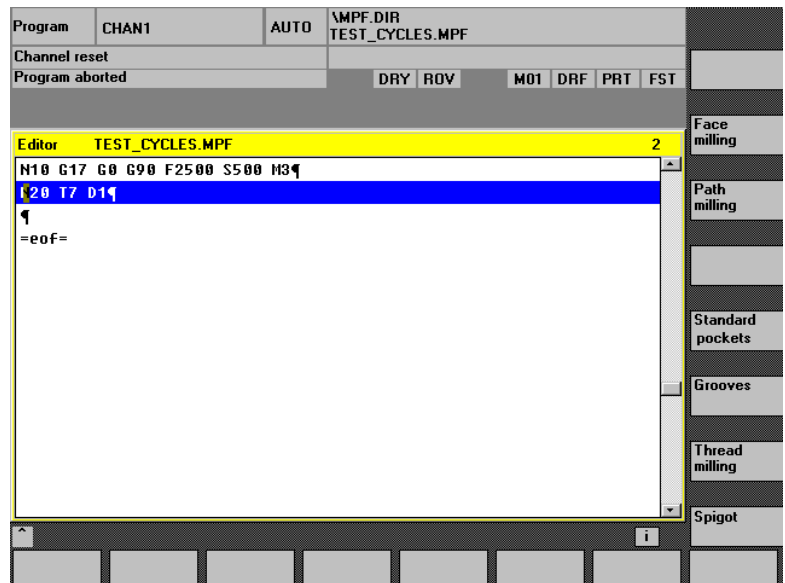


## Milling support

Milling support includes the following selection possibilities:

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

The "Standard pockets", "Grooves" and "Spigots" soft keys each branch into submenus offering a selection of pocket, groove and spigot milling cycles.



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

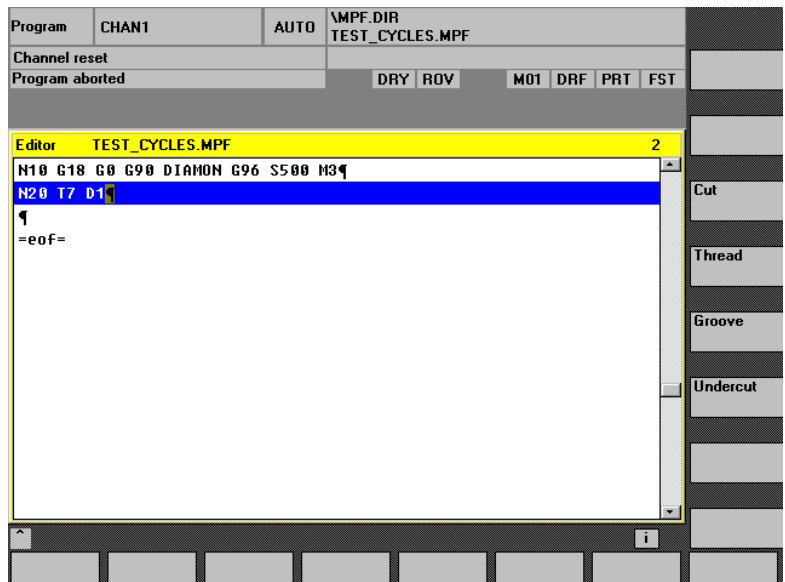


## Turning support

Turning support includes the following selection possibilities:

Cut
Thread
Groove
Undercut

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



The "Thread" soft key contains a submenu for selecting between single thread cutting and thread chaining.



### Retranslation

Retranslating program codes serves to change an existing program with the help of cycle support. Place the cursor on the line to be changed and press the "Retranslation" soft key.

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 retranslation. Therefore, consistent use of the cycle support is required and modifications are to be carried out with the help of retranslation.



### Configuring support for user cycles



#### References: /IAM/, Installation & Startup Guide HMI

- BE1 "Expand the Operator Interface"
- IM2 "Installation and Startup of HMI Embedded"
- IM4 "Installation and Startup of HMI Advanced"

## 1.5 Cycle support for user cycles

### 1.5.1 Overview of required files

The following files form 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)	pgc_<language>.pdf and pgc_<language>.txt	Standard cycles only	pdf file



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

### 1.5.2 Entry to 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") ;Operating a soft key loads the soft key bar from
;file cycproj1.com
END_PRESS
```

## 1.5 Cycle support for user cycles

A detailed description of the configuration is given in:

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

BE1 "Expand the Operator Interface"

With HMI Embedded, this soft key should be activated by making the following entry in the common.com file:

```

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

```

### 1.5.3 Configuring cycle support

#### 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/, Installation & Startup Guide HMI

BE1 "Expand the Operator Interface"

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

- dh\cst.dir
- 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" through RS 232C). 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

- Copy file arj.exe from directory \hmi\_emb\tools to an empty directory on a PC.
- Copy the separate \*.com configuration files to this directory.
- Pack each individual com file with the command  
arj a <name target file> <name source file>.  
The target files must have the extension **co\_**.  
Example: Pack configured file cycproj1.com to:  
arj a cycproj1.co\_ cycproj1.com.
- 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/, Installation & Startup Guide HMI

IM2 "Installation and Startup of HMI Embedded"

#### 1.5.4 Bitmap size and screen resolution

With SW 6.2, there are three different screen resolutions in 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 16-color bitmaps.

### 1.5.5 Storing bitmaps in data management for HMI Advanced

New paths have been set up in data management (as of HMI 6.2) for the different screen resolutions, so that the bitmaps can be stored in parallel in various 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

The search begins in the appropriate directory for the current resolution (e.g., in dh\...\hlp.dir\640.dir for 640 \* 480) and then moves to dh\...\hlp.dir. Otherwise, the search sequence cus.dir, cma.dir, cst.dir applies.

### 1.5.6 Handling bitmaps for HMI Embedded



#### Function

With HMI Embedded, the bitmaps are incorporated in the HMI software. They are grouped together in a package named **cst.arj**. Bitmaps can always be integrated in \*.bmp format. However, to save more space and display faster, use a \*.bin binary format. To do this, you will need the tools supplied with the standard cycle software in the \hmi\_emb\tools directory:

- arj.exe, bmp2bin.exe and
- sys\_conv.col
- arj\_idx.exe (as of SW 6.3)

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

- Copy all the files from directory \hmi\_emb\tools to an empty directory on a PC.
- Create a subdirectory \bmp\_file in this new directory.
- Copy your own \*.bmp bitmaps to this \bmp\_file subdirectory.
- Depending on the resolution, for which a cst.arj is being created, start mcst\_640.bat/mcst\_800.bat or mcst1024.bat.
- The cst.arj created will then be located in the same directory as the generation tools.
- 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.



cst.arj is then integrated in the HMI software as described in Subsection 1.4.6.



**References:** /BEM/, HMI Embedded Operator's Guide  
/IAM/, Installation & Startup Guide HMI

IM2 "Installation and Startup of HMI  
Embedded"

## 1.6 Installing and starting up the cycles

### 1.6.1 Machine data

The machine data below are used for the 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



*This information applies to **Siemens standard cycles only**. The relevant values have to be **added** for user cycles. 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	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.



*Axis-specific machine data MD 30200: NUM\_ENC5 must also be noted with respect to cycle CYCLE840 (tapping with compensating chuck).*



## 1.6.2 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.

To make it easier for startup engineers to compile GUDs and macros in a module without editing the original SIEMENS files, as of software version 6.3, files

- GUD7.DEF
- MAC.DEF

are supplied complete with the "standard cycles".

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 codes have been assigned by SIEMENS:

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

- 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

Other codes not listed here may be used by the system at a later date!

## 1.6 Installing and starting up the cycles



### Note

According to the definition of "defining user data" (see Programming Guide Advanced Section 3.4 "Defining user data"), modules GUD7 and SMAC.DEF are NOT available to the machine manufacturer/user. MGUD, UGUD, GUD4,8,9 and MMAC, UMAC should preferably be employed in user applications.

However, to enable users to integrate existing in-house definitions into these modules in this system, the following codes have been kept free:

- xxx\_CMA Manufacturer
- xxx\_CUS User



### Startup, upgrades for standard cycles:

- If a GUD7.DEF is already active in the control, use "Services", "Data out", "NC active data" to select the user data of GUD7 and back up the current values in an archive or onto a diskette;
- Read in files GUD7\_SC.DEF and SMAC\_SC.DEF from the diskette and load them to the NCU;
- Read in and activate GUD7.DEF and SMAC.DEF;
- Turn on the NCU;
- Read the archive of saved values back in.



### Loading an additional cycle package:

- Unload GUD7.DEF and SMAC.DEF (having backed up the values if necessary);
- Read in cycles GUD7\_xxx.DEF and SMAC\_xxx.DEF of the package and load them to the NCU;
- 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.



### Handling 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.

### 1.6.3 New form of delivery 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 directly in the relevant data management directories, but 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.

To upgrade, 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:

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

### 1.6.4 Upgrading the cycles SW 6.4 and higher in HMI Advanced SW 6.3 and higher



In HMI\_Advanced SW 6.3 and higher, the cycles are stored as archives under "Archive", "Cycle archives". Upgrading is performed by reading in an archive contained in the software supplied from directory HMI\_Adv. This unpacks the cycle archives and overwrites the existing archives of the previous cycle version.

The archives from the cycle archive of the HMI now have to be read in for the required technology (turning, milling, etc.) and language via "Data in".



**References:** For current information see:

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

## 1.7 Special functions for cycles



### Function

To provide an overview and for diagnosis of the cycle versions and their definition files, as of SW 6.3 it will be possible to display and use version screens. These can be found in the HMI under "Diagnosis" => "Service displays" => "Version" => "Cycles" or "Definitions". A LOG file can be created in ASCII format and read out via "Services" => "Diagnosis" => "LOG files".

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

Different overviews are possible with cycle version display:

- An overview of all available cycles
- An overview of the individual directories of data management for user cycles (CUS.DIR), manufacturer cycles (CMA.DIR) and Siemens cycles (CST.DIR)
- A package overview of all the cycle packages available in the control
- Details of each package and cycle file.



**References:** /BAD/, Operator's Guide HMI Advanced  
/BEM/, HMI Embedded Operator's Guide  
Service display chapter

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 is introduced for package lists

\*.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:

1. Line:       Version entry (after the vocabulary word  
                  ;**VERSION:**) and  
                  package name (after the vocabulary word  
                  ;**PACKAGE:**)
- From 2. line: List of the files associated with the  
                  cycle package with 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	MPFO
Channel RESET			Program aborted
			ROV SBL1
<b>Version data Cycles</b>			
Name	Version		
Measuring cycles	06.02.08 Mar 21, 2002		
Cycle package	01.02.03 31.10.2002		

^
NCU version
HMI version
Cycle version
Definition version

Package overview
Details
All cycles
User cycles
OEM cycles
Siemens cycles

## 1.7 Special functions for cycles

The following is displayed in the file 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

### Other information

The cycle package name following the vocabulary word PACKAGE can also be written as a string in " ", but it is then language-specific.

### Version entries in cycles

Just as with package lists, the entry after the vocabulary word ";VERSION:" is interpreted 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: February 1, 2003
October 31, 2002
;Comment
PROC CYCLE1 (REAL PAR1)
...
```



## Drilling Cycles and Drilling Patterns

2.1	Drilling cycles .....	2-48
2.1.1	Prerequisites .....	2-50
2.1.2	Drilling, centering – CYCLE81 .....	2-51
2.1.3	Drilling, counterboring – CYCLE82 .....	2-54
2.1.4	Deep-hole drilling – CYCLE83 .....	2-56
2.1.5	Rigid tapping – CYCLE84 .....	2-63
2.1.6	Tapping with compensating chuck – CYCLE840 .....	2-70
2.1.7	Boring 1 – CYCLE85 .....	2-78
2.1.8	Boring 2 – CYCLE86 .....	2-81
2.1.9	Boring 3 – CYCLE87 .....	2-85
2.1.10	Boring 4 – CYCLE88 .....	2-87
2.1.11	Boring 5 – CYCLE89 .....	2-89
2.2	Modal call of drilling cycles .....	2-91
2.3	Drill pattern cycles .....	2-94
2.3.1	Prerequisites .....	2-94
2.3.2	Row of holes – HOLES1 .....	2-95
2.3.3	Hole circle – HOLES2 .....	2-99
2.3.4	Dot matrix – CYCLE801 .....	2-102

## 2.1 Drilling cycles

The following sections describe how

- Drilling cycles and
  - Drill pattern cycles
- are programmed.

These sections are intended to guide you in selecting cycles and assigning them with parameters. After a detailed description of the function of each cycle and its parameters, you will also find a programming example at the end of each section, in order for you to familiarize yourself with using cycles.

The sections are structured as follows:

- Programming
- Parameters
- Function
- Sequence
- Explanation of parameters
- Other information
- Programming example

"Programming" and "Parameters" explain the use of cycles sufficiently for the experienced user, whereas beginners can find all the information they need to program cycles under "Function", "Sequence", "Explanation of parameters", "Other information" and the "Programming example".





Drilling cycles are motion sequences defined 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.

Five cycles are available for boring. They all follow a different technological procedure and are, therefore, parameterized differently:

Boring cycle		Special parameterization features
Boring 1 -	CYCLE85	Different feedrates for boring and retraction
Boring 2 -	CYCLE86	Oriented spindle stop, definition of retraction path, retraction in rapid traverse, definition of spindle direction of rotation
Boring 3 -	CYCLE87	Spindle stop M5 and program stop M0 at drilling depth, continued machining after NC Start, retraction in rapid traverse, definition of spindle direction of rotation
Boring 4 -	CYCLE88	As for CYCLE87 plus dwell time at drilling depth
Boring 5 -	CYCLE89	Boring and retraction at the same feedrate



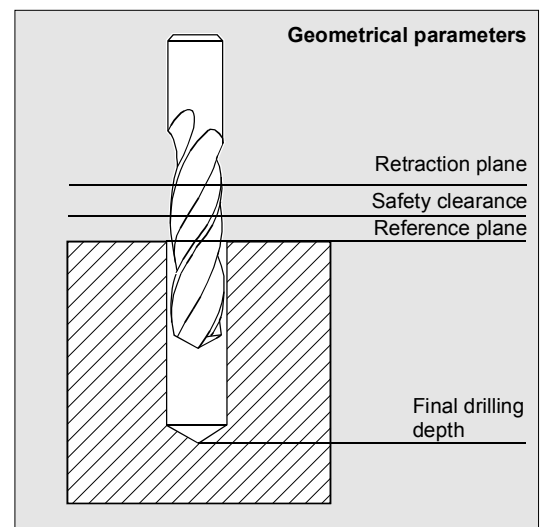
Drilling cycles can be modal, i.e., they are executed at the end of each block that contains motion commands. Other cycles written by the user can also be called modally (see Section 2.2).

There are two types of parameter:

- Geometrical parameters and
- machining parameters.

Geometrical parameters are identical for all drilling cycles, drilling pattern cycles and milling cycles. They define the reference and retraction planes, the safety distance and the absolute and relative final drilling depths. Geometrical parameters are written once in the first drilling cycle CYCLE81.

The machining parameters have a different meaning and effect in each cycle. Therefore, they are described separately in each cycle.



### 2.1.1 Prerequisites

#### Call and return conditions

Drilling cycles are programmed independently of the actual axis names. The drilling position must be approached in the higher-level program before the cycle is called.

The required values for the feedrate, spindle speed and spindle direction of rotation must be programmed in the parts program, if there are no assignment parameters for these values 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 applicate of this coordinate system.

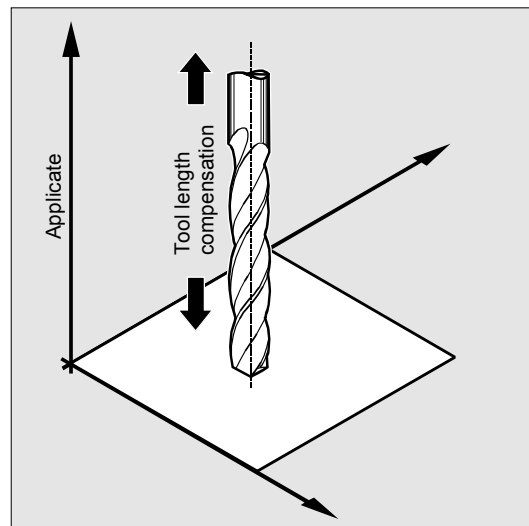
A tool length compensation must be selected before the cycle is called. Its effect is always perpendicular to the selected plane and it remains active even after the end of the cycle (see also Programming Guide).

#### Spindle handling

The drilling cycles are written in such a way that the spindle commands contained within them always refer to the active master spindle on the control. If you want to use a drilling cycle on a machine with several spindles, you must first define the spindle that is to be used for the operation as the master spindle (see also Programming Guide).

#### Dwell time programming

The parameters for the 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 are expressly stated.



## 2.1.2 Drilling, centering – CYCLE81



### Programming

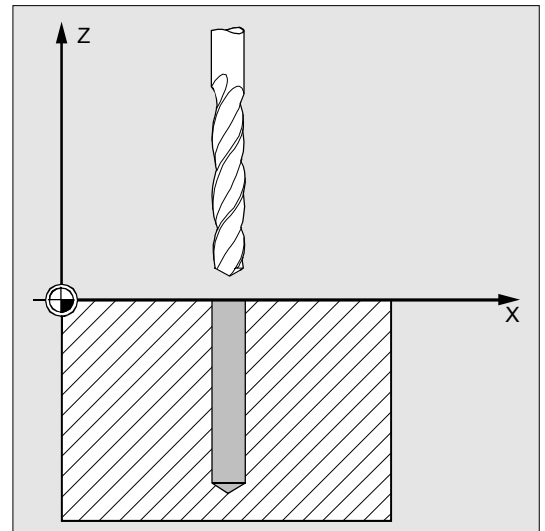
CYCLE81 (RTP, RFP, SDIS, DP, DPR)

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)



### Function

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



### Sequence

#### Position reached before the beginning of the cycle:

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

#### The cycle implements the following motion sequence:

Approach of the reference plane brought forward by the safety distance with G0.

- Traverse to final drilling depth with the feedrate (G1) programmed in the calling program.
- Retraction to retraction plane with G0.



### Explanation of parameters

#### RFP and RTP (reference plane and retraction plane)

The reference plane (RFP) and retraction plane are generally set to different values. The cycle assumes that the retraction plane precedes the reference plane. Therefore, the distance between the retraction plane and the final drilling depth is greater than the distance between the reference plane and the final drilling depth.

#### SDIS (safety distance)

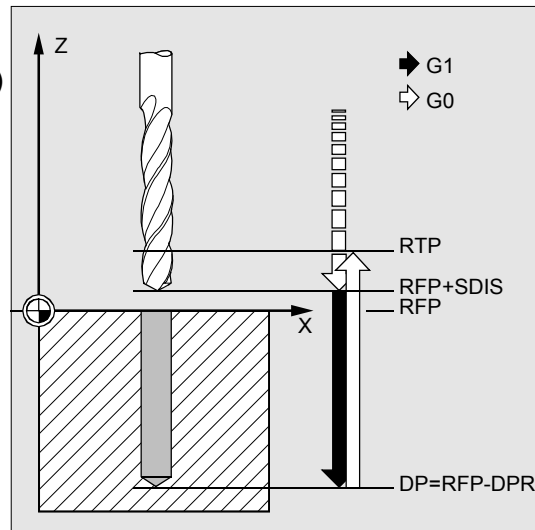
The safety distance (SDIS) is effective with regard to the reference plane, which is brought forward by the safety distance.

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

#### DP and DPR (final drilling depth)

The final drilling 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.



### Other information

If a value is entered both for the DP and the DPR, the final drilling depth is derived from the DPR. If the DP deviates from the absolute depth programmed, the message "Depth: According to value for relative depth" is displayed in the dialog line.

If the values for the reference plane and the retraction plane are identical, a relative depth must not be programmed. The following error message is output: 61101 "Reference plane incorrectly defined" and the cycle is not executed.

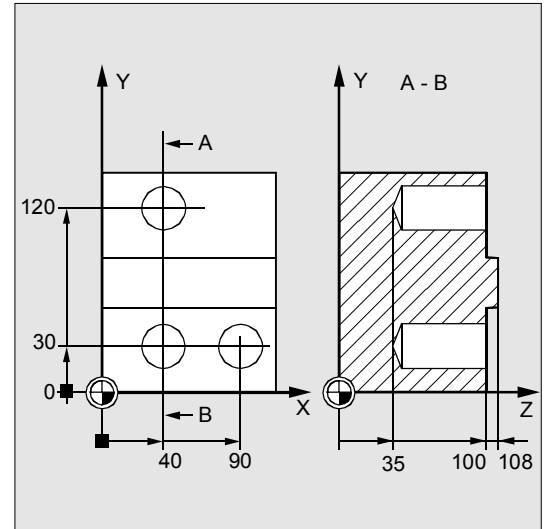
This error message is also output if the retraction plane lies behind the reference plane, i.e., the distance to the final drilling depth is smaller.



### Programming example

#### Drilling\_centering

You can use this program to make three holes using the drilling cycle CYCLE81, whereby this cycle is called with different parameter assignments. 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 final drilling ;depth, safety distance, and incomplete ;parameter list
N50 Y30	;Approach next drilling position
N60 CYCLE81 (110, 102, , 35)	;Cycle call without safety distance
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 final drilling depth ;and safety distance
N100 M30	;End of program

## 2.1.3 Drilling, counterboring – CYCLE82

**Programming**

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

**Parameters**

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)
DTB	real	Dwell time at final drilling depth (chip breaking)

**Function**

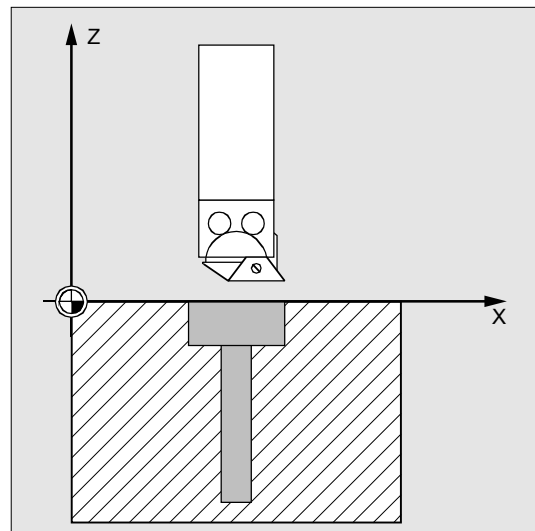
The tool drills at the programmed spindle speed and feedrate to the programmed final drilling depth. A dwell time can be allowed to elapse once the final drilling depth has been reached.

**Sequence****Position reached before the beginning of the cycle:**

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

**The cycle implements the following motion sequence:**

- Approach of the reference plane brought forward by the safety distance with G0.
- Traverse to final drilling depth with the feedrate (G1) programmed in the calling program.
- Execute dwell time at final drilling depth.
- Retraction to retraction plane with G0.





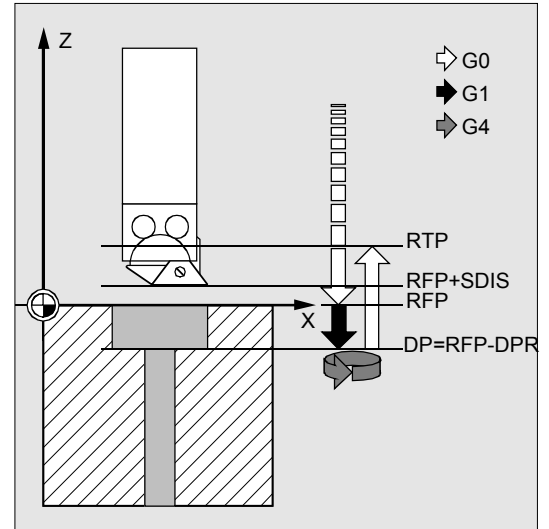
### Explanation of parameters



For parameters RTP, RFP, SDIS, DP, DPR, see Subsection 2.1.2. (Drilling, Centering – CYCLE81)

#### DTB (dwell time)

Parameter DTB is the dwell time at the final drilling depth (chip breaking) in seconds.

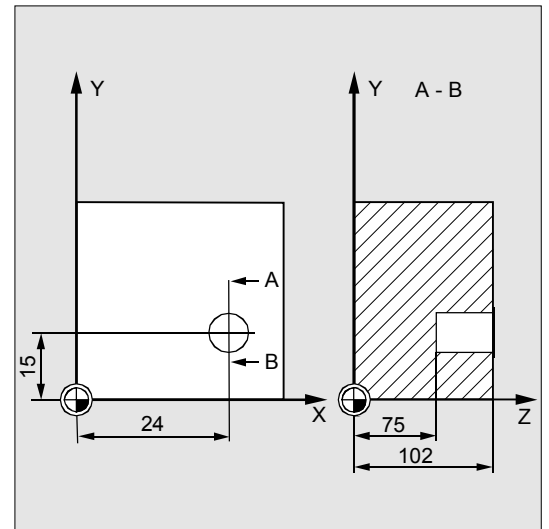


### Programming example

#### Boring\_counterboring

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

The dwell time programmed is 2 sec, the safety distance 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 final drilling ;depth and safety distance
N50 M30	;End of program

## 2.1 Drilling cycles

### 2.1.4 Deep-hole drilling – CYCLE83



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)
FDEP	real	First drilling depth (absolute)
FDPR	real	First drilling depth relative to reference plane (enter without sign)
DAM	real	Degression: (enter without sign) Values: > 0 degression as value < 0 degression factor = 0 no degression
DTB	real	Dwell time at drilling depth (chip breaking) Values: > 0 in seconds < 0 in revolutions
DTS	real	Dwell time at starting point and for swarf 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	int	Type of machining: Values: 0 chip breaking 1 swarf removal
_AXN	int	Tool axis: Values: 1 = 1st geometry axis 2 = 2nd geometry axis or else 3rd geometry axis
_MDEP	real	Minimum drilling depth
_VRT	real	Variable retraction distance for chip breaking (VARI=0): Values: > 0 is retraction distance 0 = Setting is 1 mm
_DTD	real	Dwell time at final drilling depth Values: > 0 in seconds < 0 in revolutions = 0 value as for DTB
_DIS1	real	Programmable limit distance on re-insertion in hole (VARI=1 for swarf removal) Values: > 0 programmable value applies = 0 automatic calculation





## Function

The tool drills at the programmed spindle speed and feedrate to the programmed final drilling depth. Deep-hole drilling is performed with a depth infeed of a maximum definable depth, which is executed several times, increasing gradually until the final drilling depth is reached.

The drill can either be retracted to the reference plane + safety distance after every infeed depth for swarf removal or retracted by the programmed retraction distance for chip breaking.



## Sequence

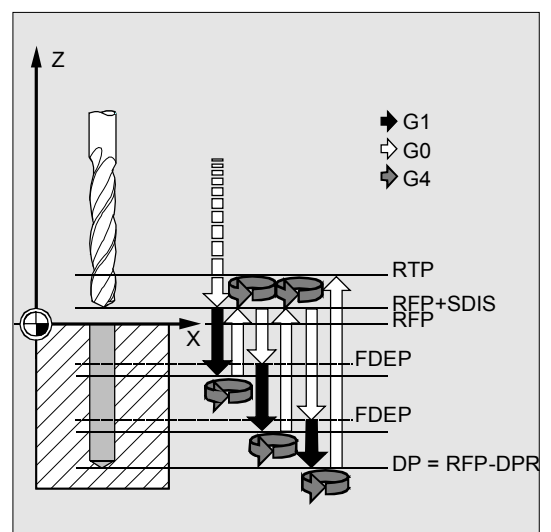
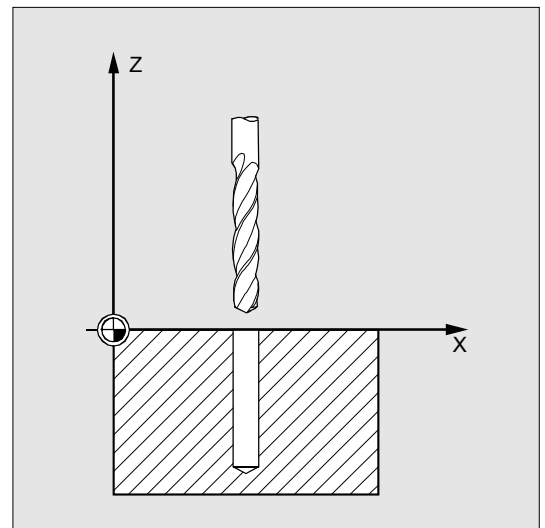
### Position reached before the beginning of the cycle:

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

### The cycle implements the following motion sequence:

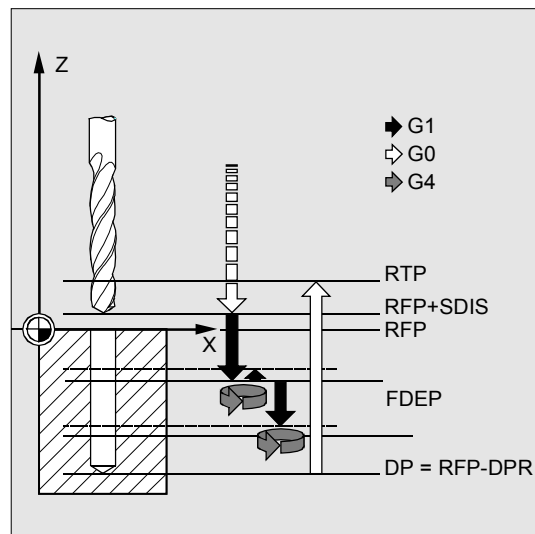
#### Deep-hole drilling with swarf removal (VARI=1):

- Approach of the reference plane brought forward by the safety distance with G0.
- Traverse to the first drilling depth with G1, the feedrate for which is derived from the feedrate defined with the program call, which is subject to parameter FRF (feedrate factor).
- Execute dwell time at final drilling depth (parameter DTB).
- Retraction to the reference plane brought forward by the safety distance with G0 for swarf removal.
- Execute dwell time at starting point (parameter DTS).
- Approach last drilling depth reached, reduced by the calculated (by cycle) or programmable limit distance with G0.
- Traverse to next drilling depth with G1 (sequence of motions is continued until the final drilling depth is reached).
- Retraction to retraction plane with G0.



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

- Approach of the reference plane brought forward by the safety distance with G0
- Traverse to the first drilling depth with G1, the feedrate for which is derived from the feedrate defined with the program call, which is subject to parameter FRF (feedrate factor)
- Execute dwell time at final drilling depth (parameter DTB)
- Variable retraction (parameter \_VRT) from the current drilling depth with G1 and the feedrate programmed in the calling program (for chip breaking)
- Traverse to next drilling depth with G1 and the programmed feedrate (sequence of motions is continued until the final drilling depth is reached)
- Retraction to retraction plane with G0

**Explanation of parameters**

See Subsection 2.1.2 (Drilling, Centering – CYCLE81) for parameters RTP, RFP, SDIS, DP, and DPR.

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

The first drilling depth is programmed by either one of these two parameters. The parameter FDP has the same effect in the cycle as parameter DPR. If the values for the reference and retraction plane 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 (degression). This allows for removal of the swarf 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 equally and traversed and are, therefore, always greater than half of the amount of degression.
- If the value for the first drilling depth is opposed to the total depth, error message 61107 "First drilling depth incorrectly defined" is generated and the cycle is not executed.

**Example:**

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

-12	is the first drilling depth
-21	the incremental difference 9 results from the first drilling depth minus the amount of degression 3
-27	previous drilling depth minus the amount of degression 3
-30, -33, -36	amount of degression
-38, -40	remaining depth divided into two cuts

## 2.1 Drilling cycles

### 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, for as long as the stroke remains larger or equal to 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 opposed to the total depth, error message 61107 "First drilling depth incorrectly defined" is generated and the cycle is not executed.
- If a value of DAM<-1 is programmed, the cycle aborts and error message 61019 "DAM parameter incorrectly defined" is generated.

Example:

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

-10                is the first drilling depth  
 -18                the incremental difference 8 is 0.8 \* first drilling depth  
 -24.4, -29.52    the previous drilling depth \* degression factor in each case  
 -34.52            minimum drilling depth MDEP takes 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.

- 0 in seconds  
    < 0 in revolutions

### DTS (dwell time)

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

- Value > 0 in seconds  
    Value < 0 in revolutions

### FRF (feedrate factor)

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

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. When VARI=1 (for swarf removal), the drill traverses in each case to the reference plane brought forward by the safety distance.

### **\_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.

Key:

\_AXN=1 1st axis of the current plane

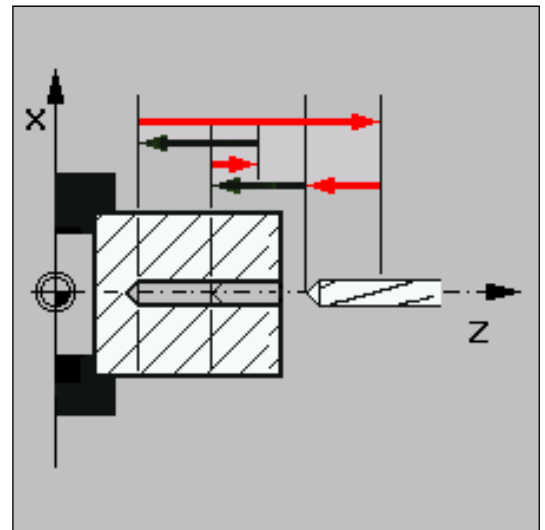
\_AXN=2 2nd axis of the current plane

\_AXN=3 3rd axis of the current plane.

For example, to machine a center drilling (in Z) in plane G18, 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 breaking with VARI=0)**

You can program the retraction path for chip breaking.

Value > 0 retraction value

Value = 0 retraction value 1 mm

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

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

Value > 0 in seconds

Value < 0 in revolutions

Value = 0 dwell time as programmed in DTB

**\_DIS1 (programmable limit distance when VARI=1)**

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

Value > 0 position at programmed value

Value = 0 automatic calculation

The limit distance is calculated within the cycle as follows:

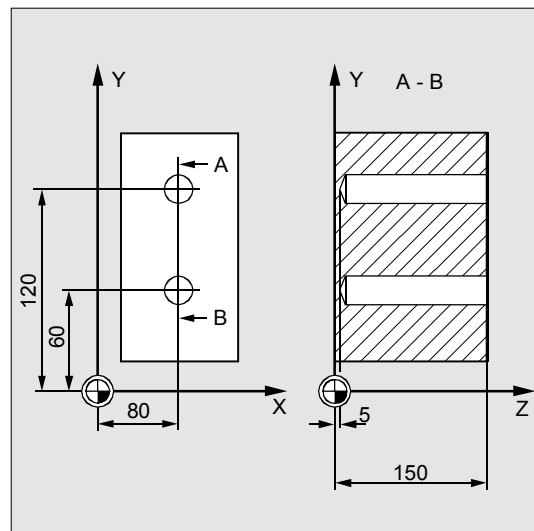
- Up to a drilling depth of 30 mm, the value is set to 0.6 mm.
- For larger drilling depths, the limit distance is calculated as  $(RFP + SDIS - \text{current depth}) / 50$ . If this calculated value >7, the limit distance is limited to a maximum of 7 mm.

**Programming example****Deep-hole drilling**

This program executes the cycle CYCLE83 at positions X80 Y120 and X80 Y60 in the XY plane. The first hole is drilled with a dwell time of zero and a machining type of 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 sec is programmed. Machining type swarf removal is selected, the final drilling depth is relative to the reference plane. The drilling axis in both cases is the Z axis.

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, DP=5, DPR=145, FDEP=100, FDPR=50, DAM=20, DTB=1, FRF=1, VARI=0, _VRT=0.8, _MDEP=10, _DIS1=0.4	;Definition of parameters
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
o40 CYCLE83 (RTP, RFP, SDIS, DP, ,-> -> FDEP, , DAM, , , FRF, VARI, , , _VRT)	;Cycle call, depth parameter with ;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, , -> -> FDPR, DAM, DTB, , FRF, VARI, , _MDEP, -> , , _DIS1)	;Cycle call with relative data for final ;drilling depth and 1st drilling depth; the ;safety distance is 1 mm and the feedrate ;is 0.5

---

```
N70 M30 ;End of program
```

---

-> Must be programmed in a single block

### 2.1.5 Rigid tapping – CYCLE84



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)
DTB	real	Dwell time at thread depth (chip breaking)
SDAC	int	Direction of rotation after end of cycle Values: 3, 4 or 5
MPIT	real	Pitch as thread size (with sign) Value range: 3 (for M3) ... 48 (for M48), the sign determines the direction of rotation in the thread
PIT	real	Pitch as value (with sign) Value range: 0.001 ... 2000.000 mm), the sign determines the direction of rotation in the thread: If PTAB=0 or 1: in mm (as before) If _PTAB=2 in thread starts per inch
POSS	real	Spindle position for oriented spindle stop in the cycle (in degrees)
SST	real	Speed for tapping
SST1	real	Speed for retraction
_AXN	int	Tool axis Values: 1 = 1st geometry axis 2 = 2nd geometry axis or else 3rd geometry axis
_PTAB	int	Evaluation of thread pitch PIT Values: 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

<code>_TECHNO</code>	int	<p>Technological settings</p> <p>UNITS DIGIT: Exact stop behavior Values: 0...As programmed before cycle call</p> <p>1...(G601) 2...(G602) 3...(G603)</p> <p>TENS DIGIT: Feedforward control Values: 0...As programmed before cycle call</p> <p>1...With feed-forward control (FFWON) 2...Without feed-forward control (FFWOF)</p> <p>HUNDREDS DIGIT: Acceleration Values: 0...As programmed before cycle call</p> <p>1...Soft acceleration of axes (SOFT) 2...Brisk acceleration of axes (BRISK) 3...Reduced acceleration of axes (DRIVE)</p> <p>THOUSANDS DIGIT: Values: 0...Activate spindle mode again (with MCALL) 1...Stay in position-control mode (with MCALL)</p>
<code>_VARI</code>	int	<p>Type of machining: Values: 0...Tapping in one pass</p> <p>1...Deep-hole drilling with chip breaking 2...Deep-hole drilling with swarf removal</p>
<code>_DAM</code>	real	<p>Incremental drilling depth Value range: 0 &lt;= Max. value</p>
<code>_VRT</code>	real	<p>Variable retraction distance for chip breakage Value range: 0 &lt;= Max. value</p>



## Function

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

With cycle `CYCLE84`, you can perform rigid tapping operations.

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



*Cycle `CYCLE84` can be used if the spindle to be used for the boring operation is technically able to go into position-controlled spindle operation.*



There is a separate cycle `CYCLE840` for tapping with compensating chuck (see Subsection 2.1.6).





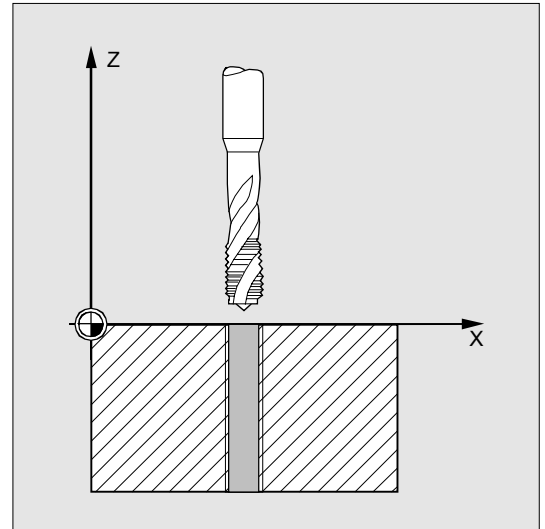
## Sequence

### Position reached before the beginning of the cycle:

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

### The cycle implements the following motion sequence:

- Approach of the reference plane brought forward by the safety distance with G0.
- Oriented spindle stop with SPOS (value in parameter POSS) and conversion of spindle to axis mode.
- Tapping to final drilling depth with G331 and speed SST.
- Execute dwell time at thread depth (parameter DTB).
- Retraction to the reference plane brought forward by the safety distance with G332, spindle speed SST1 and reversal of direction of rotation.
- Retraction to the retraction plane with G0, spindle mode is reintroduced by reprogramming the spindle speed active before the cycle was called and the direction of rotation programmed under SDAC.



## Explanation of parameters

See Subsection 2.1.2 (Drilling, Centering – CYCLE81) for parameters RTP, FRP, SDIS, and DPR.

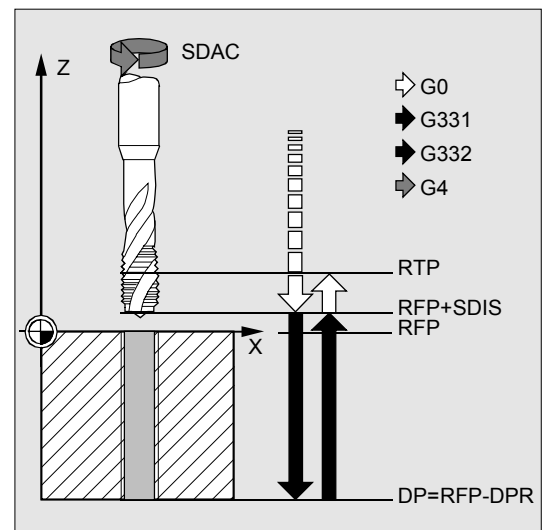
### DTB (dwell time)

You program the dwell time in seconds. It is recommended that the dwell time is omitted for the tapping of blind holes.

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

With SDAC, you program the direction of rotation after completion of the cycle.

For tapping, the direction is changed automatically by the cycle.



## 2.1 Drilling cycles

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

The value for the thread pitch can either be defined as the thread size (for metric threads between M3 and M48 only) or as a value (distance from one thread start to the next as a numerical value). The parameter not required in each case is omitted from the call or assigned the value zero.

Right or left threads are specified by the sign of the pitch parameter:

- Positive value → right (like M3)
- Negative value → left (like M4)

If the two pitch parameters have contradictory values, the cycle generates alarm 61001 "Incorrect pitch" and cycle execution is aborted.

### POSS (spindle position)

Before tapping starts in the cycle, oriented spindle stop is performed with command SPOS and the spindle is put in position control.

You program the spindle position for this spindle stop under POSS.

### SST (speed)

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

### SST1 (retraction speed)

With SST1, you program the speed for the retraction out of the thread hole in the block with G332.

If this parameter is assigned the value zero, the retraction movement is performed with the speed programmed under SST.

### \_AXN (tool axis)

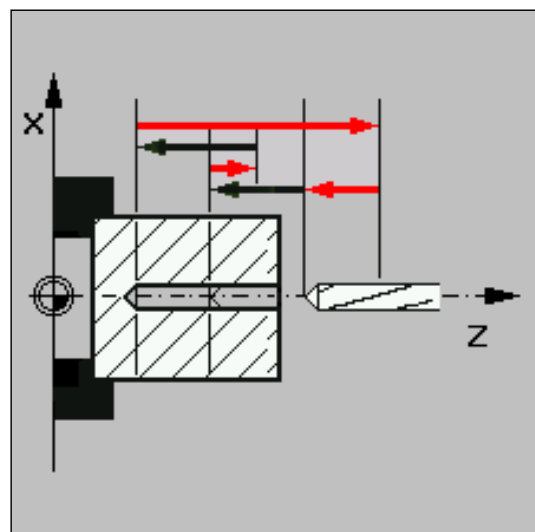
Programming the drilling axis via *AXN* enables you to omit the switchover from plane G18 to G17 when the deep-hole tapping cycle is used on turning machines.

Key:

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 center drilling (in Z) in plane G18, you program:

```
G18
_AXN=1
```



**\_PTAB (evaluation of 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 values are:

**Units digit (exact stop behavior):**

- 0 = As programmed before cycle call
- 1 = (G601)
- 2 = (G602)
- 3 = (G603)

**Tens digit (feedforward 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 = Soft acceleration of axes (SOFT)
- 2 = Brisk acceleration of axes (BRISK)
- 3 = Reduced acceleration of axes (DRIVE)

**Thousands digit:**

- 0 = Activate spindle mode again (with MCALL)
- 1 = Stay in position-control mode (with MCALL)



With input field "Technology" "Yes", both the machine manufacturer and the operator/programmer can customize the technology for tapping.

**Adaptation by the machine manufacturer**

(SW 6.4 and higher)

- Prerequisites: The manufacturer password is set, input field "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.
- This enables the machine manufacturer to customize the default settings to the characteristics of his machine.

## 2.1 Drilling cycles

### Adaptation by the operator/programmer

(SW 6.4 and higher)

- Prerequisites: The manufacturer password is deleted, input field "Technology" = "Yes"
- The modified data is 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`

Parameter `_VARI` can be set to select between simple tapping (`_VARI = 0`) and deep-hole tapping (`_VARI ≠ 0`).

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 swarf 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 depth is drilled in each pass until the remaining distance to the final drilling depth is  $< 2 * \_DAM$ .
- The remaining depth is halved and drilled in 2 passes, thereby ensuring that the minimum drilling depth is not less than  $\_DAM/2$ .

### Other information

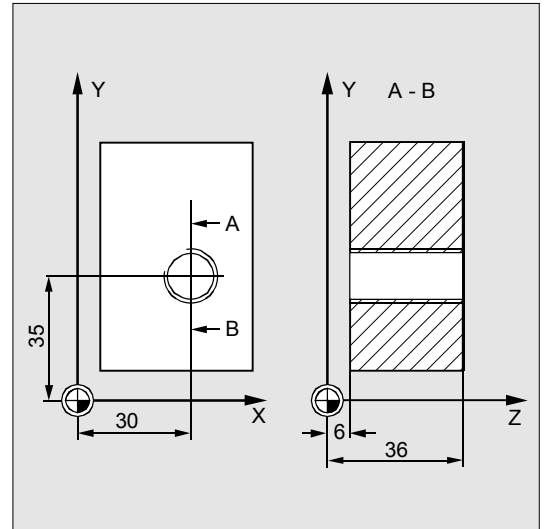
The direction of rotation is always reversed automatically for tapping in cycle.



### Programming example

#### Rigid tapping

A thread is tapped without a 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. Values must be assigned to the parameters for the direction of rotation and the pitch. 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, -> ->, 90, 200, 500)	;Cycle call, parameter PIT has been ;omitted, no value is entered for the ;absolute depth or the dwell time. ;Spindle stop at 90 degrees, speed for ;tapping is 200, speed for retraction is ;500
N40 M30	End of program
-> Must be programmed in a single block	

## 2.1.6 Tapping with compensating chuck – CYCLE840

**Programming**

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

**Parameters**

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)
DTB	real	Dwell time at thread depth: Always operative if >0 is programmed Value range: 0<=DTB
SDR	int	Direction of rotation for retraction Values: 0 (automatic reversal of direction of rotation) 3 or 4 (for M3 or M4)
SDAC	int	Direction of rotation after end of cycle Values: 3, 4 or 5 (for M3, M4 or M5)
ENC	int	Tapping with/without encoder Values: 0 = With encoder, without dwell time 1 = Without encoder, programs feedrate before cycle 11 = Without encoder, calculates the feed in the cycle 20 = With encoder, with dwell time
MPIT	real	Thread pitch as thread size Value range: 3 (for M3) ... 48 (for M48)
PIT	real	Thread pitch as value Value range: 0.001 ... 2000.000 mm If _PTAB = 0 or 1: in mm If _PTAB = 2 in thread starts per inch
_AXN	int	Tool axis: Values: 1 = 1st geometry axis 2 = 2nd geometry axis or else 3rd geometry axis
_PTAB	int	Evaluation of thread pitch PIT Values: 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

<code>_TECHNO</code>	int	Technological settings
		UNITS DIGIT: Exact stop behavior
		Values: 0...As programmed before cycle call
		1...(G601)
		2...(G602)
		3...(G603)
		TENS DIGIT: Feedforward control
		Values: 0...As programmed before cycle call
		1...With feed-forward control (FFWON)
		2...Without feed-forward control (FFWOF)
		HUNDREDS DIGIT: Brake application point
		Values: 0...Without calculation
		1...With calculation



### Function

The tool drills at the programmed spindle speed and feedrate to the programmed thread depth. This cycle can be used to perform tapping with compensating chuck

- Without encoder and
- With encoder.



### Sequence

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

#### Position reached before the beginning of the cycle:

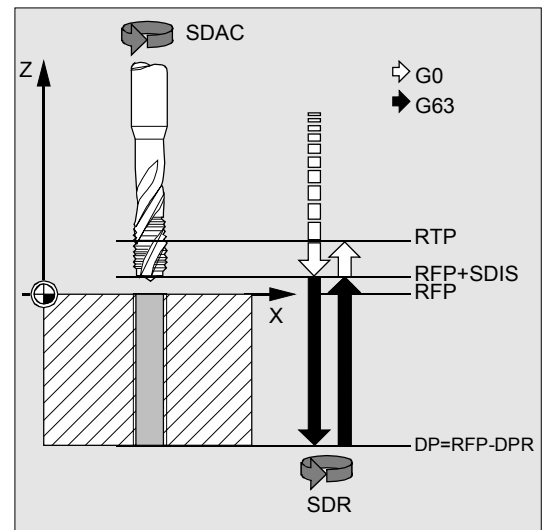
The drilling position is the position in the two axes of the selected plane.

#### The cycle implements the following motion sequence:

- Approach of the reference plane brought forward by the safety distance with G0.
- Tapping to the final drilling depth with G63.
- Retraction to the reference plane brought forward by the safety distance with G63.
- Retraction to retraction plane with G0.



The spindle override must be set to 100%.

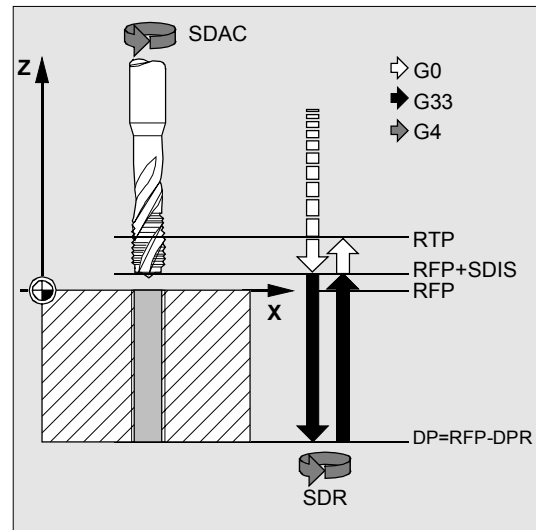


**Tapping with compensating chuck with encoder (ENC=0)****Position reached before the beginning of the cycle:**

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

**The cycle implements the following motion sequence:**

- Approach of the reference plane brought forward by the safety distance with G0.
- Tapping to the final drilling depth with G33.
- Execute dwell time at thread depth (parameter DTB); only if ENC = 20.
- Retraction to the reference plane brought forward by the safety distance with G33.
- Retraction to retraction plane with G0.

**Explanation of parameters**

See Subsection 2.1.2 (Drilling, Centering – CYCLE81) for parameters RTP, RFP, SDIS, DP, and DPR.

**DTB (dwell time)**

You program the dwell time 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. For tapping with an encoder, the direction of spindle travel can be reversed automatically, by setting SDR = 0.

**SDAC (direction of rotation)**

As the cycle can also be called modally (see Section 2.2), it requires a direction of rotation in order to tap further threads. 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.



### ENC (tapping)

To perform tapping without encoder although an encoder exists, parameter ENC must be set to x1. However, if no encoder exists and the parameter is assigned the value x0, it is ignored in the cycle.

- **Tapping without encoder with input of the 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 only has a meaning 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 either be defined as the thread size (for metric threads between M3 and M48 only) or as a value (distance from one thread start to the next as a numerical value). The parameter not required in each case is omitted from the call or assigned the value zero.

If the two thread pitch parameters have conflicting values, alarm 61001 "Thread pitch incorrect" is generated by the cycle and cycle execution is aborted.

## 2.1 Drilling cycles

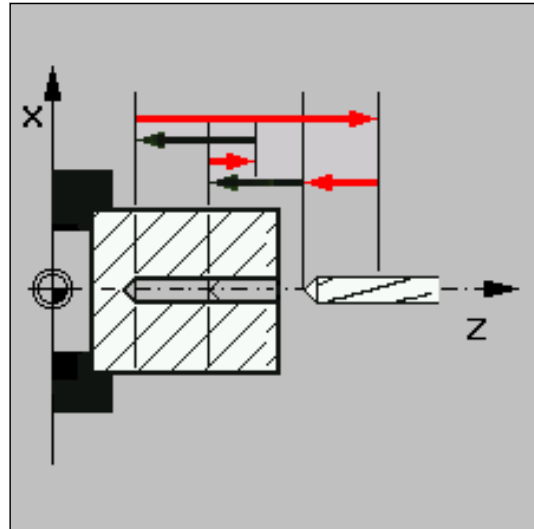
### **\_AXN (tool axis)**

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. Key:

- `_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 center drilling (in Z) in plane G18, you program:

```
G18
_AXN=1
```



### **\_PTAB (evaluation of pitch PIT)**

Parameter `_PTAB` determines the unit of measurement of the 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 required 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 values are:

#### **Units digit (exact stop behavior):**

- 0 = As programmed before cycle call
- 1 = (G601)
- 2 = (G602)
- 3 = (G603)

#### **Tens digit (feedforward control):**

- 0 = As programmed before cycle call
- 1 = With feed-forward control (FFWON)
- 2 = Without feed-forward control (FFWOF)

#### **Hundreds digit (brake application point):**

- 0 = Without calculation
- 1 = With calculation



As of SW 6.4:

With input field "Technology" "Yes", both the machine manufacturer and the operator/programmer can customize the technology for tapping.

#### **Adaptation by the machine manufacturer** (SW 6.4 and higher)

- Prerequisites: The manufacturer password is set, input field "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.

#### **Adaptation by the operator/programmer** (SW 6.4 and higher)

- Prerequisites: The manufacturer password is deleted, input field "Technology" = "Yes"
- The modified data is 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.



#### **Other information**

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 before the cycle is called.

In thread blocks with G63, the values of the feedrate override switch and spindle speed override switch are frozen at 100%.

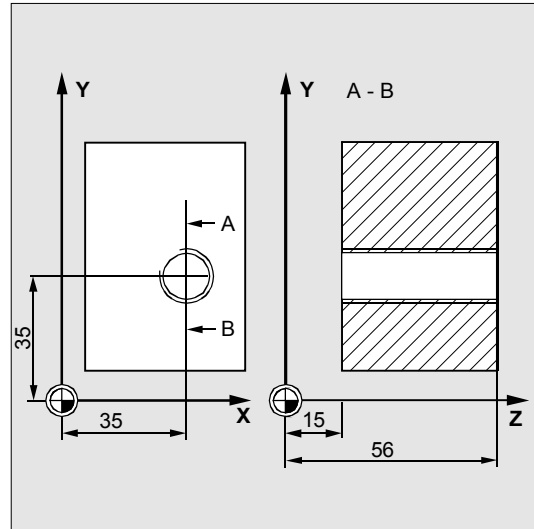
A longer compensating chuck is usually required for tapping without encoder.



### Programming example

#### Thread 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. Direction of rotation parameters SDR and SDAC must be assigned, parameter ENC is assigned the value 1, the value for the depth is absolute. Pitch parameter PIT can be omitted. A compensating chuck is used in machining.

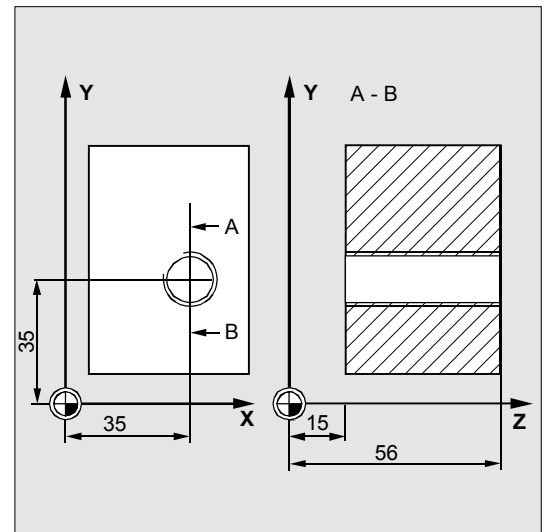


N10 G90 G0 D2 T2 S500 M3	;Specification of technology values
N20 G17 X35 Y35 Z60	;Approach drilling position
N30 G1 F200	;Setting the path feedrate
N40 CYCLE840 (59, 56, , 15, , 1, 4, 3, 1)	;Cycle call, dwell time 1 sec, SDR=4, ;SDAC=3, no safety distance, ;parameters MPIT, PIT are not ;programmed, i.e.,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



### Thread with encoder

In this program a thread is tapped with encoder at position X35, Y35 in the XY plane; the tapping 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, , , , , , -> ->, PIT)	;Cycle call, without safety distance, value ;for depth programmed as an absolute ;value, SDAC, ENC, MPIT are omitted ;(i.e., are assigned the value zero)
N40 M30	;End of program

-> Must be programmed in a single block.

## 2.1.7 Boring 1 – CYCLE85

**Programming**

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

**Parameters**

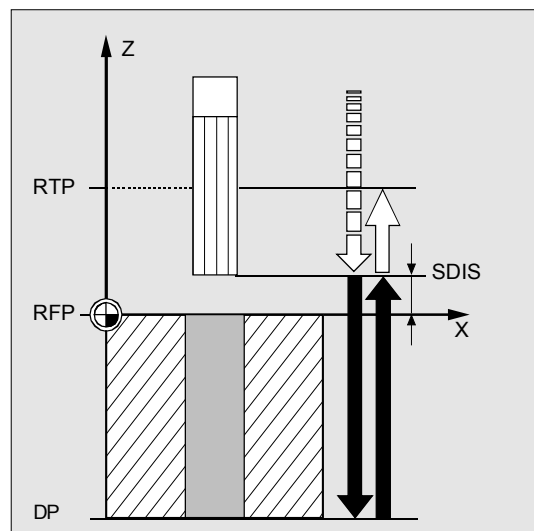
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)
DTB	real	Dwell time at final drilling depth (chip breaking)
FFR	real	Feed
RFF	real	Retraction feedrate

**Function**

The tool drills at the programmed spindle speed and feedrate to the programmed final drilling depth. The inward and outward movement is performed at the feedrate that is assigned to parameters FFR and RFF respectively. This cycle can be used to ream drill holes.

**Sequence****Position reached before the beginning of the cycle:**

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



### The cycle implements the following motion sequence:

- Approach of the reference plane brought forward by the safety distance with G0.
- Traverse to final drilling depth with G1 and at the feedrate programmed under parameter FFR.
- Execute dwell time at final drilling depth.
- Retraction to the reference plane brought forward by the safety distance with G1 and the retraction feedrate defined under parameter RFF.
- Retraction to retraction plane with G0.



### Explanation of parameters



See Subsection 2.1.2 (Drilling, Centering – CYCLE81) for parameters RTP, RFP, SDIS, DP, and DPR.

#### DTB (dwell time)

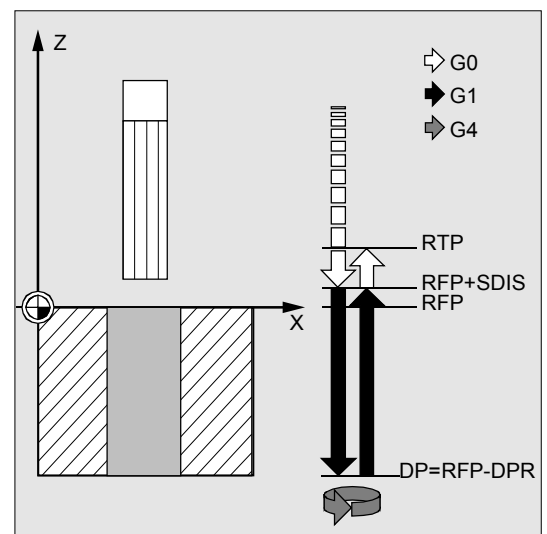
Parameter DTB is the dwell time at the final drilling depth (chip breaking) in seconds.

#### FFR (feedrate)

The feedrate value assigned to FFR is active for boring.

#### RFF (retraction feedrate)

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

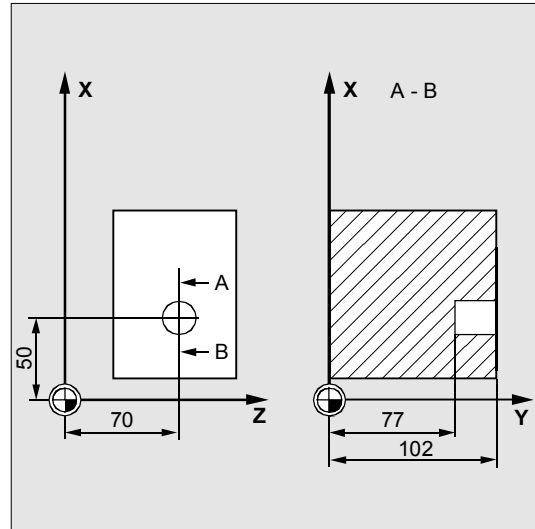




## Programming example

### First boring pass

Cycle 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 top edge of the workpiece is positioned at Y102.



DEF REAL FFR, RFF, RFP=102, DPR=25, SDIS=2	;Definition of parameters and ;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, ,-> -> FFR, RFF)	;Cycle call, no dwell time programmed
N40 M30	;End of program

-> Must be programmed in a single block.



## 2.1.8 Boring 2 – CYCLE86



### Programming

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



### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)
DTB	real	Dwell time at final drilling depth (chip breaking)
SDIR	int	Direction of rotation Value: 3 (for M3) 4 (for M4)
RPA	real	Retraction path in abscissa of the active plane (incremental, enter with sign)
RPO	real	Retraction path in ordinate of the active plane (incremental, enter with sign)
RPAP	real	Retraction path in applicate of the active plane (incremental, enter with sign)
POSS	real	Spindle position for oriented spindle stop in the cycle (in degrees)



### Function

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



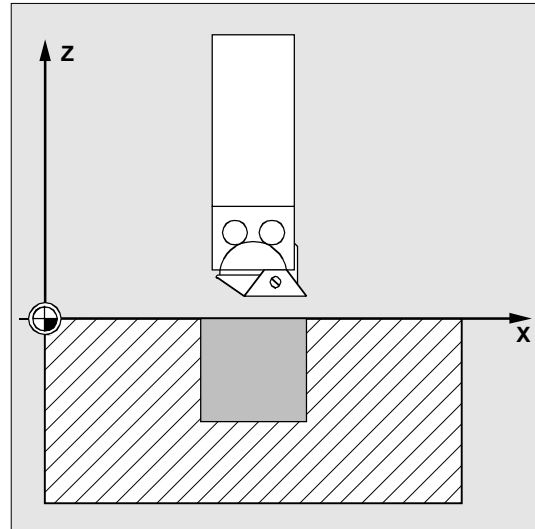
## Sequence

### Position reached before the beginning of the cycle:

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

### The cycle implements the following motion sequence:

- Approach of the reference plane brought forward by the safety distance with G0.
- Traverse to final drilling depth with G1 and the feedrate programmed before the program call.
- Dwell time at final drilling depth is executed.
- Oriented spindle stop at the spindle position programmed under POSS.
- Trav. retraction path in up to three axes with G0.
- Retraction to the reference plane brought forward by the safety distance with G0.
- Retraction to the retraction plane with G0 (initial drilling position in both axes on the plane).



## Explanation of parameters



See Subsection 2.1.2 (Drilling, Centering – CYCLE81) for parameters RTP, RFP, SDIS, DP, and DPR.

### DTB (dwell time)

Parameter DTB is the dwell time at the final drilling depth (chip breaking) in seconds.

### SDIR (direction of rotation)

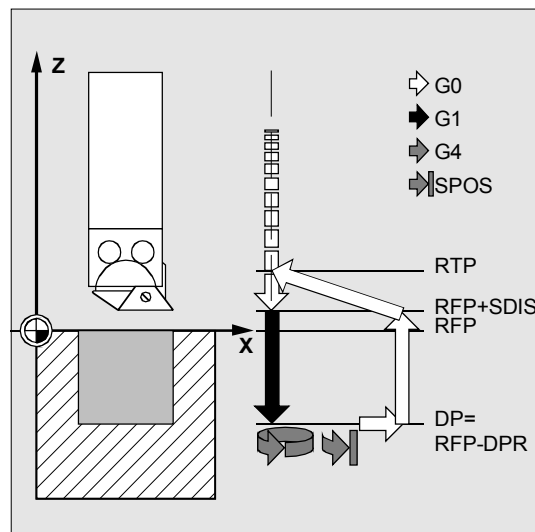
With this parameter, you determine the direction of rotation, with which boring is performed in the cycle. If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is output and the cycle is not executed.

### 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.

### RPO (retraction path, in ordinate)

Under this parameter, you define a retr. 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 applicate)**

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

**POSS (spindle position)**

Under POSS, the spindle position for the oriented spindle stop, which is performed after the final drilling depth has been reached, is programmed in degrees.

**Other information**

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



*Cycle CYCLE86 can be used if the spindle to be used for the boring operation is technically able to go into position-controlled spindle operation.*

**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.

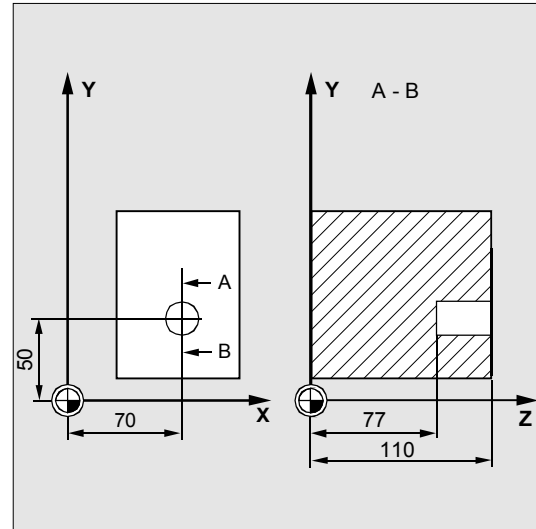
If you call the cycle without Y axis in the G18 plane, the following alarm appears: 61005 "3rd geometry axis not available", as the Y axis would then be the boring axis.



### Programming example

#### Second boring pass

Cycle CYCLE86 is called at position X70 Y50 in the XY plane; the boring axis is the Z axis. The final drilling depth is programmed as an absolute value, a safety distance is not defined. 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 turned with M3 and stops 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, -> -> -1, -1, +1, POSS)	;Cycle call with absolute drilling depth
N60 M30	;End of program

-> Must be programmed in a single block.

### 2.1.9 Boring 3 – CYCLE87



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)
SDIR	int	Direction of rotation Value: 3 (for M3) 4 (for M4)



#### Function

The tool drills at the programmed spindle speed and feedrate to the programmed final drilling depth. With Boring 3, a spindle stop without orientation M5 and then a programmed stop M0 are generated when the final drilling depth is reached. The NC START key is pressed to continue the retraction movement in rapid traverse mode until the retraction plane is reached.



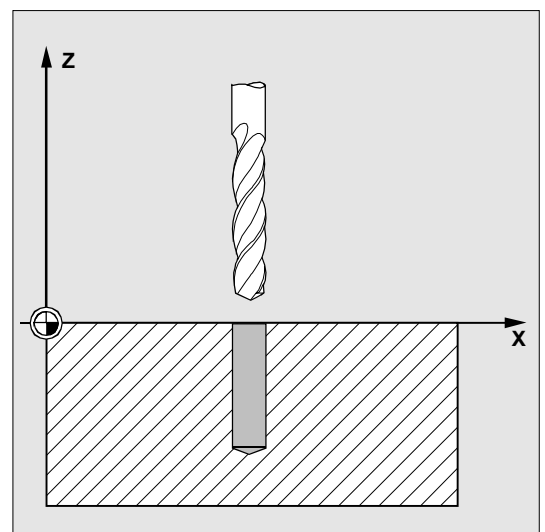
#### Sequence

##### Position reached before the beginning of the cycle:

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

##### The cycle implements the following motion sequence:

- Approach of the reference plane brought forward by the safety distance with G0.
- Traverse to final drilling depth with G1 and the feedrate programmed before the program call.
- Spindle stop with M5.
- Press NC START key.
- Retraction to retraction plane with G0.





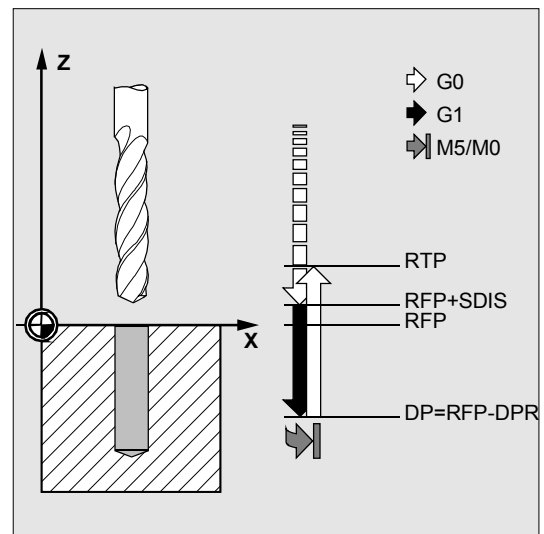
### Explanation of parameters



See Subsection 2.1.2 (Drilling, Centering – CYCLE81) for parameters RTP, RFP, SDIS, DP, and DPR.

#### SDIR (direction of rotation)

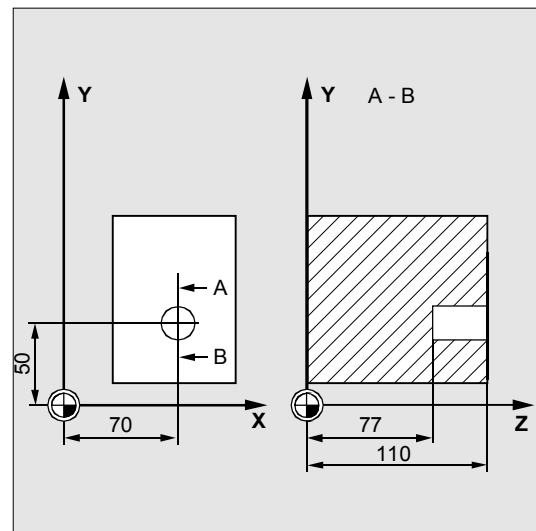
With this parameter, you determine the direction of rotation, with which boring is performed in the cycle. If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is output and the cycle is aborted.



### Programming example

#### Third boring pass

Cycle CYCLE87 is called at position X70 Y50 in the XY plane; the boring axis is the Z axis. The final drilling depth is programmed as an absolute value. The safety distance 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 spindle ;direction of rotation M3
N60 M30	;End of program

### 2.1.10 Boring 4 – CYCLE88



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)		
RFP	real	Reference plane (absolute)		
SDIS	real	Safety distance (enter without sign)		
DP	real	Final drilling depth (absolute)		
DPR	real	Final drilling depth relative to reference plane (enter without sign)		
DTB	real	Dwell time at final drilling depth		
SDIR	int	Direction of rotation	Values: 3 (for M3)	4 (for M4)



#### Function

The tool drills at the programmed spindle speed and feedrate to the programmed final drilling depth. With Boring 4, a dwell time, a spindle stop without orientation M5 and a programmed stop M0 are generated when the final drilling depth is reached. Pressing the NC START key continues the retraction movement in rapid traverse mode until the retraction plane is reached.



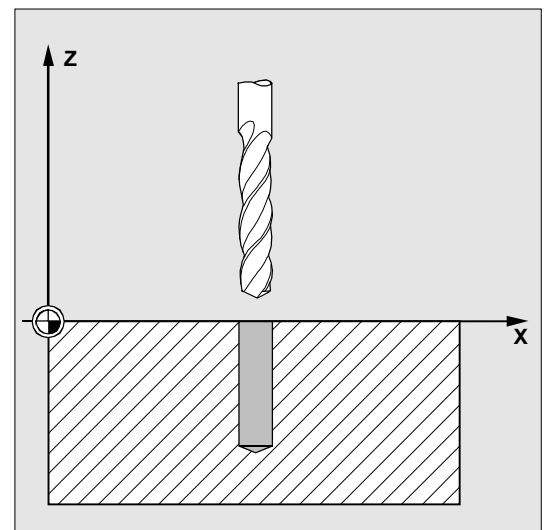
#### Sequence

##### Position reached before the beginning of the cycle:

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

##### The cycle implements the following motion sequence:

- Approach of the reference plane brought forward by the safety distance with G0.
- Traverse to final drilling depth with G1 and the feedrate programmed before the program call.
- Dwell time at final drilling depth.
- Spindle stop with M5 ( $\_ZSD[5]=1$ ) or
- Spindle and program stop with M5 M0 ( $\_ZSD[5]=0$ ). Press the NC START key after program stop.
- Retraction to retraction plane with G0.





### Explanation of parameters



See Subsection 2.1.2 (Drilling, Centering – CYCLE81) for parameters RTP, RFP, SDIS, DP, and DPR.

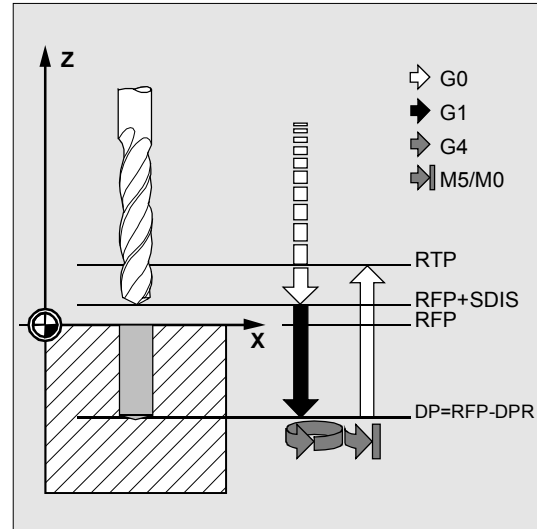
See Section 3.2 for cycle setting data `_ZSD[5]`.

#### DTB (dwell time)

Parameter DTB is the dwell time at the final drilling depth (chip breaking) in seconds.

#### SDIR (direction of rotation)

The programmed direction of rotation is active for the travel path to the final drilling depth. If values other than 3 or 4 (M3/M4) are programmed, alarm 61102 "No spindle direction programmed" is output and the cycle is aborted.

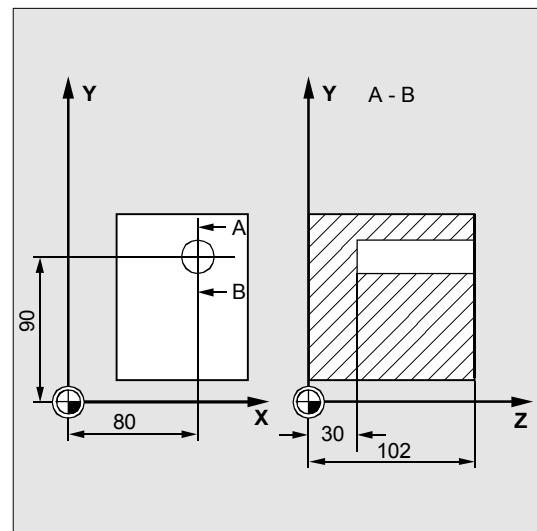


### Programming example

#### Fourth boring pass

Cycle CYCLE88 is called at position X80 Y90 in the XY plane; the boring axis is the Z axis. The safety distance is programmed as 3 mm. The final drilling depth is defined as a value relative to the reference plane.

M4 is active in the cycle.



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

-> Must be programmed in a single block.



### 2.1.11 Boring 5 – CYCLE89



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)
DTB	real	Dwell time at final drilling depth (chip breaking)



#### Function

The tool drills at the programmed spindle speed and feedrate to the programmed final drilling depth. Once the final drilling depth has been reached, a dwell time can be programmed.



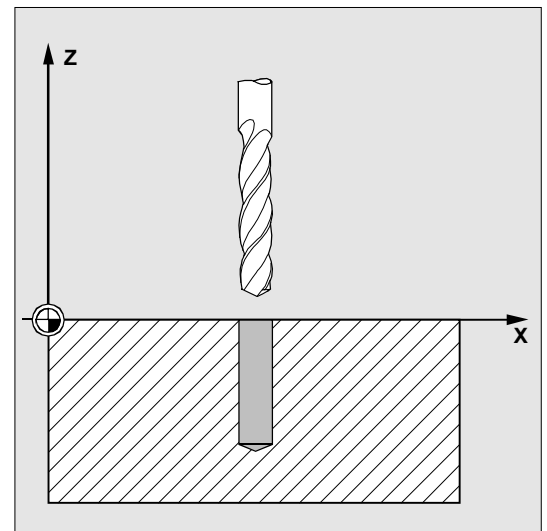
#### Sequence

##### Position reached before the beginning of the cycle:

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

##### The cycle implements the following motion sequence:

- Approach of the reference plane brought forward by the safety distance with G0.
- Traverse to final drilling depth with G1 and the feedrate programmed before the program call.
- Dwell time at final drilling depth is executed.
- Retraction to the reference plane brought forward by the safety distance with G1 and the same feedrate value.
- Retraction to retraction plane with G0.





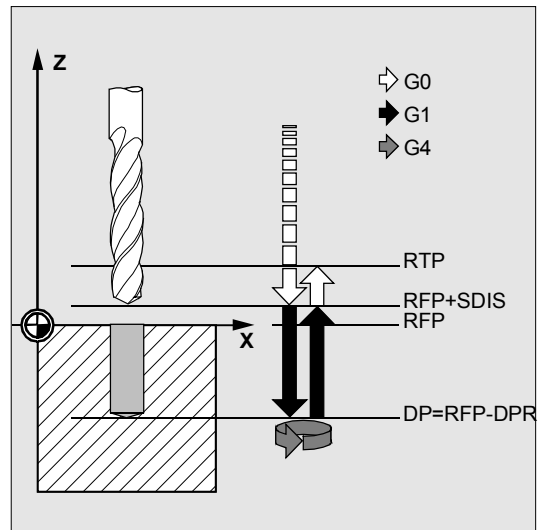
### Explanation of parameters



See Subsection 2.1.2 (Drilling, Centering – CYCLE81) for parameters RTP, RFP, SDIS, DP, and DPR.

#### DTB (dwell time)

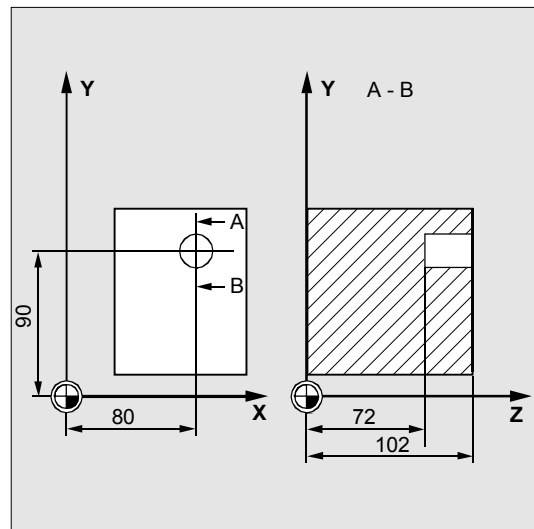
Parameter DTB is the dwell time at the final drilling depth (chip breaking) in seconds.



### Programming example

#### Fifth boring pass

Boring cycle CYCLE89 is called at position X80 Y90 in the XY plane with a safety distance of 5 mm and the final drilling depth specified as an absolute value. The boring 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	;End of program

## 2.2 Modal call of drilling cycles

With NC programming it is possible to call any subroutine modally.

This feature is of particular importance for drilling cycles.



### Programming

Modal call of a subroutine

MCALL

With drilling cycle (for example)

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



### Function

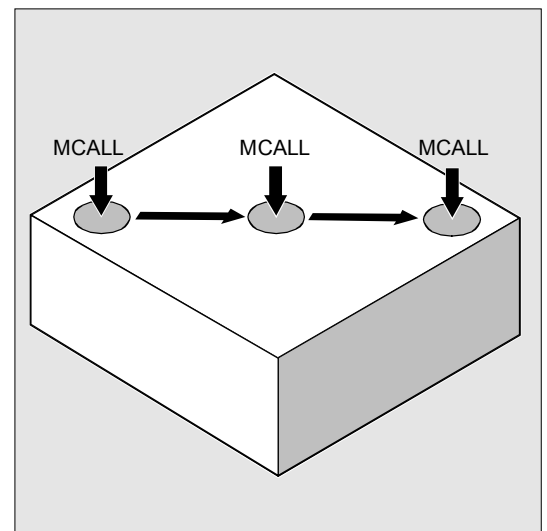
In NC programming, subroutines and cycles can be called modally.

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 path motion.

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

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.



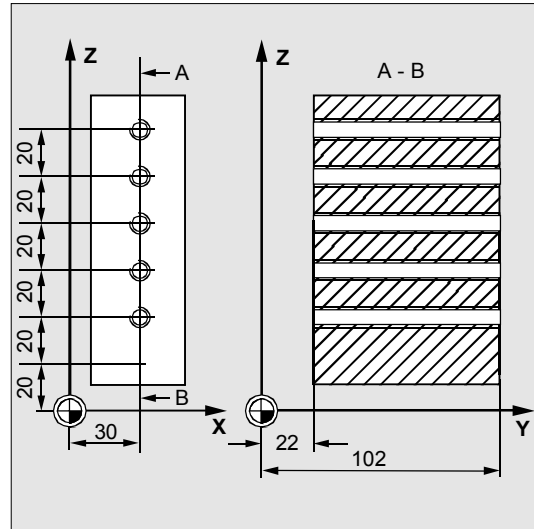
## 2.2 Modal call of drilling cycles



### Programming example

#### Row of holes\_5

With this program, you can machine a row of 5 thread holes positioned parallel to the Z axis in 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.



DEF REAL RFP=102, DP=22, RTP=105, ->	;Definition of parameters with
-> 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, SDIS, DP)	;Modal call of the drilling cycle
N50 MA1: G91 Z20	;Approach next position (ZX plane), ;cycle is executed
N60 COUNT=COUNT+1	;Loop for drilling positions along the row
N70 IF COUNT<6 GOTOB MA1	;of holes
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, ->	;Modal call of tapping cycle
-> DP , , , 3, , PIT, , 400)	
N130 MA2: G91 Z20	;Next drilling position
N140 COUNT=COUNT+1	;Loop for drilling position of the row of
N150 IF COUNT<6 GOTOB MA2	;holes
N160 MCALL	;Deselect modal call
N170 G90 X30 Y105 Z20	;Approach starting position again
N180 M30	;End of program

-> Must be programmed in a single block.



### Other information

#### 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.



The program given in the description of the drilling pattern cycles has been adapted and greatly simplified using these cycles.

The drilling pattern cycles described in the following Section 2.3 are based on the call principle.

MCALL DRILLING CYCLE (...)

DRILLING PATTERN (...).

## 2.3 Drill pattern cycles

The drill pattern cycles only describe the geometry of an arrangement of holes on a plane. The link to a drilling cycle is established via the modal call (see Section 2.2) of this drilling cycle before the drill pattern cycle is programmed.

### 2.3.1 Prerequisites

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

Drill pattern cycles can also be used for other applications without the drilling cycle first being called modally, as the drill 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 drill pattern cycle, error message 62100 "No drilling cycle active" appears.

You can acknowledge this error message with the error acknowledgment key and continue program processing by pressing the NC Start key. The drill pattern cycle then approaches each of the positions calculated from the input data one after the other, without calling a subroutine at these points.

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

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

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

Generally there are no plausibility checks for defining parameters in the drill pattern cycles, if they are not expressly declared for a parameter with a description of the corresponding response.

### 2.3.2 Row of holes – HOLES1



#### Programming

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



#### Parameters

SPCA	real	Abscissa of a reference point on the straight line (absolute)
SPCO	real	Ordinate of this reference point (absolute)
STA1	real	Angle to abscissa Value range: $-180 < STA1 \leq 180$ degrees
FDIS	real	Distance between the first hole and the reference point (enter without sign)
DBH	real	Distance between the holes (enter without sign)
NUM	int	Number of drill holes



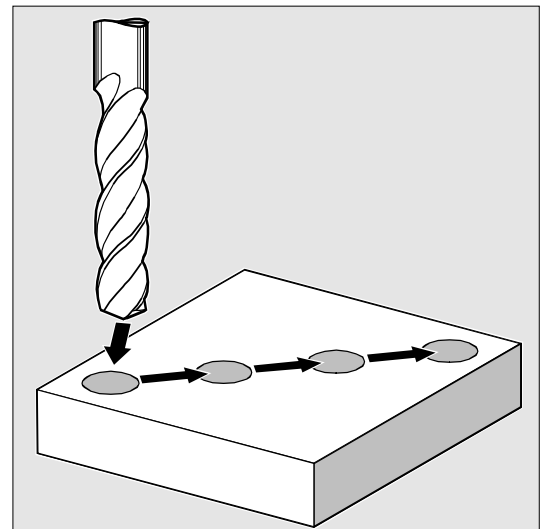
#### Function

With this cycle, you can program a row of holes, i.e., a number of holes that lie along a straight line or a grid of holes. The type of hole is determined by the drilling cycle that has already been called modally.



#### Sequence

To avoid unnecessary travel, the cycle uses the actual position of the plane axes and the geometry of the row of holes to calculate whether the row of holes is machined starting from the first hole or the last hole. The drilling positions are then approached one after the other in rapid traverse.





### Explanation of 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 distances between the holes. The distance to the first hole FDIS is defined from this point.

#### STA1 (angle)

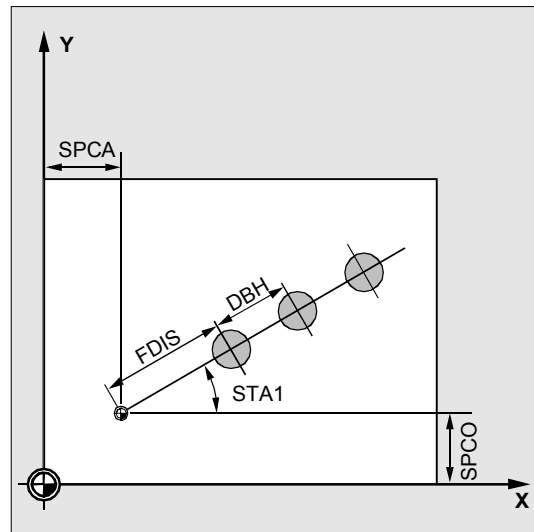
The straight line can be in any position on the plane. It is specified both by the point defined by SPCA and SPCO and by the angle contained by the straight line and the abscissa of the workpiece coordinate system that is active when the cycle is called. The angle is entered under STA1 in degrees.

#### FDIS and DBH (distance)

Under FDIS, you enter the distance between the first hole and the reference point defined under SPCA and SPCO. The parameter DBH contains the distance between any two holes.

#### NUM (number)

You determine the number of holes with the parameter NUM.



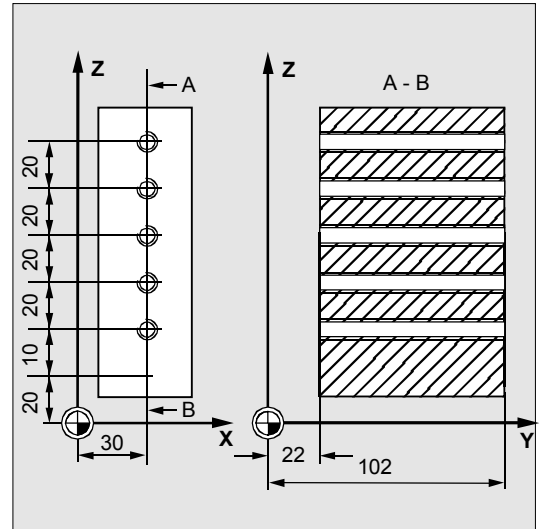




### Programming example

#### Row of holes

With this program, you can machine a row of five tapped holes positioned in parallel to the Z axis on the ZX plane, with a distance between each hole of 20 mm. The row of holes starts at Z20 and X30, the first hole in the row being 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 deep. This is the difference between the reference plane and the final drilling depth.



DEF REAL RFP=102, DP=22, RTP=105	;Definition of 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 distance and distance ;from the first hole to the reference point
N20 G90 F30 S500 M3 D1 T1	;Specification of technology values for ;the machining section
N30 G18 G0 Z20 Y105 X30	;Approach start position
N40 MCALL CYCLE81 (RTP, RFP, SDIS, DP)	;Modal call of drilling cycle
N50 HOLES1 (SPCA, SPCO, STA1, FDIS, -> -> DBH, NUM)	;Call of row of holes cycle, the cycle ;starts with the first hole. Only the drilling ;positions are approached in this 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, SDIS, DP, -> -> , 3, , 4.2, , , 400)	;Modal call of tapping cycle
N90 HOLES1 (SPCA, SPCO, STA1, FDIS, -> -> DBH, NUM)	;Call of row of holes cycle starting with ;the 5th hole in the row
N100 MCALL	;Deselect modal call
N110 M30	;End of program

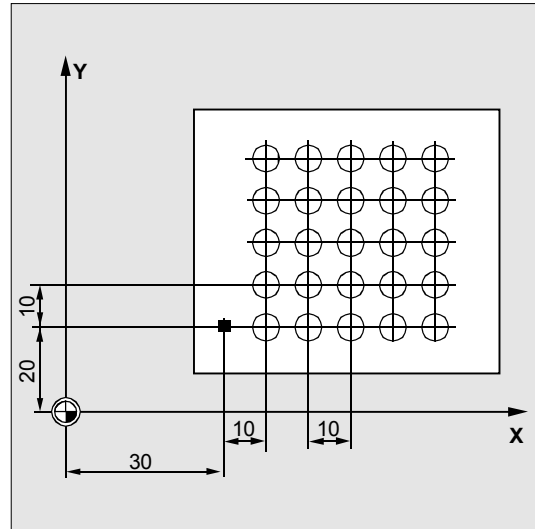
-> Must be programmed in a single block.



### Programming example

#### Grid of holes

With this program, you can machine a grid of holes consisting of five rows of five holes each that lie in the XY plane at a distance of 10 mm from one another. The starting point of the grid of holes is X30 Y20.



```

DEF REAL RFP=102, DP=75, RTP=105, SDIS=3 ;Definition of parameters with
DEF REAL SPCA=30, SPCO=20, STA1=0, -> ;value assignments
-> DBH=10, FDIS=10
DEF INT NUM=5, LINENUM=5, COUNT=0
DEF REAL LINEDIST
N10 LINEDIST=DBH ;Distance between rows = distance between
;holes
N20 G90 F300 S500 M3 D1 T1 ;Specification of technology values
N30 G17 G0 X=SPCA-10 Y=SPCO Z105 ;Approach start position
N40 MCALL CYCLE81 (RTP, RFP, SDIS, DP) ;Modal call of drilling cycle
N50 LABEL1: HOLES1 (SPCA, SPCO, STA1, -> ;Call of row of holes cycle
-> FDIS, DBH, NUM)
N60 SPCO=SPCO+LINEDIST ;Ordnate of reference point for the next line
N70 COUNT=COUNT+1 ;Jump back to LABEL1 if the condition is
N80 IF COUNT<LINENUM GOTOB LABEL1 ;fulfilled
N90 MCALL ;Deselect modal call
N100 G90 G0 X=SPCA-10 Y=SPCO Z105 ;Approach start position
N110 M30 ;End of program

```

-> Must be programmed in a single block.

### 2.3.3 Hole circle – HOLES2



#### Programming

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



#### Parameters

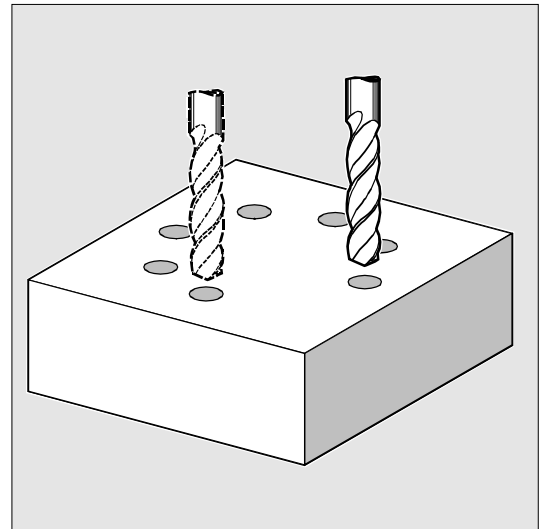
CPA	real	Center point of hole circle, abscissa (absolute)
CPO	real	Center point of hole circle, ordinate (absolute)
RAD	real	Radius of hole circle (enter without sign)
STA1	real	Initial angle Value range: $-180 < STA1 \leq 180$ degrees
INDA	real	Indexing angle
NUM	int	Number of drill holes



#### Function

A hole circle can be machined with this cycle. The machining plane must be defined before the cycle is called.

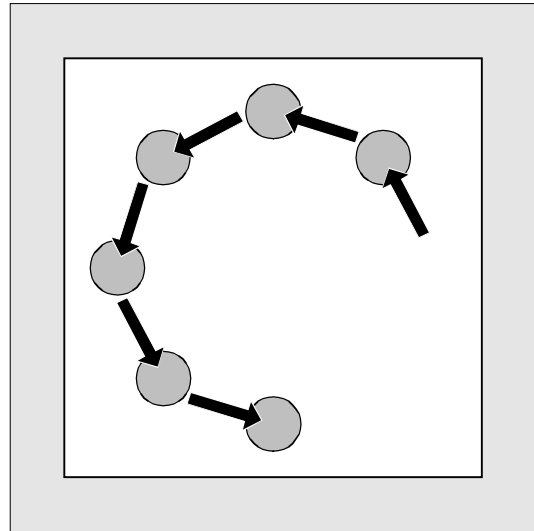
The type of hole is determined by the drilling cycle that has already been called modally.





### Sequence

In the cycle, the drilling positions on the hole circle are approached one after the other in the plane with G0.



### Explanation of parameters

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

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

#### STA1 and INDA (start angle and indexing angle)

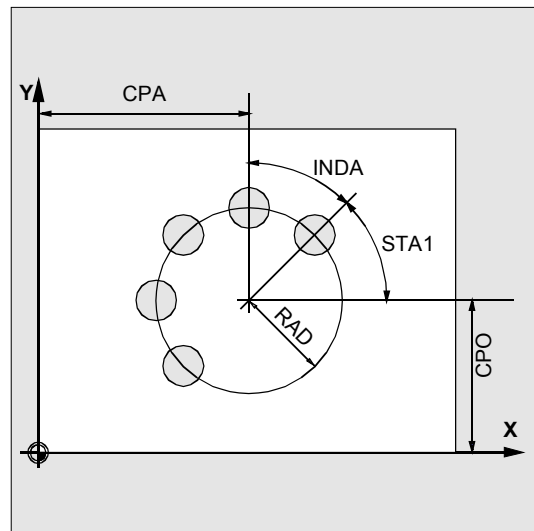
The arrangement of the holes in the circle is defined by these parameters.

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 parameter INDA is set to zero, the indexing angle is calculated internally from the number of holes so as to space them equally in a circle.

#### NUM (number)

You determine the number of holes with the parameter NUM.



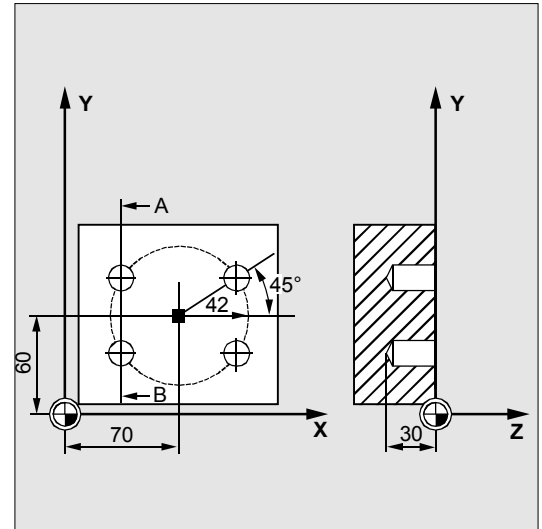


### Programming example

#### Hole circle

The program uses CYCLE82 to produce 4 holes with a depth of 30 mm. The final drilling depth is defined as a value relative to the reference plane. The circle is defined by the center point X70 Y60 and the radius 42 mm in the XY plane. The initial angle is 45 degrees.

The safety distance in the 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 hole circle cycle, the indexing ;angle is calculated internally by the cycle ;as parameter INDA has been omitted
N50 MCALL	;Deselect modal call
N60 M30	;End of program

## 2.3.4 Dot matrix – CYCLE801

**Programming**

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

**Parameters**

_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	int	Number of columns
_NUM2	int	Number of rows

**Function**

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

**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 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)**

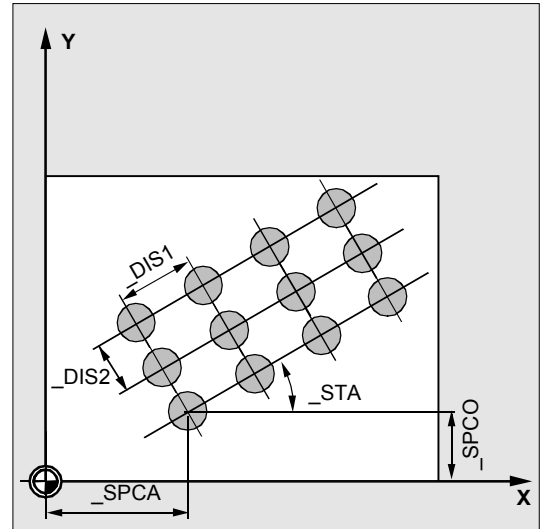
The grid of holes can be positioned at any angle in the plane. This angle is programmed in degrees in **\_STA** and refers to the abscissa of the workpiece coordinate system active as the cycle is called.

#### **\_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.

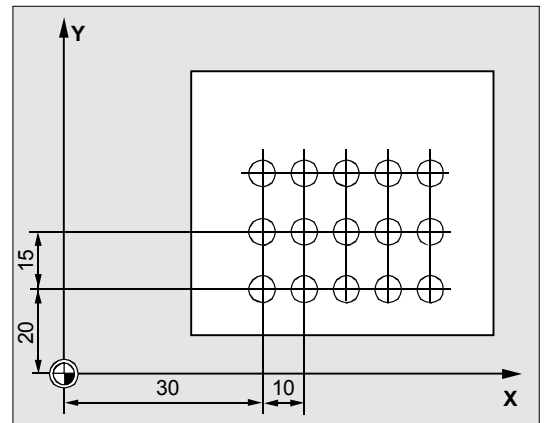
#### **\_NUM1 and \_NUM2 (number)**

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



### Programming example

Cycle CYCLE801 is used to machine a dot matrix, consisting of 15 holes arranged in three lines and five 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

## Notes



## Milling Cycles

3.1	General .....	3-107
3.2	Prerequisites .....	3-108
3.3	Thread milling - CYCLE90 .....	3-110
3.4	Elongated holes on a circle - LONGHOLE .....	3-117
3.5	Slots on a circle - SLOT1 .....	3-122
3.6	Circumferential slot - SLOT2.....	3-130
3.7	Milling rectangular pockets - POCKET1 .....	3-136
3.8	Milling circular pockets - POCKET2.....	3-140
3.9	Milling rectangular pockets - POCKET3 .....	3-144
3.10	Milling circular pockets - POCKET4.....	3-153
3.11	Face milling - CYCLE71.....	3-158
3.12	Path milling - CYCLE72 .....	3-164
3.13	Milling rectangular spigots - CYCLE76.....	3-174
3.14	Milling circular spigots - CYCLE77.....	3-179
3.15	Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75.....	3-183
3.15.1	Transfer pocket edge contour - CYCLE74.....	3-184
3.15.2	Transfer island contour - CYCLE75.....	3-186
3.15.3	Contour programming.....	3-187
3.15.4	Pocket milling with islands - CYCLE73.....	3-189
3.16	Swiveling - CYCLE800.....	3-211
3.16.1	Operation, parameter assignment, input screen form .....	3-213
3.16.2	Operating instructions, programming instructions .....	3-217
3.16.3	Parameters.....	3-218
3.16.4	Setting tools (with Cycles SW 6.5 and higher).....	3-222
3.16.5	Setting up workpieces with swiveled machining planes (Cycles SW 6.5 and higher).....	3-223
3.16.6	Operation, parameter assignment "Swiveling in JOG" (with Cycles SW 6.5 and higher).....	3-227
3.16.7	Startup of swivel cycle CYCLE800 .....	3-230
3.16.8	TOOLCARR.SPF user cycle.....	3-251
3.16.9	Error messages.....	3-255
3.17	High-speed settings - CYCLE832 (with Cycles SW 6.3 and higher) .....	3-257
3.17.1	Call CYCLE 832 in HMI menu tree .....	3-260
3.17.2	Parameters.....	3-263
3.17.3	Customizing technology.....	3-264
3.17.4	Customizing additional program parameters CYC_832T .....	3-266
3.17.5	Interfaces .....	3-268

3.17.6	Error messages .....	3-269
3.18	Engraving cycle CYCLE60 (with Cycles SW 6.4 and higher) .....	3-270

## 3.1 General

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 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.

The Sections are structured as follows:

- **Programming**
- **Parameters**
- **Function**
- **Sequence**
- **Explanation of parameters**
- **Other information**
- **Programming example**

"Programming" and "Parameters" explain the use of cycles sufficiently for the experienced user, whereas beginners can find all the information they need for programming cycles under "Function", "Sequence", "Explanation of parameters", "Other information" and the "Programming example".

## 3.2 Prerequisites

### Programs required in the control

The milling cycles call the programs

- MESSAGE.SPF and
- PITCH.SPF

internally as subroutines. Moreover, you need 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 actual axis names. You must activate a tool offset before you call the milling cycles.

The required values for the feedrate, spindle speed and spindle direction of rotation must be programmed in the parts program if no parameters are available for these in the milling cycle.

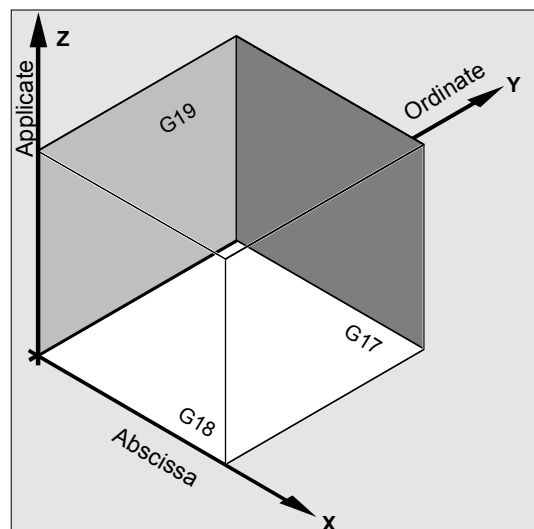
The center point coordinates of the milling pattern or the pocket to be machined are programmed in the clockwise coordinate system.

The G functions and current programmable frame active before the cycle was called 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  
/PGA/, Programming Guide Advanced



### Spindle handling

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

If you want to use a cycle on a machine with several spindles, you must first define the spindle that is to be used as the master spindle, using the SETMS command



**References:** /PG/, Programming Guide  
Fundamentals  
/PGA/, Programming Guide Advanced

### Machining status messages

Status messages are displayed on the control monitor during the processing of milling cycles.

The following messages can be displayed:

- "Elongated hole <No.>(first figure) being machined"
- "Slot <No.>(other figure) being machined"
- "Circumferential slot <No.>(last figure) being machined".

In each case <No.> stands for the number of the figure that is currently being machined.

These messages do not interrupt program processing 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 are defined in data block GUD7.DEF.

The following new cycle setting data are introduced:

<u>_ZSD[x]</u>	Value	Meaning	Cycles affected
<u>_ZSD[1]</u>	0	Depth computation in the new cycles is carried out between the reference plane + safety distance and depth ( <u>_RFP</u> + <u>_SDIS</u> - <u>_DP</u> )	POCKET1 to POCKET4, LONGHOLE,
	1	Depth computation is carried out without including safety distance	CYCLE71, SLOT1, CYCLE72, SLOT2
<u>_ZSD[2]</u>	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	
<u>_ZSD[5]</u>	0	Execute at drilling depth M5 M0	CYCLE88
	1	Execute at drilling depth M5	

### 3.3 Thread milling - CYCLE90



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)
DIATH	real	Nominal diameter, outside diameter of thread
KDIAM	real	Core diameter, inside diameter of thread
PIT	real	Thread pitch; value range: 0.001 ... 2000.000 mm
FFR	real	Feedrate for thread milling (enter without sign)
CDIR	int	Direction of rotation for thread milling Values: 2 (for thread milling with G2) 3 (for thread milling with G3)
TYPTH	int	Thread type: values: 0= Inside thread 1=Outside thread
CPA	real	Center point of circle, abscissa (absolute)
CPO	real	Center point of circle, ordinate (absolute)



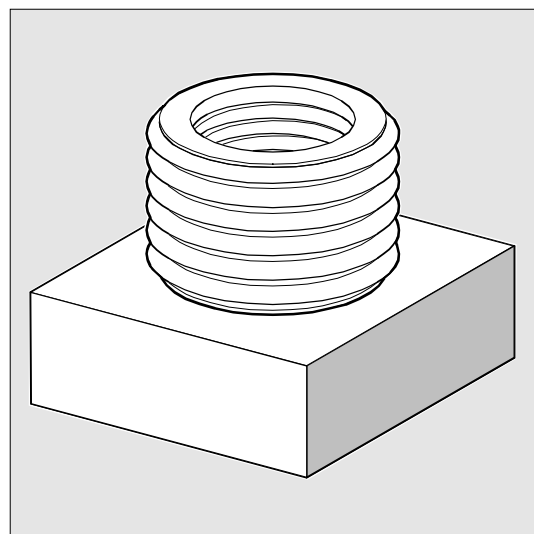
#### Function

You can produce inside and outside threads with cycle CYCLE90. The path in thread milling is based on helical interpolation. All three geometry axes of the current plane, which you define before calling the cycle, are involved in this movement.

The programmed feedrate  $F$  depends on the axis grouping defined in the FGROUPE instruction before the cycle call.



**References:** /PG/, Programming Guide  
Fundamentals  
/PGA/, Programming Guide Advanced





## Sequence

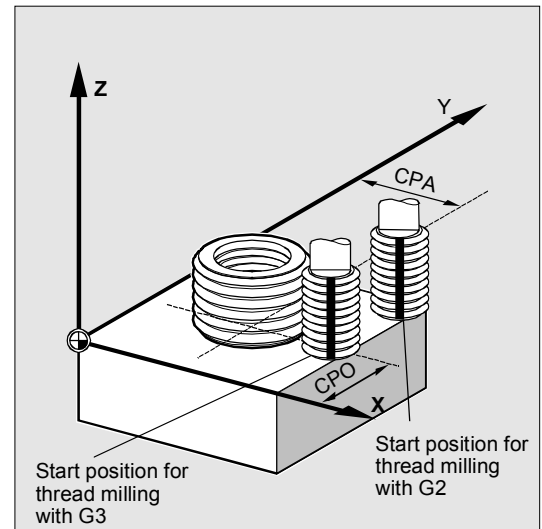
### Outside threads

#### Position reached before the beginning of the cycle:

This can be any position, from which the starting position on the outside diameter of the thread at the retraction plane level 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).

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



#### The cycle implements the following motion sequence:

- Travel to the starting point with G0 at the retraction plane level in the applicate of the current plane.
- Infeed to the reference plane brought forward by the safety distance with G0.
- Travel-in movement to the thread diameter along a circular path opposite to direction G2/G3 defined in CDIR.
- Thread milling along a helical path with G2/G3 and feedrate FFR.
- Travel-out movement along a circular path in the opposite direction G2/G3 and the reduced feedrate FFR.
- Retraction to retraction plane in the applicate with G0.

### Inside threads

#### **Position reached before the beginning of the cycle:**

The starting position can be any position, from which the center point of the thread at the retraction plane level can be approached without collision.

#### **The cycle implements the following motion sequence:**

- Travel to the center point of the thread with G0 at the retraction plane level in the applicate of the current plane.
- Infeed to the reference plane brought forward by the safety distance with G0.
- Approach with G1 and the reduced feedrate FFR along an approach circle calculated in the cycle.
- Travel-in movement to the thread diameter along a circular path in the direction G2/G3 defined in CDIR.
- Thread milling along a helical path with G2/G3 and feedrate FFR.
- Travel-out movement along 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 retraction plane in the applicate with G0.

#### **Thread from bottom to top**

For technological reasons, it may be preferable to machine the thread from the bottom to the top. The retraction plane RTP is then below the thread depth DP.

This machining operation is possible if the depth data are programmed as absolute values before cycle start and the machine is positioned on the retraction plane or one position behind the retraction plane.





### Programming example

(thread from bottom to top)

A thread must be milled starting from -20 up to 0, with a 3 mm pitch. The retraction plane is at 8.

```
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
```

The hole must have a depth of at least -21.5 (half pitch in excess).

### Overshoot in the thread longitudinal direction

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 thus:

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

$\Delta z$     Overshoot, internal

$p$        Pitch

$WR$      Tool radius

$DIATH$  Outside diameter of the thread

$RDIFF$  Radius difference for travel-out circle

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

For outside threads  $RDIFF = DIATH/2 + WR$ .

### Function extension with SW 6.3 and higher

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

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

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

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

$\Delta z$  Overshoot, internal

p Pitch

WR Tool radius

DIATH Outside diameter of the thread

KDIAM Core diameter of the thread



### Explanation of parameters



See Subsection 2.1.2 (Drilling, Centering – CYCLE81) for parameters RTP, RFP, SDIS, DP, and DPR.

### DIATH, KDIAM and PIT (nominal diameter, core diameter and thread pitch)

You define thread data such as nominal diameter, core diameter and pitch using these parameters. Parameter DIATH is the outside diameter and KDIAM the inside diameter of the thread. The travel-in and travel-out movements are generated by the cycle based on these parameters.

### FFR (feedrate)

The value of parameter FFR is defined as the current feedrate value for thread milling. In thread milling it is active for the movement along the helical path.

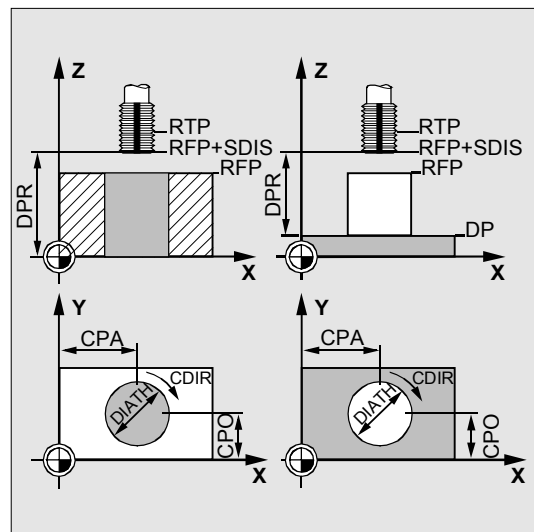
This value is reduced in the cycle for the travel-in and travel-out movements. Retraction is performed outside the helical path with G0.

### CDIR (direction of rotation)

You define the value for the machining direction of the thread in this parameter.

If the parameter is assigned an invalid value, the following alarm is output:

"Incorrect mill direction: G3 will be generated".



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

**TYPTH (thread type)**

With parameter TYPTH, you determine whether an outside or inside thread is to be machined.

**CPA and CPO (center point)**

You define the center point of the hole or spigot, on which the thread is to be machined, using these parameters.

**Other information**

The milling cutter radius is taken into account by the cycle. Therefore, a tool offset must be programmed before the cycle is called. Otherwise, the following alarm is output:

61000 "No tool offset active"  
and the cycle is aborted.

When the tool radius equals zero or a negative value, the cycle is also aborted with this alarm.

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

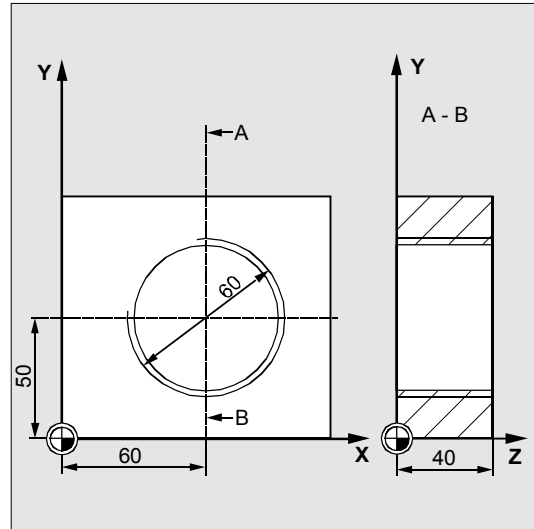
## 3.3 Thread milling - CYCLE90



## Programming example

## Inside thread

With this program, you can machine an inside thread at position X60 Y50 on 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	;End of program
-> Must be programmed in a single block	

### 3.4 Elongated holes on a circle - LONGHOLE



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Elongated hole final drilling depth (absolute)
DPR	real	Elongated hole final drilling depth relative to reference plane (enter without sign)
NUM	integer	Number of elongated holes
LENG	real	Length of elongated hole (enter without sign)
CPA	real	Center point of circle, abscissa (absolute)
CPO	real	Center point of circle, ordinate (absolute)
RAD	real	Radius of circle (enter without sign)
STA1	real	Initial angle
INDA	real	Indexing angle
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for infeed (enter without sign)



*The cycle requires a milling cutter with an "end tooth cutting over center" (DIN 844).*



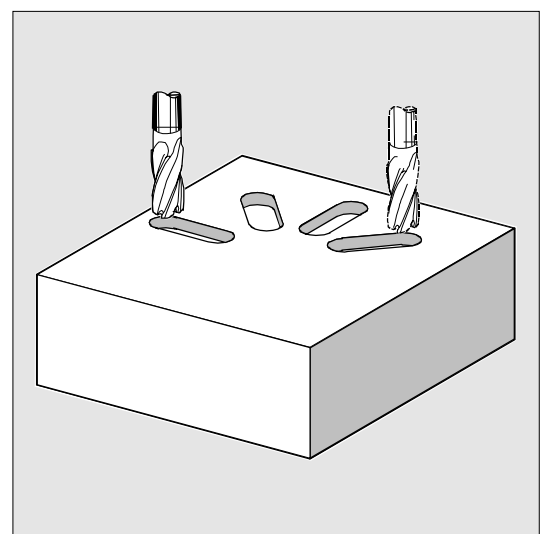
#### Function

Elongated holes arranged on a circle can be machined with this cycle. The longitudinal axis of the elongated holes is arranged radially.

Unlike the slot, the width of the elongated hole is determined by the diameter of the tool.

To avoid unnecessary travel, the cycle calculates the optimum path for the tool. If several depth infeed movements are required to machine an elongated hole, the infeed is performed at alternate end points.

The path to be traversed in the plane along the longitudinal axis of the elongated hole changes direction after every infeed. The cycle automatically looks for the shortest path when changing to the next elongated hole.





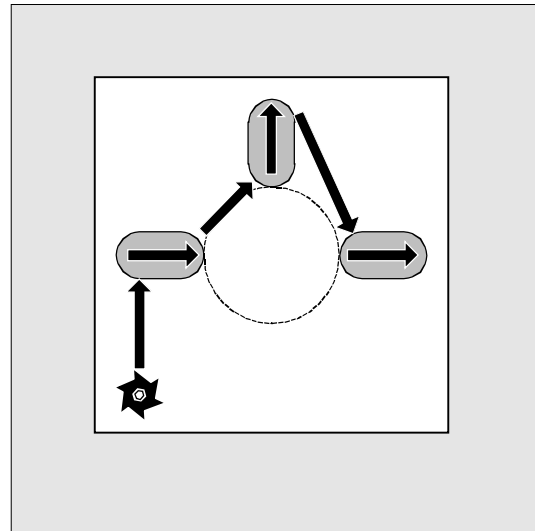
### Sequence

#### Position reached before the beginning of the cycle:

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

#### The cycle implements the following motion sequence:

- The starting position of a cycle is approached with G0. The nearest end point of the first elongated hole to be machined is approached in both axes of the current plane at the retraction plane level in the applicate of this plane and then lowered in the applicate to the reference plane brought forward by the safety distance.
- Each elongated hole is milled in a reciprocating movement. Machining is performed in the plane with G1 and the feedrate programmed under FFP1. At each reversal point, the infeed to the next machining depth calculated by the cycle is performed with G1 and the feedrate FFD, until the final depth is reached.
- Retraction to the retraction plane with G0 and approach to the next elongated hole along the shortest path.
- When the last elongated hole has been machined, the tool is traversed from the last position reached in the machining plane to the retraction plane with G0 and the cycle is terminated.





### Explanation of the parameters

See Subsection 2.1.2. (Drilling, Centering –CYCLE81) for a description of parameters RTP, RFP and SDIS. See Section 3.2 for cycle setting data `_ZSD[1]`.

#### DP and DPR (elongated hole depth)

The elongated hole final drilling 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.

#### NUM (number)

The number of elongated holes is determined with the parameter NUM.

#### LENG (elongated hole length)

The elongated hole length is programmed under LENG.

If, during the cycle run, it is detected that this length is less than the cutter diameter, then the cycle is aborted with alarm 61105 "Cutter radius too large".

#### MID (infeed depth)

The maximum infeed depth is defined with this parameter.

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

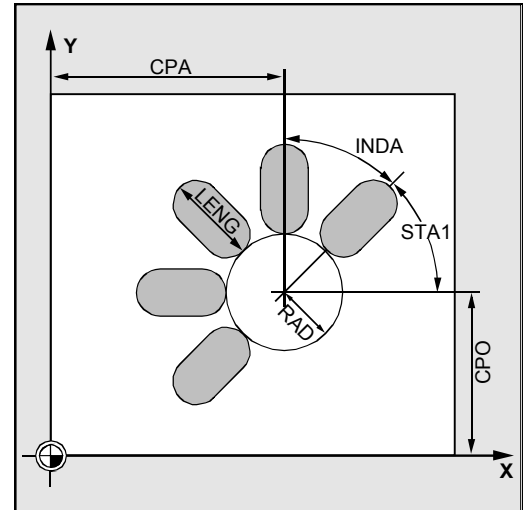
Using MID and the total depth, the cycle automatically calculates this infeed, which lies between 0.5 x maximum infeed depth and the maximum infeed depth. The minimum possible number of infeed steps is used as the basis. MID=0 means that the cut to pocket depth is made with one infeed. The depth infeed commences at the reference plane brought forward by the safety distance (as a function of `_ZSD[1]`).

#### FFD and FFP1 (feedrate depth and plane)

Feedrate FFP1 is active for all traversing movements performed in the plane at feedrate. FFD is active for infeeds that are perpendicular to this plane.

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

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



### 3.4 Elongated holes on a circle - LONGHOLE

#### STA1 and INDA (initial angle and indexing angle)

The arrangement of the elongated holes around the circle is defined by these parameters.

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



#### Other information

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with alarm 61000 "No tool offset active".

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

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

When the cycle is completed, the workpiece coordinate system is again in the same position as before the cycle was called.

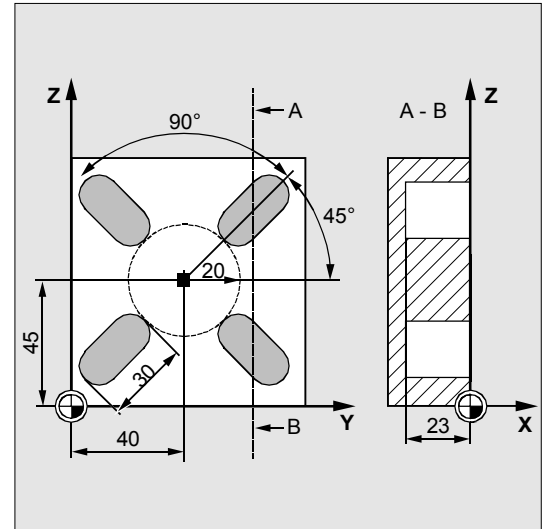




### Programming example

#### Machining elongated holes

With this program you can machine four elongated holes, which are 30 mm in length and with a relative depth of 23 mm (difference between the reference plane and the base of the elongated hole), that lie in a circle with the center point Z45 Y40 and a radius of 20 mm in the YZ plane. The initial angle is 45 degrees, the indexing angle is 90 degrees. The maximum infeed depth is 6 mm, the safety distance is 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, ->  
-> 40, 45, 20, 45, 90, 100 ,320, 6)

;Cycle call

N40 M30

;End of program

-> Must be programmed in a single block

### 3.5 Slots on a circle - SLOT1



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (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 circle (enter without sign)
STA1	real	Initial angle
INDA	real	Indexing angle
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for infeed (enter without sign)
CDIR	integer	Mill direction for machining the slot Values: 0...Down-cut milling (as spindle rotation) 1...Up-cut milling 2...With G2 (independent of spindle direction) 3...With G3
FAL	real	Final machining allowance on slot edge (enter without sign)
VARI	integer	Machining type (enter without sign) UNITS DIGIT: Values: 0...Complete machining 1...Roughing 2...Finishing TENS DIGIT: Values: 0...Perpendicular with G0 1...Perpendicular with G1 3...Oscillation with G1
MIDF	real	Maximum infeed depth for finishing
FFP2	real	Feedrate for finishing
SSF	real	Speed for finishing

<code>_FALD</code>	real	Final machining allowance on the base of slot
<code>_STA2</code>	real	Maximum insertion angle for oscillation movement



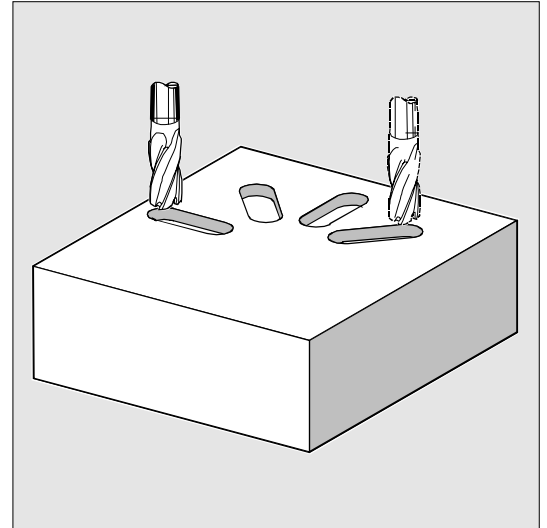
The cycle requires a milling cutter with an "end tooth cutting over center" (DIN 844).



### Function

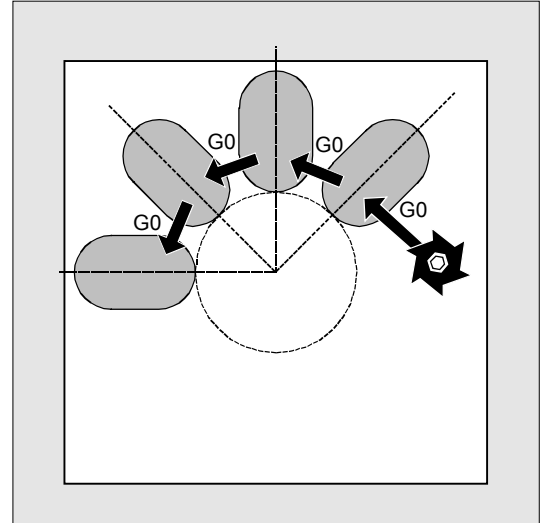
Cycle SLOT1 is a combined roughing-finishing cycle.

With this cycle, you can machine slots arranged on a circle. The longitudinal axis of the slots is arranged radially. Unlike the elongated hole, a value is defined for the slot width.



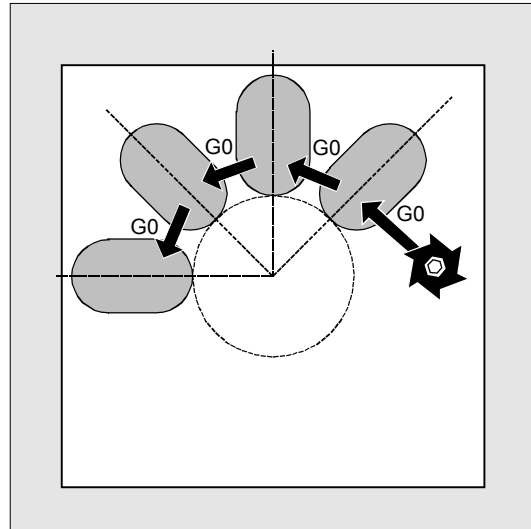
### Sequence

Position reached before the beginning of the cycle:  
The starting position can be any position, from which each of the slots can be approached without collision.



### The cycle implements the following motion sequence:

- Approach the position marked in the figure on the right at the beginning of the cycle with G0.
- Complete machining of a slot is performed in the following stages:
  - Approach to reference plane brought forward by the safety distance with G0
  - Infeed to the next machining depth as programmed under VAR1 and at feedrate value FFD
  - Solid machining of the slot to the final machining allowance on slot base and slot edge at feedrate value FFP1.
 Subsequent finishing at feedrate value FFP2 and spindle speed SSF along the contour, according to the machining direction programmed under CDIR.
  - The vertical depth infeed of G0/G1 always takes place at the same position in the machining plane until the final depth of the slot is reached.
  - With reciprocating machining, select the starting point such that the end point always reaches the same position in the machining plane.
- Retract tool to retraction plane and move to next slot with G0.
- After machining the last slot, the tool moves to the retraction plane with G0 and the cycle ends.

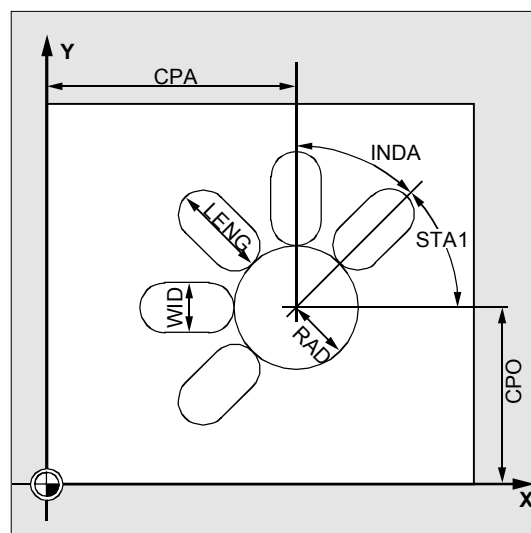


### Explanation of parameters

See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP and SDIS.  
See Section 3.2 for cycle setting data `_ZSD[1]`.

#### DP and DPR (slot depth)

The slot 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.



**NUM (number)**

The number of slots is determined with the parameter NUM.

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

The shape of a slot in the plane is determined with parameters LENG and WID. The milling cutter diameter must be smaller than the slot width. Otherwise, alarm 61105 "Cutter radius too large" is activated and the cycle aborted.

The milling cutter diameter must not be smaller than half of the slot width. This is not checked.

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

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

**STA1 and INDA (initial angle and indexing 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 indexing angle is calculated from the number of slots so that they are equally distributed around the circle.

**FFD and FFP1 (feedrate depth and plane)**

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

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

**MID (infeed depth)**

The maximum infeed depth is defined with this parameter. 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 infeed.

### 3.5 Slots on a circle - SLOT1

The depth infeed commences at the reference plane brought forward by the safety distance (as a function of `_ZSD[1]`).

#### CDIR (mill direction)

You define the slot machining direction in this parameter.

Under parameter CDIR, the mill direction can be programmed as

- Directly "2 for G2" and "3 for G3" or
- Alternatively "up-cut milling" or "down-cut milling"

Up-cut milling or down-cut milling is determined within the cycle via the spindle direction activated prior to the cycle call.

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

#### FAL (final machining allowance at slot edge)

With this parameter, you can program a final machining allowance on the slot edge. FAL does not affect the depth infeed. If the value of FAL is greater than permitted for the specified width and the milling cutter used, FAL is automatically reduced to the maximum possible value. In the case of rough machining, 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)

You can define the type of machining with parameter VARI.

Possible values are:

UNITS DIGIT

- 0=Complete machining in two parts
  - Machining of the slot (SLOT1, SLOT2) or pocket (POCKET1, POCKET2) to the final machining allowance is performed at the spindle speed programmed before the cycle was called and with feedrate FFP1. Depth infeed is defined with MID.
  - Solid machining of the remaining final machining allowance is performed at the spindle speed preset by SSF and feedrate FFP2. With parameter MIDF, it is possible to program a depth infeed other than that programmed for roughing. However, this is only effective for finishing at the edge. If MIDF=0, the infeed is equal to the final depth.

If FFP2 is not programmed, feedrate FFP1 applies. The same applies if there is no message from SSF, i.e., the speed programmed before the cycle was called is applicable.

- 1=Roughing  
The slot (SLOT1, SLOT2) or pocket (POCKET1, POCKET2) is solid machined up to the final machining allowance at the speed programmed before the cycle call and feedrate FFP1. The depth infeed is programmed in MID.
- 2=Finishing  
The cycle requires that the slot (SLOT1, SLOT2) or pocket (POCKET1, POCKET2) have already been machined to a remaining final machining allowance, so it is only necessary to machine the final machining allowance. If neither FFP2 nor SSF are 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=Perpendicular with G0
- 1=Perpendicular with G1
- 3=Oscillation with G1

If another value is programmed for parameter VARI, the cycle aborts after outputting alarm 61102 "Operating mode not defined correctly".

#### **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 (final machining allowance on slot base)**

A separate final machining allowance on the base is taken into account in roughing operations.

#### **\_STA2 (insertion angle)**

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

- **Vertical insertion (VARI=0X, VARI=1X)**  
Vertical depth insertion is always performed at the same position on the machining plane down to the final depth of the slot.

### 3.5 Slots on a circle - SLOT1

- Insertion with oscillation on the center axis of the slot (VARI=3X)**  
 means that the mill center point oscillates along an oblique linear path until it reaches the next current depth. The maximum insertion angle is programmed under `_STA2`, 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`.

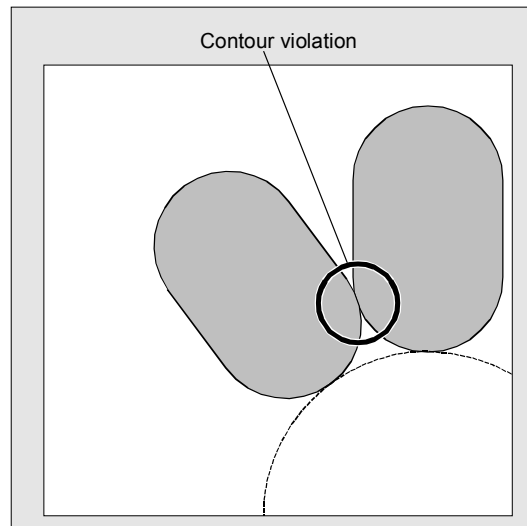
#### Other information

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with Alarm 61000 "No tool offset active".

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

During the cycle, the workpiece coordinate system is shifted 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.

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







### Programming example

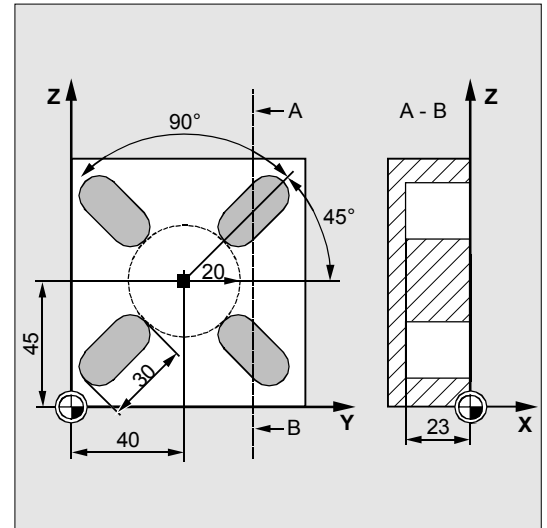


#### Slots

This program produces the same arrangement of four slots on a circle as the program for elongated hole machining (see Section 3.4).

The slots have the following dimensions: Length 30 mm, width 15 mm and depth 23 mm. The safety distance is 1 mm, the final machining allowance is 0.5 mm, the mill direction is G2, the maximum infeed in the depth is 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
```

```
;End of program
```

-> Must be programmed in a single block

## 3.6 Circumferential slot - SLOT2

### 3.6 Circumferential slot - SLOT2



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (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)
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 circle (enter without sign)
STA1	real	Initial angle
INDA	real	Indexing angle
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for infeed (enter without sign)
CDIR	integer	Mill direction for machining the circumferential slot Values: 2 (for G2) 3 (for G3)
FAL	real	Final machining allowance on slot edge (enter without sign)
VARI	integer	Machining type UNITS DIGIT: Values: 0=Complete machining 1=Roughing 2=Finishing TENS DIGIT (SW 6.3 and higher) Values: 0=Positioning from slot to slot in a straight line with G0 1=Positioning from slot to slot in a circular path with feedrate
MIDF	real	Maximum infeed depth for finishing
FFP2	real	Feedrate for finishing
SSF	real	Speed for finishing
_FFCP (SW 6.3 and higher)	real	Feedrate for intermediate positioning on a circular path, in mm/min



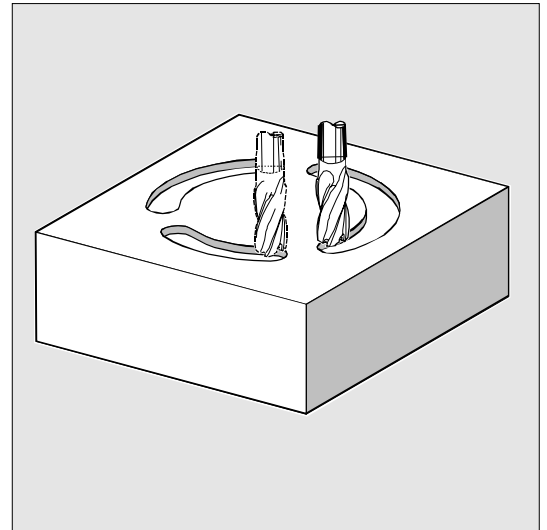
*The cycle requires a milling cutter with an "end tooth cutting over center" (DIN 844).*



### Function

Cycle SLOT2 is a combined roughing-finishing cycle.

With this cycle you can machine circumferential slots arranged on a circle.



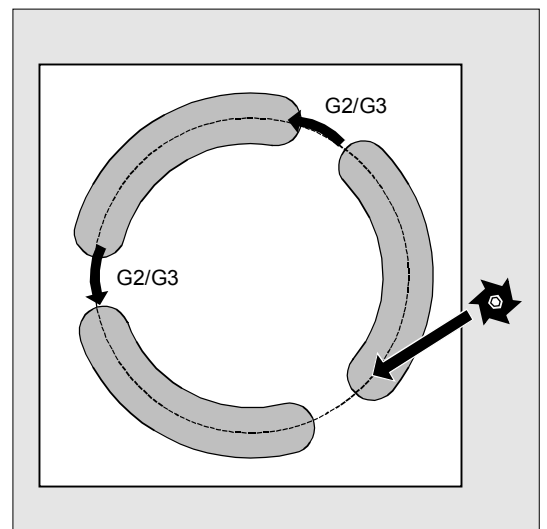
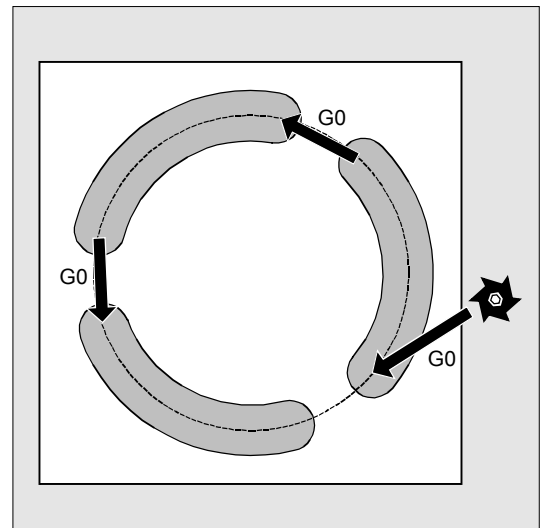
### Sequence

#### Position reached before the beginning of the cycle:

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

#### The cycle implements the following motion sequence:

- Approach the position marked in the figure on the right at the beginning of the cycle with G0.
- 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.
- When the last slot has been machined, the tool is traversed to the end position in the machining plane specified in the display to the retraction plane with G0 and the cycle is terminated.



## 3.6 Circumferential slot - SLO2



### Explanation of the parameters



See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP and SDIS.



See Section 3.5 (SLOT1) for a description of parameters DP, DPR, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2 and SSF.

See Section 3.2 for cycle setting data `_ZSD[1]`.

#### NUM (number)

The number of slots is determined with the parameter NUM.

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

With parameters AFSL and WID, you define the shape of a slot in the plane. The cycle checks whether the slot width is violated by the active tool. If this is the case, alarm 61105 "Cutter radius too large" is output and the cycle is aborted.

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

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

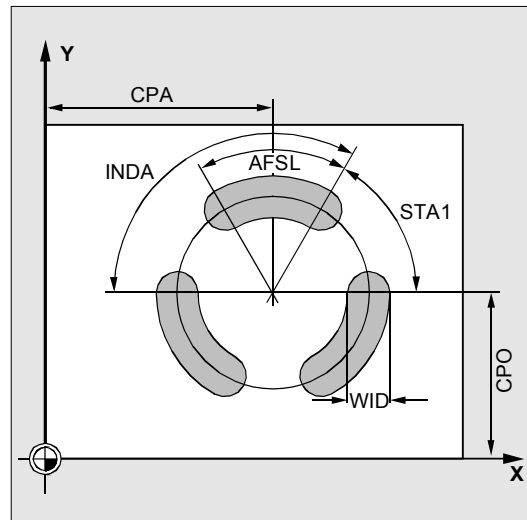
#### STA1 and INDA (initial angle and indexing angle)

The arrangement of 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 indexing angle is calculated from the number of circumferential slots, so that they are equally distributed around the circle.



## New machining types in SW 6.3 and higher:

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

- There is a new selection, "Finishing the edge". 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. The slot 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 the 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.

### Other information

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with alarm 61000 "No tool offset active".

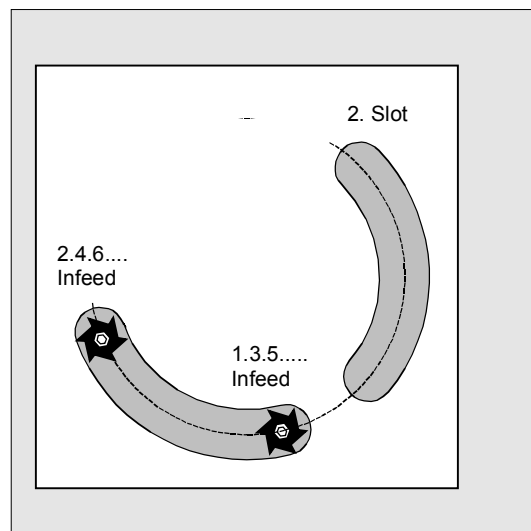
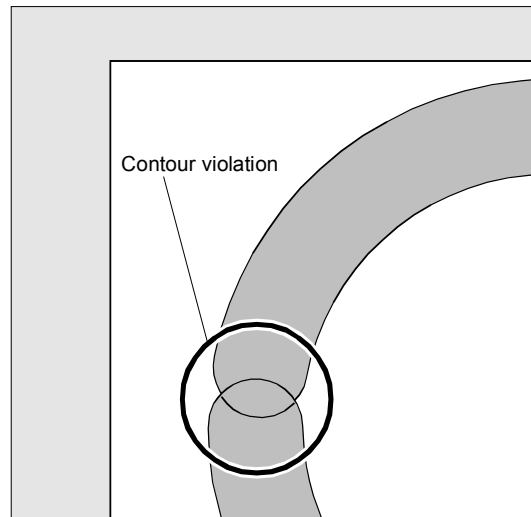
If incorrect values are assigned to the parameters that determine the arrangement and size of the slot, thus causing mutual contour violation of the slots, the cycle is not started.

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

During the cycle, the workpiece coordinate system is shifted 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. When the cycle is completed, the workpiece coordinate system is again in the same position as before the cycle was called.

#### Special case: Slot width = cutter diameter

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

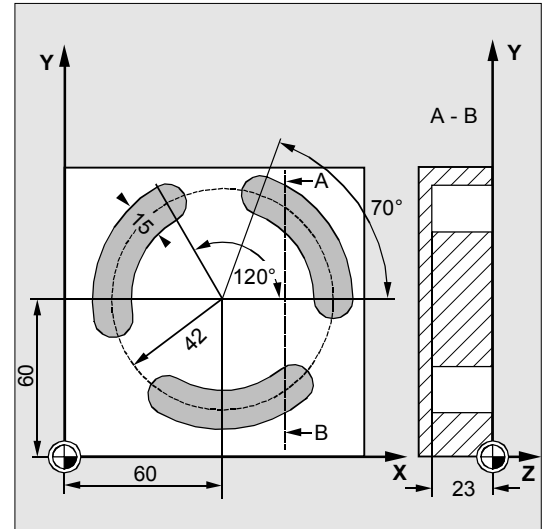




### Programming example

#### Slots2

With this program, you can machine three circumferential slots arranged on a circle, whose center point is 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 indexing angle is 120 degrees. The slot contours are machined to a final machining allowance of 0.5 mm, the safety distance in infeed axis Z is 2 mm, the maximum depth infeed is 6 mm. The slots are to be completely machined. The same speed and feedrate are used for finishing. Infeed during finishing is performed straight to the base of the slot.



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, -> -> 15, 60, 60, 42, , 120, FFD, -> -> FFD+200, 6, 2, 0.5)	;Cycle call ;Reference plane + SDIS=retraction plane ;means: Lower in infeed axis with ;G0 to reference plane + SDIS no longer ;applicable, parameters VAR, MIDF, ;FFP2 and SSF omitted
N40 M30	;End of program

-> Must be programmed in a single block

### 3.7 Milling rectangular pockets - POCKET1



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (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 Value range: $0 \leq \text{STA1} < 180$ degrees
FFD	real	Feedrate for depth infeed
FFP1	real	Feedrate for surface machining
MID	real	Maximum infeed depth for infeed (enter without sign)
CDIR	integer	Mill direction for machining the pocket Values: 2 (for G2) 3 (for G3)
FAL	real	Final machining allowance on 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 for finishing



*The cycle requires a milling cutter with an "end tooth cutting over center" (DIN 844).*



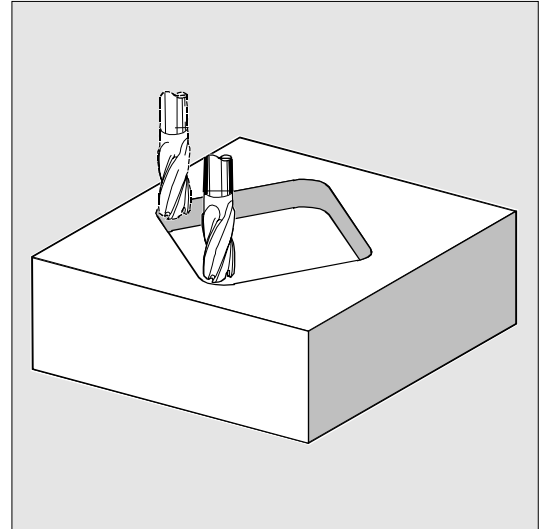
The pocket milling cycle POCKET3 can be performed with any tool.





### Function

The cycle is a combined roughing-finishing cycle. With this cycle, you can machine rectangular pockets in any position in the machining plane.



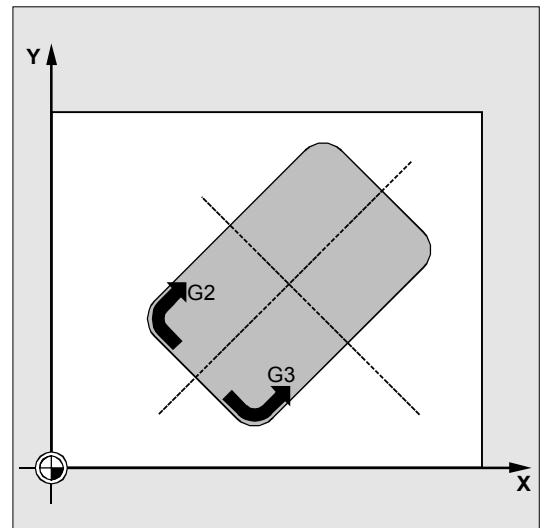
### Sequence

#### Position reached before the beginning of the cycle:

The starting position can be any position, from which the center point of the pocket at the retraction plane level can be approached without collision.

#### The cycle implements the following motion sequence:

- With G0, the pocket center point is approached at the retraction plane level and then, from this position, the reference plane brought forward by the safety distance is approached with G0. Complete machining of the pocket is performed in the following stages:
  - Infeed to the next machining depth with G1 and feedrate FFD
  - Pocket milling up to the final machining allowance with feedrate FFP1 and the spindle speed that was active before the cycle was called.
- After roughing is completed:
  - Infeed to the machining depth defined by MIDF
  - Final machining along the contour with feedrate FFP2 and speed SSF
  - The machining direction is defined by 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.



### 3.7 Milling rectangular pockets - POCKET1



#### Explanation of the parameters

See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP and SDIS.

See Section 3.5 (SLOT1) for a description of parameters FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2 and SSF.

See Section 3.2 for cycle setting data `_ZSD[1]`.

#### 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.

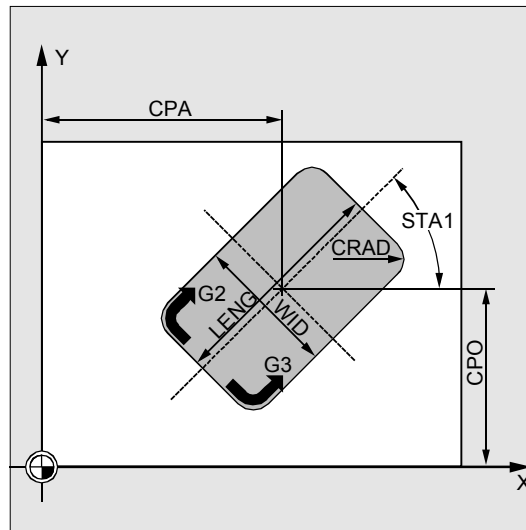
If it is not possible to traverse to the programmed corner radius with the active tool because its radius is larger, the corner radius of the completed pocket corresponds to the tool radius. If the milling cutter radius is greater than half the length or width of the pocket, the cycle is aborted and 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.



#### Other information

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with alarm 61000 "No tool offset active".

A new current workpiece coordinate system that influences the actual value display is used in the cycle. The zero point of this coordinate system lies on the pocket center point.

The original coordinate system becomes active again after the end of the cycle.

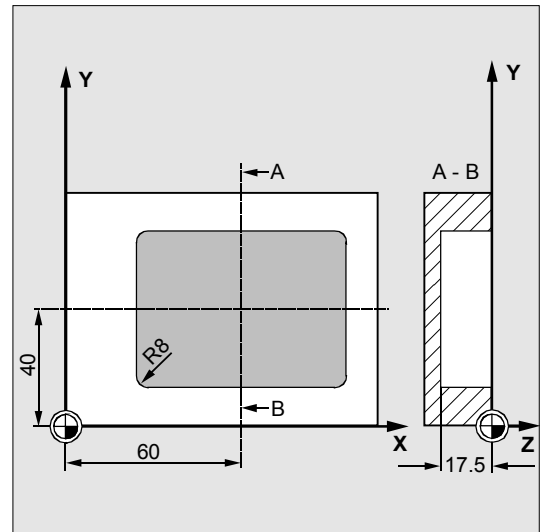


### Programming example

#### Pocket

With this program, you can machine a pocket that is 60 mm long, 40 mm wide, 17.5 mm deep (difference between the reference plane and the base of the pocket) and, which has a corner radius of 8 mm, in the XY plane. The angle to the X axis is 0 degrees. 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.

Only roughing is to be performed.



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	;End of program
-> Must be programmed in a single block	

### 3.8 Milling circular pockets - POCKET2



#### Programming

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



#### Parameters

RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (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 infeed (enter without sign)
CDIR	integer	Mill direction for machining the pocket Values: 2 (for G2) 3 (for G3)
FAL	real	Final machining allowance on 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 for finishing



*The cycle requires a milling cutter with an "end tooth cutting over center" (DIN 844).*

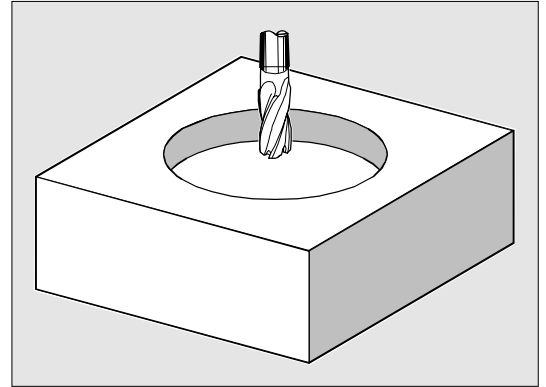


The pocket milling cycle POCKET4 can be performed with any tool.



### Function

The cycle is a combined roughing-finishing cycle. With this cycle, you can machine circular pockets in the machining plane.



### Sequence

#### Position reached before the beginning of the cycle:

The starting position can be any position, from which the center point of the pocket at the retraction plane level can be approached without collision.

#### The cycle implements the following motion sequence:

- With G0, the pocket center point is approached at the retraction plane level and then, from this position, the reference plane brought forward by the safety distance is approached with G0. Complete machining of the pocket is performed in the following stages:
  - Infeed perpendicular to the pocket center to the next machining depth with feedrate FFD
  - Pocket milling up to the final machining allowance with feedrate FFP1 and the spindle speed that was active before the cycle was called.
- 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 defined by CDIR.
- When machining is completed, the tool is traversed to the pocket center point in the retraction plane and the cycle is terminated.



### Explanation of the parameters



See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP and SDIS.

See Section 3.7 for a description of parameters DP and DPR.



See Section 3.5 (SLOT1) for a description of parameters FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, and SSF.

See Section 3.2 for cycle setting data `_ZSD[1]`.

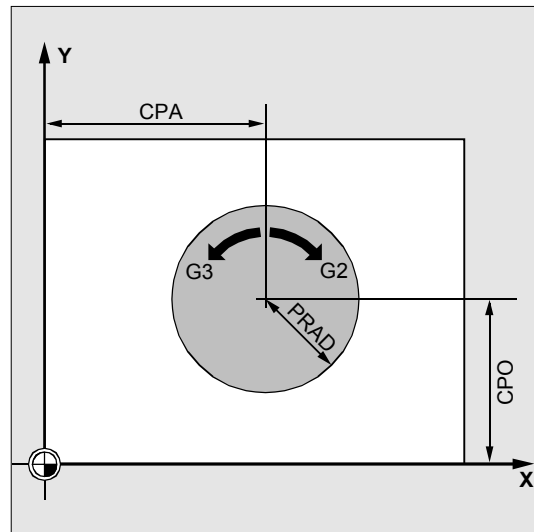
#### PRAD (pocket radius)

The shape of the circular pocket is determined by the radius only.

If the radius is less than the tool radius of the active tool, the cycle is aborted after 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.



### Other information

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with alarm 61000 "No tool offset active".

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

A new current workpiece coordinate system that influences the actual value display is used in the cycle. The zero point of this coordinate system lies on the pocket center point.

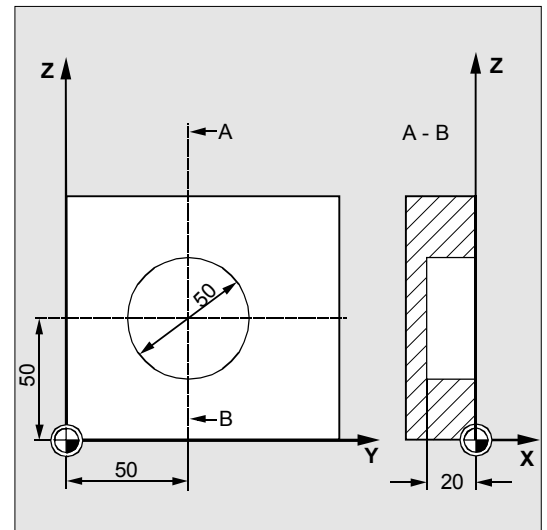
The original coordinate system becomes active again after the end of the cycle.



### Programming example

#### Circular pocket

With this program, you can machine a circular pocket in the YZ plane. The center point is defined as Y50 Z50. The infeed axis for the depth infeed is the X axis, the pocket depth is entered as an absolute value. Neither a final machining allowance nor a safety distance is defined.



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	;End of program

-> Must be programmed in a single block

### 3.9 Milling rectangular pockets - POCKET3



The POCKET3 cycle is available with SW 4 and higher.



#### Programming

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



#### Parameters

The following input parameters are always required:

_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety distance (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); Value range: $0^\circ \leq \_STA < 180^\circ$
_MID	real	Maximum infeed depth (enter without sign)
_FAL	real	Final machining allowance on pocket edge (enter without sign)
_FALD	real	Final allowance at base (enter without sign)
_FFP1	real	Feedrate for surface machining
_FFD	real	Feedrate for depth infeed
_CDIR	integer	Mill direction: (enter without sign) Values: 0...Down-cut milling (as spindle rotation) 1...Up-cut milling 2...With G2 (independent of spindle direction) 3...With G3
_VARI	integer	Type of machining: (enter without sign) UNITS DIGIT: Values: 1...Roughing 2...Finishing TENS DIGIT: Values: 0...Perpendicular to the pocket center with G0 1...Perpendicular to the pocket center with G1 2...Along a helix 3...Oscillation along the pocket longitudinal axis



The other parameters can be selected as options. They define the insertion strategy and overlapping for solid machining: (enter without sign)

_MIDA	real	Maximum infeed width during solid machining in the plane
_AP1	real	Basic size pocket length
_AP2	real	Basic size pocket width
_AD	real	Basic pocket depth 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

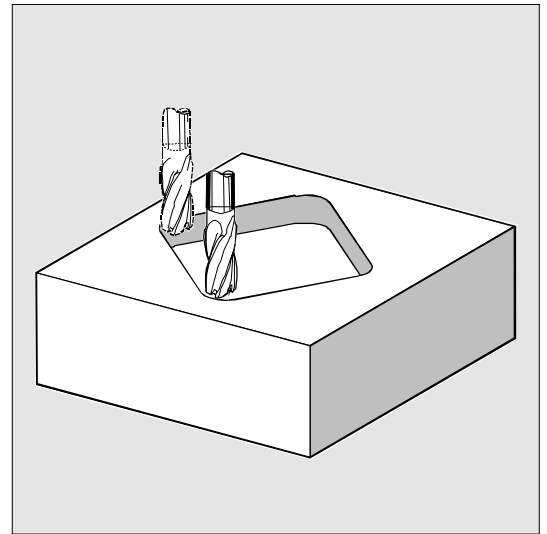


### Function

The cycle can be applied to roughing and finishing. A face cutter is required for finishing. 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 mill direction can be defined by a G instruction (G2/G3), up-cut milling or down-cut milling from the spindle direction.
- For solid machining, the maximum infeed width in the plane can be programmed.
- Final machining allowance for the pocket base
- Three different insertion strategies:
  - Perpendicular at the pocket center point
  - Along a helical path around the pocket center
  - Oscillating around the pocket central axis
- Short approach paths in the plane for finishing
- Consideration of a blank contour in the plane and a basic size at the base (optimum processing of pre-formed pockets possible)





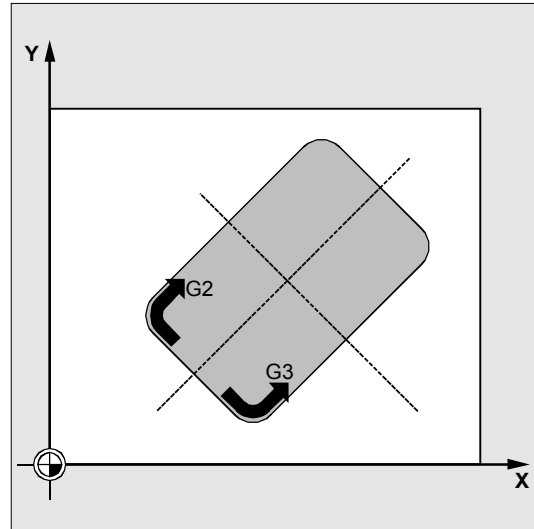
### Sequence

#### Position reached before the beginning of the cycle:

The starting position can be any position, from which the center point of the pocket at the retraction plane level can be approached without collision.

#### Motion sequence when roughing (VARI=X1):

With G0, the pocket center point is approached at the retraction plane level and then, from this position, the reference plane brought forward by the safety distance is approached with G0. Pocket machining is then performed according to the selected insertion strategy and in consideration of the programmed base size.



#### Insertion strategies:

- Vertical insertion to pocket center (VARI=0X, VARI=1X)** means that the current infeed depth internally calculated in the cycle ( $\leq$  programmed maximum infeed depth through `_MID`) is executed in one block with G0 or G1.
- Insertion along helical path (VARI=2X)** means that the milling center point travels on the helical path determined by radius `_RAD1` and depth per revolution `_DP1`. The feedrate is always programmed through `_FFD`. The sense of rotation of this helical path corresponds to the direction to be used for machining the pocket. The depth programmed under `_DP1` on insertion is calculated as the maximum depth and is always calculated as a whole number of revolutions of the helical path. When the current depth for the infeed (these may be several revolutions on the helical path) has been calculated, a full circle is made to remove the slope on insertion. Pocket solid machining then starts in this plane and continues until it reaches the final machining allowance. The starting point of the helical path described is on the pocket longitudinal axis in the "plus direction" and is approached with G1.

- **Oscillating insertion on center axis of pocket (VARI=3X)**

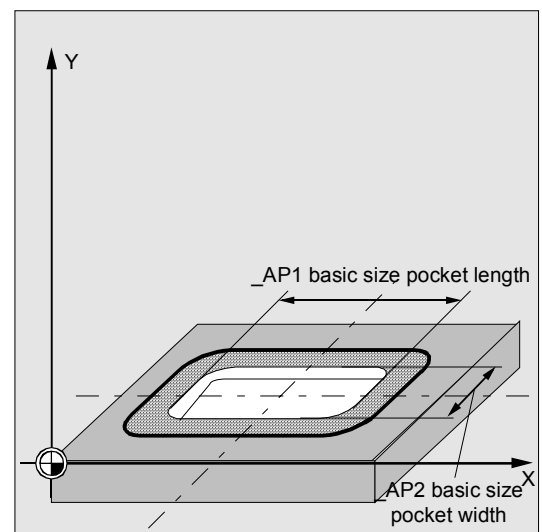
means that the mill center point oscillates along an oblique linear path until it has reached the next current depth. The maximum insertion angle is programmed under `_RAD1`, the length of the oscillation path is calculated within the cycle. When the current depth has been reached, the path is traversed again without depth infeed, in order to remove the slope caused by insertion. The feedrate is programmed through `_FFD`.

#### Accounting for blank dimensions

During solid machining, it is possible to take blank dimensions (for example, in the machining of precast workpieces) into account.

The basic sizes for the length and width (`_AP1` and `_AP2`) are programmed without sign and their symmetrical positions around the pocket center computed in the cycle. They define the part of the pocket that does not have to be solid-machined. The basic size for the depth (`_AD`) is also programmed without a sign and computed in the direction of the pocket depth from the reference plane.

When accounting for blank dimensions, depth infeed is carried out according to the programmed type (helical path, oscillating, vertical). If the cycle detects that the blank contour and the radius of the active tool would provide enough space in the pocket center, infeed is continued vertically downward to the pocket center for as long as possible to avoid time-consuming insertion paths outside the material. The pocket is solid machined beginning from the top and proceeding in the downward direction.



### 3.9 Milling rectangular pockets - POCKET3

#### Motion sequence when finishing (VARI=X2)

Finishing is performed in the following sequence: finishing on the edge to final machining allowance on the base, then finishing on the base. If one of the final machining allowances is equal to zero, this part of the finishing process is skipped.

- Finishing the edge  
While finishing on the edge, the pocket is only machined 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 available, equals the difference between the corner radius and the cutter 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 + final machining allowance + safety distance. From this point onward, the tool is always fed in **vertically** at the depth infeed feedrate (since a tool with a front cutting edge is used for base finishing). The base surface of the pocket is machined once.



#### Explanation of the parameters



See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters `_RTP`, `_RFP` and `_SDIS`.

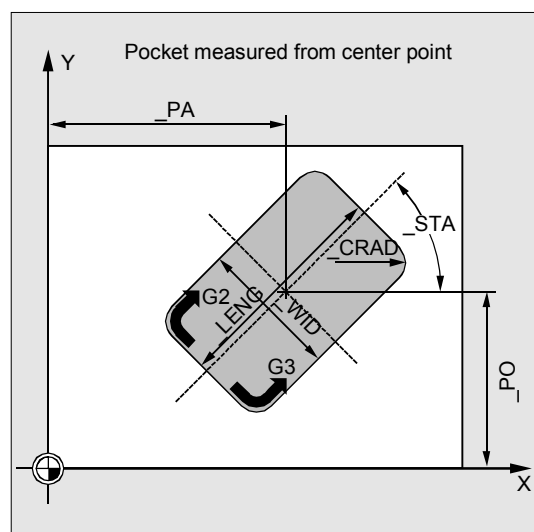
See Section 3.7 for a description of parameter `_DP`.

See Section 3.2 for cycle setting data `_ZSD[1]`, `_ZSD[2]`.

#### `_LENG`, `_WID` and `_CRAD` (pocket length, pocket width and corner radius)

The shape of a pocket in the plane is determined with parameters `_LENG`, `_WID` and `_CRAD`.

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 it is not possible to traverse to the programmed corner radius with the active tool because its radius is larger, the corner radius of the completed pocket corresponds to the tool radius.

If the cutter radius is greater than half the length or width of the pocket, the cycle is aborted after outputting alarm

61105 "Cutter radius too large".

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

The reference point of the pocket in the abscissa and ordinate is defined with parameters \_PA and \_PO.

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.

#### **\_MID (infeed depth)**

With this parameter, you determine the maximum infeed depth when roughing.

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

The cycle automatically calculates this infeed using \_MID and the total depth. The minimum possible number of infeed steps is used as the basis.

\_MID=0 means that the cut to pocket depth is made with one infeed.

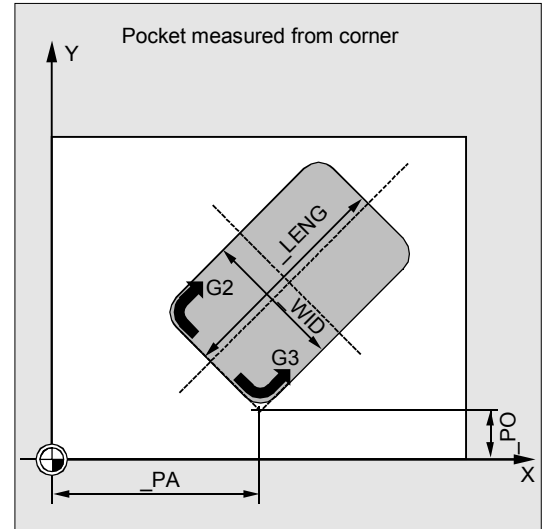
#### **\_FAL (final machining allowance at the edge)**

The final machining allowance only affects machining of the pocket in the plane at the edge.

When the final machining allowance  $\geq$  tool diameter, the pocket will not necessarily be machined completely.

The following message appears:

"Caution: Final machining allowance  $\geq$  tool diameter"  
is output but the cycle is continued.



### 3.9 Milling rectangular pockets - POCKET3

#### **\_FALD (final machining allowance on the base)**

For roughing, a separate final machining allowance is considered on the base (POCKET1 does not normally consider any final machining allowance on the base).

#### **\_FFD and \_FFP1 (feedrate depth and plane)**

Feedrate `_FFD` is used for insertion into the material. Feedrate `_FFP1` is used for all movements in the plane traversed at feedrate when machining.

#### **\_CDIR (mill direction)**

The value for the machining direction of the pocket is defined in this parameter.

Using parameter `_CDIR`, the mill direction can be programmed as

- Directly "2 for G2" and "3 for G3" or
- Alternatively "up-cut milling" or "down-cut milling"

Up-cut milling or down-cut milling is determined within the cycle via the spindle direction activated prior to the cycle call.

<b>Down-cut</b>	<b>Up-cut</b>
M3 → G3	M3 → G2
M4 → G2	M4 → G3

#### **\_VARI (machining type)**

You can define the type of machining with parameter `_VARI`.

Possible values are:

##### **Units digit:**

- 1=Roughing
- 2=Finishing

##### **Tens digit (infeed):**

- 0=Perpendicular to the pocket center with G0
- 1=Perpendicular to the pocket center with G1
- 2=Along a helical path
- 3=Oscillating along the pocket longitudinal axis

If another value has been programmed for parameter `_VARI`, the cycle is aborted after alarm 61002 "Machining type incorrectly defined" is output.

#### **\_MIDA (max. infeed width)**

With this parameter, you define the maximum infeed width for solid machining in the plane. In the same way as the known calculation of the infeed depth (equal distribution of the overall depth using the largest possible value), the width is evenly divided, using the value programmed in `_MIDA` as a maximum value.

If this parameter is not programmed, or if its value is 0, the cycle uses 80% of the milling cutter diameter as the maximum infeed width.



### Other information

Applies if the width infeed determined from edge machining is recalculated on reaching the full pocket depth; otherwise, the width infeed calculated at the start is retained for the full 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)**

With the parameter \_RAD1, you define the radius of the helical path (i.e., the tool center point path) or the maximum insertion angle for reciprocating movement.

#### **\_DP1 (insertion depth)**

With the parameter \_DP1, you define the infeed depth for insertion on the helical path.

### 3.9 Milling rectangular pockets - POCKET3

#### Other information

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with alarm 61000 "No tool offset active".

A new current workpiece coordinate system that influences the actual value display is used in the cycle. The zero point of this coordinate system lies on the pocket center point.

The original coordinate system becomes active again after the end of the cycle.

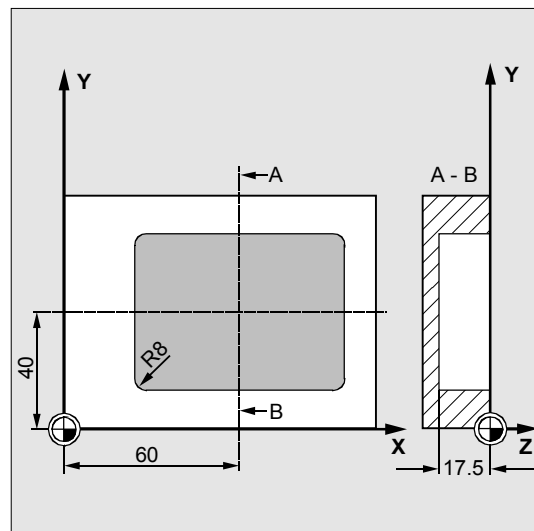
#### Programming example

##### Pocket

With this program, you can machine a pocket that is 60 mm long, 40 mm wide, 17.5 mm deep in the XY plane, and which has a corner radius of 8 mm. The angle in relation to the X axis is 0 degrees. The final machining allowance of the pocket edges is 0.75 mm, 0.2 mm at the base, 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.

Down-cut milling uses the spindle rotation direction as the direction of machining.

Only roughing is to be performed.



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, -> -> 40, 8, 60, 40, 0, 4, 0.75, 0.2, -> -> 1000, 750, 0, 11, 5)	;Cycle call
N40 M30	;End of program

-> Must be programmed in a single block



### 3.10 Milling circular pockets - POCKET4



The cycle POCKET4 is available with SW 4 and higher.



#### Programming

POCKET4 (\_RTP, \_RFP, \_SDIS, \_DP, \_PRAD, \_PA, \_PO, \_MID, \_FAL, \_FALD, \_FFP1, \_FFD, \_CDIR, \_VARI, \_MIDA, \_AP1, \_AD, \_RAD1, \_DP1)



#### Parameters

The following input parameters are always required:

_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety distance (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 (enter without sign)
_FAL	real	Final machining allowance on 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	Mill direction: (enter without sign) Values: 0...Down-cut milling (as spindle rotation) 1...Up-cut milling 2...With G2 (independent of spindle direction) 3...With G3
_VARI	integer	Type of machining: (enter without sign) UNITS DIGIT: Values: 1...Roughing 2...Finishing TENS DIGIT: Values: 0...Perpendicular to the pocket center with G0 1...Perpendicular to the pocket center with G1 2...Along a helical path

The other parameters can be selected as options. They define the insertion strategy and overlapping for solid machining: (enter without sign)

_MIDA	real	Maximum infeed width value during solid machining in the plane
_AP1	real	Basic size pocket radius
_AD	real	Basic pocket depth from reference plane
_RAD1	real	Radius of the helical path during insertion (related to the tool center point path)
_DP1	real	Insertion depth per 360° revolution on insertion along helical path

### 3.10 Milling circular pockets - POCKET4



#### Function

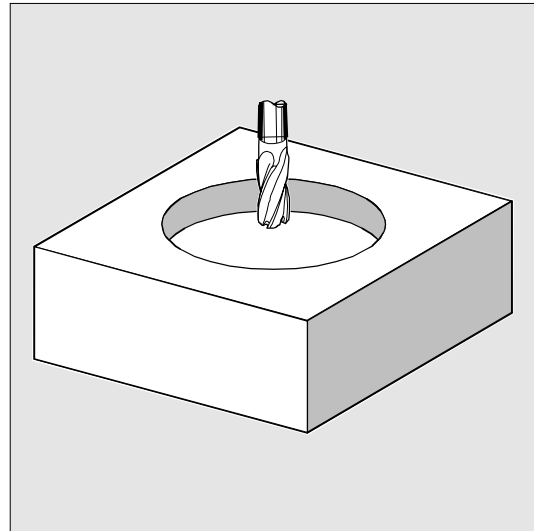
With this cycle, you can machine circular pockets in the machining plane.

A face cutter is required for finishing.

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 mill direction can be defined by a G instruction (G2/G3), up-cut milling or down-cut milling from the spindle direction.
- For solid machining, the maximum infeed width in the plane can be programmed.
- Final machining allowance for the pocket base
- Two different insertion strategies:
  - Vertically from the pocket center point
  - Along a helical path around the pocket center
- Short approach paths in the plane for finishing.
- Consideration of a blank contour in the plane and a basic size at the base (optimum processing of pre-formed pockets possible).
- `_MIDA` is recalculated when machining the edge.



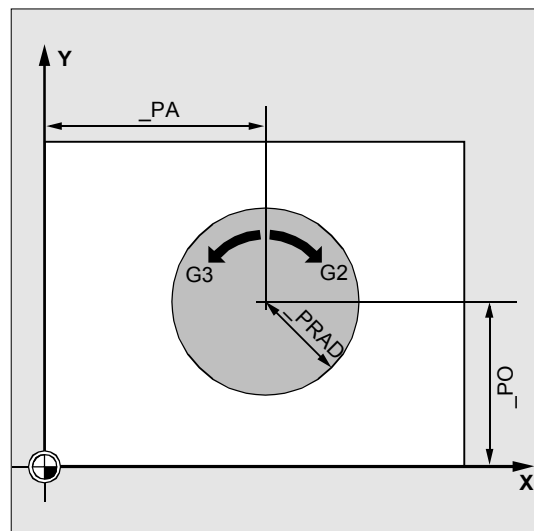
#### Sequence

##### Position reached before the beginning of the cycle:

The starting position can be any position, from which the center point of the pocket at the retraction plane level can be approached without collision.

##### Motion sequence when roughing (VARI=X1):

With G0, the pocket center point is approached at the retraction plane level and then, from this position, the reference plane brought forward by the safety distance is approached with G0. Pocket machining is then performed according to the selected insertion strategy and in consideration of the programmed base size.



**Insertion strategies:**

See Section 3.9 (POCKET3).

**Accounting for blank dimensions**

During solid machining, it is possible to take blank dimensions (for example, in the machining of precast workpieces) into account.

For circular pockets, the basic size `_AP1` at the edge is also circular (with a smaller radius than the pocket radius).



For additional explanations, see Section 3.9 (POCKET3).

**Motion sequence when finishing (VARI=X2):**

Finishing is performed in the following sequence: finishing on the edge to final machining allowance on the base, then finishing on the base. If one of the final machining allowances is equal to zero, this part of the finishing process is skipped.

- Finishing the edge  
While finishing on the edge, the pocket is only machined once.  
For finishing on the edge, the path includes one quadrant reaching the pocket radius. The radius of this path is less than or equal to 2 mm or, if "less space" is available, equals the difference between the pocket radius and the cutter 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 + final machining allowance + safety distance. From this point onward, the tool is always fed in **vertically** at the depth infeed feedrate (since a tool with a front cutting edge is used for base finishing).  
The base surface of the pocket is machined once.



### Explanation of the parameters

See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters `_RTP`, `_RFP` and `_SDIS`.

See Section 3.7 (POCKET1) for a description of parameter `_DP`.

See Section 3.9 (POCKET3) for a description of parameters `_MID`, `_FAL`, `_FALD`, `_FFP1`, `_FFD`, `_CDIR`, `_MIDA`, `_AP1`, `_AD`, `_RAD1`, and `_DP1`.  
See Section 3.2 for cycle setting data `_ZSD[1]`.

#### `_PRAD` (pocket radius)

The shape of the circular pocket is determined by the radius only.

If the radius is less than the tool radius of the active tool, the cycle is aborted after alarm 61105 "Cutter radius too large" is output.

#### `_PA`, `_PO` (pocket center point)

With parameters `_PA` and `_PO`, you define the center point of the pocket. Circular pockets are always measured from the center.

#### `_VARI` (machining type)

You can define the type of machining with parameter `_VARI`.

Possible values are:

##### Units digit:

- 1=Roughing
- 2=Finishing

##### Tens digit (infeed):

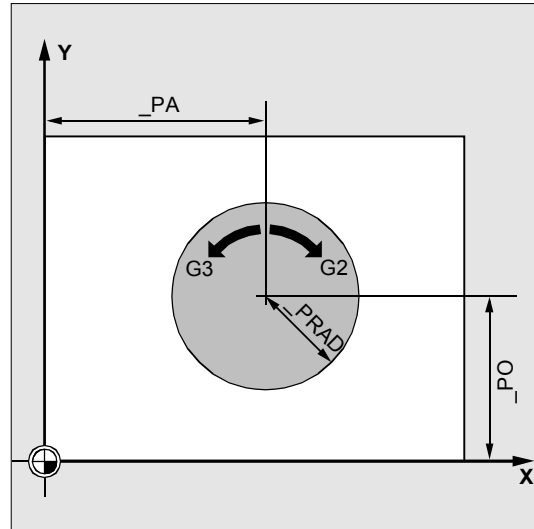
- 0=Perpendicular to the pocket center with G0
- 1=Perpendicular to the pocket center with G1
- 2=Along a helical path

If another value has been programmed for parameter `_VARI`, the cycle is aborted after alarm 61002 "Machining type incorrectly defined" is output.

#### Edge-only roughing with POCKET4

POCKET4 (circular pocket) can now machine only the edge when roughing.

Define the rough dimension in the depth (parameter `_AD`) so that it is at least the same size as the pocket depth (`DP`) minus the final machining allowance depth (`_FALD`).



Example:

Requirement: Depth calculation without including safety distance ( $\_ZSD[1]=1$ )

$\_RTP=0$  Reference plane  
 $\_SDIS=2$  Safety distance  
 $\_DP=-21$  Pocket depth  
 $\_FALD=1.25$  Depth allowance  
 $\rightarrow \_AD \geq 19.75$  Rough dimension for depth  
 must be greater than or equal to pocket depth incremental minus allowance  
 depth, i.e.,  $21 - 1.25 = 19.75$

### Other information

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with Alarm 61000 "No tool offset active".

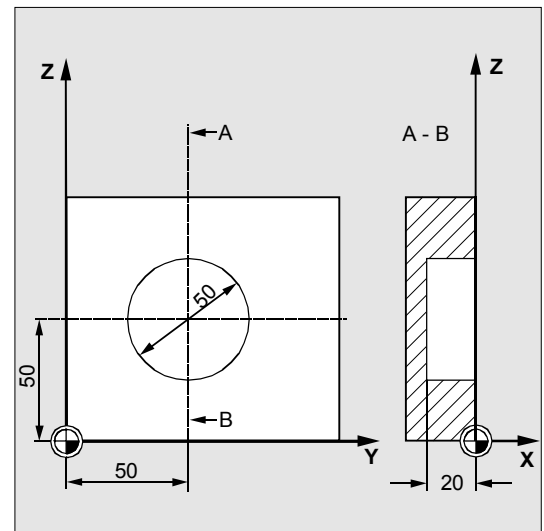
A new current workpiece coordinate system that influences the actual value display is used in the cycle. The zero point of this coordinate system lies on the pocket center point.

The original coordinate system becomes active again after the end of the cycle.

### Programming example

#### Circular pocket

With this program, you can machine a circular pocket in the YZ plane. The center point is defined by Y50 Z50. The infeed axis for the depth infeed is the X axis. Neither a final machining allowance nor a safety distance is defined. The pocket will be machined using up-cut milling. Infeed occurs 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 ;End of program
```

-> Must be programmed in a single block

### 3.11 Face milling - CYCLE71



The cycle CYCLE71 is available with SW 4 and higher.



#### Programming

CYCLE71 (\_RTP, \_RFP, \_SDIS, \_DP, \_PA, \_PO, \_LENG, \_WID, \_STA, \_MID, \_MIDA, \_FDP, \_FALD, \_FFP1, \_VARI, \_FDP1)



#### Parameters

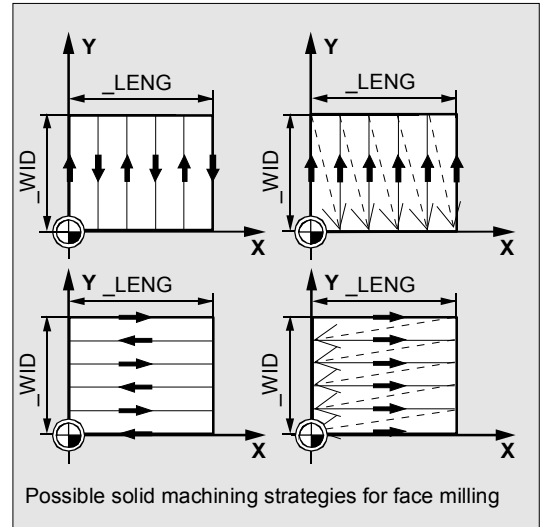
The following input parameters are always required:

_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety distance (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	Rectangle length along the 1st axis, incremental. The corner, from which dimensions are measured, is given by the sign.
_WID	real	Rectangle length along the 2nd axis, incremental. The corner, from which dimensions are measured, is given by the sign.
_STA	real	Angle between the longitudinal axis of the rectangle and the first axis of the plane (abscissa, enter without sign); Value range: $0^\circ \leq \_STA < 180^\circ$
_MID	real	Maximum infeed depth (enter without sign)
_MIDA	real	Maximum infeed width value for solid 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	Type of machining: (enter without sign) UNITS DIGIT: Values: 1...Roughing 2...Finishing TENS DIGIT: Values: 1...Parallel to the abscissa, in one direction 2...Parallel to the ordinate, in one direction 3...Parallel to the abscissa, in alternate directions 4...Parallel to the ordinate, in alternate directions
_FDP1	real	Overrun travel in direction of plane infeed (incremental, enter without sign)



### Function

With cycle CYCLE71, you can 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). Maximum infeed can be defined in width and depth. The cycle operates without cutter radius compensation. The depth infeed is programmed in the open.



### Sequence

#### Position reached before the beginning of the cycle:

The starting position can be any position, from which the infeed point at the retraction plane level can be reached without collision.

#### The cycle implements the following motion sequence:

- 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, infeed to machining plane. G0 is possible, since infeed occurs in the open. There are several roughing strategies (paraxial in one direction or back and forth).
- Motion sequence when roughing (VARI=X1): Face milling is possible on several planes according to the programmed values `_DP`, `_MID` and `_FALD`. Machining is performed in the downward direction, i.e., by removing stock on one plane at a time, and then executing the next depth infeed in open space (parameter `_FDP`). The traversing paths for solid machining in the plane are determined by the settings in parameters `_LENG`, `_WID`, `_MIDA`, `_FDP`, `_FDP1` and the cutter radius of the active tool. The first path to be milled is always selected so that the infeed width is exactly `_MIDA`, such that no width exceeds the maximum possible width infeed.

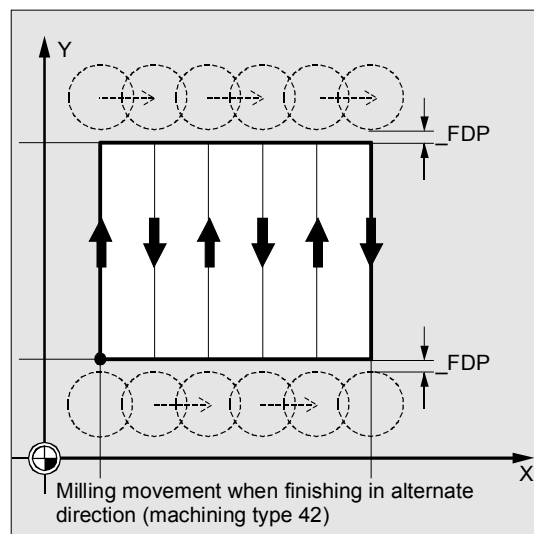
### 3.11 Face milling - CYCLE71

Thus, the tool center point does not always travel exactly to the edge (only if  $\_MIDA =$  cutter radius). The dimension, by which the tool traverses outside the edge always equals

cutter diameter -  $\_MIDA$ ,

even when only 1 surface cut is performed, i.e., surface width + overrun is less than  $\_MIDA$ . The other paths for width infeed are calculated internally so as to produce a uniform path width ( $\leq \_MIDA$ ).

- Motion sequence when finishing ( $VARI=X2$ ):  
When finishing, the surface is milled once in the plane. The final machining allowance for roughing must also be selected so that the remaining depth can be machined in one pass with the finishing tool.  
After each surface milling pass in the plane, the tool retracts completely. The retraction travel is programmed by 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 infeed depth is performed towards the same point as for roughing.  
After finishing has been completed, the tool retracts from the last position reached to the retraction plane  $\_RTP$ .



#### Explanation of the parameters

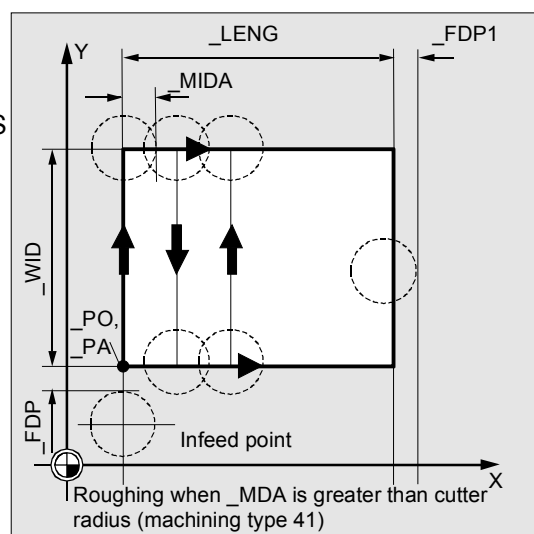
See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters  $\_RTP$ ,  $\_RFP$  and  $\_SDIS$   
See Section 3.9 (POCKET3) for a description of parameters  $\_STA$ ,  $\_MID$  and  $\_FFP1$ .  
See Section 3.2 for cycle setting data  $\_ZSD[1]$ .

#### $\_DP$ (depth)

The depth can be defined as an absolute value ( $\_DP$ ) in relation to the reference plane.

#### $\_PA$ , $\_PO$ (starting point)

With parameters  $\_PA$  and  $\_PO$ , you define the starting point of the surface in the abscissa and ordinate.





**\_LENG, \_WID (length)**

With parameters `_LENG` and `_WID`, you determine the length and width of the rectangle in the plane. The sign determines the position of the rectangle relative to `_PA` and `_PO`.

**\_MIDA (max. infeed width)**

With this parameter, you define the maximum infeed width for solid machining in the plane. In the same way as the known calculation of the infeed depth (equal distribution of the overall depth using the largest possible value), the width is evenly divided, using the value programmed in `_MIDA` as a maximum value. If this parameter is not programmed, or if its value is 0, the cycle uses 80% of the cutter diameter as the maximum infeed width.

**\_FDP (retraction travel)**

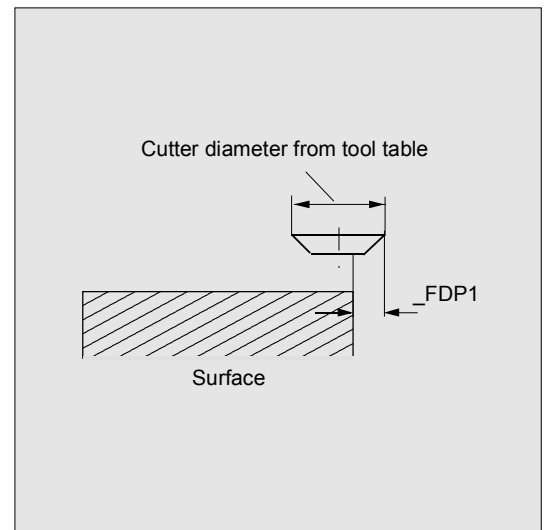
This parameter defines the dimension for retraction travel in the plane. This parameter must be programmed with a value greater than zero.

**\_FDP1 (overrun travel)**

This parameter can be used to define an overrun travel in the direction of the plane infeed (`_MIDA`), which enables the deviation between the current cutter radius and the cutting edge (e.g., tool nose radius or inclined cutting tips) to be compensated for. Therefore, the last cutter center point path always corresponds to `_LENG` (or `_WID`) + `_FDP1` - tool radius (from correction table).

**\_FALD (final machining allowance)**

During roughing, the depth final machining allowance used is defined by this parameter. The residual material designated as the final machining allowance must always be specified for finish cutting so as to ensure that the tool can be lifted and inserted at the starting point of the next cut without the risk of collision.



### 3.11 Face milling - CYCLE71

#### **\_VARI (machining type)**

You can define the type of machining with parameter `_VARI`.

Possible values are:

#### **Units digit:**

- 1=Roughing to final machining allowance
- 2=Finishing

#### **Tens digit:**

- 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 another value has been programmed for parameter `_VARI`, the cycle is aborted after alarm 61002 "Machining type incorrectly defined" is output.

#### **Other information**

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with alarm 61000 "No tool offset active".



## Programming example

### Face milling

#### Parameters for cycle call:

- Retraction plane: 10 mm
- Reference plane: 0 mm
- Safety distance: 2 mm
- Milling depth: -11 mm
- Max. infeed depth 6 mm
- No final machining allowance -
- Starting point of the rectangle X = 100 mm  
Y = 100 mm
- Rectangle dimensions X = +60 mm  
Y = +40 mm
- Angle of rotation in the plane 10 degrees
- Max. infeed width 10 mm
- Retract travel at end of milling path: 5 mm
- Feedrate for surface machining 4000 mm/min.
- Type of machining: roughing parallel to the X axis with alternating direction
- Overrun on last cut as determined by the cutting edge geometry 2 mm

```

%_N_TSTCYC71_MPF ;Program for face milling with CYCLE71
; $PATH=/_N_MPF_DIR
;*
$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 ;End of program
-> Must be programmed in a single block

```

### 3.12 Path milling - CYCLE72



The cycle CYCLE72 is available with SW 4 and higher (not for FM-NC).



#### Programming

CYCLE72 (\_KNAME, \_RTP, \_RFP, \_SDIS, \_DP, \_MID, \_FAL, \_FALD, \_FFP1, \_FFD, \_VARI, \_RL, \_AS1, \_LP1, \_FF3, \_AS2, \_LP2)



#### Parameters

The following input parameters are always required:

_KNAME	string	Name of the contour subroutine
_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety distance (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	Final machining allowance at the edge contour (enter without sign)
_FALD	real	Final machining 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	Type of machining: (enter without sign) UNITS DIGIT: Values: 1...Roughing 2...Finishing TENS DIGIT: Values: 0...Intermediate paths with G0 1...Intermediate paths with G1 HUNDREDS DIGIT: Values: 0...Retraction at end of contour to _RTP 1...Retraction at end of contour to _RFP + _SDIS 2...Retraction by _SDIS at end of contour 3...No retraction at end of contour
_RL	integer	Contouring is centric, on right or left (with G40, G41 or G42, enter without sign) Values: 40...G40 (approach and return, straight line only) 41...G41 42...G42

---

_AS1	integer	Specification of approach direction/path: (enter without sign) UNITS DIGIT: Values: 1...Straight tangential line 2...Quadrant 3...Semicircle TENS DIGIT: Values: 0...Approach to the contour in the plane 1...Approach to the contour in a spatial path
------	---------	--

---

_LP1	real	Length of the approach travel (along a straight line) or radius of the approach arc (along a circle) (enter without sign)
------	------	---

---

The other parameter can be preset optionally  
(enter without sign).

---

_FF3	real	Return feedrate and feedrate for intermediate positioning in the plane (when retracting)
------	------	--

---

_AS2	integer	Specification of return direction/path: (enter without sign) UNITS DIGIT: Values: 1...Straight tangential line 2...Quadrant 3...Semicircle TENS DIGIT: Values: 0...Retraction from the contour in the plane 1...Retraction from the contour in a spatial path
------	---------	--

---

_LP2	real	Length of the return travel (along a straight line) or radius of the return arc (along a circle) (enter without sign)
------	------	---

---

## 3.12 Path milling - CYCLE72



### Function

With the cycle CYCLE72, it is possible to mill along any contour defined in a subroutine. 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 of milling and must be in the same plane.** In addition, it must consist of at least 2 contour blocks (start and end points), since the contour subroutine is invoked within the cycle.

### Cycle functions:

- Selection of roughing (single-pass parallel to the contour, in consideration of a final machining allowance at several depths, if necessary, up to the final machining allowance) and finishing (single-pass of final contour at several depths, if necessary)
- Flexible approach and retraction to/from the contour either tangentially or radially (quadrant or semicircle)
- Programmable depth infeeds
- Intermediate motions either with rapid traverse or at feedrate.



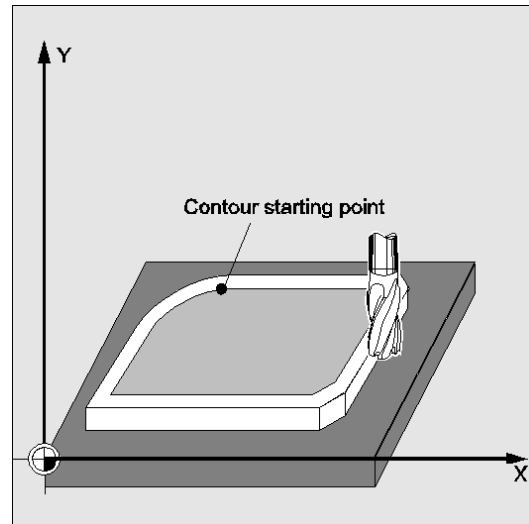
The requirement for executing a cycle is NC SW 4.3 or higher, including the "Flexible approach and return" function.



### Sequence

#### Position reached before the beginning of the cycle:

The starting position can be any position, from which the start of the contour at the retraction plane level can be reached without collision.



**The cycle creates the following motion sequence when roughing (VARI=XX1):**

The depth infeeds are divided evenly using the highest possible value according to the preset parameters.

- Travel to starting point for initial cut with G0/G1 (and \_FF3). This point is calculated in the control and depends on
  - The contour starting point (first point in subroutine),
  - The direction of the contour at the starting point,
  - The approach mode and corresponding 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 processing depth results from
  - The overall depth,
  - The final machining allowance and
  - The maximum possible depth infeed.
- Vertically approach the contour with depth feedrate and then in the plane with the feedrate programmed for surface machining or 3D with the feedrate programmed in \_FAD, in accordance with flexible approach programming.
- Milling along the contour with G40/G41/G42.
- Flexible retraction from the contour with G1, still with the feedrate for surface machining by a lift calculated within the cycle.
- Retraction with G0 /G1 (and feedrate for intermediate travel \_FF3) depending on program.
- Return to depth infeed point with G0/G1 (and \_FF3).
- This operating sequence is repeated on the next machining plane, until the final machining allowance is reached in depth.

When roughing is over, the tool is positioned at the contour retraction point (calculated within the control unit) at the retraction plane level.

### 3.12 Path milling - CYCLE72

#### **The cycle creates the following motion sequence when finishing (VARI=XX2):**

During finishing, milling is performed at the relevant infeed along the base of the contour until the final dimension is reached.

Approaching and retracting from the contour is performed in a flexible way according to the corresponding preset parameters. The corresponding path is calculated within the control unit.

At the end of the cycle, the tool is positioned at the contour retraction point at the retraction plane level.

#### **Contour programming**

For programming the contour, please note 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 and deselected from the upper level cycle; no G40, G41, G42 is then programmed in the contour subroutine.





### Explanation of the parameters



See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters `_RTP`, `_RFP` and `_SDIS`.



See Section 3.9 for a description of parameters `_MID`, `_FAL`, `_FALD`, `_FFP1` and `_FFD`. See Section 3.11 for a description of parameter `_DP`.

See Section 3.2 for cycle setting data `_ZSD[1]`.

#### `_KNAME` (name)

The contour to be milled is completely programmed in a subroutine. With parameter `_KNAME`, you can define 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. In this case, the program name and labels/block number are identified by an ":".

Examples:

`_KNAME="CONTOUR_1"`

The milling contour is the complete program "Contour\_1".

`_KNAME="START:END"`

The milling contour is defined as the section starting from the block labeled START to the block labeled END in the calling routine.

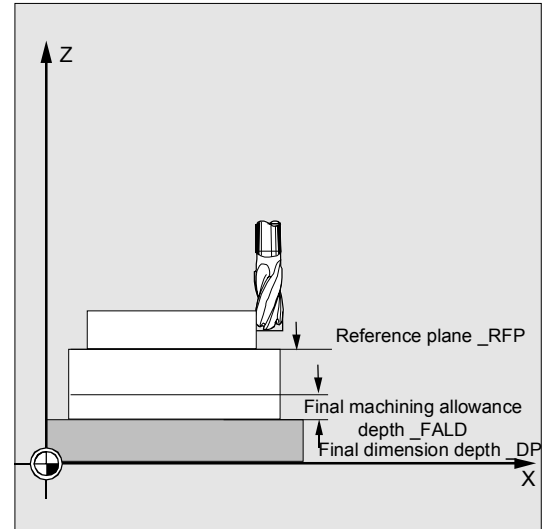
`_KNAME=`

`"/_N_SPF_DIR/_N_CONTOUR_1_SPF:N130:N210"`

The milling contour is defined in blocks N130 to N210 in the program CONTOUR\_1. The program name must be entered complete with path and extension, see description of call in References: /PGA/ Programming Guide Advanced



If the section is defined by block numbers, it must be noted that these block numbers must be adjusted for the section in `_KNAME` if the program is modified and subsequently renumbered.



### 3.12 Path milling - CYCLE72

#### **\_VARI (machining type)**

You can define the type of machining with parameter `_VARI`. For possible values, see "Parameter CYCLE72".

**If another value has been programmed for parameter `_VARI`, the cycle is aborted after alarm 61002 "Machining type incorrectly defined" is output.**

#### **\_RL (travel around the contour)**

Parameter `_RL` is set to define how the tool must travel around the contour, i.e., along the center path or on the left or right-hand side with G40, G41 or G42. For possible values, see "Parameter CYCLE72".

#### **\_AS1, \_AS2 (direction of approach/approach path, direction of return/return path)**

You can specify the approach path with parameter `_AS1` and the return path with `_AS2`. For possible values, see "Parameter CYCLE72". If `_AS2` is not programmed, then the behavior programmed for the approach path will also apply to the return path. The flexible approach of the contour along a 3-D path (helix or straight line) should be programmed only if the tool is suitable and not yet engaged.



With center path travel (G40), the tool must approach and return along a straight line.

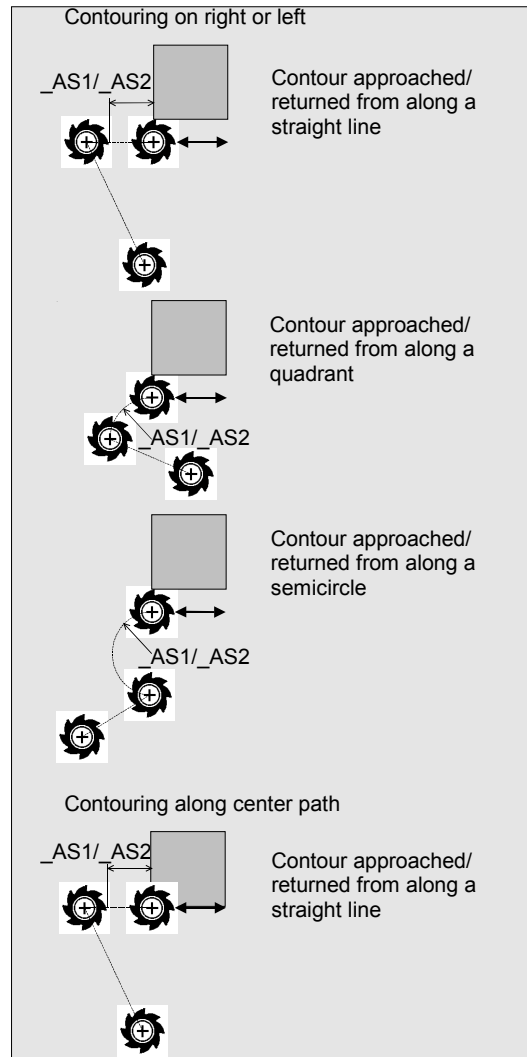
#### **\_LP1, \_LP2 (length, radius)**

Parameter `_LP1` is set to program the approach path or approach radius (distance between tool outer edge and contour starting point) and `_LP2` to program the return path or return radius (distance between tool outer edge and end point of contour).

Parameters `_LP1`, `_LP2` must be set to  $>0$ . A setting of zero generates error message 61116 "Approach or return path=0."



When G40 is programmed, the approach or return path corresponds to the distance between the tool center point and the starting or end point of the contour.



**\_FF3 (retraction feedrate)**

Parameter \_FF3 is used to define a retraction feedrate for intermediate positioning in the plane (in the open) when intermediate motions are to be performed with feedrate (G01). If no feedrate is programmed, the intermediate motions are carried out with the surface feedrate for G01.

**Other information**

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with alarm 61000 "No tool offset active".

## 3.12 Path milling - CYCLE72



### Programming example 1

#### Externally milling around a closed contour

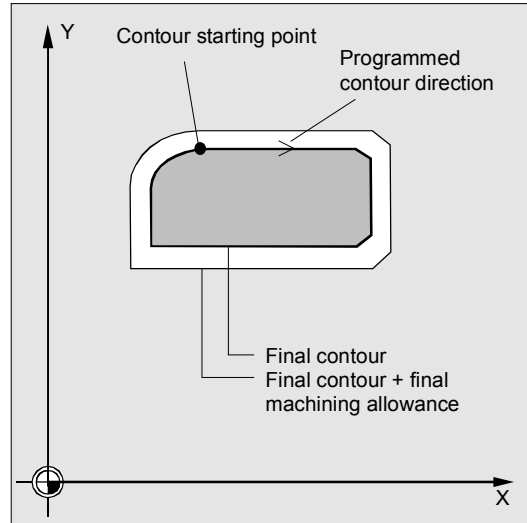
This program is used to mill a contour as shown in the figure.

#### Parameters for cycle call:

- Retraction plane 250 mm
- Reference plane 200
- Safety distance 3 mm
- Depth 175 mm
- Maximum depth infeed 10 mm
- Finishing allowance in depth 1.5 mm
- Feedrate depth infeed 400 mm/min
- Finishing allowance in plane 1 mm
- Feedrate in the plane 800 mm/min
- Machining:           Roughing up to the final machining allowance, Intermediate paths with G1, for intermediate paths Retraction in Z to `_RFP + _SDIS`

#### Parameters for the approach:

- G41 - to the left of the contour, i.e., external machining
- Approach and return on quadrant in plane 20 mm radius
- Retraction feedrate 1000 mm/min



```

%_N_EDGECONTOUR1_MPF           ;Program for milling around a contour
; $PATH=/_N_MPF_DIR             ;with CYCLE72
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                           ;End of program

```

-> Must be programmed in a single block

```

%_N_MYCONTOUR_SPF ;Subroutine for contour milling (for example)
; $PATH=/_N_SPF_DIR
N100 G1 G90 X150 Y160 ;Start point of contour
N110 X230 CHF=10
N120 Y80 CHF=10
N130 X125
N140 Y135
N150 G2 X150 Y160 CR=25
N160 M17

```



### Programming example 2

Milling round the outside of a closed contour as described in sample program 1, with the contour defined 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 starting 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

```

### 3.13 Milling rectangular spigots - CYCLE76



#### Programming

CYCLE76 (\_RTP, \_RFP, \_SDIS, \_DP, \_DPR, \_LENG, \_WID, \_CRAD, \_PA, \_PO, \_STA, \_MID, \_FAL, \_FALD, \_FFP1, \_FFD, \_CDIR, \_VARI, \_AP1, \_AP2)



#### Parameters

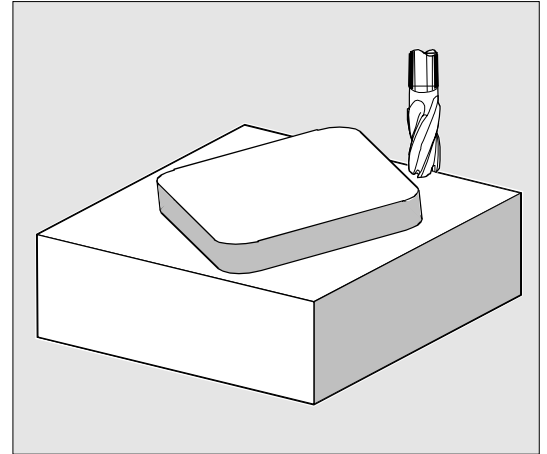
The following input parameters are always required:

_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety distance (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	Spigot reference point, abscissa (absolute)
_PO	real	Spigot reference point, ordinate (absolute)
_STA	real	Angle between longitudinal axis and 1st axis of plane
_MID	real	Maximum depth infeed (incremental, enter without sign)
_FAL	real	Final machining allowance on edge contour (incremental)
_FALD	real	Final machining allowance at the base (incremental, enter without sign)
_FFP1	real	Feedrate on contour
_FFD	real	Feedrate for depth infeed
_CDIR	integer	Mill direction: (enter without sign) Values: 0...Down-cut milling 1...Up-cut milling 2...With G2 (independent of spindle direction) 3...With G3
_VARI	integer	Type of machining: Values: 1...Roughing to final machining allowance 2...Finishing (allowance X/Y/Z=0)
_AP1	real	Length of blank spigot
_AP2	real	Width of blank spigot



### Function

With this cycle, you can machine rectangular spigots in the machining plane. A face cutter is required for finishing. Depth infeed is always performed in the position reached prior to semicircular positioning on the contour.



### Sequence

#### Position reached before the beginning of the cycle:

The starting point is a position in the positive range of the abscissa with integrated approach semicircle and consideration of the programmed, abscissa-related blank dimension.

#### Motion sequence for roughing ( $\_VARI=1$ )

##### *Approach and return from contour:*

The retraction plane ( $\_RTP$ ) is approached in rapid traverse in order that the tool can be positioned at the starting point in the machining plane at this level. The starting point is defined as 0 degrees in relation to the abscissa.

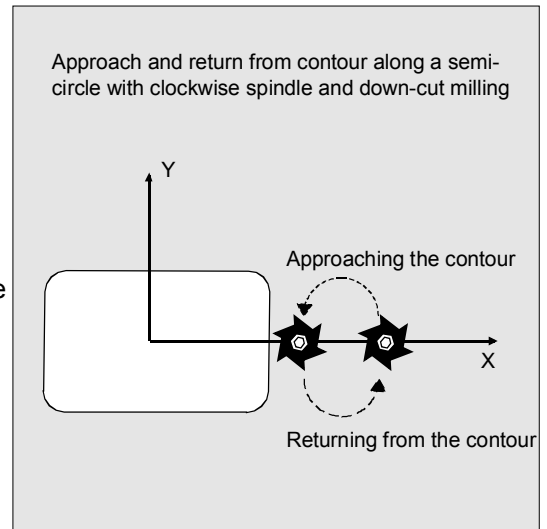
The tool is fed in at rapid traverse to the safety distance ( $\_SDIS$ ) and then traverses to machining depth at normal feedrate.

The tool approaches the spigot contour along a semicircular path.

The mill direction can be defined as up-cut or down-cut milling in relation to the spindle direction.

If the spigot has been circumnavigated once, the tool lifts off the contour in the plane along a semicircular path and is then fed into the next machining depth.

The contour is then reapproached along a semicircle and the spigot is circumnavigated once. The process is repeated until the programmed spigot depth has been reached. The tool then approaches the retraction plane ( $\_RTP$ ) in rapid traverse.



### 3.13 Milling rectangular spigots - CYCLE76

#### Depth infeed:

- Infeed to safety distance
- Insertion to machining depth

The first machining depth is the product of:

- The total depth,
- The final machining allowance and
- The maximum possible depth infeed.

#### Motion sequence for finishing (**\_VARI=2**)

Depending on the setting of parameters **\_FAL** and **\_FALD**, a finishing operation is performed on the contour surface or the base or both. The approach strategy matches the motions in the plane executed for roughing operations.



#### Explanation of the parameters

See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters **RTP**, **RFP**, **SDIS**, **DP**, and **DPR**.

See Section 3.9 for a description of parameters **\_MID**, **\_FAL**, **\_FALD**, **\_FFP1**, and **\_FFD**.

See Section 3.2 for cycle setting data **\_ZSD[2]**.

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

The shape of a spigot in the plane is determined with parameters **\_LENG**, **\_WID** and **\_CRAD**.

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 abscissa (with a plane angle of zero degrees).

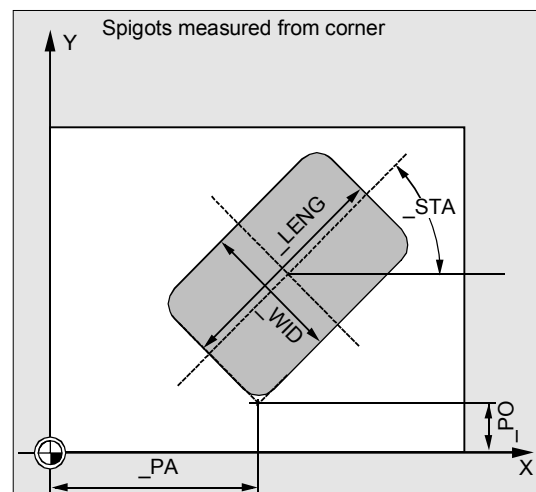
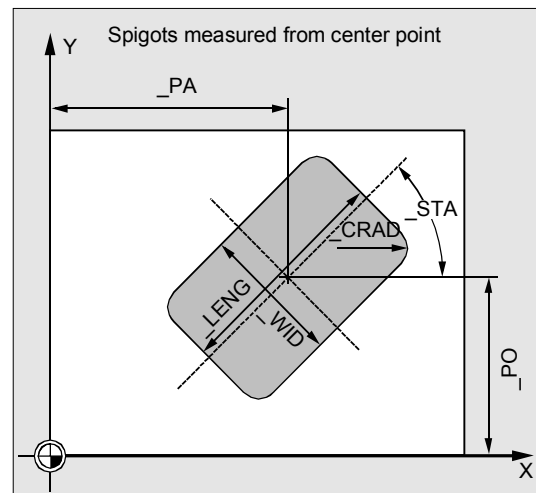
#### **\_PA**, **\_PO** (reference point)

Parameters **\_PA** and **\_PO** are set to define the reference point of the spigot in abscissa and ordinate.

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.

**\_CDIR (mill direction)**

The machining direction of the spigot is defined in this parameter.

Using parameter \_CDIR, the mill direction can be programmed as

- Directly "2 for G2" and "3 for G3" or
- Alternatively "up-cut milling" or "down-cut milling"

Up-cut milling or down-cut milling is determined within the cycle via the spindle direction activated prior to the cycle call.

<b>Down-cut</b>	<b>Up-cut</b>
M3 → G3	M3 → G2
M4 → G2	M4 → G3

**\_VARI (machining type)**

You can define the type of machining with parameter \_VARI.

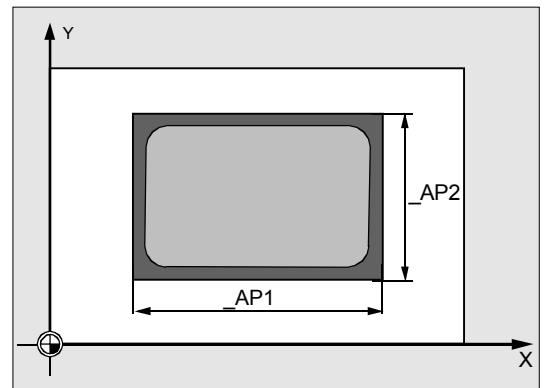
Possible values are:

- 1=Roughing
- 2=Finishing

**\_AP1, \_AP2 (blank dimensions)**

Blank dimensions (e.g., in the case of precast workpieces) can be taken into account when machining the spigot.

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 is dependent on this dimension.

**Other information**

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with alarm 61009 "Active tool number=0".

A new current workpiece coordinate system that influences the actual value display is used in the cycle.

The zero point of this coordinate system lies on the pocket center point.

The original coordinate system becomes active again after the end of the cycle.

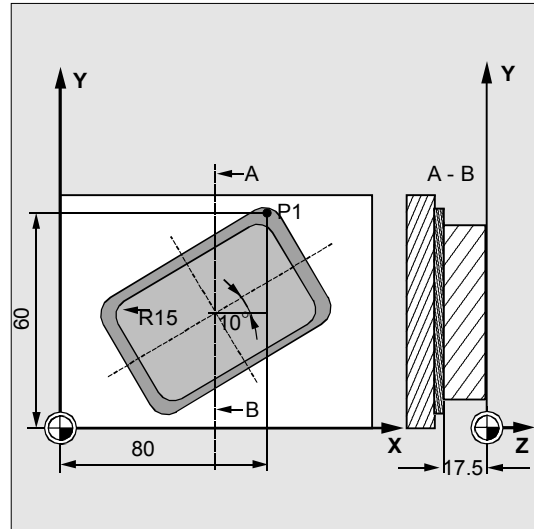
### 3.13 Milling rectangular spigots - CYCLE76



#### Programming example

##### Spigots

This program allows you to machine a spigot that is 60 mm long, 40 mm wide and with a corner radius of 15 mm in the XY plane. The spigot has an angle of 10 degrees in relation to the X axis and is programmed from a corner point P1. When a spigot is dimensioned with reference to corners, the length and width must be entered with a sign in order to define a unique position for the spigot. The spigot is premachined with a length allowance of 80 mm and a width allowance of 50 mm.



```

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 ;End of program

```

-> Must be programmed in a single block

### 3.14 Milling circular spigots - CYCLE77



#### Programming

CYCLE77 (\_RTP, \_RFP, \_SDIS, \_DP, \_DPR, \_PRAD, \_PA, \_PO, \_MID, \_FAL, \_FALD, \_FFP1, \_FFD, \_CDIR, \_VARI, \_AP1)



#### Parameters

The following input parameters are always required:

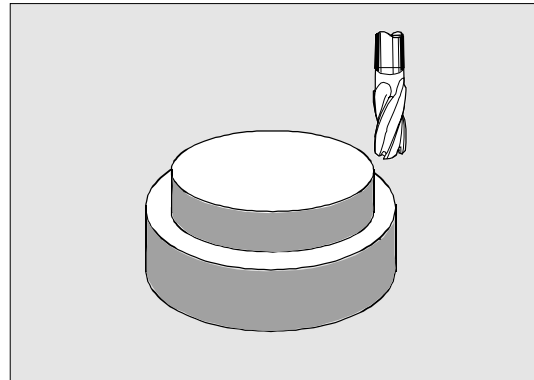
_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety distance (to be added to the reference plane, enter without sign)
_DP	real	Depth (absolute)
_DPR	real	Depth relative to the reference plane (enter without sign)
_PRAD	real	Diameter of spigot (enter without sign)
_PA	real	Spigot center point, abscissa (absolute)
_PO	real	Spigot center point, ordinate (absolute)
_MID	real	Maximum depth infeed (incremental, enter without sign)
_FAL	real	Final machining allowance on edge contour (incremental)
_FALD	real	Final machining allowance at the base (incremental, enter without sign)
_FFP1	real	Feedrate on contour
_FFD	real	Feedrate for depth infeed (or spatial infeed)
_CDIR	integer	Mill direction: (enter without sign) Values: 0...Down-cut milling 1...Up-cut milling 2...With G2 (independent of spindle direction) 3...With G3
_VARI	integer	Machining type Values: 1...Roughing to final machining allowance 2...Finishing (allowance X/Y/Z=0)
_AP1	real	Diameter of blank spigot

### 3.14 Milling circular spigots - CYCLE77



#### Function

With this cycle, you can machine circular spigots in the machining plane. A face cutter is required for finishing. Depth infeed is always performed in the position reached prior to semicircular positioning on the contour.



#### Sequence

##### Position reached before the beginning of the cycle:

The starting point is a position in the positive range of the abscissa with integrated approach semicircle and consideration of the programmed blank dimension.

##### Motion sequence for roughing ( $\_VARI=1$ )

###### Approach and return from contour:

The retraction plane ( $\_RTP$ ) is approached in rapid traverse in order that the tool can be positioned at the starting point in the machining plane at this level. The starting point is defined as 0 degrees in relation to the abscissa axis.

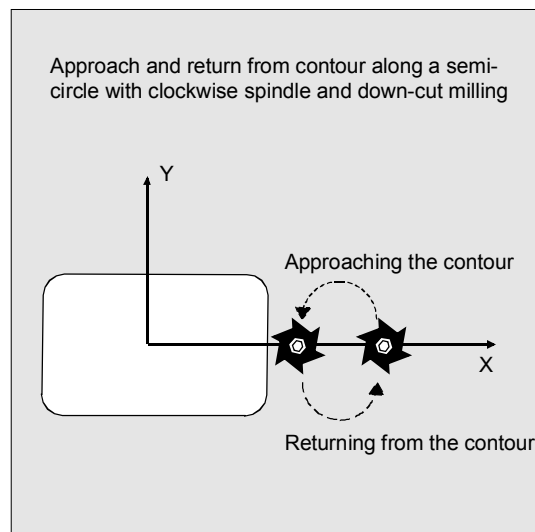
The tool is fed in at rapid traverse to the safety distance ( $\_SDIS$ ) and then traverses to machining depth at normal feedrate. The spigot contour is approached along a semicircular path, making allowance for the programmed blank spigot.

The mill direction can be defined as up-cut or down-cut milling in relation to the spindle direction.

If the spigot has been circumnavigated once, the tool lifts off the contour in the plane along a semicircular path and is then fed into the next machining depth.

The contour is then reapproached along a semicircle and the spigot is circumnavigated once. The process is repeated until the programmed spigot depth has been reached.

The tool then approaches the retraction plane ( $\_RTP$ ) in rapid traverse.



**Depth infeed:**

- Infeed to safety distance
- Insertion to machining depth

The first machining depth is the product of:

- The total depth,
- The final machining allowance and
- The maximum possible depth infeed.

**Motion sequence for finishing (\_VARI=2)**

Depending on the setting of parameters \_FAL and \_FALD, a finishing operation is performed on the contour surface or the base or both. The approach strategy matches the motions in the plane executed for roughing operations.

**Explanation of the parameters**

See Subsection 2.1.2 (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, SDIS, DP, and DPR.

See Section 3.9 for a description of parameters \_MID, \_FAL, \_FALD, \_FFP1, and \_FFD.

**\_PRAD (diameter of spigot)**

The diameter must be entered without a sign.

**\_PA, \_PO (spigot center point)**

With parameters \_PA and \_PO you define the reference point of the spigot.

Circular spigots are always measured from the center.

**\_CDIR (mill direction)**

The machining direction of the spigot is defined in this parameter.

Using parameter \_CDIR, the mill direction can be programmed as

- Directly "2 for G2" and "3 for G3" or
- Alternatively "up-cut milling" or "down-cut milling"

Up-cut milling or down-cut milling is determined within the cycle via the spindle direction activated prior to the cycle call.

<b>Down-cut</b>	<b>Up-cut</b>
M3 → G3	M3 → G2
M4 → G2	M4 → G3

**\_VARI (machining type)**

You can define the type of machining with parameter \_VARI.

Possible values are:

- 1=Roughing
- 2=Finishing

### 3.14 Milling circular spigots - CYCLE77

#### **\_AP1 (diameter of blank spigot)**

This parameter defines the blank dimension of the spigot (without sign). The internally calculated radius of the approach semicircle is dependent on this dimension.

#### **Other information**

A tool offset must be activated before the cycle is called. Otherwise, the cycle is aborted with alarm 61009 "Active tool number=0".

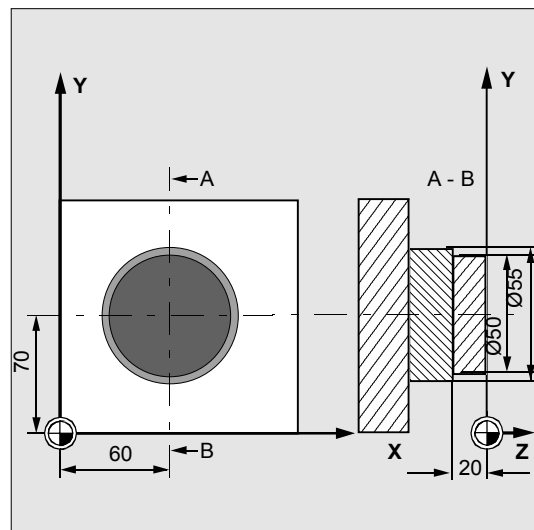
A new current workpiece coordinate system that influences the actual value display is used in the cycle. The zero point of this coordinate system lies on the pocket center point.

The original coordinate system becomes active again after the end of the cycle.

#### **Programming example**

##### **Circular spigot**

Machine a spigot from a blank with a diameter of 55 mm and a maximum infeed of 10 mm per cut. Enter a final machining allowance for finishing the spigot surface. The entire spigot is machined in an up-cut milling operation.



```
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)
```

```
N40 M30 ;End of program
```

-> Must be programmed in a single block

### 3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75



Pocket milling with islands is an option, which requires SW6 in both the NCK and HMI Advanced.

#### Requirement

To use the pocket milling cycle with islands, you need specific machine data settings.



**References:** For current information see:

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



#### 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 create a machining program, which it then executes. To ensure correct program processing, it is important to program cycle calls in the proper sequence.

- CYCLE74( ) ;Transfer edge contour
- CYCLE75( ) ;Transfer island contour 1
- CYCLE75( ) ;Transfer island contour 2
- ...
- CYCLE73( ) ;Machining pocket

### 3.15.1 Transfer pocket edge contour - CYCLE74



Pocket milling with islands is an option, which requires SW6 in both the NCK and HMI Advanced.



#### Programming

CYCLE74 (\_KNAME, \_LSANF, \_LSEND)



#### Parameters

_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



#### 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.





### Explanation of parameters

The edge contour can be programmed either in a separate program or in the main program that calls the routine. The contour is transferred to the cycle by parameter `_KNAME`, name of program or `_LSANF`, `_LSEND` and the program section from ... to identified by block numbers or labels.

Therefore, there are three options for contour programming:

- **Contour is defined in a separate program,** in which case only `_KNAME` needs to be programmed;  
e.g., `CYCLE74 ("EDGE", "", "")`
- **Contour is defined in the calling program,** in which case only `_LSANF` and `_LSEND` need to be programmed;  
e.g., `CYCLE74 ("", "N10", "N160")`
- **The edge contour is part of a program but not part of the program that calls the cycle,** in which case all three parameters need to be programmed;  
e.g., `CYCLE74("EDGE", "LABEL_START", "LABEL_END")`

The program name can be describable by its path name and program type.

Example:

```
_KNAME="/N_WKS_DIR/_N_EXAMPLE3_WPD/_N_EDGE_MPF"
```

**3.15.2 Transfer island contour - CYCLE75**

Pocket milling with islands is an option, which requires SW6 in both the NCK and HMI Advanced.

**Programming**

CYCLE75 (\_KNAME, \_LSANF, \_LSEND)

**Parameters**

<code>_KNAME</code>	string	Name of contour subroutine of island contour
<code>_LSANF</code>	string	Block number/label identifying start of contour definition
<code>_LSEND</code>	string	Block number/label identifying end of contour definition

**Function**

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.

**Explanation of parameters**

The number and meaning of parameters are the same as for CYCLE74.



(See CYCLE74)

### 3.15.3 Contour programming



Pocket edge and island contours must always be closed, i.e., the start and end points are identical.

The start point, i.e., the first point on a contour, must always be programmed with G0, and all other contour elements via G1 to G3.

When the contour is programmed, the last contour element (block with label or block number at end of contour) 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.

When the pocket machining program is calculated, only the geometries in the plane are taken into account.

If other axes or functions (T., D., S., M., etc.) are programmed in the contour sections, they are skipped during contour preparation within 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.).

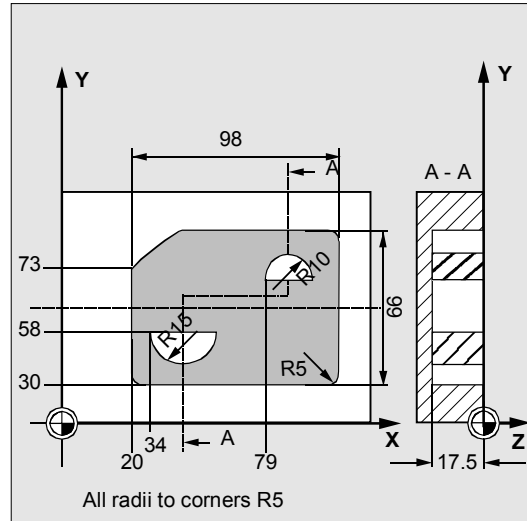
Every island to be repeated always has to be programmed again with the offsets calculated in the coordinates.

## 3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75



## Programming example

Sample program 1.mpf (pocket with islands)



```

%_N_SAMPLE1_MPF
;$PATH=/_N_MPF_DIR
; Example_1: Pocket with islands
;
$TC_DP1[5,1]=120 $TC_DP6[5,1]=6 $TC_DP3[5,1]=111 ;Tool offset cutter T5 D1
$TC_DP1[2.2]=120 $TC_DP6[2.2]=5 $TC_DP3[2.2]=130
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 _ENDEEDGE: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", "_ENDEEDGE") ;Transfer edge contour
CYCLE75 ("EXAMPLE1", "_ISLAND1", "_ENDISLAND1") ;Transfer island contour 1
CYCLE75 ("EXAMPLE1", "_ISLAND2", "_ENDISLAND2") ;Transfer island contour 2
ENDLABEL:
M30

```

### 3.15.4 Pocket milling with islands - CYCLE73



Pocket milling with islands is an option, which requires SW6 in both the NCK and HMI Advanced.



#### Programming

CYCLE73 (\_VARI, \_BNAME, \_PNAME, \_TN, \_RTP, \_RFP, \_SDIS, \_DP, \_DPR, \_MID, \_MIDA, \_FAL, \_FALD, \_FFP1, \_FFD, \_CDIR, \_PA, \_PO, \_RAD, \_DP1)



#### Parameters

_VARI	integer	Type of machining: (enter without sign) UNITS DIGIT (select machining): Values: 1...Roughing (solid machining) from solid material 2...Roughing residual material 3...Finishing edge 4...Finishing base 5...Predrilling  TENS DIGIT (select insertion strategy): Values: 1...Perpendicular with G1 2...Along a helical path 3...Oscillate  HUNDREDS DIGIT (select liftoff mode): Values: 0...To retraction plane (_RTP) 1...By safety distance (_SDIS) via reference plane (_RFP)  THOUSANDS DIGIT (select start point): Values: 1...Automatic 2...Manual
_BNAME	string	Name for program of drill positions
_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 distance (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 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

**3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75**

<code>_CDIR</code>	integer	Mill direction for machining the pocket: (enter without sign) Values: 0...Down-cut milling (as spindle rotation) 1...Up-cut milling 2...With G2 (independent of spindle direction) 3...With G3
<code>_PA</code>	real	Start point in first axis (only with manual selection of start point)
<code>_PO</code>	real	Start point in second axis (only with manual selection of start point)
<code>_RAD</code>	real	Radius of center-point path on insertion along helical path or max. insertion angle for oscillating insertion motion
<code>_DP1</code>	real	Insertion depth per 360° revolution on insertion along helical path

**Function**

Cycle CYCLE73 enables you to machine pockets 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.

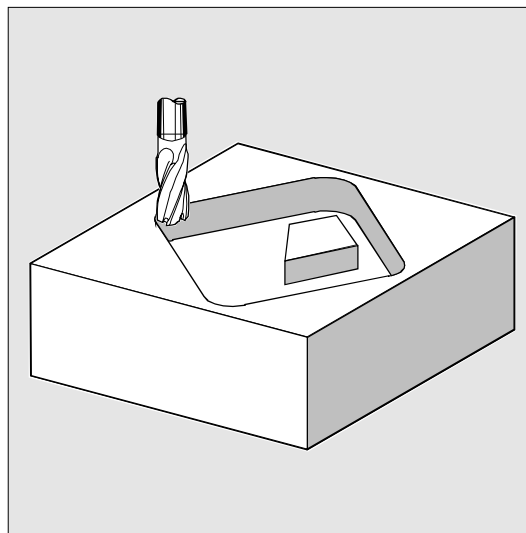
The cycle is executed once for each operation according to the programmed machining type (`_VARI`). In other words, in applications requiring roughing and finishing, or an additional rough-cut residual material operation, CYCLE73 must be called a second time.

**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.

No check is made on completion of the cycle for any further residual material 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).



### 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.

#### Example

Predrill, with solid machining

ACCEPTANCE4_CONT:	;Marker with name for beginning of ;pocket contour
CYCLE74 ("EDGEA01", , )	;Definition of contour for pocket edge
CYCLE75 ("INS11A01", , )	;Definition of contour for 1st island
CYCLE75 ("INS1A01", , )	
CYCLE75 ("INS2A01", , )	
CYCLE75 ("INS3A01", , )	
ENDLABEL:	;Marker for end of a pocket contour
T4 M6	
D1 M3 F1000 S4000	
MCALL CYCLE81(10,0,1,-3)	;Modal call of drilling cycle
REPEAT ACCEPTANCE4_MACH ACCEPTANCE4_MACH_END	;Execute drilling position program
MCALL	;Deselect drilling cycle modally
GOTOF ACCEPTANCE4_MACH_END	;Skip to solid machine pocket
ACCEPTANCE4_MACH:	;Start of Generate programs section
;REPEAT ACCEPTANCE4_CONT ENDLABEL	;Required only if there is more than one ;pocket contour



```

CYCLE73 (1015, "ACCEPTANCE4_DRILL", "ACCEPTANCE4_
MILL1", "3", 10, 0, 1, -12, 0, , 2, 0.5, , 9000, 400, 0, , , , )
ACCEPTANCE4_MACH_END: ;End of Generate programs section
T3 M6
D1 M3 S2000
;REPEAT ACCEPTANCE4_CONT ENDLABEL ;Required only if there is more than one
; pocket contour
CYCLE73 (1011, "ACCEPTANCE4_DRILL", "ACCEPTANCE
4_MILL1", "3", 10, 0, 1, -
12, 0, , 2, 0.5, , 9000, 400, 0, , , , ) ;Solid machine pocket

```



### 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 brought forward by the safety distance.
- 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. The pocket can be split if the ratio between the cutter diameter and the clearance between islands or between islands and edge contours is not ideal.  
For this purpose, the cycle calculates additional start points for cutter insertion.
- 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 distance via the reference plane, depending on the selected liftoff 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 edge 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 distance 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.
- Liftoff 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 values are:

#### **Units digit:**

- 1=Roughing (solid machine) from solid material
- 2=Roughing residual material
- 3=Finishing edge
- 4=Finishing base
- 5=Predrilling

When "Roughing from solid material" is set, 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:**

- 1=Perpendicular with G1
- 2=Along a helical path
- 3=Oscillation

#### **Selection of insertion strategies:**

- **Insert vertically (`_VARI=XX1X`)**  
means that the current infeed depth calculated internally is executed in one block.
- **Insert along helical path (`_VARI=XX2X`)**  
means that the mill center point traverses along the helical path determined by radius `_RAD` and depth per revolution `_DP1`. The feedrate is always programmed through `_FFD`. The sense of rotation of this helical path corresponds to the direction to be used for machining the pocket.  
The depth programmed under `_DP1` on insertion is calculated as the maximum depth and is always calculated as a whole number of revolutions of the helical path.

When the current depth for the infeed (this may be several revolutions on the helical path) has been calculated, a full circle is made to remove the slope created by insertion.

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

### 3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75

- **Insertion with oscillation (\_VARI=XX3X)**  
means that the mill center point oscillates along an oblique linear path until it has reached the next current depth. The maximum insertion angle is programmed under `_RAD`, the position of the oscillation path is calculated within the cycle. When the current depth has been reached, the path is traversed again without depth infeed in order to remove the slope caused by insertion. The feedrate is programmed through `_FFD`.

#### Hundreds digit: (\_VARI=X1XX)

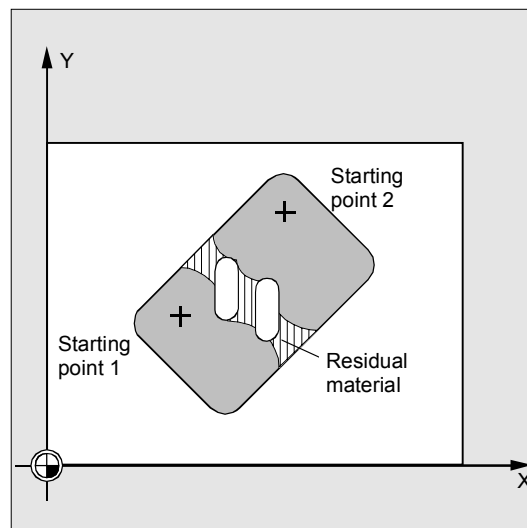
- 0=To retraction plane (`_RTP`)
- 1=By safety distance (`_SDIS`) via reference plane (`_RFP`).

#### Thousands digit: (\_VARI=1XXX)

- 1=Start point automatic
- 2=Start point manual

When the starting point is selected automatically, the cycle internally calculates the starting point for machining itself.

**Caution:** Manually specified start positions must not be too close to the island surface. Manually specified start positions are not internally monitored. 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. 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)**

This parameter must be set 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  
CYCLE73(1015,"PART1\_DRILL","PART1\_MILL",  
"MILL3",...)
- Without tool management  
CYCLE73(1015,"PART1\_DRILL","PART1\_MILL",  
"3",...)

Parameter **\_TN** is defined as a compulsory parameter with a maximum length of 16 characters. It must, therefore, be assigned to the solid machining tool in every subsequent **CYCLE73** call. When the residual material machining operation is used more than once, the tool from the last residual material removal process must be used.

**TOOL AND OFFSET:**

It must be ensured that the tool offset is processed exclusively by D1. Replacement tool strategies may not be used.

**\_RFP and \_RTP (reference plane and retraction plane)**

The reference plane (RFP) and retraction plane (RTP) are generally set to different values. The cycle assumes that the retraction plane precedes the reference plane. Therefore, the distance between the retraction plane and the final drilling depth is greater than the distance between the reference plane and the final drilling depth.

**\_SDIS (safety distance)**

The safety distance (SDIS) is effective with regard to the reference plane, which is brought forward by the safety distance.

The direction, in which the safety distance 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)**

The maximum infeed depth is defined with this parameter. 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 cut to pocket depth is made with one infeed.

**\_MIDA (max. infeed depth in the plane)**

With this parameter, you define the maximum infeed width for solid machining in the 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 final machining allowance only affects machining of the pocket in the plane at the edge.

When the final machining allowance  $\geq$  tool diameter, the pocket will not necessarily be machined completely.

**\_FALD (final machining allowance on 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)**

Feedrate `_FFD` is used for insertion into the material.

Feedrate `_FFP1` is used for all movements in the plane traversed at feedrate when machining.

**\_CDIR (mill direction)**

The value for the machining direction of the pocket is defined in this parameter.

Using parameter `_CDIR`, the mill direction can be programmed as

- Directly "2 for G2" and "3 for G3" or
- Alternatively "up-cut milling" or "down-cut milling"

Up-cut milling or down-cut milling is determined within the cycle via the spindle direction activated prior to the cycle call.

Down-cut	Up-cut
M3 $\rightarrow$ G3	M3 $\rightarrow$ G2
M4 $\rightarrow$ G2	M4 $\rightarrow$ G3

**\_PA, \_PO (start point for 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.

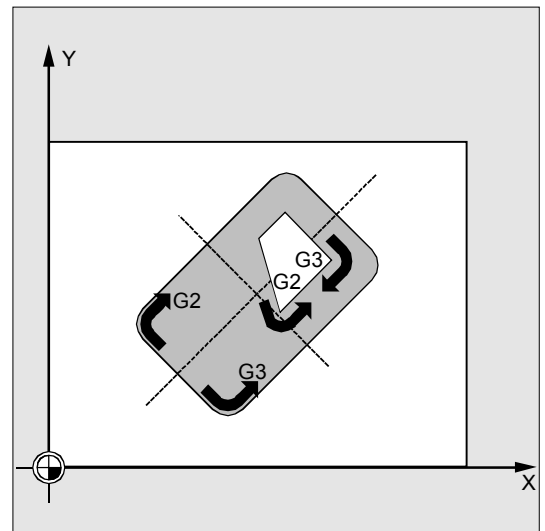
Please note that only one start point can be programmed (see description of parameter `_VARI`).

**\_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)**

With the parameter `_DP1`, you define the infeed depth for insertion on the helical path.





## Other information

### 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 on 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.



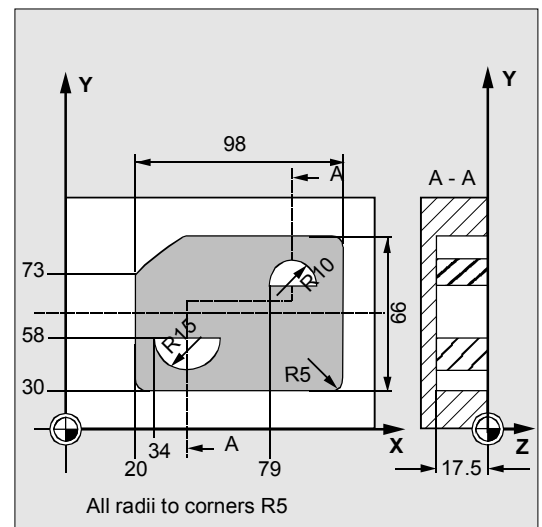
### Programming example 1

The machining task involves machining a pocket with two islands from solid material and then finishing the pocket in plane X, Y.

Sample program 1.mpf (pocket with islands)

```
%_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 ;Tool offset cutter T5 D1
$TC_DP6[5,1]=4
$TC_DP1[2,1]=120 $TC_DP3[2,1]=130
$TC_DP6[2,1]=5
N100 G17 G40 G90 ;Initial conditions G code
N110 T5 D1 ;Load milling tool
N120 M6
```



**3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75**

```

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 _ENEDGE: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:
SAMPLE1_CONT:
CYCLE74 (“”, “_EDGE”, “_ENEDGE”)
CYCLE75 (“”, “_ISLAND1”, “_ENDISLAND1”)
CYCLE75 (“”, “_ISLAND2”, “_ENDISLAND2”)
ENDLABEL:

;Programming Mill Pocket
CYCLE73 (1021, “”, “SAMPLE1_MILL1”, “5”, 10, 0, 1,
-17.5, 0, , 2, 0.5, , 9000, 3000, 0, , , 4, 3)

T2 D1 M6
S3000 M3

;Programming Finish Pocket
CYCLE73 (1113, “”, “SAMPLE1_MILL3”, “5”, 10, 0, 1,
-17.5, 0, , 2, , , 8000, 1000, 0, , , 4, 2)
M30

```



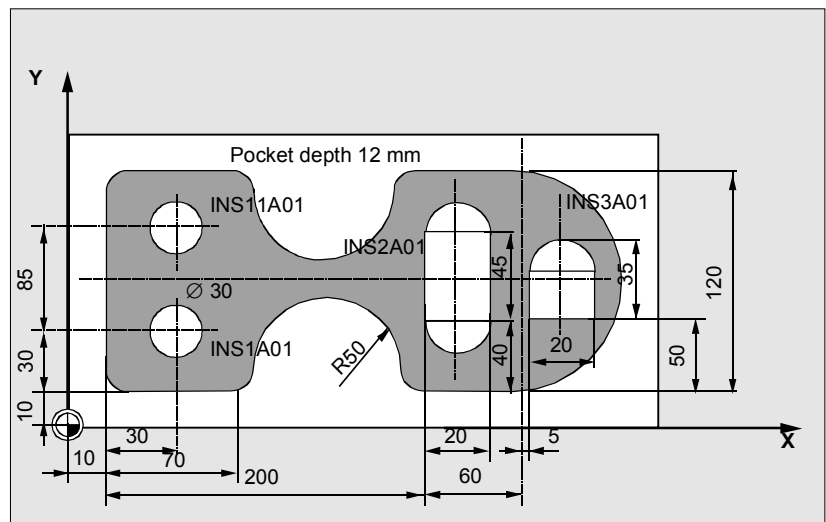
### Programming example 2

Machining task:

Before the pocket is milled, the workpiece must be predrilled to ensure optimum insertion of the milling tool.

- Center for predrilling
- Drilling
- Solid machine pocket with islands, cutter radius 12 mm
- Solid machine residual material, cutter radius 6 mm
- Finish pocket, cutter radius 5 mm

Sketch of machining operation



### Machining program:

```

%_N_SAMPLE2_MPF
; $PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
; Example_2: Pocket with islands
; 2*predrill, machine, machine resid. mat., finish
;
; Tool offset data
$TC_DP1[2,1]=220 $TC_DP6[2,1]=10
$TC_DP1[3,1]=120 $TC_DP6[3,1]=12
$TC_DP1[4,1]=220 $TC_DP6[4,1]=3
$TC_DP1[5,1]=120 $TC_DP6[5,1]=5
$TC_DP1[6.1]=120 $TC_DP6[6.1]=6
TRANS X10 Y10

; Defining machining contours
ACCEPTANCE4_CONT:
CYCLE74("EDGEA01", ,)
CYCLE75("INS11A01", ,)
CYCLE75("INS1A01", ,)
CYCLE75("INS2A01", ,)
CYCLE75("INS3A01", ,)
ENDLABEL:

```

**3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75**


---



---

```
;Program centering
```

```
T4 M6
```

```
D1 M3 F1000 S4000
```

```
MCALL CYCLE81 (10,0,1,-3,)
```

```
REPEAT ACCEPTANCE4_MACH ACCEPTANCE4_MACH_END
```

```
MCALL
```

---

```
;Program drilling
```

```
T2 M6
```

```
D1 M3 F2222 S3000
```

```
MCALL CYCLE81 (10,0,1,-12,)
```

```
REPEAT ACCEPTANCE4_MACH ACCEPTANCE4_MACH_END
```

```
MCALL
```

---

```
GOTOF ACCEPTANCE4_MACH_END
```

```
ACCEPTANCE4_MACH:
```

```
REPEAT ACCEPTANCE4_CONT ENDLABEL
```

```
CYCLE73(1015,"ACCEPTANCE4_DRILL","ACCEPTANCE4_MILL1",  
"3",10,0,1,-12,0,,2,0.5,,2000,400,0,,,,)
```

```
ACCEPTANCE4_MACH_END:
```

---

```
;Program solid machining
```

```
T3 M6
```

```
D1 M3 S4000
```

```
REPEAT ACCEPTANCE4_CONT ENDLABEL
```

```
CYCLE73(1011,"","ACCEPTANCE4_MILL1","3",10,0,1,  
-12,0,,2,0.5,,2000,400,0,,,,)
```

```
;Program solid machining of residual material
```

```
T6 M6
```

```
D1 M3 S4000
```

```
REPEAT ACCEPTANCE4_CONT ENDLABEL
```

```
CYCLE73(1012,"","ACCEPTANCE4_2_MILL4","3",10,0,1  
'  
-12,0,,2,0.5,,1500,800,0,,,,)
```

```
;Program finishing
```

```
T5 M6
```

```
D1 M3 S4500
```

```
REPEAT ACCEPTANCE4_CONT ENDLABEL
```

```
CYCLE73(1013,"","ACCEPTANCE4_MILL3","3",10,0,1,  
-12,0,,2,,,3000,700,0,,,,)
```

```
M30
```

---



**3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75**

```
%_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
```



### Programming 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
$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 M8
N30 G0 Z20

; Machining contours pocket 1
GOTOF ENDLABEL
POCKET1_CONT:
CYCLE74 ("EDGE", "", "")
CYCLE75 ("ISLAND1", "", "")
CYCLE75 ("ISLAND2", "", "")
ENDLABEL:

```

**3.15 Pocket milling with islands - CYCLE73, CYCLE74, CYCLE75**

```

;Machining contours pocket 2
GOTOF ENDLABEL
SAMPLE2_CONT:
CYCLE74("EDGEA01", ,)
CYCLE75("INS11A01", ,)
CYCLE75("INS1A01", ,)
CYCLE75("INS2A01", ,)
CYCLE75("INS3A01", ,)
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:
REPEAT POCKET1_CONT ENDLABEL
CYCLE73(1025,"TASCHE1_DRILL","TASCHE1_MILL1","3",10,0,1,-8,0,0,2,0,0,2000,400,0,0,0,3,4)
POCKET1_MACH_END:
REPEAT POCKET1_CONT ENDLABEL
CYCLE73(1021,"POCKET1_DRILL","POCKET1_MILL1","3",10,0,1,-8,0,0,2,0,0,2000,400,0,0,0,3,4)

;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,-8,0,0,2,0,0,2000,400,0,0,0,3,4)
SAMPLE2_MACH_END:
REPEAT SAMPLE2_CONT ENDLABEL
CYCLE73(1011,"SAMPLE2_DRILL","SAMPLE2_MILL1","3",10,0,1,-8,0,0,2,0,0,2000,400,0,0,0,3,4)

```



```

;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

```

**Explanation****Alarms source CYCLE73...CYCLE75**

Alarm number	Alarm text	Explanation, remedy
61703	"Internal cycle error while deleting file"	
61704	"Internal cycle error while writing file"	
61705	"Internal cycle error while reading file"	
61706	"Internal cycle error during checksum formation"	
61707	"Error in ACTIVATE on MMC"	
61708	"Error in READYPROG on MMC"	
61900	"No contour"	
61901	"Contour is not closed"	
61902	"No more free memory"	
61903	"Too many contour elements"	
61904	"Too many intersections"	
61905	"Cutter radius too small"	
61906	"Too many contours"	
61907	"Circle without center point measurement"	
61908	"No starting point specified"	
61909	"Helical radius too small"	
61910	"Helix violates contour"	
61911	"More insertion points required"	
61912	"No path to generate"	
61913	"No residual material generated"	
61914	"Programmed helix violates contour"	
61915	"Approach/return motion violates contour"	
61916	"Ramp path too short"	
61917	"Residual corners may be left with less than 50 percent overlap"	
61918	"Cutter radius too large for residual material"	
61980	"Error in island contour"	
61981	"Error on edge contour"	
61982	"Infeed width in plane too large"	
61983	"Pocket edge contour missing"	
61984	"Tool parameter _TN not defined"	
61985	"Name of drilling position program missing"	
61986	"Machine pocket program missing"	
61987	"Drilling position program missing"	
61988	"Name of program for machining pocket missing"	
61989	"D1 not programmed as active tool cutting edge"	

### 3.16 Swiveling - CYCLE800



#### Calling swiveling – CYCLE800

Entry area Programs / Milling

The  →

soft key is displayed if the swivel data record has been set up (MD 18088: MM\_NUM\_TOOL\_CARRIER >0).



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
- Orientational toolholder

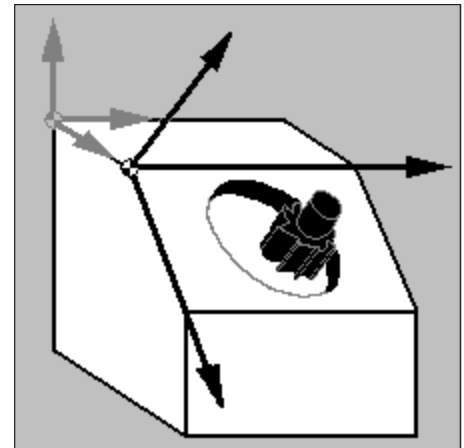


**References:** Description of Functions 840D/840Di/810D  
/W1/ "Tool offset"  
/R2/ "Rotary axes"  
/K2/ "System frames"



#### 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.



#### 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



See also the "Offset rotary axis vectors (machine kinematics)" figures in Subsection 3.16.7.

**Important**

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.

### 3.16.1 Operation, parameter assignment, input screen form



#### Explanation of the parameters

Input screen form CYCLE800 on standard interface

Swivel cycle/CYCLE800		Reference point for rotation around Y axis		
Name:				
Retract:		Z		
Schwenken:		Yes		
Swivel plane:		new		
Ref. point:	X0		0.000	
	Y0		25.000	← Reference point before rotation
	Z0		0.000	
Swivel mode:		axis by ax.		
Rot. around	X (A)		-15.000	
Rot. around	Y (B)		0.000	← Rotation
Rot. around	Z (C)		0.000	
Zero point:	X1		0.000	
	Y1		0.000	
	Z1		0.000	← Zero point after rotation
Direction:			Minus	
Tracking TL			No	

#### 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, no name needs to be declared.

"0" → Swivel data record deselection.

#### Retraction \_FR (prior to swiveling rotary axes)

- Do not retract
- Retract Z axis<sup>1)</sup>
- Retract axis Z, XY<sup>1)2)</sup>
- Maximum retraction in tool direction (with Cycles SW 6.5 and higher)<sup>2)</sup>
- Incremental retraction in tool direction (with Cycles SW 6.5 and higher)<sup>2)</sup>

The incremental value for the travel path in the tool direction must be entered in the input field.

The retraction positions can be entered in the CYCLE800 startup menu.



- 1) The retraction positions are traversed absolutely. If another sequence or incremental positioning is desired, this can be modified during installation and setup in the TOOLCARR user cycle.
- 2) Only if the CYCLE800 startup menu is active.

**Note:**

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
- **Minus/plus direction**  
Reference to rotary axis 1 or 2 when direction of travel is selected in the swivel cycle. Due to the angular range of the rotary axes of machine kinematics, the NCU can calculate two possible solutions. Usually, one of these solutions is technologically suitable.  
The rotary axis (1st or 2nd rotary axis), to which the two solutions must refer, is selected in the CYCLE800 startup menu (see Subsection 3.16.7).  
Selecting the "minus" or "plus" direction decides, which of the two possible solutions is to be applied.

Note the machine manufacturer's instructions!

**Swivel plane \_ST**

- **New**  
Previous swivel frames and programmed frames are deleted and a new swivel frame 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.

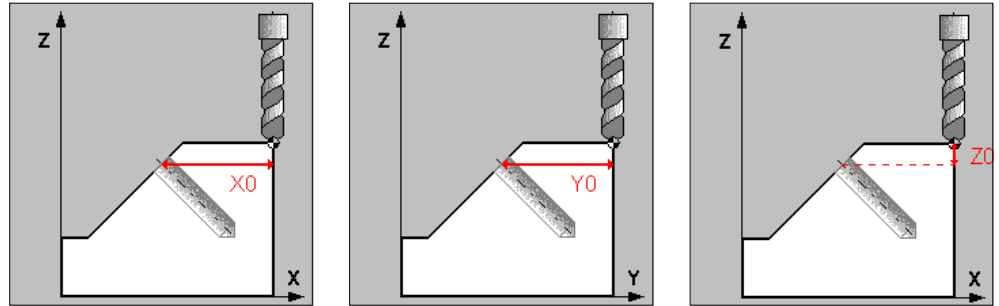


If the currently active ZO contains rotations, e.g., due to previous workpiece gauging, they will be taken into account in the swivel cycle.

The following help displays relate to machining plane G17 (tool axis Z).

### Reference points before rotation X0, Y0, Z0

Reference points



### Swivel mode \_MODE

This parameter defines the swivel mode for the axis.

- Axial
- Projection angle<sup>1)2)</sup>
- Solid angle<sup>1)</sup>



Swivel mode always refers to the coordinate system of the workpiece and does not, therefore, depend on the machine.

You can set the available swivel modes in the CYCLE800 startup menu.

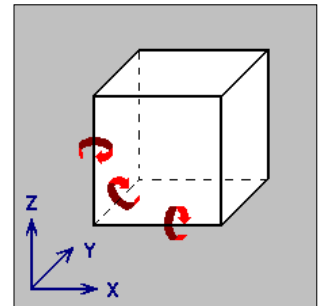
- With the axial swiveling option, the tool is rotated about the individual axes of the coordinate system in succession, with each rotation starting from the previous rotation. The axis sequence can be freely selected.
- When swiveling using the projection angle, the angle value of the swiveled surface is projected onto the first two axes of the coordinate system. The third rotation starts from the previous rotation. The axis sequence can be freely selected.
- 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.

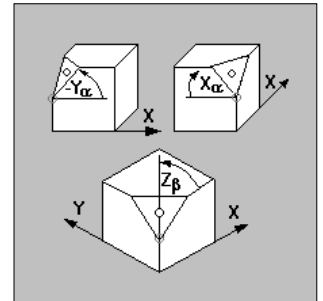


- 1) Only available if selected by the machine manufacturer in the CYCLE800 startup menu.
- 2) When programming from X and Y, the new X axis lies in the old ZX plane.  
When programming from Z and X, the new Z axis lies in the old YZ plane.  
When programming from Y and Z, the new Y axis lies in the old XY plane.  
The third rotation takes place on the swiveled plane.

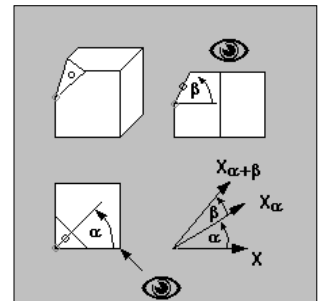
### Axial



### Projection angle



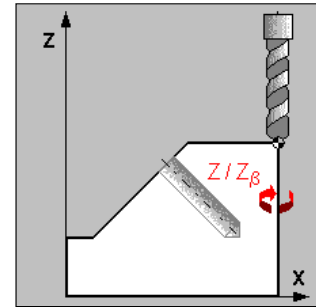
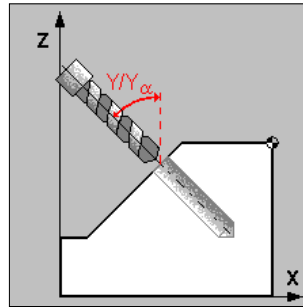
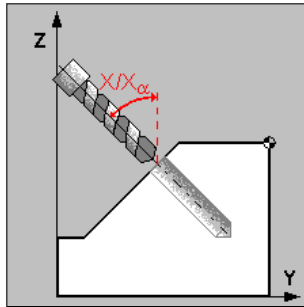
### Solid angle



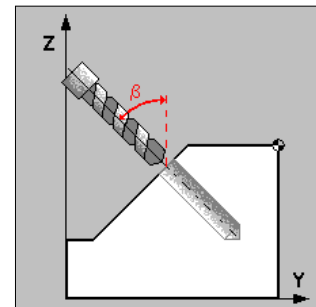
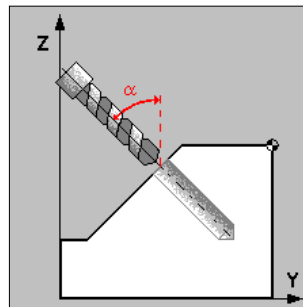
## 3.16 Swiveling - CYCLE800

## Rotation about A, B, C

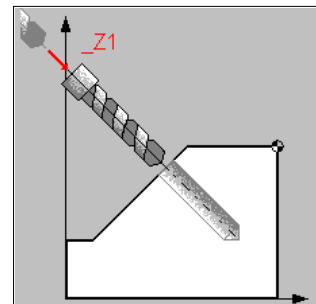
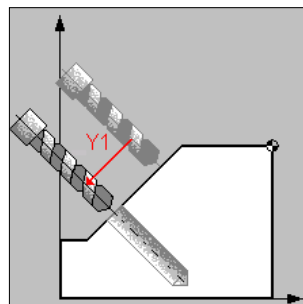
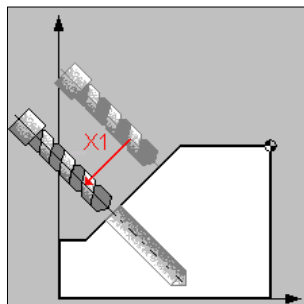
- Rotations (axial, projection angle)



- Rotation (solid angle)



## Zero point after rotation X1, Y1, Z1



## Correct tool\_TC\_N\_WZ

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.

Prerequisites:

  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 Operating instructions, programming 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 ;(Machine kinematic types "T" and "M" only)
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 "high speed settings" cycle CYCLE832 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.
- It is possible to deselect the swivel data record and cancel the swivel frame (WPFRAME, PARTFRAME, TOOLFRAME) by programming **CYCLE800()**.
- In CYCLE800, parameters can also be transferred as input values (e.g., result variable of measuring cycles `_OVR[19]`).
- If rotary axis values are entered in the active ZO (basic rotation of the workpiece), they will be taken into account when swiveling.

When swiveling to the **setup basic setting** (see Subsection 3.16.5) 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).

## 3.16.3 Parameters



## Programming

CYCLE800(\_FR, \_TC, \_ST, \_MODE, \_X0, \_Y0, \_Z0, \_A, \_B, \_C, \_X1, \_Y1, \_Z1, \_DIR, \_FR\_I)



## Parameters

_FR	integer	Retraction Values: 0...Do not retract 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 UNITS DIGIT: Values: 0... New 1... Additive TENS DIGIT <sup>2)</sup> : Requirement: TRAORI option must be available. Values: 0x... Do not correct tool tip 1x... Correct tool tip HUNDREDS DIGIT: Values: 1xx... Set tool																																																																																																																																																									
_MODE	integer	Swivel mode Evaluation of angle: Decimal parameter mode, binary coding <table style="margin-left: 40px; border-collapse: collapse;"> <tr> <td style="text-align: center; border-bottom: 1px solid black;">7</td> <td style="text-align: center; border-bottom: 1px solid black;">6</td> <td style="text-align: center; border-bottom: 1px solid black;">5</td> <td style="text-align: center; border-bottom: 1px solid black;">4</td> <td style="text-align: center; border-bottom: 1px solid black;">3</td> <td style="text-align: center; border-bottom: 1px solid black;">2</td> <td style="text-align: center; border-bottom: 1px solid black;">1</td> <td style="text-align: center; border-bottom: 1px solid black;">0</td> <td></td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td></td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">01: Rotation around 1<sup>st</sup> axis</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">10: Rotation around 2<sup>nd</sup> axis</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">11: Rotation around 3<sup>rd</sup> axis</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">} Angle of rotation 1</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">01: Rotation around 1<sup>st</sup> axis</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">10: Rotation around 2<sup>nd</sup> axis</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">11: Rotation around 3<sup>rd</sup> axis</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">} Angle of rotation 2</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">01: Rotation around 1<sup>st</sup> axis</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">10: Rotation around 2<sup>nd</sup> axis</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">11: Rotation around 3<sup>rd</sup> axis</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">} Angle of rotation 3</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">00: Swivel angle per axis (_A, _B, _C)</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">01: Solid angle (_A, _B)</td> </tr> <tr> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td colspan="2" style="border-left: 1px solid black; border-right: 1px solid black;"></td> <td style="border-left: 1px solid black; border-right: 1px solid black;">10: Angle of projection (_A, _B, _C)</td> </tr> </table>	7	6	5	4	3	2	1	0																			01: Rotation around 1 <sup>st</sup> axis									10: Rotation around 2 <sup>nd</sup> axis									11: Rotation around 3 <sup>rd</sup> axis									} Angle of rotation 1									01: Rotation around 1 <sup>st</sup> axis									10: Rotation around 2 <sup>nd</sup> axis									11: Rotation around 3 <sup>rd</sup> axis									} Angle of rotation 2									01: Rotation around 1 <sup>st</sup> axis									10: Rotation around 2 <sup>nd</sup> axis									11: Rotation around 3 <sup>rd</sup> axis									} Angle of rotation 3									00: Swivel angle per axis (_A, _B, _C)									01: Solid angle (_A, _B)									10: Angle of projection (_A, _B, _C)
7	6	5	4	3	2	1	0																																																																																																																																																				
								01: Rotation around 1 <sup>st</sup> axis																																																																																																																																																			
								10: Rotation around 2 <sup>nd</sup> axis																																																																																																																																																			
								11: Rotation around 3 <sup>rd</sup> axis																																																																																																																																																			
								} Angle of rotation 1																																																																																																																																																			
								01: Rotation around 1 <sup>st</sup> axis																																																																																																																																																			
								10: Rotation around 2 <sup>nd</sup> axis																																																																																																																																																			
								11: Rotation around 3 <sup>rd</sup> axis																																																																																																																																																			
								} Angle of rotation 2																																																																																																																																																			
								01: Rotation around 1 <sup>st</sup> axis																																																																																																																																																			
								10: Rotation around 2 <sup>nd</sup> axis																																																																																																																																																			
								11: Rotation around 3 <sup>rd</sup> axis																																																																																																																																																			
								} Angle of rotation 3																																																																																																																																																			
								00: Swivel angle per axis (_A, _B, _C)																																																																																																																																																			
								01: Solid angle (_A, _B)																																																																																																																																																			
								10: Angle of projection (_A, _B, _C)																																																																																																																																																			
_X0, _Y0, _Z0	real	Reference point prior to rotation																																																																																																																																																									

- 1) – If declared in swiveling startup.  
 – Can be adjusted in the TOOLCARR user cycle.  
 – Note the machine manufacturer's instructions.
- 2) See note following parameter list.

<code>_A</code>	real	1. Axis angle (axial swivel mode) 2. Angle of rotation about the Z axis in XY plane (solid angle swivel mode) 3. Axis angle (projection angle swivel mode) sequence of axes
<code>_B</code>	real	1. Axis angle (axial swivel mode) 2. Angle of rotation in space about the Y axis (swivel mode 'solid angle')
<code>_C</code>	real	Axis angle (axial, projection angle swivel modes)
<code>_X1, _Y1, _Z1</code>	real	Zero point after rotation
<code>_DIR</code>	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 movement of rotary axes (calculation only)
<code>_FR_I</code>	real	Incremental retraction value in tool direction (with Cycles SW 6.5 and higher)

2)

**Retraction in tool direction (with Cycles SW 6.5 and higher)**

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 incremental input value (`_FR_I`).

Maximum and incremental retraction in tool direction always occurs in a **positive tool direction** (away from workpiece).

**Retraction in tool direction is always executed prior to swiveling the rotary axes.**

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. When retracting 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 (with Cycles SW 6.5 and higher)	Retract:	Tool max.
Incremental retraction in tool direction (with Cycles SW 6.5 and higher)	Retract:	Tool inc., e.g., 100.00

## 3.16 Swiveling - CYCLE800



## Programming example 1

## Set swivel plane ZERO

```

%_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

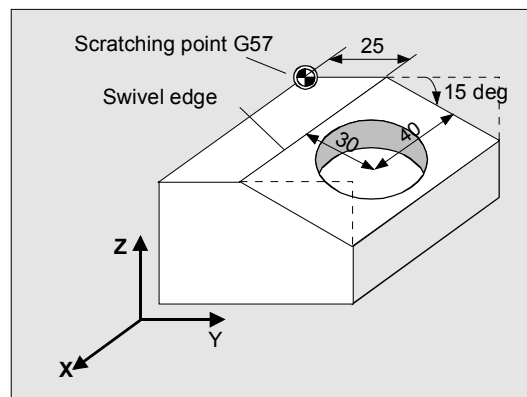
```

Swivel cycle/CYCLE800		Reference point for rotation around X axis	
Name:			
Retract:		Z	
Schwenken:		Yes	
Swivel plane:		new	
Ref. point:	X0	0.000	
	Y0	0.000	
	Z0	0.000	
Swivel mode:		axis by ax.	
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	



## Programming example 2

## Face milling and milling a circular pocket on a machining plane swiveled through 15 degrees



```

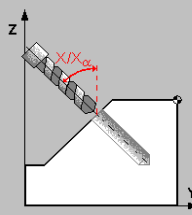
%_N_SWIVEL_CIRCULARPOCKET_SPF
; $PATH=/_N_WCS_DIR/_N_HAA_SWIVEL_WPD
N12 T="MILL_26mm"
N14 M6
N16 G57
N18 CYCLE800(1, "", 0, 57, 0, 0, 0, 0, 0, 0, 0, 0, 1)
N20 M3 S5000
N22 CYCLE71(50, 2, 2, 0, 0, 0, 80, 60, 0, 4, 15, 5, 0, 2000, 31, 5)

```

;Face milling

N24 CYCLE800(1, "", 0, 57, 0, 25, 0, -15, 0, 0, 0, 0, -1)

Swivel cycle/CYCLE800		Angle of 1st rotation
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 <input checked="" type="checkbox"/>
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	

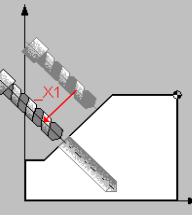


N26 CYCLE71(50, 12, 2, 0, 0, 0, 80, 60, 0, 4, 10, 5, 0, 2000, 31, 5)

;Face milling

N28 CYCLE800(1, "", 1, 57, 0, 0, 0, 0, 0, 0, 40, 30, 0, 1)

Swivel cycle/CYCLE800		Zero point of X axis swivel plane
Name:		
Retract:	Z	
Schwenken:	Yes	
Swivel plane:	new	
Ref. point:	X0	0.000
	Y0	0.000
	Z0	0.000
Swivel mode:	axis by ax.	
Rot. around	X (A)	0.000
Rot. around	Y (B)	0.000
Rot. around	Z (C)	0.000
Zero point:	X1	40.000 <input checked="" type="checkbox"/>
	Y1	30.000
	Z1	0.000
Direction:	Minus	
Tracking TL	No	



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, , , , , )

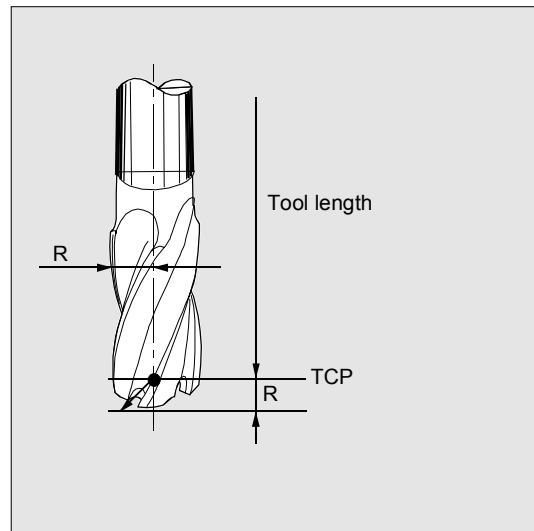
N40 M2

## 3.16.4 Setting tools (with Cycles SW 6.5 and higher)

After swiveling, 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 rotation sequence can be freely selected.



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).



## Call

Milling >> Setting tool (4<sup>th</sup> vertical soft key)

Input screen form

Approach TL/CYCLE800		Angle of 2nd rotation
Name:	TC40_TABLE_AC	
Retract:	No	
Swivel plane:	Additive	
Swivel mode:	axis by ax.	
Rot. around	X (A)	8.0000
Rot. around	Y (B)	-8.0000
Tracking TL	No	

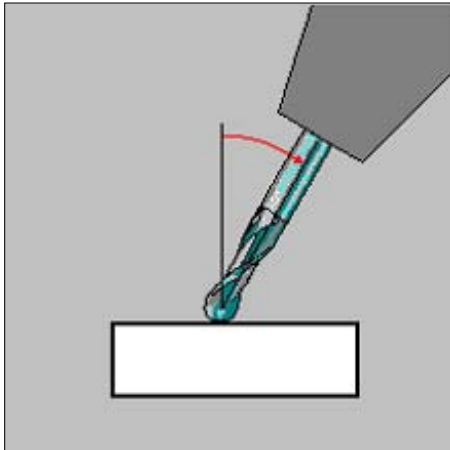
The "Setting tool" input screen form corresponds to the CYCLE800 input screen form, which has been shut down.

On the input screen form, the following applies:

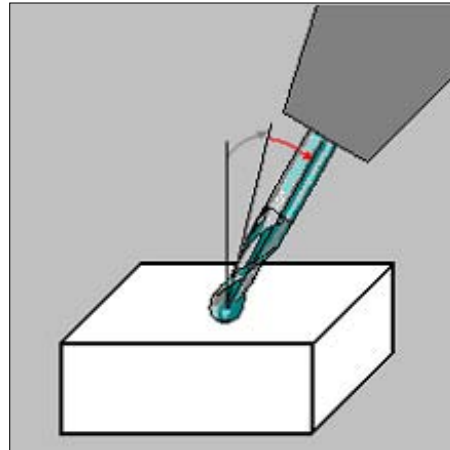
<b>Name _TC:</b>	Current swivel data record
<b>Retraction _FR:</b>	As for CYCLE800
<b>Swivel plane _ST:</b>	Additive (display only)
<b>Swivel mode _MODE:</b>	Axially (display only)
<b>Rotation about:</b>	Rotation about a maximum of 2 axes in the WCS XY, YZ, XZ
<b>Correct tool:</b>	As for CYCLE800 (for display option, see CYCLE800 startup)



Setting tool help displays



Setting tool: 1st rotation



Setting tool: 2nd rotation

### 3.16.5 Setting up workpieces with swiveled machining planes (Cycles SW 6.5 and higher)



#### Swiveling in JOG mode

The "Swiveling in JOG" function is used to set up workpieces with swiveled machining planes and represents a prerequisite for further setup procedures (scratch/measure edge/corner).

Swiveling in JOG mode is supported by its own input screen form.

The rotary axes are positioned with NC start (turned with manual rotary axes) and a swivel frame is calculated for this plane.

The operations below can be carried out with "Swiveling in JOG":

- Compensation of basic rotation of the workpiece on machines with swivel tables.
- Swiveled plane set to a new zero plane.
- Swiveled plane refers additively to the current zero frame and is 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.).



### Compensation of basic rotation in the active tool plane

In the following, the term “**basic rotation**” applies to the rotation of the workpiece about the tool axis, for example, rotation of a rotary table about the Z tool axis (C rotary axis).

By rotating the rotary table, operators/machine setters wish to compensate the basic rotation of the workpiece, thus allowing them to machine the workpiece axially.

This can be achieved by:

1. Retracting the workpiece edge with the dial indicator, whereby the rotary axis is rotated such that the dial indicator's deviation approaches 0.  
The operator can enter the rotary axis position value into the zero offset (ZO) of the rotary axis.
2. Measuring “Arrangement of workpieces by measuring points, holes or spigots” in JOG mode.

When using the process described here to compensate the basic rotation, the operator can define the **basic setting** in the “Swiveling” input screen form in JOG mode and the rotary axis is positioned accordingly using NC start. Positioning the rotary axis in order to compensate the basic rotation has no effect on the WCS. The WCS remains unchanged, if the rotary axis rotates around the tool axis.



### Swiveled plane set to a new zero plane

Operators/machine setters must be aware of the machine position, which 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 machine kinematics basic setting when setting up the “Swiveling” or 5-axis transformation function.

When setting up workpieces with swiveled (inclined) machining planes or an inclined workpiece setting, the term **basic rotation** is extended to **both rotary axes**.

Operators/machine setters must categorically define the plane (surface) to be defined as the **new zero plane**. This can also be the basic setting of the machine kinematics. Therefore, the swiveled-in surface becomes the zero plane, such that the rotations are transferred from the “Swiveling in JOG” input screen form to the current zero offset.

This results in a new **setup basic setting**. The **setup basic setting** can be approached both as “new” and “additive”.

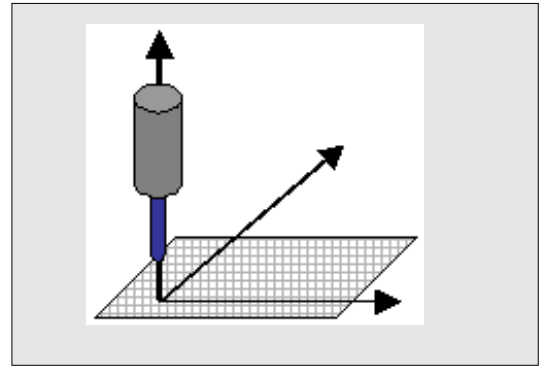
After approaching the **setup basic setting**, the workpiece is either set up via the “Scratching” menu or by using the available probe in “Measuring in JOG”.

All subsequent actions such as scratching, measuring with probe and swiveling with NC workpiece programs refer to this zero plane or to the **setup basic setting**.

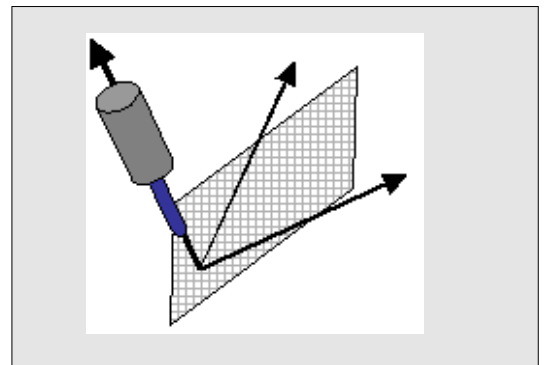


**Machine kinematics basic setting**

- 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.

**Example:**

The machine kinematics basic setting of the tool is vertical on the rotary axis C (vertical tool setting) 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 with horizontal tool orientation, the tool must first be positioned horizontally using a rotation about Y = 90 degrees (possibly in conjunction with a rotation about Z = 180 degrees) in "Swiveling in JOG".

The ZO is then defined using Scratching or Measuring in JOG, with horizontal tool orientation.



### Data transfer of swivel data in "Swiveling in JOG"

Only rotations are written to the active frame chain with the "Swiveling in JOG" function.

Depending on the possible rotation combinations or the zero plane, the following apply to the transfer of swivel data to the current frame chain:

- **Swivel plane: New -> New zero plane**
  - Deletes and links the rotations of the active ZO<sup>1)</sup> to geometry axes XYZ
  - Deletes the workpiece reference component of the swivel frame (WPFRAME)
- **Swivel plane: Additive -> New zero plane**
  - Describes the rotations of the active ZO<sup>1)</sup> of geometry axes XYZ
  - Deletes the workpiece reference component of the swivel frame (WPFRAME)
- **Swivel plane: New** (on current zero plane)<sup>2)</sup>
  - Deletes and links the rotations of the workpiece reference (WPFRAME)
- **Swivel planes: Additive** (on current zero plane)<sup>2)</sup>
  - Links the rotations of the workpiece reference (WPFRAME)



#### 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.

#### 2) Only with ShopMill/ShopTurn.

If the swivel plane refers to the current zero plane, the rotations of geometry axes XYZ of the current ZO are retained.

With "Swiveling in JOG", the translational components of geometry axes XYZ of the active ZO remain unconsidered and unchanged. This means that the position of the tool on the swiveled zero plane is undefined and is only defined by subsequent operator actions such as Scratching and Measuring.

The translational components of the rotary axes of the active ZO also remain unchanged and are considered in CYCLE800, e.g., basic rotation of the C axis on machines with swivel tables.

### 3.16.6 Operation, parameter assignment “Swiveling in JOG” (with Cycles SW 6.5 and higher)



#### Explanation of input screen form parameters

- Horizontal soft key 8 is the entry soft key to “Swiveling in JOG” in JOG mode.
- Horizontal soft key 3 is the entry soft key in "Measuring in JOG".

- The “Swiveling in JOG” input screen form corresponds to a shortened version of the CYCLE800 Swiveling input screen form.
- The swivel mode (direct) and the “Basic setting” or “Teach-in” (direct only) soft keys have been expanded.
- The “Retraction”, “Direction” and “Correct tool” input fields can be set using display options in the “Swiveling” startup figure.
- Active ZO display, which is to be defined for the new zero plane or **setup basic setting**.
- Display of rotations of the active ZO and rotations of the swivel frame (\$P\_WPFFRAME)>

#### Name

The name of the swivel data record corresponds to parameter:

- `_TC` (CYCLE800) or
- `TRAORI`

With an active 5-axis transformation and deselected swivel data record (TCARR=0), `TRAORI` or `TRAORI(2)` is displayed. In this case, the swivel plane is approached by means of the 5-axis transformation.

**Retraction**

Corresponds to the CYCLE800 parameter `_FR` (retraction).

**Swivel plane**

1. New swivel plane → New zero plane
2. Additive swivel plane → New zero plane

**Swivel mode**

Selection:

- **Axis by axis**

Rotation about X, Y or Z, as for CYCLE800

- **Direct**

This allows rotary axis positions to be entered directly.

After rotary axes have been positioned (or manual rotary axes have been turned), 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 rotary axis actual values.

“Direct” swivel mode		
Swivel mode		Direct
Rotary axis 1	C	-15.00
Rotary axis 2	B	0.000

**Direction**

Corresponds to the CYCLE800 parameter `_DIR` (direction).

- Minus direction corresponds to `_DIR=-1`.
- Plus direction corresponds to `_DIR=1`.

The Swiveling and startup menu display options can be used to skip the “Direction” input field. In Direct swivel mode, the “Direction” input field is not displayed.

**Correct tool**

Corresponds to the CYCLE800 parameter `_ST=1x` (correct tool tip).

The Swiveling startup menu display options can be used to skip the “Correct tool” input field.

- **“Correct tool” no**

→ Tool tip is not corrected during swiveling.

- **“Correct tool” yes**

→ Activate/deactivate TRAORI in the TOOLCARR.SPF cycle while the rotary axes are traversed, if this has been provided by the machine manufacturer.



Note the machine manufacturer’s instructions!

**Basic setting (vertical soft key 3)**

Approaching the **setup basic 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: Z
- Swivel mode: Axis by axis
- Rotation about Z 0.0
- Rotation about Y 0.0
- Rotation about X 0.0

**Abort (vertical soft key 7)**

→ Input screen form is exited.

The swivel plane is not changed and the machine is not traversed.

**OK (vertical soft key 8)**

→ Input parameters are accepted.

After confirming acceptance of the input parameters with **OK**, the text message "Start traversing swivel axes -> press NC start" requests the operator/machine setter to approach the swivel plane using NC start.

**Note on the active 5-axis transformation (TRAORI)**

With active 5-axis transformation, the "Swiveling in JOG" input screen form fields below are displayed:

- Name: 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 Z
- Rotation about Y
- Rotation about X

**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]).



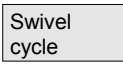
If the rotary axes are positioned, they are also traversed with G0 and conventional rapid traverse for JOG (MD 32010 \$MA\_JOG\_VELO\_RAPID[AX]). Only the TOOLCARR.SPF cycle is used for traversing, which can be adjusted by the machine manufacturer.

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., with G17, C3=1).

### 3.16.7 Startup of swivel cycle CYCLE800



Entry

soft key  →  → 

Only available if CYCLE800 startup menu has been activated.



When CYCLE800 is started up, the data (swivel data record) is set in the tool data \$TC\_CARR1....

This is grouped to form Swivel startup menus.

→ "Startup" operating area; "Swivel cycle" soft key



**References:** For current information see:

- The "siemensd.txt" file on the delivery 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" (3/2 axes inclined machining)

The following cycles must be loaded:

- CYCLE800.SPF, CYCPE\_SC.SPF (standard cycles)
- TOOLCARR.SPF (standard cycle)
- PROG\_EVENT.SPF (manufacturer cycle)<sup>1)</sup>

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 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.



#### 1) Note on PROG\_EVENT.SPF cycle

The PROG\_EVENT.SPF cycle is supplied as a manufacturer cycle and is used to preposition rotary axes following a block search. The machine manufacturer can expand PROG\_EVENT functionality by adding separate manufacturer cycles. CYCPE\_MA.SPF should be used as the name of the manufacturer cycle. A corresponding exit marker should be prepared in PROG\_EVENT.SPF for the CYCPE\_MA.SPF manufacturer cycle.

**Activating “Swiveling in JOG” (with Cycles SW 6.5 and higher)**

Horizontal soft key 8 is the “Swiveling in JOG” entry soft key in machine/JOG mode.

Soft key 8 is activated in the MA\_JOE.COM file, which is located in the “Standard cycles” directory.

Delete the “;” in front of the following lines in the MA\_JOE.COM file:

```
HS5= ($82119, , se1)
;PRESS (HS8)
;LM ("SWIVEL3", "jog_c800.com")
;END_PRESS
```

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.

For this, MD24006 \$MC\_CHSFRAME\_RESET\_MASK BIT 4 = 1 must be set.



### Machine data

To use the swiveling cycle, the machine data below must be set as follows (minimum requirement):

- Machine data with an exact value assignment (G)  
→ Machine data that must not be changed
- Machine data with a variable value assignment (V)  
→ Machine data, for which the default value can be set to a higher or lower value.

MD number	MD identifier	Value	Comment	Modifiable
10602	\$MN_FRAME_GEOAX_CHANGE_MODE	1	<sup>1)</sup>	V
11450	\$MN_SEARCH_RUN_MODE	Bit 1=1	Activate PROG_EVENT after block search	G
18088	\$MN_MM_NUM_TOOL_CARRIER	n>0	n → Number of swivel data records <sup>1)</sup>	G
20110	\$MC_RESET_MODE_MASK	'H4041'	Bit 14=1	G
20112	\$MC_START_MODE_MASK	'H400'	-	G
21100	\$MC_ORIENTATION_IS_EULER	0	Angles of rotations are interpreted as RPY	G
21110	\$MC_X_AXIS_IN_OLD_X_Z_PLANE	1		G
20126	\$MC_TOOL_CARRIER_RESET_VALUE	0...n	Described in CYCLE800	V
20150	\$MC_GCODE_RESET_VALUES[41]	1	TCOABS <sup>1)</sup>	G
20150	\$MC_GCODE_RESET_VALUES[51]	2	PAROT <sup>1)</sup>	G
20150	\$MC_GCODE_RESET_VALUES[52]	1	TOROTOF <sup>1)</sup> (for kinematics types T and M only)	V
20152	\$MC_GCODE_RESET_MODE[41]	0	(Default) <sup>1)</sup>	G
20152	\$MC_GCODE_RESET_MODE[51]	0	(Default) <sup>1)</sup>	V
20152	\$MC_GCODE_RESET_MODE[52]	0	(Default) <sup>1)</sup>	V
20180	\$MC_TOCARR_ROT_ANGLE_INCR[0]	0	(Default) <sup>1)</sup>	G
20180	\$MC_TOCARR_ROT_ANGLE_INCR[1]	0	(Default) <sup>1)</sup>	G
20182	\$MC_TOCARR_ROT_ANGLE_OFFSET[0]	0	(Default) <sup>1)</sup>	G
20182	\$MC_TOCARR_ROT_ANGLE_OFFSET[1]	0	(Default) <sup>1)</sup>	G
20184	\$MC_TOCARR_BASE_FRAME_NUMBER	-1	(Default) <sup>1)</sup>	G
22530	\$MC_TOCARR_CHANGE_M_CODE	0	<sup>1)</sup>	V
24006	\$MC_CHSFRAME_RESET_MASK	Bit 4 = 1	If system frame \$P_WPFRAME is to remain active after a reset	V
24007	\$MC_CHSFRAME_RESET_CLEAR_MASK	Bit 2,3=0		G
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 \$P_WPFRAME, \$P_TOOLFRAME and \$P_PARTFRAME	G

1) For note on machine data, see next pages.





MD number	MD identifier	Value	Comment	Modifiable
28083	\$MC_MM_SYSTEM_DATAFRAME_MASK	Bit 4, 3, Default='H7F' 2=1		G
30455	MISC_FUNCTION_MASK	Bit 2, 0=1	For rotary axes defined as modulo axes <sup>1)</sup>	V
32010	\$MA_JOG_VELO_RAPID[AX]		Rapid traverse in JOG mode for swiveling in JOG, with Cycles SW 6.5 and higher. Machine data is for retracted axes and relates to rotary axes.	V



### Setting data

SD no.	SD identifier	Value	Comment	Modifiable
42980	\$SC_TOFRAME_MODE	2000	See notes	V

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/ Installation & Start-Up Guide HMI



**For notes on machine and setting data, see this page and next.**

#### Note on MD 10602:

If the ZO is not to be reprogrammed following TRAORI, e.g., for tool correction.

#### Note on 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.

#### Note on MD 20180/MD 20182:

For rotary axes with the Hirth tooth system, the relevant values are entered in the CYCLE800 startup menu.

### 3.16 Swiveling - CYCLE800

**Note on 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

By outputting M commands, the PLC can, for example, limit or invert the spindle speed.

**Note on MD 30455**

This allows the rotary axis to traverse for G90 with DC (shortest path); see TOOLCARR.SPF user cycle.

**Note on 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 and on machine data MD 21110: \$MC\_X\_AXIS\_IN\_OLD\_X\_Z\_PLANE=1 (default).

For kinematics (swivel head/mixed kinematics) with the Hirth tooth system, set setting data SD 42980: \$SC\_TOFRAME\_MODE=2000.

To ensure that the compensating frame (TOOLFRAME) is retained after a reset or end of program, enter the following value 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

The CYCLE800 startup process is supported by the following menu displays:

### Startup of kinematic chain

Kinematic channel1			
Kinematics	<b>Inclinable head</b>	Name:	HEAD_1 No.: 1
Retract:	X	Y	Z
Retract position			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 opt.			
Swivel mode:	axial +projection angle		
Direction:	Rot. axis 1		
Tracking TL	No		

You must set up a swivel data record for every swivel head, swivel table or combination of the two. Swivel data records can be declared in several channels.

The number of swivel data records is limited by the following machine data:

- MD 18088: MM\_NUM\_TOOL\_CARRIER or
- MD: NUM\_CHANNELS (number of channels option)
- MD 28085: MM\_LINK\_TOA\_UNIT

The swivel data record is assigned to the tool data by \$TC\_CARR1[n] to \$TC\_CARR40[n] or, with NCU SW 7.1 or higher, to \$TC\_CARR65[n].

The meaning of the parameters in the "Kinematics" screen form is as follows:

**Name:** Swivel data record \$TC\_CARR34[n]

n → No. of swivel data record

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 +/-).



Swivel data record names may only contain permitted NC programming characters (A...Z, 0...9 and \_)!

**Kinematic type** \$TC\_CARR23[n]

- Swivel head (type T)
- Swivel table (type P)
- Swivel head + swivel table (type M)

**Retraction/retraction position**

\$TC\_CARR38[n] X; \$TC\_CARR39[n] Y; \$TC\_CARR40[n] Z

n → No. of swivel data record

The startup engineer determines whether the 'Retract Z axis' and 'Retract Z,X,Y axes' or 'Retract in tool direction' (with Cycles SW 6.5 and higher) may be included in the input menu for the swivel cycle.

The mode of retraction can be modified in the TOOLCARR.SPF user cycle (see Subsection 3.16.8), if necessary. If the TOOLCARR.SPF user cycle is not modified, the retraction position is approached as an absolute machine position.



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 rotary axis vectors (machine kinematics)**

\$TC\_CARR1[n] ... \$TC\_CARR20[n]

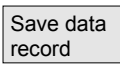
The vectors always contain 3 components, which represent the reference to the X, Y, Z machine axes.

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).

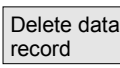
Offset vectors I1 to I4 refer to the **non-swiveled state of the rotary axes** (machine kinematics basic setting).

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.

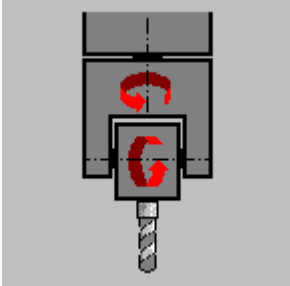
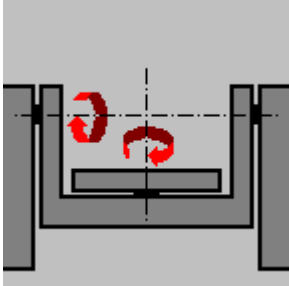
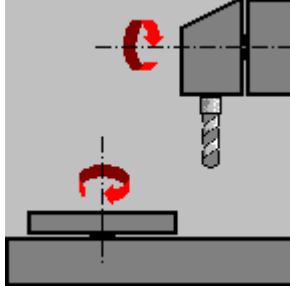
Manually adjustable rotary axes (manual mode) are possible with or without measuring systems and can be used with "plain machines".

Soft key 

Current swivel data record is saved as the parts program. The parts program corresponds to the name of the swivel data record.

Soft key 

Current swivel data record is deleted.

Swivel head (type T)	Swivel table (type P)	Swivel head + swivel table (type M)
		
Offset vector I1	Offset vector I2	Offset vector I1
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**.



#### References:

- Description of Functions 840D/840Di/810D
  - /W1/ "Tool offset"
  - (3/2 axes inclined machining)
- ISO 841-2001 or DIN 66217

#### Swivel head

- I3 distance from the tool adapter to the pivot point/intersection of the 2<sup>nd</sup> rotary axis
- I2 distance from the pivot point/intersection of the 2<sup>nd</sup> rotary axis to the pivot point/intersection of the 1<sup>st</sup> 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 1<sup>st</sup> rotary axis
- I3 distance from the pivot point/intersection of the 1<sup>st</sup> rotary axis to the pivot point/intersection of the 2<sup>nd</sup> 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 1<sup>st</sup> 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 2<sup>nd</sup> 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 and the rotary axis vectors result from the specifications of the axis directions according to ISO 841-2001 or DIN 66217 (right-hand rule).



### TOOLCARRIER data ↔ 5-axis transformation



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.

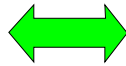
#### Swivel head (swivel-mounted tool)

\$TC\_CARR23[1]="T"

I1 \$TC\_CARR1...3[n]

I2 \$TC\_CARR4...6[n]

I3 \$TC\_CARR15...17[n]



\$MC\_TRAFO\_TYPE\_1=24

\$MC\_TRAFO5\_PART\_OFFSET\_1[0...2]

\$MC\_TRAFO5\_JOINT\_OFFSET\_1[0...2]

\$MC\_TRAFO5\_BASE\_TOOL\_1 [0...2]

Close I1=-**(I2+I3)** vector chain<sup>1)</sup>

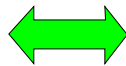
Swivel table (swivel-mounted workpiece)

\$TC\_CARR23[1]="P"

I2 \$TC\_CARR4...6[n]

I3 \$TC\_CARR15...17[n]

I4 \$TC\_CARR18...20[n]



\$MC\_TRAFO\_TYPE\_1=40

\$MC\_TRAFO5\_BASE\_TOOL\_1 [0..2]

\$MC\_TRAFO5\_JOINT\_OFFSET\_1[0...2]

\$MC\_TRAFO5\_PART\_OFFSET\_1[0...2]

Close I4=-**(I2+I3)** vector chain<sup>1)</sup>

#### Swivel head + swivel table (swivel-mounted tool + workpiece)

\$TC\_CARR23[1]="M"

I1 \$TC\_CARR1...3[n]

I2 \$TC\_CARR4...6[n]

I3 \$TC\_CARR15...17[n]

I4 \$TC\_CARR18...20[n]



\$MC\_TRAFO\_TYPE\_1=56

\$MC\_TRAFO5\_JOINT\_OFFSET\_1[0...2]

\$MC\_TRAFO5\_BASE\_TOOL\_1 [0...2]

\$MC\_TRAFO5\_JOINT\_OFFSET\_PART\_1[0...2]

\$MC\_TRAFO5\_PART\_OFFSET\_1[0...2]

Close I1=-I2 I4=-I3 vector chain<sup>1)</sup>



1) With fixed-mounted machine kinematics

### Display options

- **\$TC\_CARR37[n]** (n → swivel data record)

If the relevant display options are not set, the value will not be displayed in the input screen form (see Subsection 3.16.1).

8	7	6	5	4	3	2	1	0	(Decimal places)
								0:	Axis by axis
								1:	Axis by axis + angle of projection
								2:	Axis by axis + angle of projection + solid angle
									Rotary axis 1
								0:	Automatic
								1:	Manual
									Rotary axis 2
								0:	Automatic
								1:	Manual
									Selection of preferred axis direction
								0:	No
								1:	Reference to rotary axis 1
								2:	Reference to rotary axis 2
									Correcting tool tip
								0:	No
								1:	Yes
									Reserved
									Retraction mode
								0:	Z axis
								1:	Z axis or ZXY axis
									Retracting in tool directions (with Cycles SW 6.5 and and higher) <sup>3)</sup>
								(left)	(right)
								Z	Max. tool direction
								Z, X, Y	Inc. tool direction
								Z + Z, X, Y	Max. + inc. tool direc.
									Swivel data record change / tool change <sup>1)</sup>
								0:	No
								1:	Manual
								2:	Automatic
								3:	No
								4:	Manual
								5:	Automatic
									2)
									Automatic
									Automatic
									2)
									Manual
									Manual

1) Only relevant for ShopMill/ShopTurn.

2) If no swivel data record change is declared, the automatic/manual tool change setting is irrelevant.

3) For retraction mode coding, see table below.

**Coding of retraction modes Z, ZXY, maximum, incremental tool direction**

Z	Z, X, Y	Max. tool direction	Inc. tool direction	Coding of \$TC_CARR37 xXXxxxxxx
1	0	0	0	00
1	1	0	0	01
0	1	0	0	02 <sup>1)</sup>
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

1) With Cycles SW 6.5 and higher, codings 02 to 15 apply



The following display options affect the input screen form for the swivel cycle:

- **Swivel mode**
  - **Axis by axis**
  - **Axial and projection angle**
  - **Axial and projection angle and solid angle**

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**

When the direction of travel is selected on the swivel cycle input screen form, reference to **rotary axis 1** or **2**.

Because of the angular range of the **rotary axes** of machine kinematics, the NCU calculates 2 possible solutions. Usually, one of these solutions is technologically suitable.

- The **rotary axis**, to which the two solutions must refer, is selected in the startup menu.
- The solution to be applied is selected in the input screen form for the swivel cycle.
- With **no**, the direction parameter is not displayed on the operating screen.



- **Correct tool**

→ No

→ Yes

"Correct tool" displayed in the swivel cycle input screen form.

The correct tool function requires the 5-axis transformation (TRAORI) option.



See TOOLCARR.SPF programming example.

### Starting up rotary axis parameters

Rotary axis channel1					
Kinematics	Inclinable head	Name:	HEAD_2	No.:	2
<b>Rot. axis 1</b>	Identifier	<input type="text" value="B"/>	Mode	Manual	
	Angle area	<input type="text" value="0.000"/>	-	<input type="text" value="360.000"/>	deg
	Hirth teeth	<input type="text" value="Yes"/>	Angle offset	<input type="text" value="0.000"/>	deg
	Aut.override	<input type="text" value="No"/>	Angle grid	<input type="text" value="2.500"/>	deg
<b>Rot. axis 2</b>	Identifier	<input type="text" value="C"/>	Mode	Manual	
	Angle area	<input type="text" value="-90.000"/>	-	<input type="text" value="90.000"/>	deg
	Hirth teeth	<input type="text" value="Yes"/>	Angle offset	<input type="text" value="0.000"/>	deg
	Aut.override	<input type="text" value="No"/>	Angle grid	<input type="text" value="2.500"/>	deg
	Swivel data record change	<input type="text" value="No"/>			
	Tool changing	<input type="text" value="automatically"/>			



### Input of data for rotary axes 1 and 2 that are relevant to the swivel cycle

Name/kinematics → see CYCLE800 "Kinematics" startup menu

#### Axis identifiers of the rotary axes

\$TC\_CARR35[n] rotary axis 1

\$TC\_CARR36[n] rotary axis 2

The following identifiers should be chosen, where possible:

Axis rotates about machine axis X --> A

Axis rotates about machine axis Y --> B

Axis rotates about machine axis Z --> 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 designation (up to 6 letters or numbers).

**Mode**

\$TC\_CARR37[n] see display options

- **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 angles 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., 1<sup>st</sup> rotary axis automatic, 2<sup>nd</sup> rotary axis manual) and "incomplete" machine kinematics (e.g., 1<sup>st</sup> rotary axis rotates about axis X) are permitted. If machine kinematics are to be implemented with just one rotary axis, this must always be declared as the 1st rotary axis.



For the swivel angle display, see display messages  
CYCLE800 → 62180/62181

**Angular range**

\$TC\_CARR30[n] .. \$TC\_CARR33[n]

A valid angular range 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, enter a traversing range of between **0 and 360 degrees**.

**Offset kinematics (with Cycles SW 6.3 and higher)**

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]

**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 on SW 6.3 and higher)

**Swivel data record change (only relevant for ShopMill/ShopTurn)**

- **No**
- **Automatic**
- **Manual**

**Tool change (only relevant for ShopMill/ShopTurn)**

- **Automatic**
- **Manual**

"Tool change" display for kinematic type T and M only

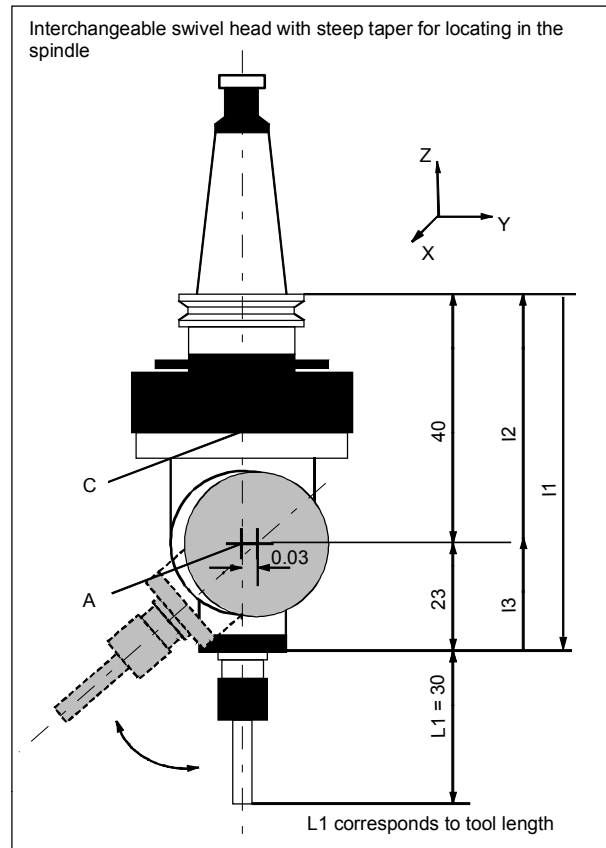


### Startup examples for machine kinematics

#### 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)



#### Swivel head kinematics setup

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		

#### Rotary axes

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 about Y **and** about Z

Offset vector I1: Closure of vector chain with fixed-mounted swivel head  $I1 = -(I2 + I3)$

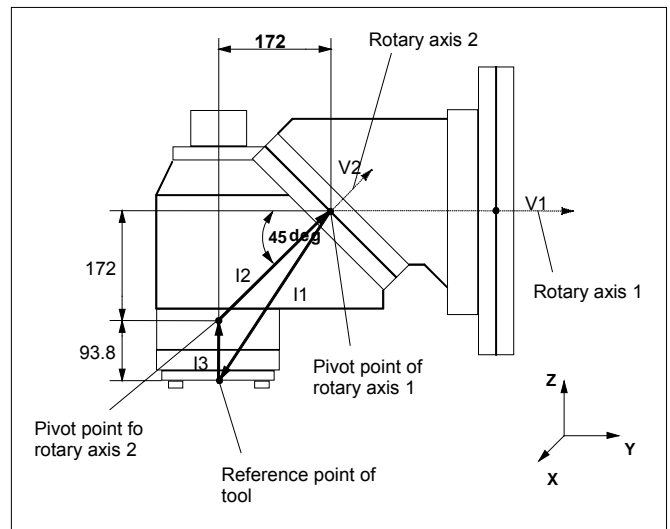
Offset vector I2: Distance between pivot point of **rotary axis 1** and pivot point of **rotary axis 2**

Offset vector I3: Distance between the reference point of the tool and the 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).



### Swivel head kinematics setup

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
Display options			
Swivel mode	Axis by axis		
Direction	Rotary axis 2		
Correct tool	No		

### Rotary axes

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		
Hirth tooth system	Yes	Angular grid	1.000



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

$$V_{2y} = \sin(45) = 0.7071$$

$$V_{2z} = \cos(45) = 0.7071$$

V<sub>2y</sub> and V<sub>2z</sub> can be normalized to 1.



### Example 3: Cardanic table "TABLE\_45"

**Vectors relate to the basic setting of the kinematics.**

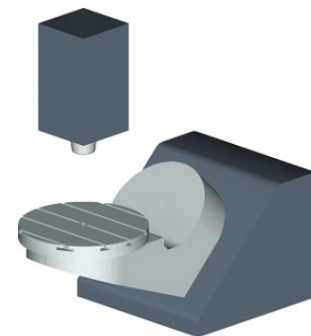
Rotary axis vector V1: Rotary axis B rotates about Y and about Z.

Rotary axis vector V2: Rotary axis C rotates about Z.

Offset vector I2: Distance from the machine reference point to the pivot point/intersection of **rotary axis 1**

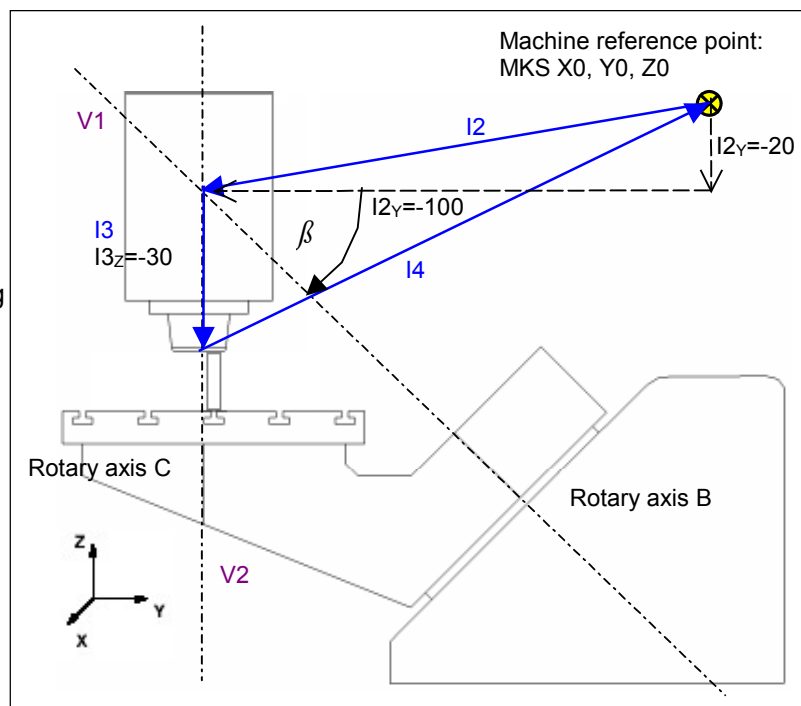
Offset vector I3: Distance from the pivot point/intersection of **rotary axis 1** to the pivot point/intersection of **rotary axis 2**

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.



### Swivel head kinematics setup

Kinematics	Swivel table		TABLE_45
	X	Y	Z
Offset vector I2	0.000	-100.000	-20.000
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	50.000
Display options			
Swivel mode	Axis by axis		
Direction	Rotary axis 2		
Correct tool	No		

### Rotary axes

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

$$V1_y = \sin(-45) = -0.7071$$

$$V1_z = \cos(-45) = 0.7071$$

$V1_y$  and  $V1_z$  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 about Y and about Z.

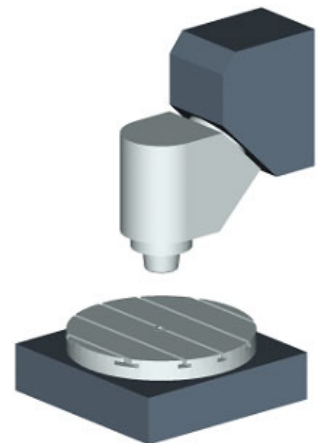
Rotary axis vector V2: Rotary axis C rotates about Z.

Offset vector I2: Distance from the reference point of the tool adapter to the pivot point/intersection of **rotary axis 1**

Offset vector I1: Closure of vector chain I1=-12

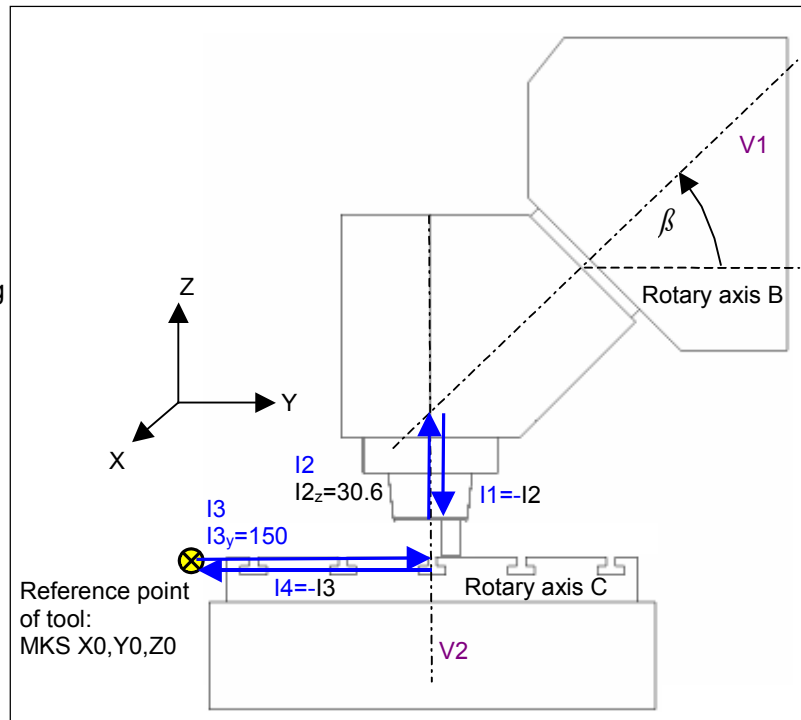
Offset vector I3: Distance from reference point of the machine to the pivot point/intersection of **rotary axis 2**

Offset vector I4: Closure of vector chain I4=-12



**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.

**Swivel head kinematics setup**

Kinematics	Swivel table		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	Yes		

**Rotary axes**

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

$$V1_y = \sin(45) = 0.7071$$

$$V1_z = \cos(45) = 0.7071$$

$V1_y$  and  $V1_z$  can be normalized to 1.





### Example 5: Swivel table "TABLE\_5"

Vectors relate to the basic setting of the kinematics.

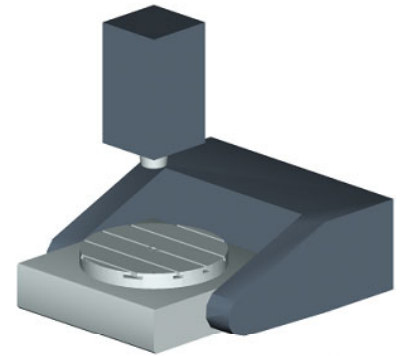
Rotary axis vector V1: Rotary axis A rotates about X.

Rotary axis vector V2: Rotary axis C rotates about Z.

Offset vector I2: Distance from the machine reference point to the pivot point/intersection of **rotary axis 1**

Offset vector I3: Distance from the pivot point of rotary axis 1 to the pivot point/intersection of **rotary axis 2**

Offset vector I4: Closure of vector chain  $I4 = -(I2 + I3)$

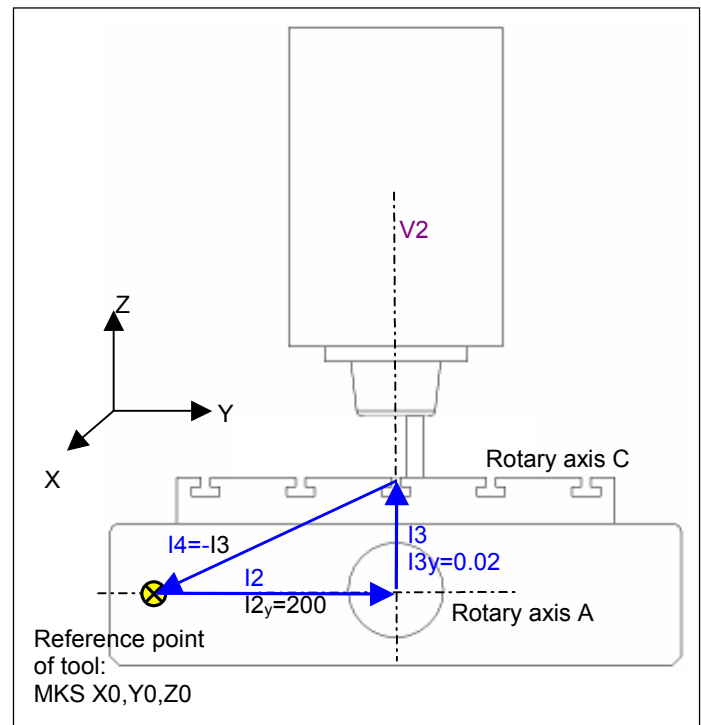


#### Side view of the machine

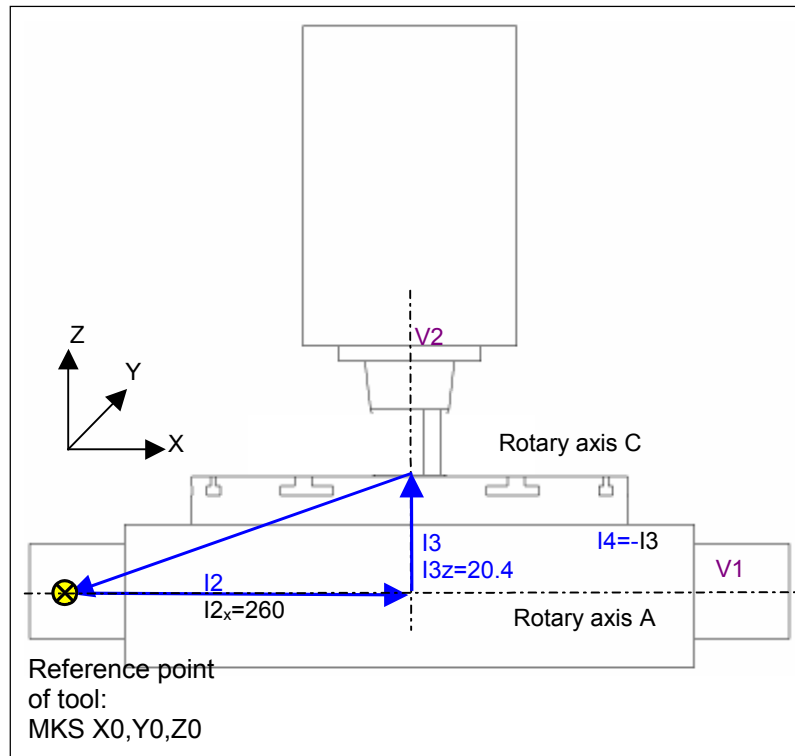
##### from the X direction:

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.



### Front view of the machine from the Y direction



### Swivel head kinematics setup

Kinematics	Swivel table		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		

### Rotary axes

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 TOOLCARR.SPF user cycle



#### Programming

TOOLCARR(\_MODE, \_TC1, \_A1, \_A2, \_TC2)



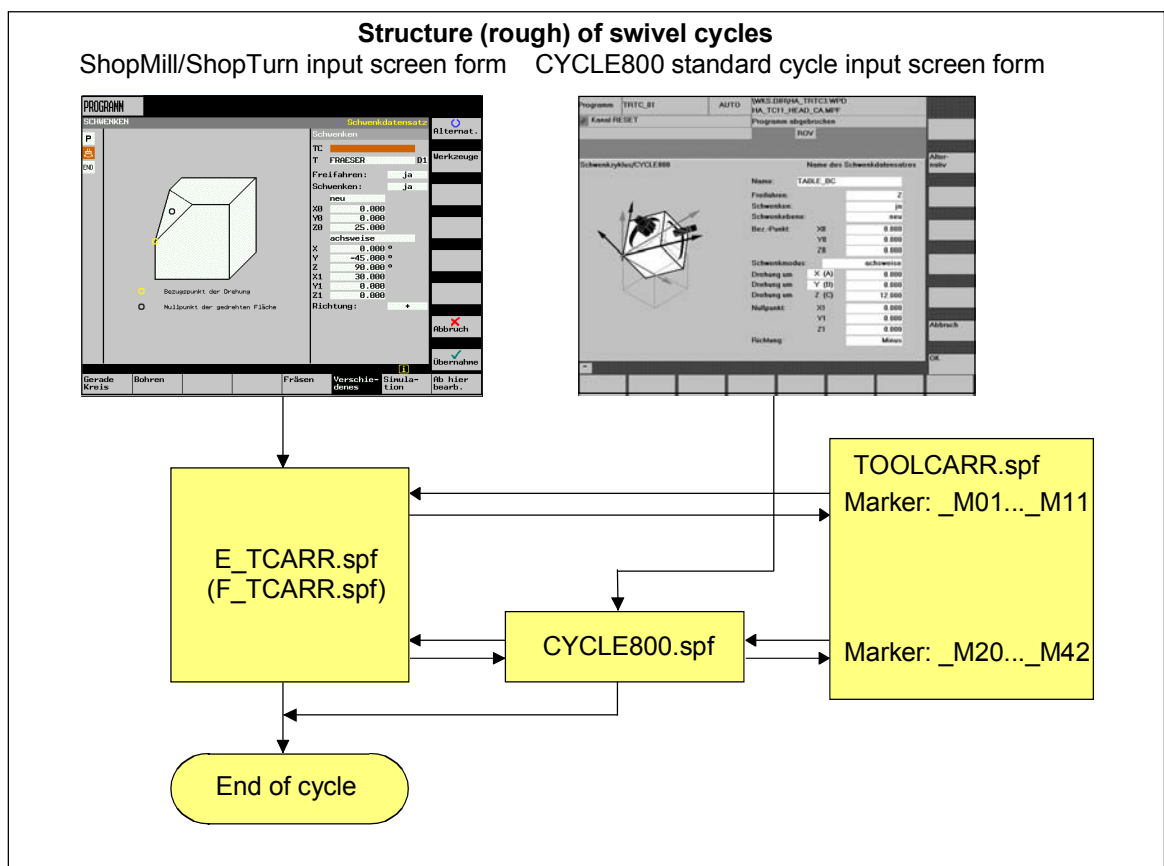
#### Parameters

_MODE	Corresponds to the markers of the structure described below
_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)



#### Customization by the machine manufacturer

During swiveling, all axis positions are traversed using the TOOLCARR.SPF cycle. This is either called from the CYCLE800 swivel cycle, 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 user cycle is not modified, the Z axis (marker \_M41) or Z axis is moved first during retraction before swiveling, followed by the X, Y axes (marker \_M42). The positions correspond to the CYCLE800 "Kinematics" → retraction positions startup menu. If INCH is the basic system of the machine, the TOOLCARR cycle must be modified accordingly.

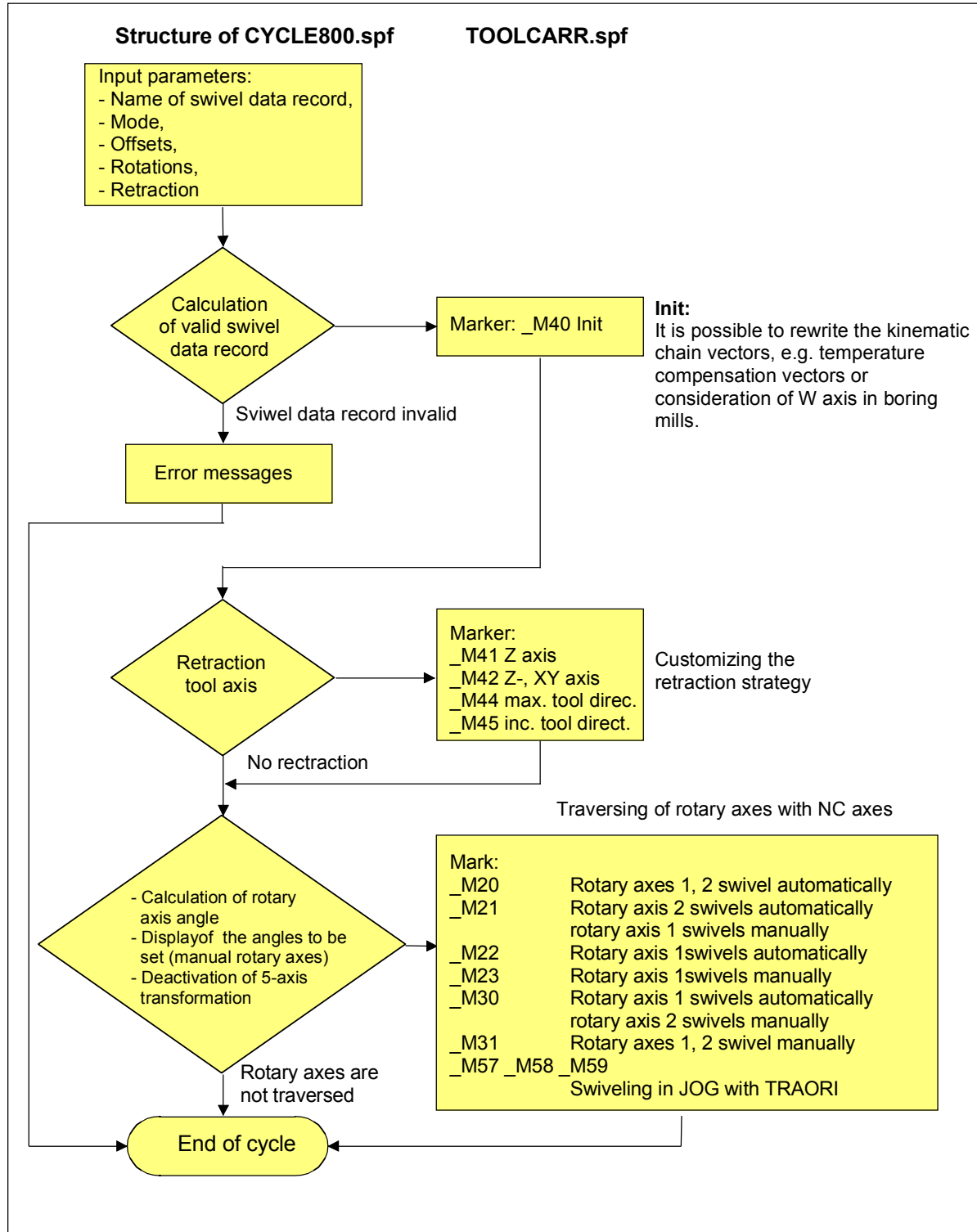


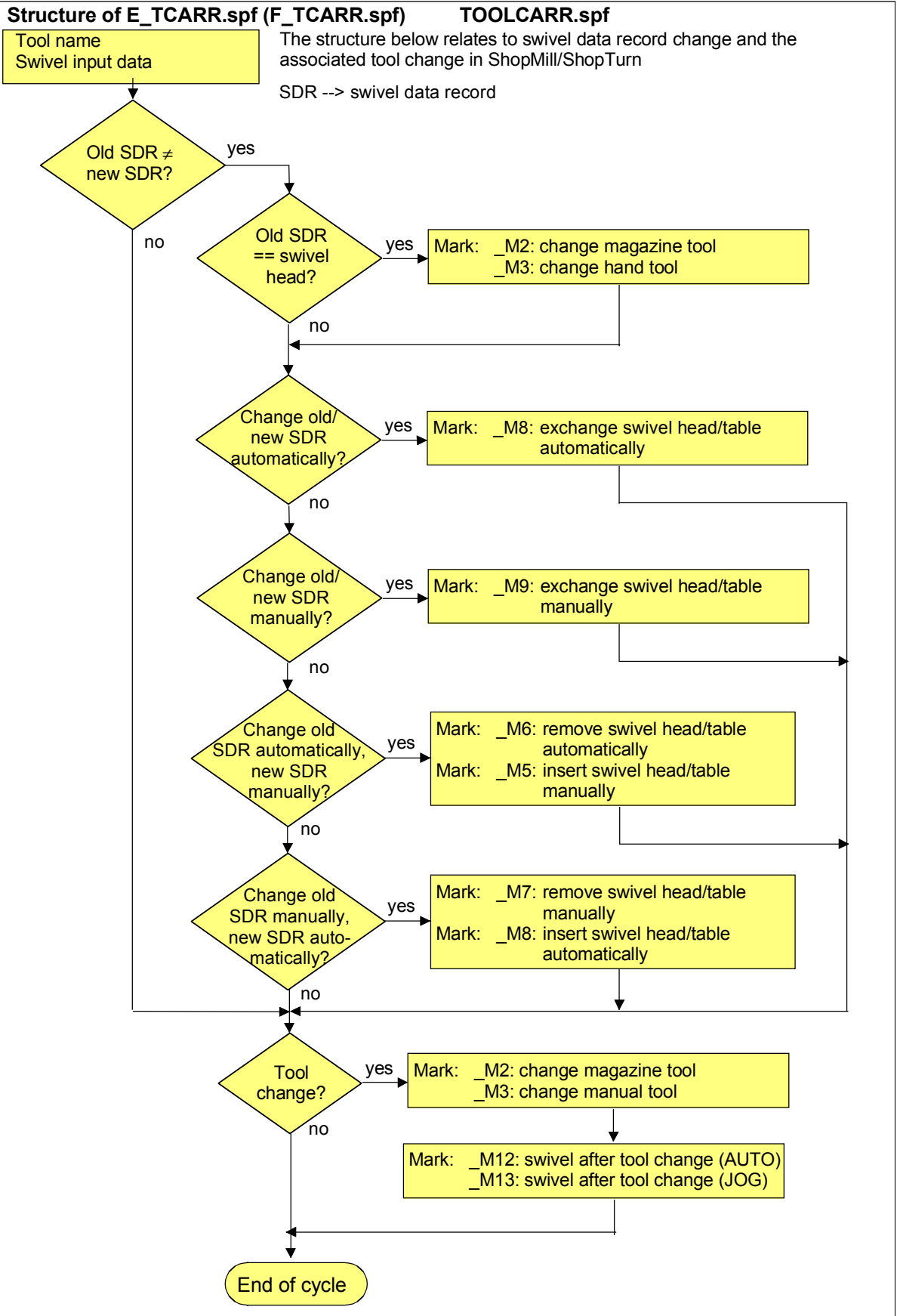
Note the machine manufacturer's instructions.

**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.





### 3.16 Swiveling - CYCLE800



#### Note on markers `_M20` to `_M37`

Markers `_M20` to `_M37` 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.



#### Note on ShopMill/ShopTurn

The `E_SWIV_H` or `F_SWIV_H` cycle is called in ShopMill/ShopTurn (see markers `_M2` to `_M9`) in the `TOOLCARR.SPF` user 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 labels.

If the rotary axes are to move to a certain position, an angle value can be transferred to parameters Par 2, Par 3.



#### Note on “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.

## 3.16.9 Error messages



## Explanation

## Alarm source CYCLE800

Alarm number	Alarm text	Explanation, remedy
61180	"No name assigned to swivel data record even though machine data \$MN_MM_NUM_TOOL_CARRIER > 1"	No name assigned to swivel data record even though several swivel data records exist (\$MN_MM_NUM_TOOL_CARRIER>0) or no swivel data record defined (\$MN_MM_NUM_TOOL_CARRIER=0)
61181	"NCK software version too old (no TOOLCARRIER functionality)"	TOOLCARRIER functionality with NCU 6.3xx and higher
61182	"Name of swivel data record unknown"	See Swivel cycle startup CYCLE800 → Kinematics Name (swivel data record)
61183	"Retraction mode GUD7_TC_FR outside value range 0...2"	See Swivel cycle startup CYCLE800 → Retraction; 1st transfer parameter CYCLE800(x,...) is faulty >2
61184	"No solution can be found with current angle inputs"	Or coding of _MODE parameter incorrect, e.g., axis by axis rotation YXY
61185	"Angle ranges declared for rotary axes missing or incorrect (min.>max.)"	Check startup of swivel cycle CYCLE800
61186	"Invalid rotary axis vectors → check startup of swivel cycle CYCLE800"	Startup of swivel cycle CYCLE800: missing or incorrect entry for rotary axis vector V1 or V2
61188	"No axis name declared for 1 <sup>st</sup> rotary axis → check startup of swivel cycle CYCLE800"	Startup of swivel cycle CYCLE800: missing entry under rotary axis 1 identifier
61189	"Swiveling not possible"	G500 active Check basic zero offset of rotary axes
61190	"Retracting in tool direction not possible"	See following page
61196	"No swiveling in JOG"	5-axis transformation and TOOLCARRIER activated simultaneously
61199	"Tool setting and swivel data record change (TOOLCARRIER) not permitted"	
62180	"Set rotary axes x.x [deg]"	Angles to be set for manual rotary axes
62181	"Set rotary axes x.x [deg]"	Angle to be set for manual rotary axis
62185	"Angle adapted to angular grid: %4"	%4 differential value with Hirth tooth system
62186	"No swiveling in JOG"	Active ZO G %4 and basic frames contain rotations %4 No. of active ZO
62187	"No swiveling in JOG"	Several active basic frames (G500) contain rotations

**3.16 Swiveling - CYCLE800****Note on 61190**

61190 Retracting in tool direction not possible --> error code: %4

Error code %4 = abcd

a=0xxx -> CYCLE800 \_FR parameter incorrect or retraction option not set up in CYCLE800 startup

a=1xxx -> No applicate present \$P\_AXN3

a=2xxx -> Maximum retraction travel incorrect, see GUD \_TC\_P[8]

a=3xxx -> Incremental retraction travel incorrect, see GUD \_TC\_P[8]

b= \_FR\*100 input parameter

cd=\$P\_TCARR37[] startup parameter (6<sup>th</sup>, 7<sup>th</sup> decimal place)

**Note on 62186 and 62187**

Using the \_TC\_FR GUD7 parameter to set activation of error messages 62186 and 62187:

100<sup>th</sup> place 0xx -> No error analysis of 62186 62187

1xx -> Error analysis 62186 – Active ZO G%4 and basic frames contain rotations

2xx -> Error analysis 62187 – Several active basic frames (G500) contain rotations

3xx -> Error analysis 62186 and 62187



Typical display of the swivel angle to be set for a manual rotary axis in CYCLE800:

---

62181	“Rotary axis B: set to 32.5 [deg]”
-------	------------------------------------

---



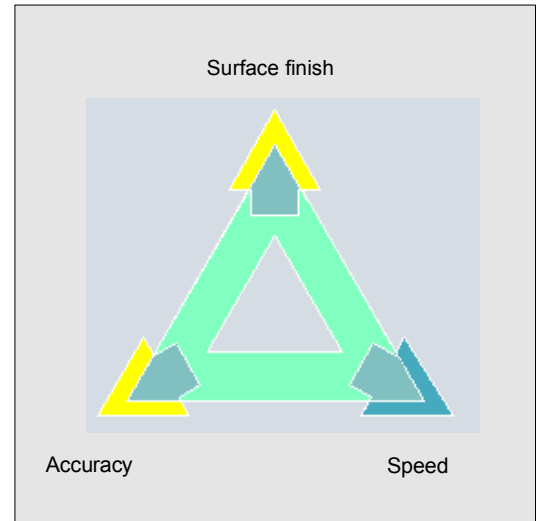
### 3.17 High-speed settings - CYCLE832 (with Cycles SW 6.3 and higher)



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:

- Technological support when machining free-form contours (surfaces) in the 3- or 5-axis high-speed machining range (High Speed Cutting - HSC)
- Predominantly used in the HSC milling range (possibility of using for turning or grinding)
- Groups together the important G codes and machine and setting data that are needed for HSC machining
- Separating 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** with great **accuracy** in the mm range 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**, **speed** is the priority due to smoothing the contour.

When **finishing**, **accuracy** is the priority.

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.

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, TRAFOF)<sup>1)</sup>
- B-SPLINE<sup>1)</sup>

1) Only if the relevant option is set.



## Function

The CYCLE832 cycle groups together the important G codes and the machine or setting data required for HSC machining. 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 relevant weighting. Different tolerances and settings (customizing technology) can be assigned to the four machining types.

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.



Note the machine manufacturer's instructions!

Typical 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

---



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.).

#### 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 Subsection 3.17.5) 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.

### 3.17.1 Call CYCLE 832 in HMI menu tree



#### Explanation of the parameters

Entry area Programs/Milling

soft key  →

is displayed.

CYCLE832 input screen form on standard interface

High Speed Settings		Machining selection	
Surface finish	Machining	Finishing	<input checked="" type="checkbox"/>
	Tolerance	_TOL	0.005
	Transformation		TRAORI
	Adjustment		Yes
	Compression		COMPCAD
	Path control		G642
	Feedforward control		FFWON SOFT

#### Machining (\_TOLM)

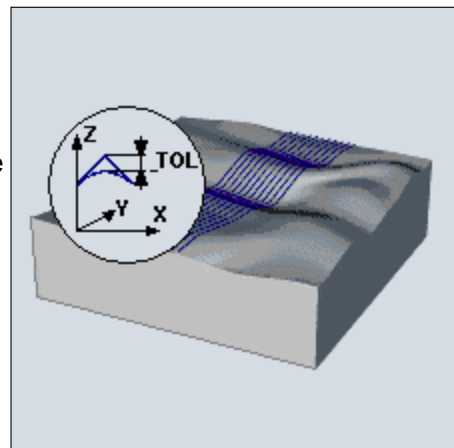
- Finishing (default)
- Rough-finishing
- Roughing
- Deselection

For the coding of variable \_TOLM, see Subsection 3.17.2, Parameters.

#### 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.) (see Subsection 3.1.7.5).

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.

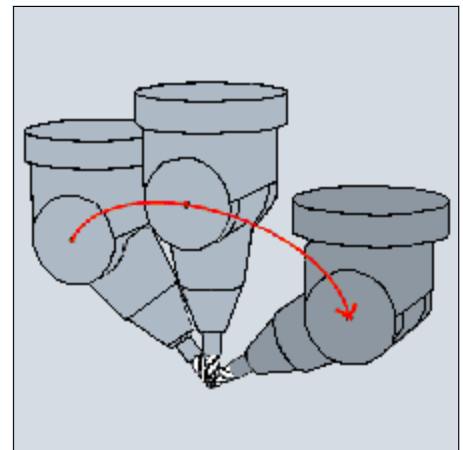
### Transformation (\_TOLM)

The transformation input field is only displayed when the NC option is set (5 axes machining package set).

- **No**
- **TRAORI** → 1. Activate 5-axis transformation
- **TRAORI(2)** → 2. Activate 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 the technology**

(see subsection 3.17.3)

- **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)**

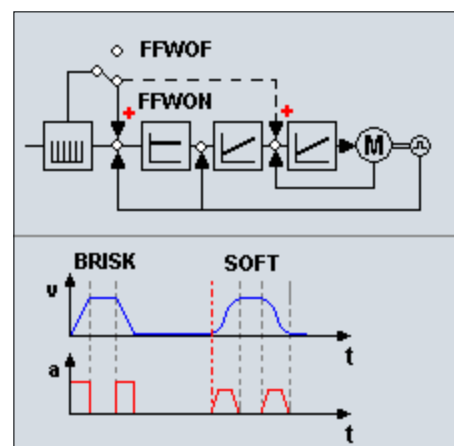
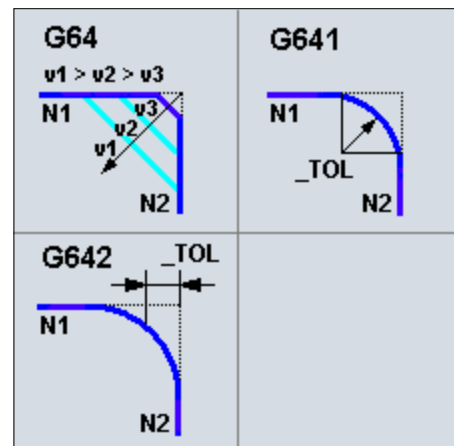
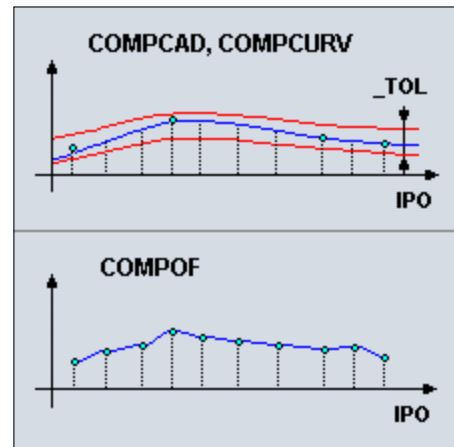
- **G642** (default)
- **G641**
- **G64**

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.



## 3.17.2 Parameters



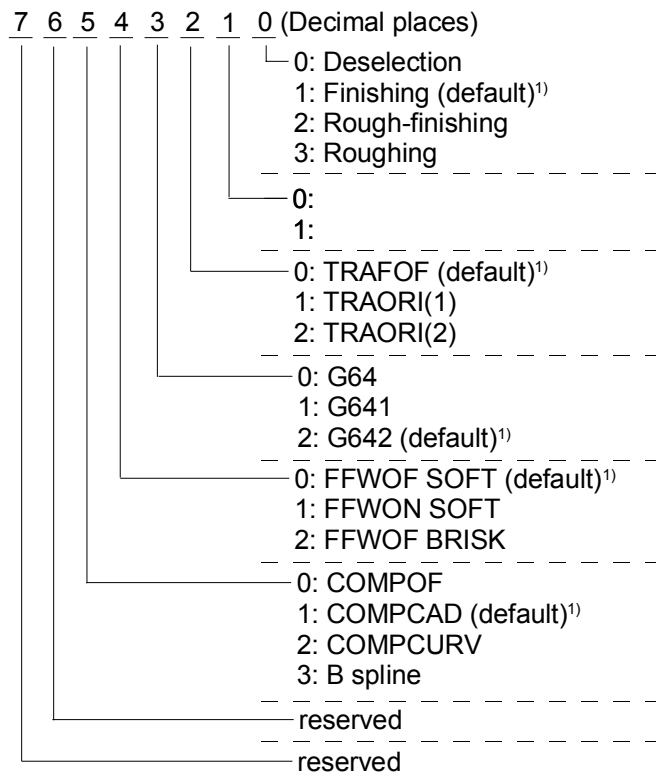
## Programming

CYCLE832(\_TOL, \_TOLM)



## Parameters

_TOL	real	Tolerance of machining axes → unit. mm/inch; deg
_TOLM	integer	Tolerance mode



1) Machine manufacturer can change the setting, see "Customizing technology" section.

### 3.17.3 Customizing technology



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.

#### Customization by the machine manufacturer

Prerequisites:



- **The manufacturer password is set,**
- "Customizing technology" input field → "Yes"

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

1. 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:

#### Computation of the velocity jumps overload factor of all the 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 CYCLE832.

2. 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.

#### Customization by the machine setter/programmer

Prerequisites:



- **The manufacturer password is deleted**
- Keyswitch position 2 or 3
- Protection level 5, 4, 3, 2
- "Customizing technology" input variable → "Yes"

In order to customize the technology, the machine setter/programmer must have precise knowledge of the subsequent CAM machining program.

The modified data is used to generate CYCLE832 and applies to the current call of CYCLE832.

The machine manufacturer's default settings are not altered.

### 3.17.4 Customizing additional program parameters CYC\_832T



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.



#### Parameters

CYC\_832T(\_ASVS, \_FACTOR, \_OVL\_on)

_ASVS	Machine-specific customization is possible at markers _M0 to _M4. _M4 = Init CYCLE832 call made before the actual program execution 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 adjust MD \$MA_MAX_ACCEL_OVL_FACTOR overload factor

1) The \_FACTOR and \_OVL\_on parameters are only effective at marker \_M4(INIT).



#### Programming example

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).

```

%_N_CYC_832T_SPF
; $PATH=/_N_CST_DIR).
PROC CYC_832T(INT _ASVS,VAR INT
_FACTOR,VAR INT _OVL_on) SAVE DISPLOF
...
N801 CASE _ASVS OF 0 GOTOF _M0 1 GOTOF
_M1 2 GOTOF _M2 3 GOTOF _M3 4 GOTOF
_M4 DEFAULT GOTOF _MEND
...
_M4:
_FACTOR=12 ;Init
GOTOF _MEND

_M0: ;Deselection
$MC_MAX_PATH_JERK=1000
$MA_MAX_AX_JERK[X]=15
$MA_MAX_AX_JERK[Y]=15
$MA_MAX_AX_JERK[Z]=15
GOTOF _MEND

_M1: ;Finishing
_M2: ;Rough-finishing
_M3: ;Roughing
$MC_MAX_PATH_JERK=15
$MA_MAX_AX_JERK[X]=150
$MA_MAX_AX_JERK[Y]=150
$MA_MAX_AX_JERK[Z]=150

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 Interfaces



#### 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 no.	MD identifier	Comment
10071	\$MN_IPO_CYCLE_TIME	IPO cycle
20480	\$MC_SMOOTHING_MODE	
20482	\$MC_COMPRESSOR_MODE	
24100	5-axis transformation machine data	
to 24462		

The following machine data is rewritten in CYCLE832:

MD no.	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	Corresponds to the tolerance of the linear axes
42466	\$SC_SMOOTH_ORI_TOL	Corresponds to the tolerance of the rotary axes
42475	\$SC_COMPRESS_CONTUR_TOL	For COMPCURV only
42476	\$SC_COMPRESS_ORI_TOL	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. 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
<code>_TOLT2 [2]</code>	STRING[32]	„ (default)	Program name of subroutine for calling the 5-axis transformation
<code>_TOLT [4]</code>	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 <code>_TOLM</code> variable (see parameters)
<code>_TOLV [4]</code>	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: 0.01 deselection (GUD7.def) 0.01 finishing 0.05 rough-finishing 0.1 roughing

### 3.17.6 Error messages



#### Explanation

##### Alarm source CYCLE832

Alarm number	Alarm text	Explanation, remedy
61191	"5-axis transformation not set up"	1. Machining package five axes or multi-axis interpolation option not set. 2. Check MD 24100: \$MC_TRAFO_TYPE_1 to \$MC_TRAFO_TYPE_8 for a valid 5-axis transformation type
61192	"Second 5-axis transformation not set up"	
61193	"Compressor option not set up"	Set spline interpolation option (A-, B- and C-splines/compressor function)
61194	"Spline interpolation option not set up"	

### 3.18 Engraving cycle CYCLE60 (with Cycles SW 6.4 and higher)

#### 3.18 Engraving cycle CYCLE60 (with Cycles SW 6.4 and higher)



##### Programming

CYCLE60 (\_TEXT, \_RTP, \_RFP, \_SDIS, \_DP, \_DPR, \_PA, \_PO, \_STA, \_CP1, \_CP2, \_WID, \_DF, \_FFD, \_FFP1, \_VARI, \_CODEP)



##### Parameters

_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>
_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 (equiv. _VARI hundred thousands digit) <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

## 3.18 Engraving cycle CYCLE60 (with Cycles SW 6.4 and higher)

_VARI	integer	<p>Type of machining: (enter without sign)</p> <p>UNITS DIGIT: Reference point Values: 0...Rectangular (Cartesian) 1...Polar</p> <p>TENS DIGIT: Text orientation Values: 0...Text in a line 1...Text on an upward arc 2...Text on a downward arc</p> <p>HUNDREDS DIGIT: Reserved</p> <p>THOUSANDS DIGIT: Reference point of the text horizontal Values: 0...Left 1...Central 2...Right</p> <p>TEN THOUSANDS DIGIT: Reference point of the text vertical Values: 0...Bottom 1...Central 2...Top</p> <p>HUNDRED THOUSANDS DIGIT: Text width Values: 0...Character spacing 1...Overall text width (linear text only) 2...Arc angle (only for arced text)</p> <p>7th digit from right (million): circle center point Values: 0...Rectangular (Cartesian) 1...Polar</p>
_CODEP	integer	<p>Code page number for character set entered 1252...Code page for Central European languages (value 0 is internally treated as equivalent to 1252)</p>

### 3.18 Engraving cycle CYCLE60 (with Cycles SW 6.4 and higher)

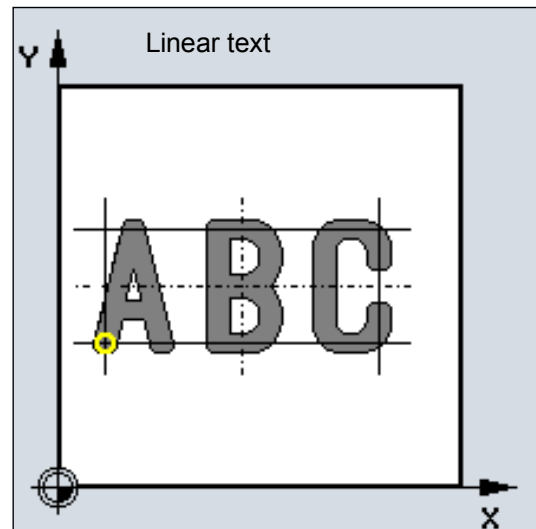


#### Function

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.

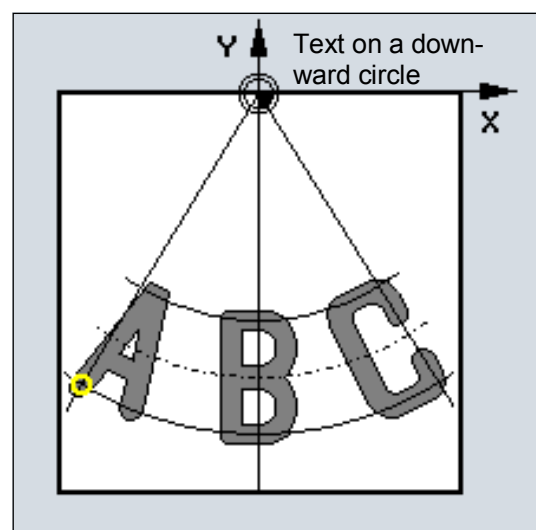
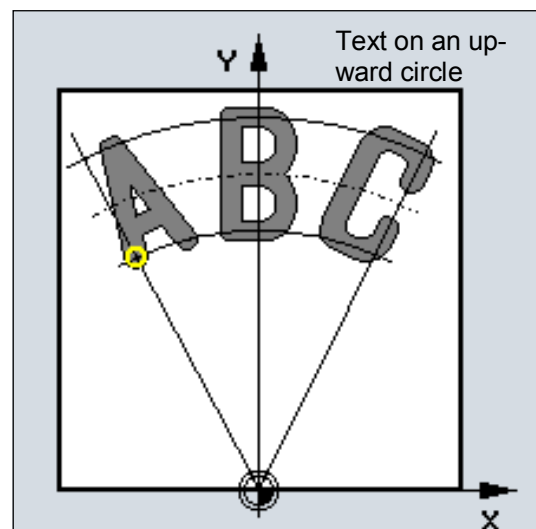


#### Sequence

Position reached before the beginning of the cycle:  
The starting position can be any position, from which the starting position of the first character can be approached without collision.

#### The cycle implements the following motion sequence:

- 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 federate 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



See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters `_RTP`, `_RFP` and `_SDIS`.

#### **`_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 "Non-existent character".

#### **`_DP`, `_DPR` (character depth)**

The character depth can be specified as either an absolute value (`_DP`) or incremental value (`_DPR`) in relation 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.

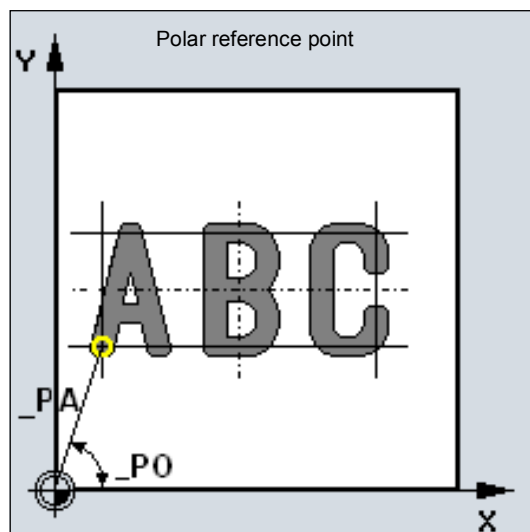
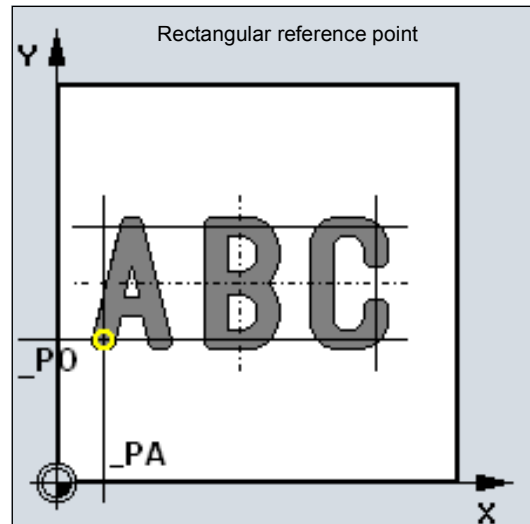
The character depth is reached with infeed; it is not divided into separate infeed steps. Depth infeed for each character is vertical with G1.

### 3.18 Engraving cycle CYCLE60 (with Cycles SW 6.4 and higher)

#### **\_PA, \_PO (reference point for text orientation)**

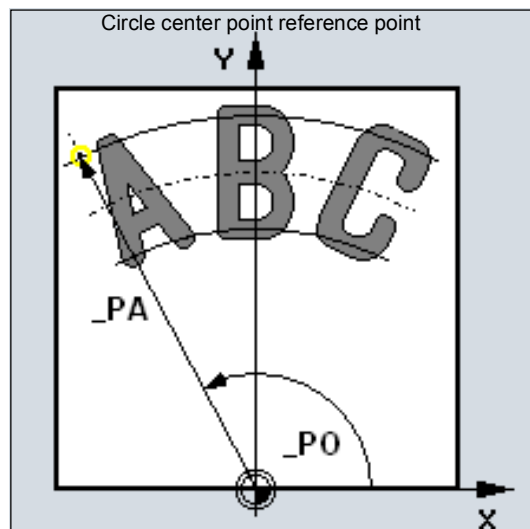
Irrespective of the linear or circular orientation of the text, the reference point can either be programmed as rectangular (Cartesian) or polar.

If the text is of a linear orientation, the reference point is always based on the current workpiece zero.



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`.

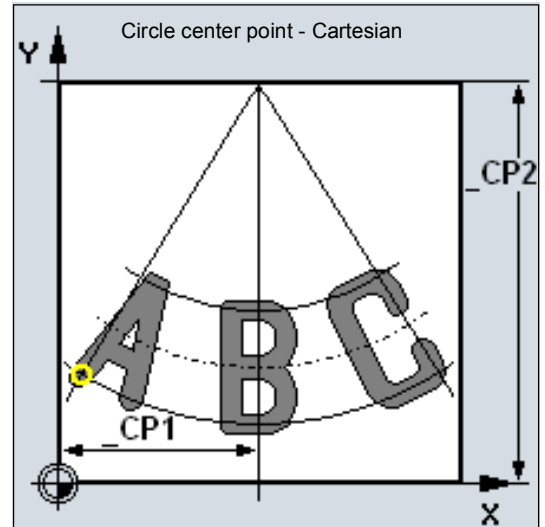


**\_CP1, \_CP2 (circle center point)**

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.

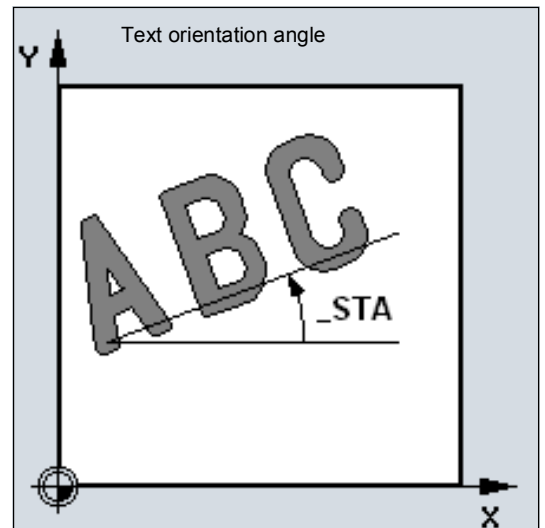
**\_STA (text orientation angle)**

`_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.

**\_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.

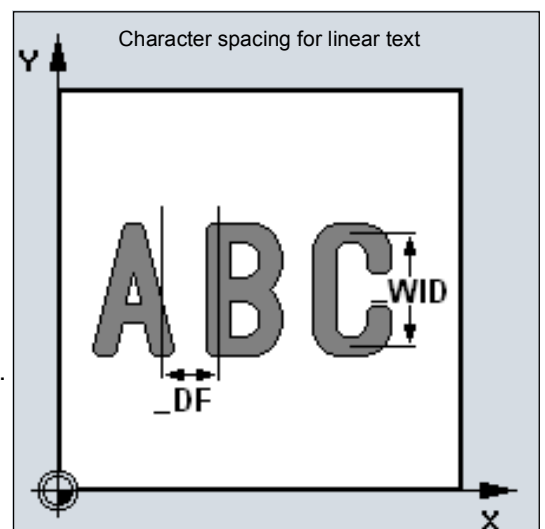
**\_DF (character spacing)**

For linear text, either the character spacing or the overall text width may be specified. These are incremental values.

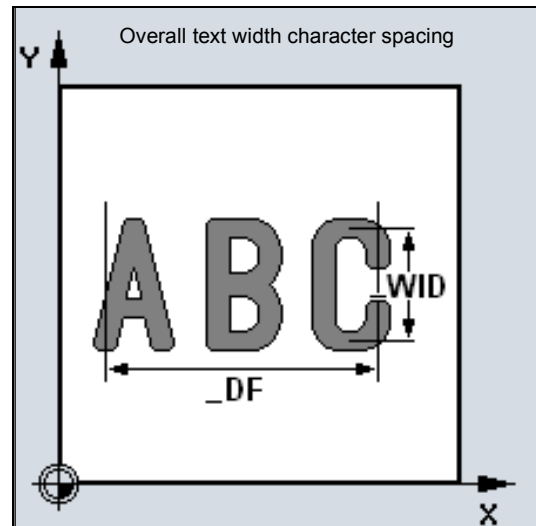
The cycle monitors whether it is possible to implement the programmed overall width, i.e., whether the sum of the character widths exceeds it.

In case of error, the alarm:

61176 "\_DF text length programmed to small" is output.



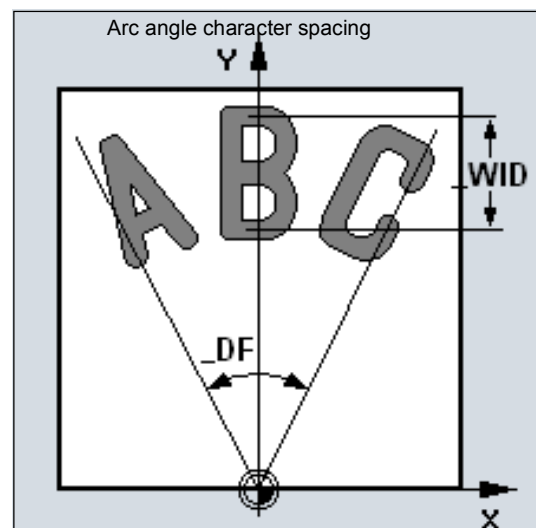
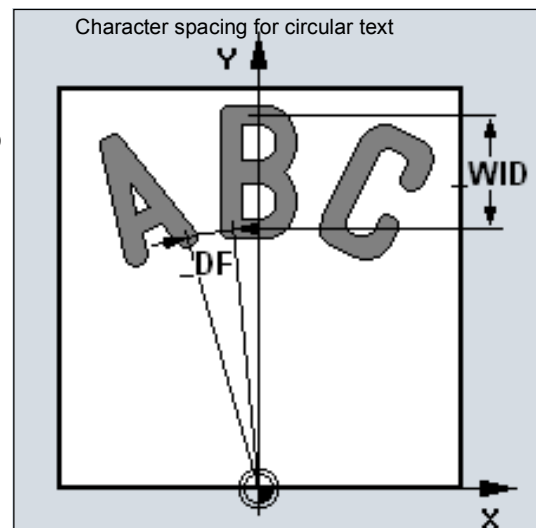
### 3.18 Engraving cycle CYCLE60 (with Cycles SW 6.4 and higher)



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`.



To distribute the characters evenly in a circle, simply program `_DF=360`. 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: character cycle: code page not available"

is output 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	°	
+	-	*	/	\	÷	x	<	>	=	~	
€	\$	£	§	&	%	@	©	®			
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	ø	Ð	þ	Ɔ	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 ' ' ' .

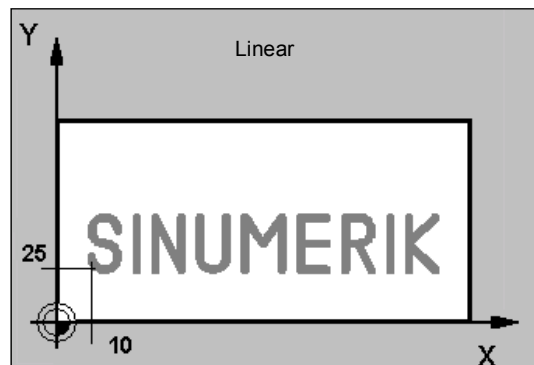
### 3.18 Engraving cycle CYCLE60 (with Cycles SW 6.4 and higher)



#### Programming example

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.



```

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

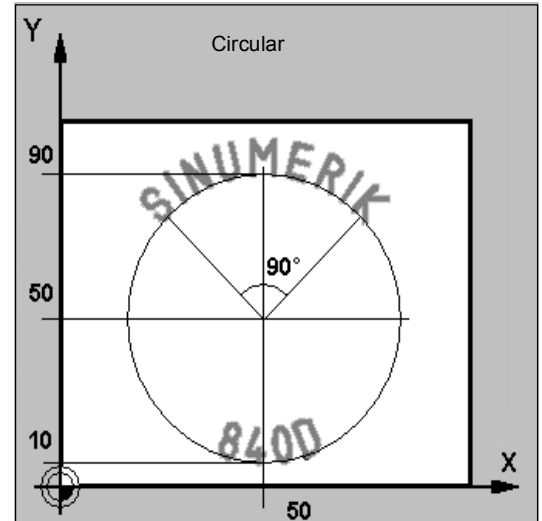
```



### Programming example 2

This program engraves two texts on a circle: "SINUMERIK" upward and "840D" downward. 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.




---

```
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
```



## Notes



## Turning Cycles

4.1	General .....	4-282
4.2	Prerequisites .....	4-283
4.3	Grooving cycle – CYCLE93 .....	4-286
4.4	Undercut cycle – CYCLE94 .....	4-295
4.5	Stock removal cycle – CYCLE95 .....	4-299
4.6	Thread undercut – CYCLE96.....	4-312
4.7	Thread cutting – CYCLE97 .....	4-316
4.8	Thread chaining – CYCLE98 .....	4-323
4.9	Thread recutting .....	4-329
4.10	Extended stock removal cycle – CYCLE950 .....	4-331

## 4.1 General

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.

The sections are structured as follows:

- **Programming**
- **Parameters**
- **Function**
- **Sequence**
- **Explanation of parameters**
- **Other information**
- **Programming example**

"Programming" and "Parameters" explain the use of cycles sufficiently for the experienced user, whereas beginners can find all the information they need for programming cycles under "Function", "Sequence", "Explanation of parameters", "Other information", and the "Programming example".

## 4.2 Prerequisites

### Data block for turning cycles

Turning cycles require the GUD7.DEF module. This is supplied, along with the cycles, on a diskette.

### 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 before the cycle is called. In the case of turning, this is usually the G18 (ZX) plane. The two axes of the turning plane are referred to below as the longitudinal axis (first axis of this plane) and the facing axis (second axis of this plane).

If diameter programming is active, the second axis of the plane is always taken as the facing axis:

**References:** /PG/ Programming Guide.

### 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

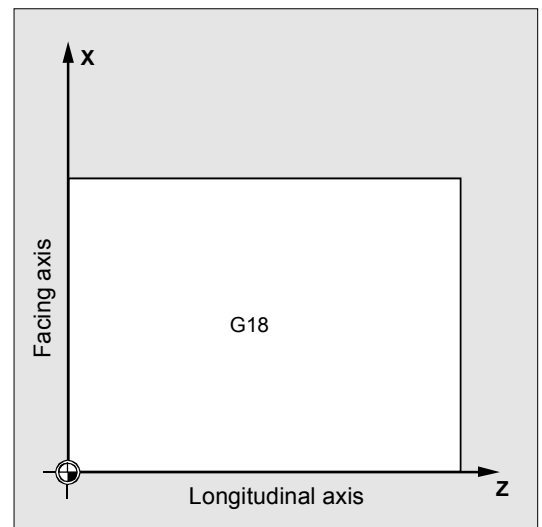
### Machining status messages

Status messages are displayed on the control monitor while the turning cycles are processed. The following messages can be displayed:

- "Thread start <No.> - longitudinal thread machining"
- "Thread start <No.> - face thread machining"

In each case, <No.> stands for the number of the figure that is currently being machined.

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



## 4.2 Prerequisites

### 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 (as before)

There is an option in `_ZSD[6]` to set mirroring behavior for the CYCLE93 grooving cycle.

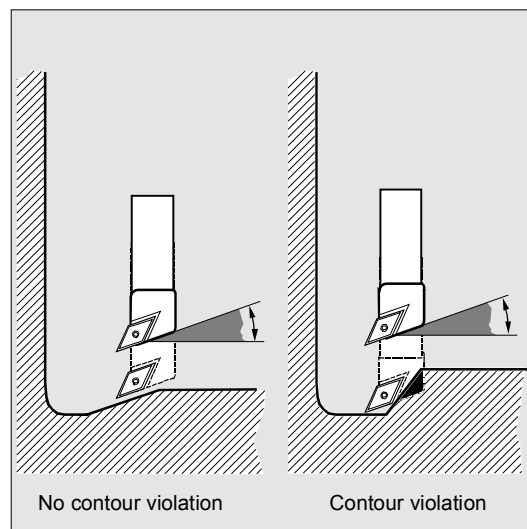
- `_ZSD[6]=0` The tool offsets are exchanged in the cycle when mirroring is active (to be used without an orientatable toolholder).
- `_ZSD[6]=1` The tool offsets are not exchanged in the cycle when mirroring is active (to be used with an orientatable toolholder).

### Contour monitoring with respect to tool clearance angle

Some turning cycles, in which travel movements with relief cutting are generated, monitor the tool clearance angle of the active tool for possible contour violation.

This angle is entered as a value in the tool offset (under parameter P24 in the D offset).

An angle between 0 and 90 degrees is entered without a sign.



When entering the tool clearance angle, remember that this depends on whether machining is longitudinal or facing. If a tool is to be used for longitudinal and face machining, two tool offsets must be applied if the tool clearance angles are different.

A check is made in the cycle to determine whether the programmed contour can be machined with the selected tool.

If machining is not possible with this tool, then

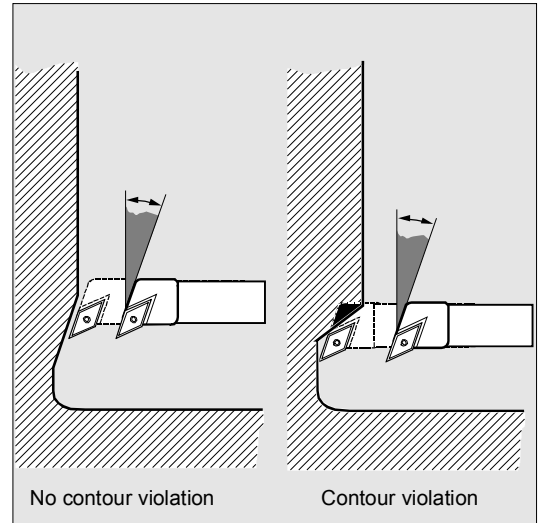
- The cycle is terminated with an error message (while cutting) or
- Contour machining continues and a message is output (in undercut cycles). The tool nose geometry then determines the contour.

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 as zero in the tool offset, this monitoring function is deactivated. The precise responses are described in the various 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



#### Programming

CYCLE93 (SPD, SPL, WIDG, DIAG, STA1, ANG1, ANG2, RCO1, RCO2, RCI1, RCI2, FAL1, FAL2, IDEP, DTB, VARI, \_VRT)



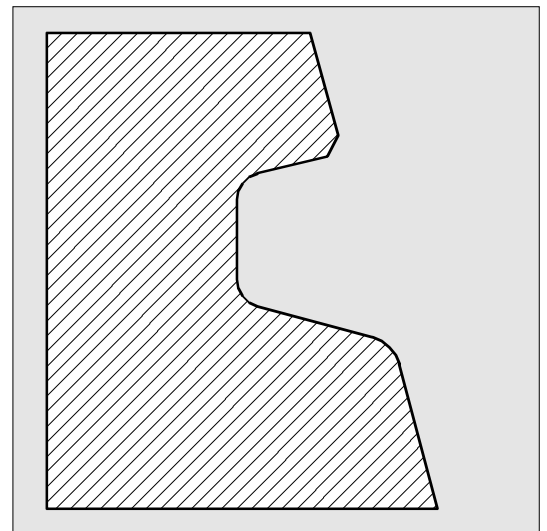
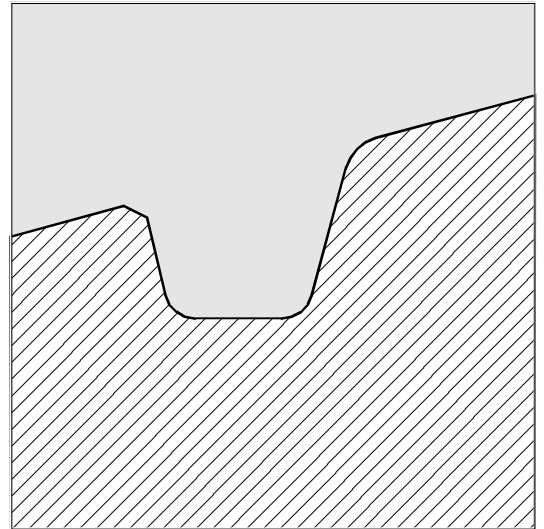
#### Parameters

SPD	real	Starting point in the facing axis (enter without sign)
SPL	real	Starting point in the longitudinal axis
WIDG	real	Width of groove (enter without sign)
DIAG	real	Depth of groove (enter without sign)
STA1	real	Angle between contour and longitudinal axis Value range: $0 \leq \text{STA1} \leq 180$ degrees
ANG1	real	Flank angle 1: on the side of the groove defined by the starting point (enter without sign) Value range: $0 \leq \text{ANG1} < 89.999$ degrees
ANG2	real	Flank angle 2: on the other side (enter without sign) Value range: $0 \leq \text{ANG2} < 89.999$
RCO1	real	Radius/chamfer 1, outside: on the side defined by the starting point
RCO2	real	Radius/chamfer 2, outside
RCI1	real	Radius/chamfer 1, inside: on the starting point side
RCI2	real	Radius/chamfer 2, inside
FAL1	real	Final machining allowance on the base of groove
FAL2	real	Final machining allowance on the flanks
IDEP	real	Infeed depth (enter without sign)
DTB	real	Dwell time at base of groove
VARI	int	Machining type Value range: 1...8 and 11...18
_VRT	real	Variable retraction distance from contour, incremental (enter without sign)



### 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.



### Sequence

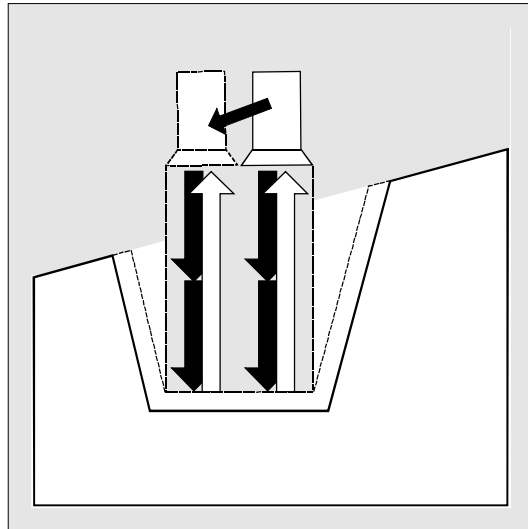
The depth infeed (towards the base of the groove) and infeed across the width (from groove to groove) are distributed evenly and with the greatest possible value.

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. The safety distance to the contour is calculated in the cycle.

### 4.3 Grooving cycle – CYCLE93

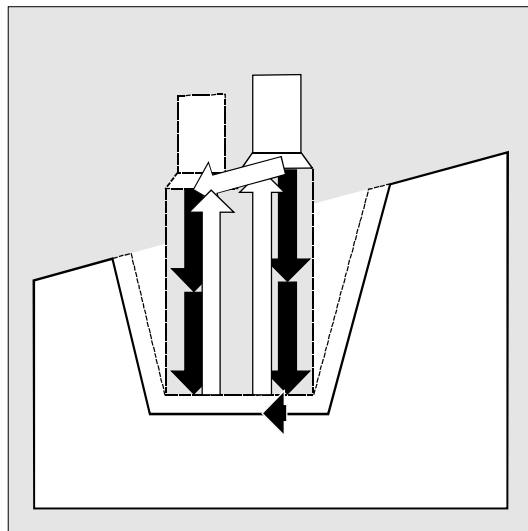
#### 1st step

Paraxial roughing to the base of the groove in single infeed steps. After each infeed, the tool is retracted for chip breaking.



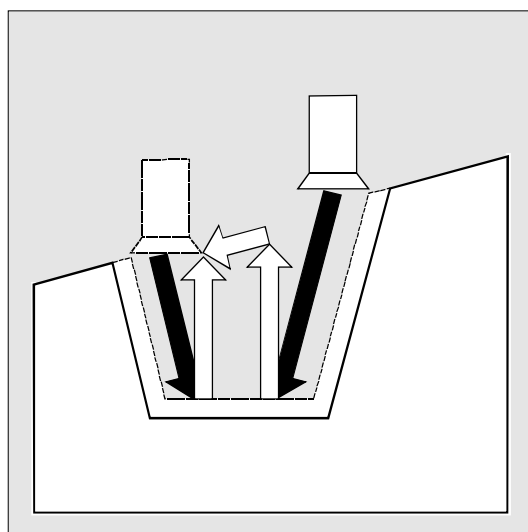
#### 2nd step

The groove is machined perpendicular to the infeed direction in one or more cuts. Each cut is again divided up according to the infeed depth. The tool is withdrawn by 1 mm from the second cut along the groove width, before it is fully retracted.



#### 3rd step

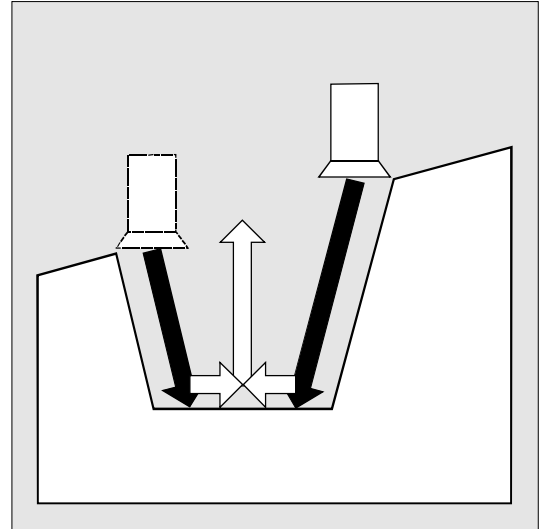
Cutting of the flanks in one step, if angles are programmed under ANG1 or ANG2. If the flank width is greater, the infeed along the groove width is performed in several steps.





**4th step**

Cutting of final machining allowance parallel to the contour from the edge to the center of the groove. The tool radius compensation is automatically selected and deselected by the cycle.





### Explanation of 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 can be created on curved surfaces in a variety of ways. Depending on the shape and radius of the curve, either a paraxial straight line can be placed on the maximum of the curve or a tangential oblique line can be placed on one of the edge points of the groove.

Radii and chamfers should only be programmed on the groove edge of a curved contour if the edge point in question is positioned on the straight line defined for the cycle.

#### WIDG and DIAG (groove width and groove depth)

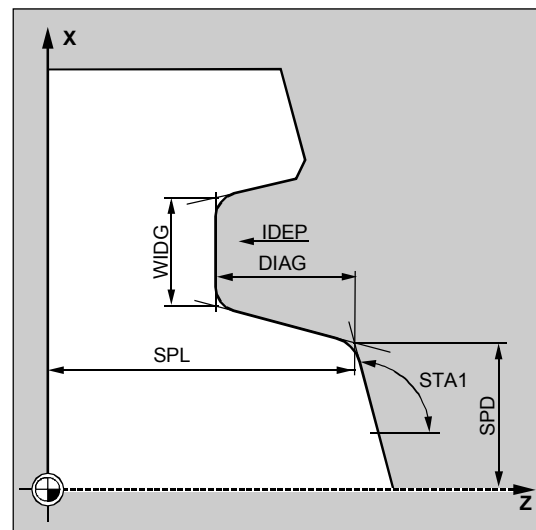
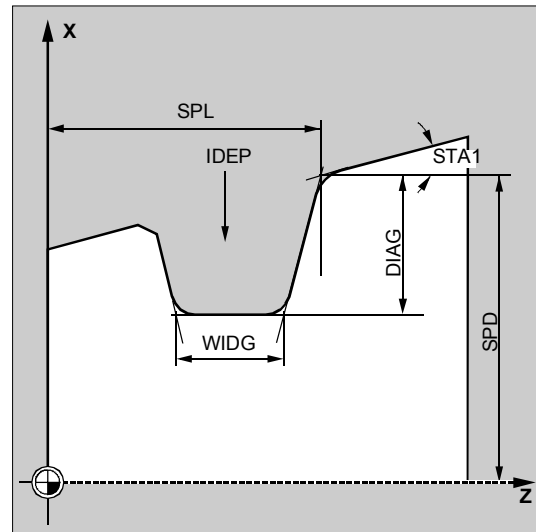
The shape of the groove is defined with the groove width (WIDG) and groove depth (DIAG) parameters. The cycle always starts its calculation from the point programmed with SPD and SPL.

If the groove is wider than the active tool, the width is machined in several steps. The total width is divided into equal sections by the cycle. The maximum infeed is 95% of the tool width after subtracting the tool nose radii. This ensures a cut overlap.

If the programmed groove width is less than the actual tool width, the following error message is output:  
61602 "Tool width incorrectly defined".

The cycle is not started and machining is aborted.

The alarm is also output if the value zero has been entered for the tool nose width.



**STA1 (angle)**

The angle of the oblique surface, on which the groove is to be machined, is programmed with the STA1 parameter. The angle can have any value between 0 and 180 degrees and always refers to the longitudinal axis.

**ANG1 and ANG2 (flank angle)**

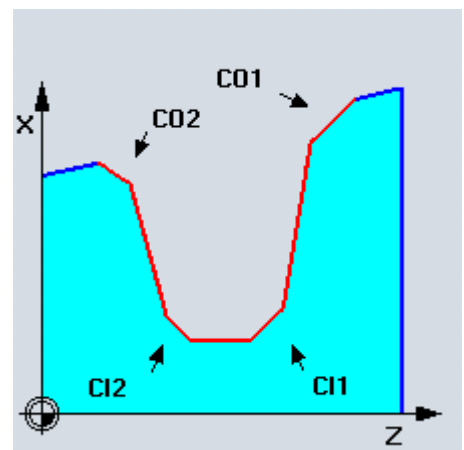
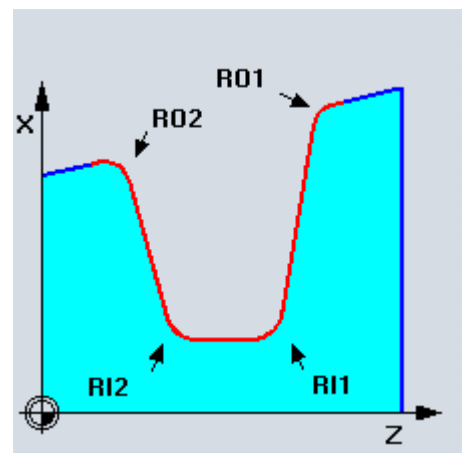
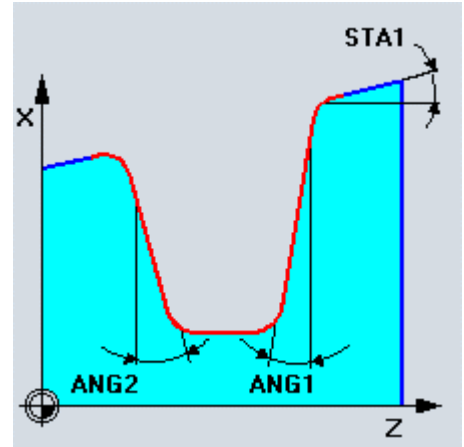
Asymmetrical grooves can be described by separate flank angles. The angles can be assigned any value between 0 and 89.999 degrees.

**RCO1, RCO2 and RC11, RC12 (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).



### 4.3 Grooving cycle – CYCLE93

#### FAL1 and FAL2 (final machining 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 is to be programmed in any case.

#### DTB (dwell time)

A dwell time at the base of the groove should be chosen that allows at least one spindle revolution. The dwell time is programmed in seconds.

#### VARI (machining type)

The units digit of the VARI parameter determines the type of processing for the groove. This parameter can be assigned any of the values shown in the figure.

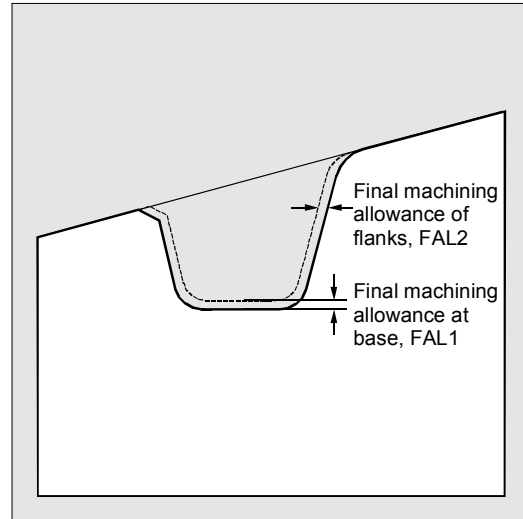
The tens value of the VARI parameter determines the type of calculation for the chamfer.

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:

1. Field: longitudinal/face
2. Field: external/internal
3. Field: starting point left/right (for longitudinal) or top/bottom (for face)



VARI		Selection in cycle support
1/11		longitudinal, external, left
5/15		longitudinal, external, right
3/13		longitudinal, internal, left
7/17		longitudinal, internal, right
6/16		facing, external, top
8/18		facing, external, bottom
2/12		facing, internal, top
4/14		facing, internal, bottom

If the parameter is assigned another value, the cycle is aborted and alarm

61002 "Machining type incorrectly programmed" is output.

The contour monitoring performed by the cycle ensures that a realistic groove contour results. This is not the case if the radii/chamfers touch each other at the base of the groove or overlap, or if an attempt at face grooving is made on a section of the contour that runs parallel to the longitudinal axis. In these cases, the cycle is aborted with alarm 61603 "Groove form incorrectly defined".

#### **\_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), liftoff 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.



#### **Other information**

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 determines, which of the two tool offsets it requires for which machining step, and activates them automatically. After the cycle is completed, the offset number programmed before the cycle call becomes active again. If no D number has been programmed for a tool offset when the cycle is called, the cycle is aborted with alarm 61000 "No tool offset 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 exchanged internally in the cycle (without orientatable toolholder).

`_ZSD[6]=1` TO is not exchanged internally in the cycle (with orientatable toolholder).



### Programming example

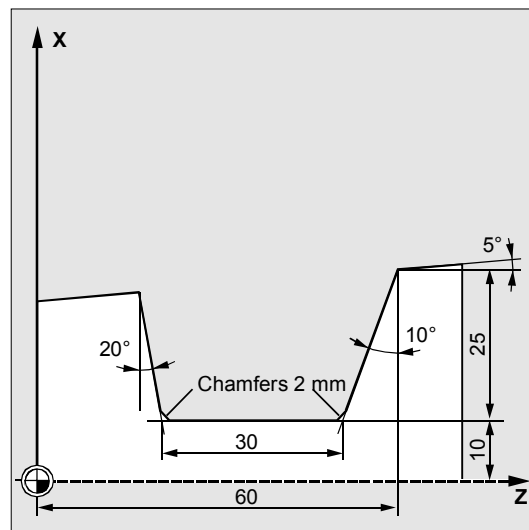
#### Grooving

This program machines a groove on an oblique surface (longitudinal, outside).

The starting point is at X35 Z60.

The cycle uses tool offsets D1 and D2 of tool T1.

The grooving tool must be defined accordingly.



```

DEF REAL SPD=35, SPL=60, WIDG=30, -> ;Definition of parameters with
-> DIAG=25, STA1=5, ANG1=10, ANG2=20, ;value 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 ;End of program

```

-> Must be programmed in a single block

## 4.4 Undercut cycle – CYCLE94



### Programming

CYCLE94 (SPD, SPL, FORM, \_VARI)



### Parameters

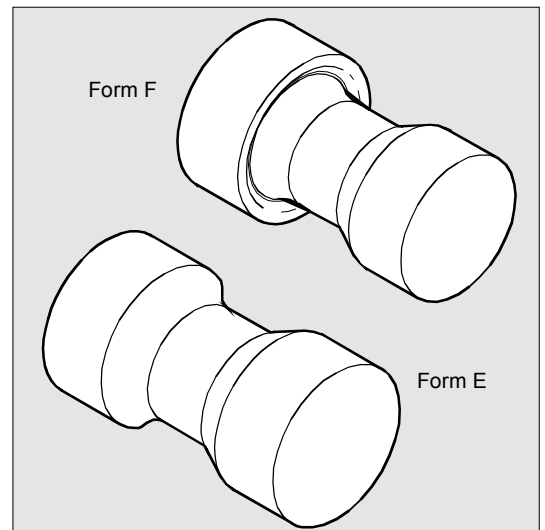
SPD	real	Starting point in the facing axis (enter without sign)
SPL	real	Starting point of the contour in the longitudinal axis (enter without sign)
FORM	char	Definition of the form Values: E (for form E) F (for form F)
_VARI	int	Specification of undercut position Values: 0 According to tool point direction 1...4 Define position



### 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 4.6).



### Sequence

#### Position reached before the beginning of the cycle:

The starting position can be any position, from which any undercut can be approached without collision.

#### The cycle implements the following motion sequence:

- Approach the starting point calculated in the cycle with G0.
- Selection of tool nose radius compensation according to active tool point direction and traversal of undercut contour at feedrate programmed prior to cycle call.
- Retraction to the starting point with G0 and deselection of the tool nose radius compensation with G40.

## 4.4 Undercut cycle – CYCLE94



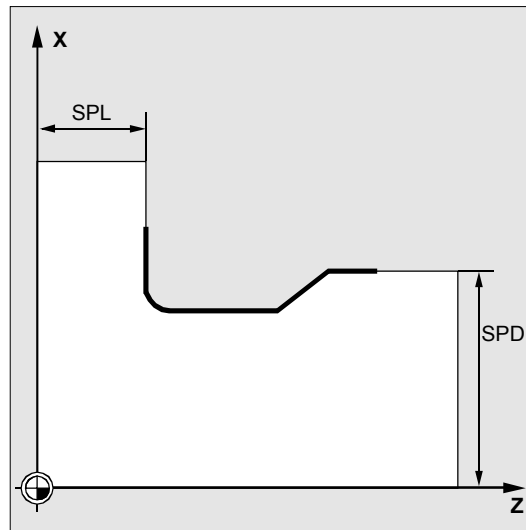
### Explanation of parameters

#### 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 the value programmed for SPD results in a final diameter that is <3 mm, the cycle is aborted and alarm

61601 "Finished part diameter too small" is output.

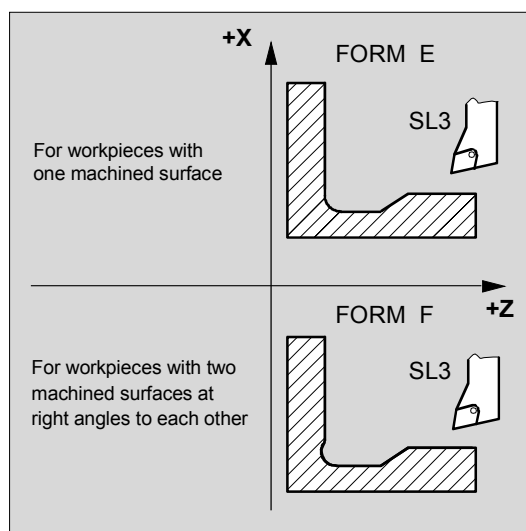


#### FORM (definition)

Form E and Form F are defined in DIN509 and determined by this parameter.

If a value other than E or F is assigned to the parameter, the cycle is aborted after outputting alarm

61609 "Form incorrectly defined".



#### \_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 cycle automatically determines the tool point direction from the active tool offset. The cycle can then only work with tool point directions 1 to 4.

If the cycle detects a tool point direction of 5 to 9, alarm

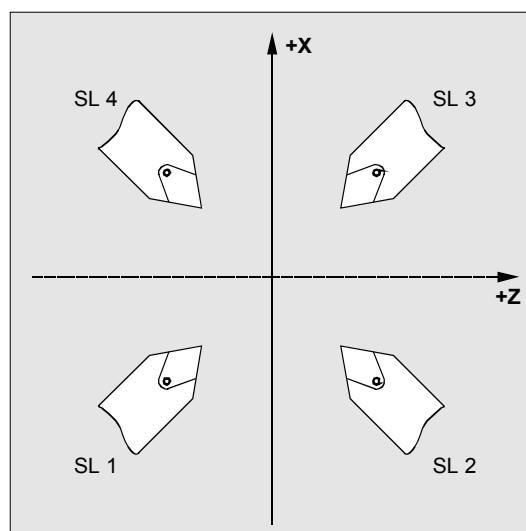
61608 "Incorrect tool point direction programmed" is output and the cycle is aborted.

\_VARI=1...4: Definition of undercut position

The cycle automatically determines the starting point.

This lies 2 mm from the final diameter and

10 mm from the final dimension in the longitudinal axis.





The position of this starting point in relation to the programmed coordinate values is determined by the tool point direction of the active tool.

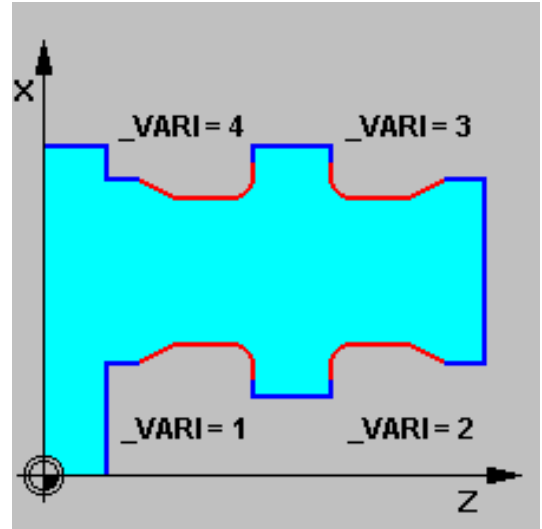
The cycle monitors the clearance angle of the active tool if a value has been assigned to the relevant parameter of the tool offset. If the cycle ascertains that the undercut form cannot be machined with the selected tool because the clearance angle is too small, the message

"Altered undercut form" is displayed on the control.

However, machining is continued.

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 orientatable toolholder 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.



### Other information

You must activate a tool offset before you call the cycles. Otherwise, the following alarm is output: 61000 "No tool offset active" and the cycle is aborted.

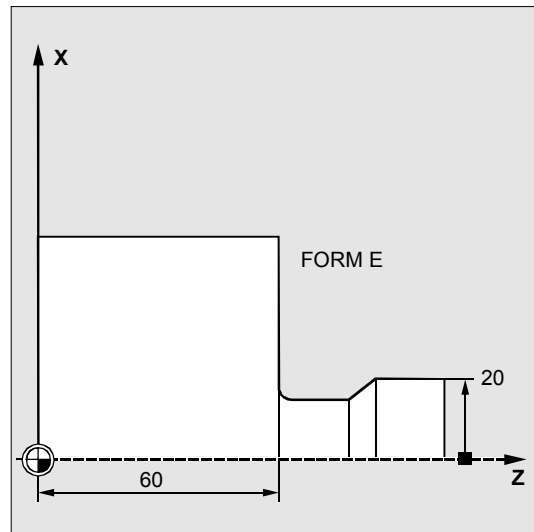
## 4.4 Undercut cycle – CYCLE94



## Programming example

## 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

## 4.5 Stock removal cycle – CYCLE95



### Programming

CYCLE95 (NPP, MID, FALZ, FALX, FAL, FF1, FF2, FF3, VARI, DT, DAM, \_VRT)



### Parameters

NPP	string	Name of the contour subroutine
MID	real	Infeed depth (enter without sign)
FALZ	real	Final machining allowance in the longitudinal axis (enter without sign)
FALX	real	Final machining allowance in the facing axis (enter without sign)
FAL	real	Final machining allowance along contour (enter without sign)
FF1	real	Feedrate for roughing without relief cut
FF2	real	Feedrate for insertion into relief cut elements
FF3	real	Feedrate for finishing
VARI	int	Machining type Value range: 1 ... 12 HUNDREDS DIGIT: Values: 0...With rounding at the contour No residual corners remain, the contour is rounded overlapping. That means the contours are rounded over several points of intersection. 2...Without rounding at the contour Contours are always rounded to the previous roughing intersection and then lifted off. Depending on the ratio of tool radius to infeed depth (MID), residual corners may be left.
DT	real	Dwell time for chip breaking during roughing
DAM	real	Path length, after which each roughing cut is interrupted for chip breaking
_VRT	real	Retraction distance from contour for roughing, incremental (enter without sign)

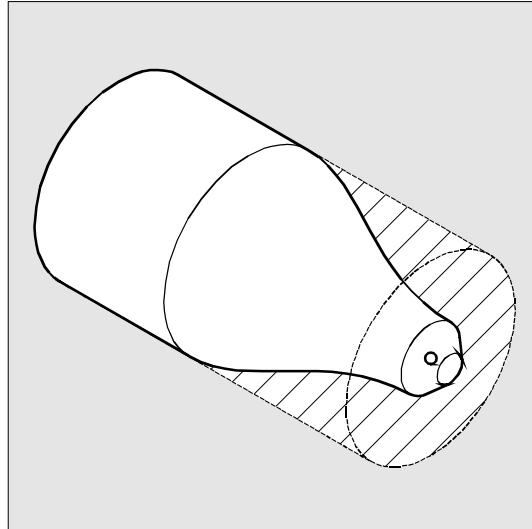
## 4.5 Stock removal cycle – CYCLE95



### Function

With this stock removal cycle, you can machine a contour programmed in a subroutine from a blank with paraxial stock removal. Relief cut elements can be included in the contour. With this cycle, contours can be machined in the longitudinal and facing directions, inside and outside. The technology can be freely selected (roughing, finishing, complete machining). While roughing the contour, paraxial cuts are generated from the maximum programmed infeed depth and when a point of intersection with the contour is reached, the residual corners are immediately removed cutting parallel to the contour. Roughing is performed up to the final machining allowance programmed.

Finishing is performed in the same direction as roughing. The tool radius compensation is automatically selected and deselected by the cycle.



### Sequence

#### Position reached before the beginning of the cycle:

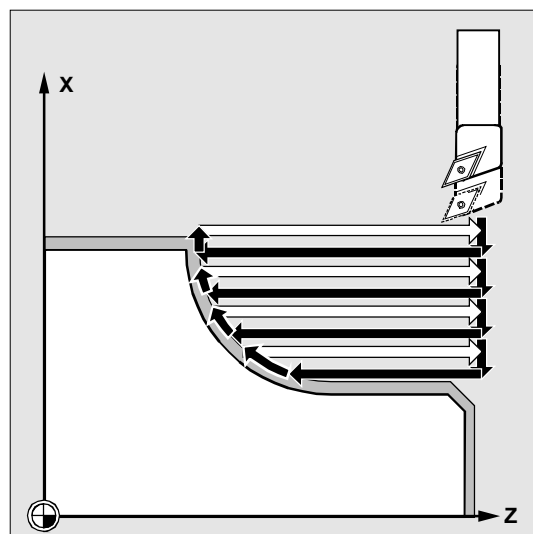
The starting position can be any position, from which the starting point of the contour can be approached without collision.

#### The cycle implements the following motion sequence:

- The cycle starting point is calculated in the cycle and then approached in both axes simultaneously with G0.

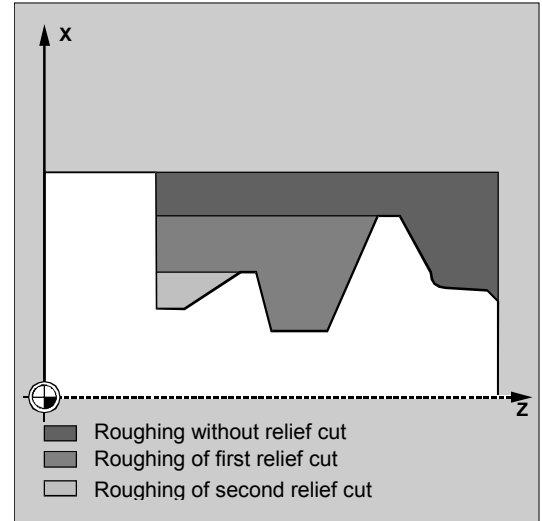
#### Roughing without relief cut elements:

- Paraxial infeed to the current depth is calculated internally and then approached with G0.
- Approach roughing intersection paraxially with G1 and 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 contour by the amount programmed in `_VRT` in every axis and retract with G0.
- This procedure is repeated until the total depth of the machining section has been reached.
- When roughing without relief cut elements, retraction to the cycle starting point is effected axis by axis.



**Roughing the relief cut elements:**

- Approach the starting point for the next relief cut axis by axis with G0. An additional safety distance is calculated internally.
- Infeed parallel to the contour + final machining allowance with G1/G2/G3 and FF2.
- Approach roughing intersection paraxially with G1 and feedrate FF1.
- Round to the last roughing intersection. Lift and retract as in the first machining section.
- If further relief cut elements are to be machined, repeat the above procedure for each relief cut element.

**Finishing:**

- The calculated cycle starting point is approached in both axes simultaneously with G0 and tool nose radius compensation is selected.
- Both axes then continue to move; with G0 up to final machining allowance + tool nose radius + 1 mm safety distance 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 parameters

#### NPP (name)

Enter the name of the contour subroutine under this parameter. This contour subroutine must not be a subroutine with a parameter list.

Please use the name conventions described in the Programming Guide when naming the contour subroutine.



#### 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. In this case, the program name and labels/block number are identified by an ":".

Examples:

NPP="CONTOUR\_1"

The machining contour is the "Contour\_1" complete program.

NPP="START:END"

The machining contour is defined as the section starting from the block labeled START to the block labeled END in the calling routine.

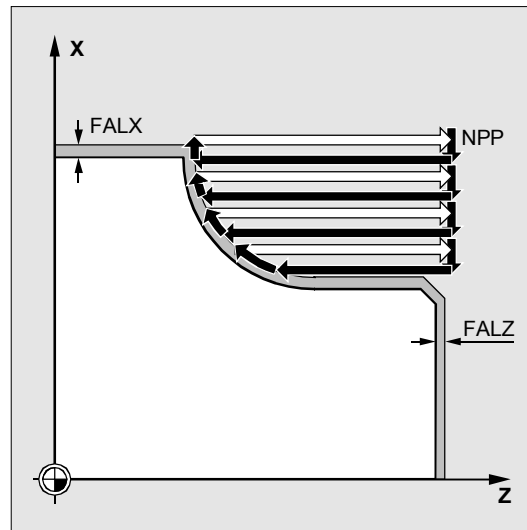
NPP="/\_N\_SPF\_DIR/\_N\_CONTOUR\_1\_SPF:N130:N210"

The machining 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



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 4.2).

The cycle automatically calculates the current infeed depth for roughing.

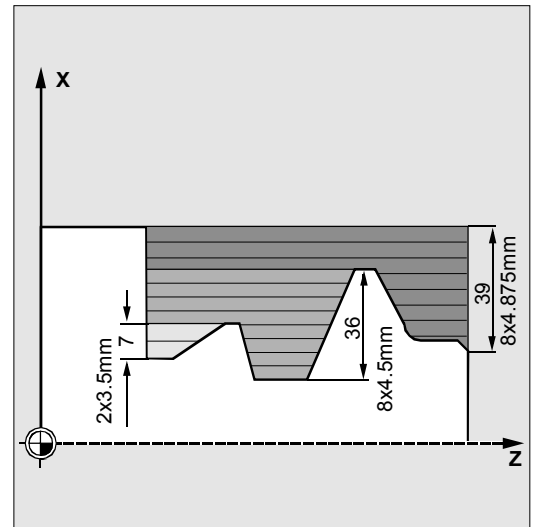
Where contours with relief cut elements are to be machined, the cycle divides up the roughing operation into single roughing steps. The cycle recalculates the current infeed depth for every roughing step. This current infeed depth always lies between the programmed infeed depth and half its value. The number of required roughing cuts is derived from the total depth of a roughing section and the programmed maximum infeed depth. The total depth to be machined is then divided equally amongst these roughing cuts. This method ensures optimum cutting conditions. The machining steps shown in the figure above are used for roughing this contour.

Example of calculating the current infeed depths:

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 performed with an infeed of 4.875 mm.

In machining section 2, eight roughing cuts, each with an infeed of 4.5 mm are also executed (total difference 36 mm).

Machining section 3 is roughed twice with a current infeed of 3.5 (total difference 7 mm).



## 4.5 Stock removal cycle – CYCLE95

### FAL, FALZ and FALX (final machining allowance)

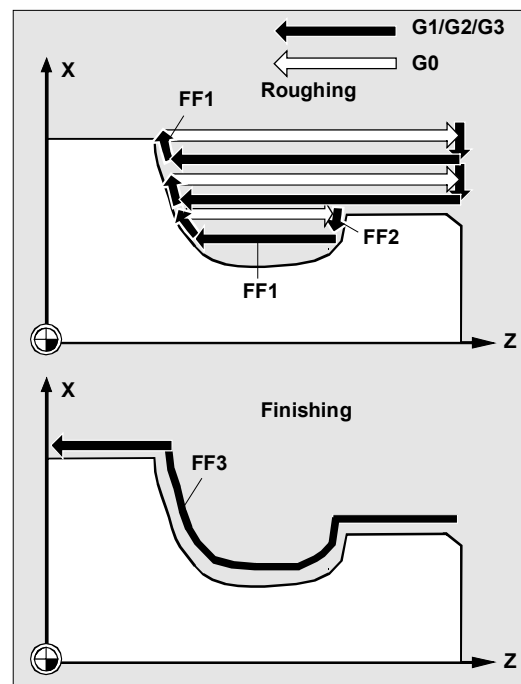
The final machining allowance for the roughing operation is either defined in the FALZ and FALX parameters, if you wish to enter different final machining allowances for each axis, or in the FAL parameter, if you wish to enter a final machining allowance that follows a contour. In this case, this value is used for the final machining allowance in both axes. The programmed values are not subjected to a plausibility check. If all three parameters are assigned values, all the final machining allowances are calculated by the cycle. It is, however, advisable to decide on one or the other form of final machining allowance definition.

**Roughing** is always performed to these final machining 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 final machining allowances have been programmed, roughing is performed 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 types as follows:

Machining   
 (Roughing/Finishing/Complete machining)  
 Selection   
 (Longitudinal/face)  
 Selection   
 (External/internal)

The machining types are shown in the table below.

Value	Machining	Selection	Selection
1	Roughing	long.	outside
2	Roughing	transv.	outside
3	Roughing	long.	inside
4	Roughing	transv.	inside
5	Finishing	long.	outside
6	Finishing	transv.	outside
7	Finishing	long.	inside
8	Finishing	transv.	inside
9	Complete machining	long.	outside
10	Complete machining	transv.	outside
11	Complete machining	long.	inside
12	Complete machining	transv.	inside

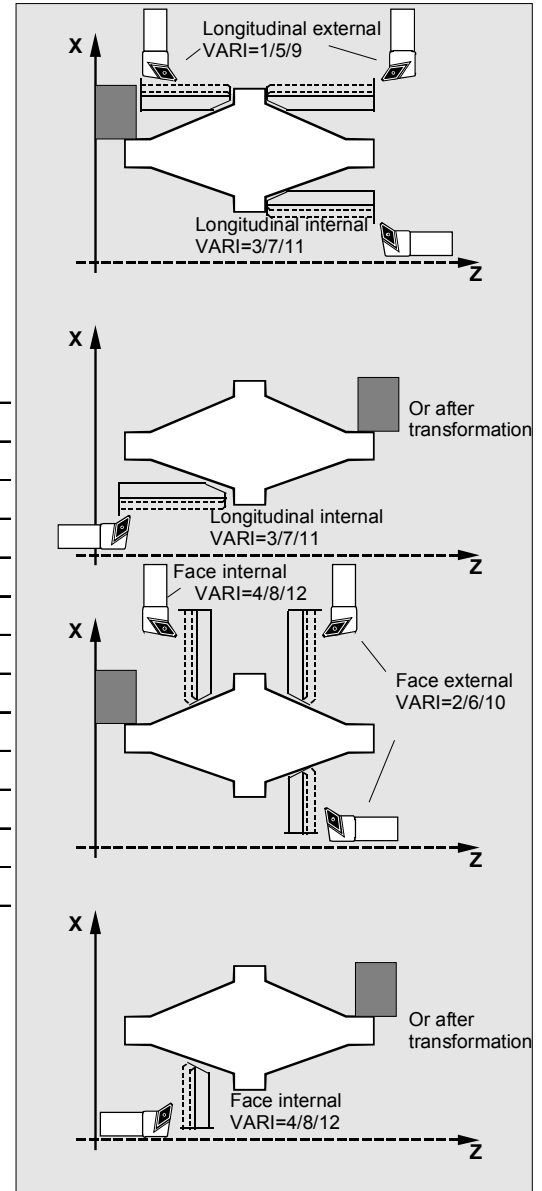


*Infeed is always performed in the facing axis with longitudinal machining and in the longitudinal axis with face machining.*

*External machining means that infeed is performed in the direction of the negative axis. In internal machining, infeed is performed 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 implemented in the VARI parameter.*

A plausibility check is performed on the VARI parameter. If you select an invalid value, the cycle is aborted and alarm 61002 "Incorrect machining type defined" is output.



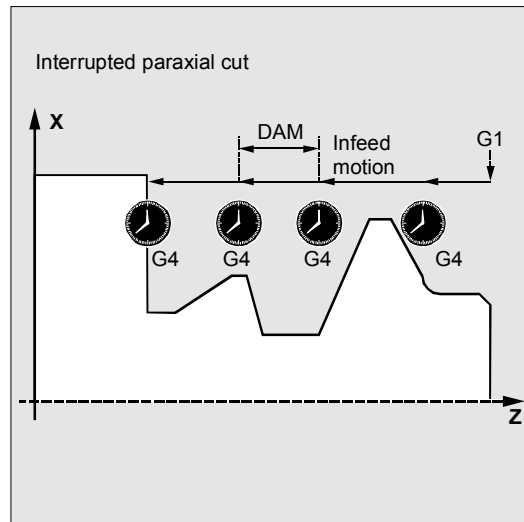
## 4.5 Stock removal cycle – CYCLE95

### 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 only apply to roughing. You define the maximum displacement, after which chip breaking is to be performed, in the DAM parameter. In DT, you can also program a dwell time to be executed at each of the interruption points. If no displacement has been specified for cut interruption (DAM = 0), then uninterrupted roughing cuts without dwell times are generated.

### \_VRT (lift-off path)

The \_VRT parameter can be used to program the amount of liftoff during roughing in both axes. If \_VRT=0 (parameter not programmed), liftoff is 1 mm. The lift-off path is always measured in inches or metric depending on the programmed measuring system, i.e., \_VRT=1 for inch → programming produces a liftoff of 1 inch.



### Other information

#### Contour definition

The contour is programmed in a subroutine, whose name is defined as a parameter.

The contour subroutine must contain at least 3 blocks with movements in both 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 subroutine.

If the contour subroutine is shorter, alarms 10933 "The contour subroutine contains too few contour blocks" and 61606 "Error in preprocessing contour" are output and the cycle is aborted.

Relief cut elements can be programmed consecutively. Blocks without movement in the plane are not subject to any limitations.

All the traversing blocks for the first two axes in the current plane are processed in the cycle, as only these axes are involved in cutting. Movements for other axes can be included in the contour subroutine, but their travel paths are suppressed during the cycle run.

The only geometry permitted in the contour are straight line and circular programming with G0, G1, G2, and G3. Commands for fillets and chamfers can also be programmed. If any other motion commands are programmed in the contour, the cycle is aborted with alarm 10930 "Illegal interpolation type in the machining contour".

The first block containing a traversing movement in the current machining plane must contain a travel command (G0, G1, G2, or G3), otherwise the cycle is aborted with the alarm 15800 "Incorrect starting conditions for CONTPRON".

This alarm is also activated when G41/G42 is active. The starting point of the contour is the first position on the machining plane programmed in the contour subroutine.

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 memory can hold, the cycle is aborted with the alarm 10934 "Overflow contour table".

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.

## 4.5 Stock removal cycle – CYCLE95

If the maximum diameter in one contour subroutine is not located within the programmed end point or starting point of the contour, the cycle automatically extends a paraxial straight line 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.

Programming

- Radius compensation plane with G17/G18/G19
- A frame
- Traversing of an axis of the plane, in which machining is performed, as a positioning axis
- Selection of tool radius compensation with G41/G42

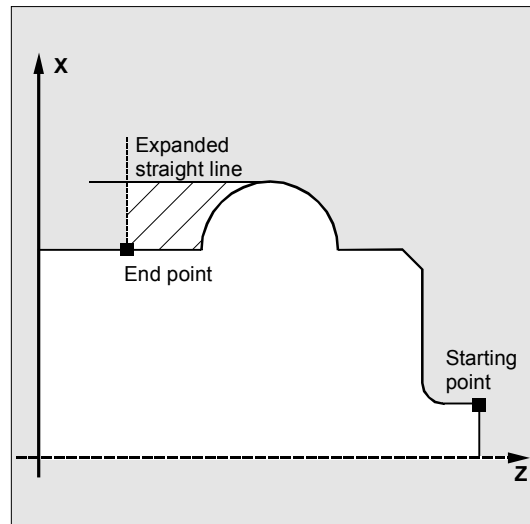
in the contour program causes the alarm 10931 "Incorrect machining contour" to be output and the cycle to be aborted.

### Contour direction

The direction, in which the stock removal contour can be programmed, can be freely selected. The machining direction is automatically defined in the cycle. With complete machining operations, the contour is finished in the same direction, in which roughing took place.

If only finishing is selected, the contour is always traversed in the programmed direction.

The first and last programmed contour points are the criteria for selecting the machining direction. 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 angles of the active tool
- Circular programming of arcs with an arc angle > 180 degrees

In the case of relief cut elements, the cycle checks whether machining is possible with the active tool. If the cycle detects that this machining operation will lead to a contour violation, it outputs alarm 61604 "Active tool violates the programmed contour" and machining is aborted.

Contour monitoring is not performed if the clearance angle has been defined as zero in the tool offset.

If the arcs in the offset are too large, the following alarm is output:  
10931 "Incorrect machining contour".

Overhanging contours cannot be machined by CYCLE95. Contours of this type are not monitored by the cycle and consequently there is no alarm.

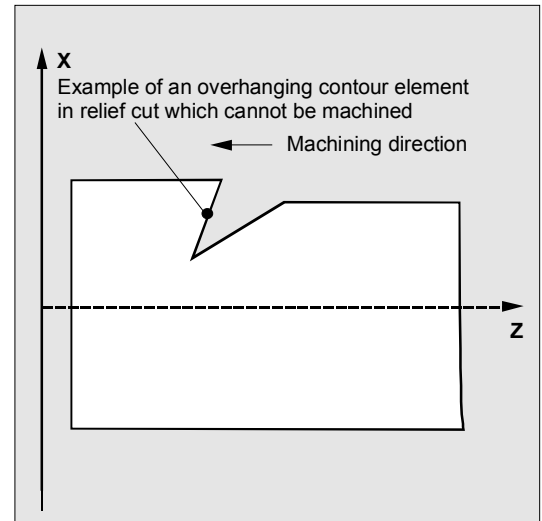
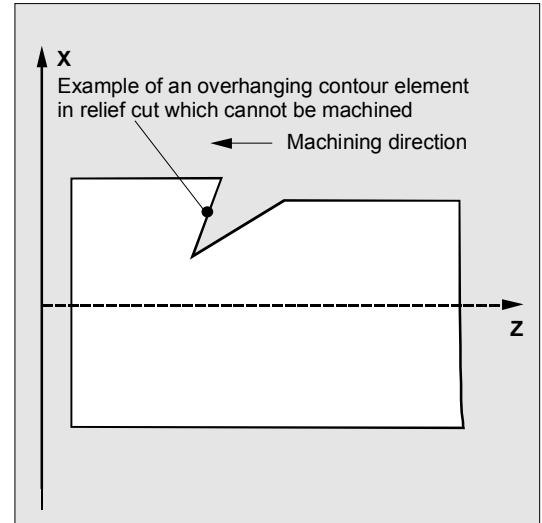
### Starting point

The cycle automatically determines the starting point of the machining operation. The starting point is positioned on the axis, in which infeed is performed, at the following distance from the contour: final machining allowance + lift-off path (parameter `_VRT`). In the other axis, it is positioned at a distance corresponding to final machining allowance + `_VRT` in front of the contour starting point. The tool nose radius compensation is selected internally in the cycle while the starting point is being approached.

Therefore, the last point before the cycle is called must be selected such that it can be approached without risk of collision and adequate space is available for the compensating movement.

### Approach strategy of the cycle

The starting point calculated by the cycle is always approached in the two axes simultaneously for roughing and one axis at a time for finishing. In finishing, the infeed axis is the first to travel.



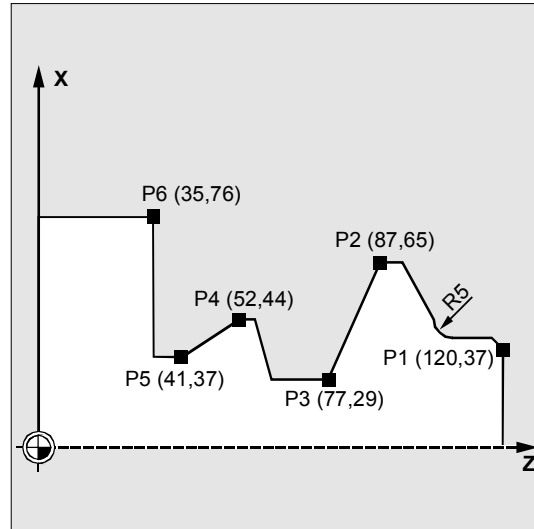
## 4.5 Stock removal cycle – CYCLE95



## Programming example 1

## Stock removal cycle

The contour illustrated in the figure explaining the assignment parameters must be machined completely (longitudinal, external). Axis-specific final machining allowances have been defined. No interruption between cuts has been programmed. 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, , -> -> 0 .2, 0.1, 0.2, 9, , , 0.5)	;Cycle call
N30 G0 G90 X81	;Reapproach starting position
N40 Z125	;Traverse axis by axis
N50 M30	;End of program
PROC CONTOUR_1	;Beginning of contour subroutine
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

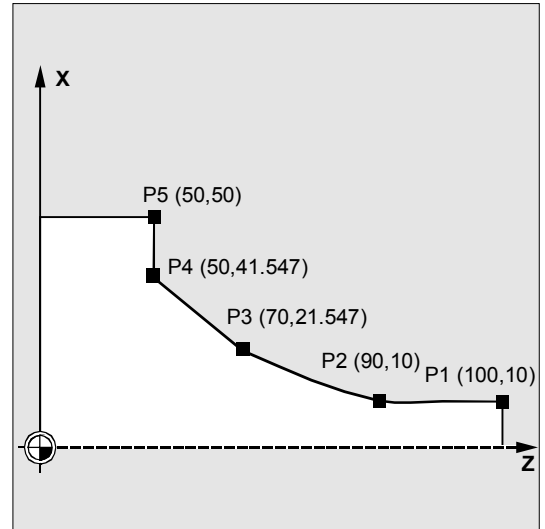
-> Must be programmed in a single block



### Programming example 2

#### Stock removal cycle

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 ("START:END",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

```

## 4.6 Thread undercut – CYCLE96

### 4.6 Thread undercut – CYCLE96



#### Programming

CYCLE96 (DIATH, SPL, FORM, \_VARI)



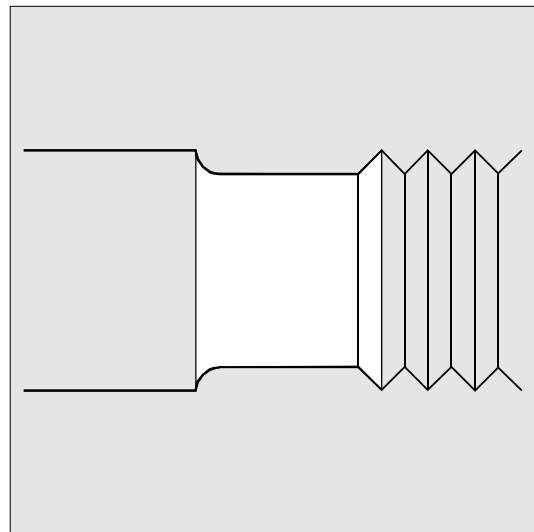
#### Parameters

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	int	Specification of undercut position Values: 0 According to tool point direction 1...4 Define position



#### Function

This cycle is for machining thread undercuts in accordance with DIN 76 on parts with a metric ISO thread.







### Sequence

#### Position reached before the beginning of the cycle:

The starting position can be any position, from which any thread undercut can be approached without collision.

#### The cycle implements the following motion sequence:

- Approach the starting point calculated in the cycle with G0.
- Selection of the tool radius compensation for the active tool point direction. Retraction along the undercut contour at the feedrate programmed before cycle call.
- Retraction to the starting point with G0 and deselection of tool radius compensation with G40.



### Explanation of parameters

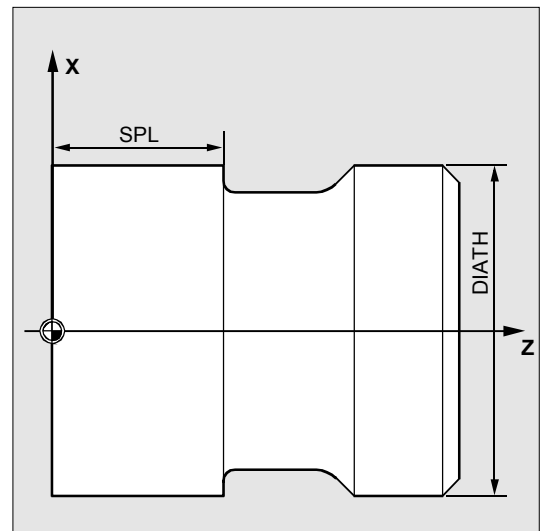
#### DIATH (nominal diameter)

With this cycle, you can machine thread undercuts for metric ISO threads from M3 to M68.

If the value programmed in DIATH results in a final diameter of <3 mm, the cycle is aborted and alarm 61601 "Finished part diameter too small" is output. If the parameter is assigned a value other than that defined by DIN76 Part 1, the cycle is again aborted and the following alarm is output: 61001 "Thread pitch incorrectly defined".

#### SPL (starting point)

You define the final dimension in the longitudinal axis with the SPL parameter.



**FORM (definition)**

Thread undercuts of forms A and B are defined for external threads, form A for normal thread run-outs, form B for short thread run-outs.

Thread undercuts of forms C and D are used for internal threads, form C for a normal thread run-out, form D for a short thread run-out.

If the parameter is assigned a value other than A to D, the cycle is aborted after outputting alarm 61609 "Form incorrectly defined".

The tool radius compensation is automatically selected by the cycle.

**\_VARI (undercut position)**

The position of the undercut can be either specified directly or derived from the tool point direction, using the `_VARI` parameter.

See `_VARI` with CYCLE94.

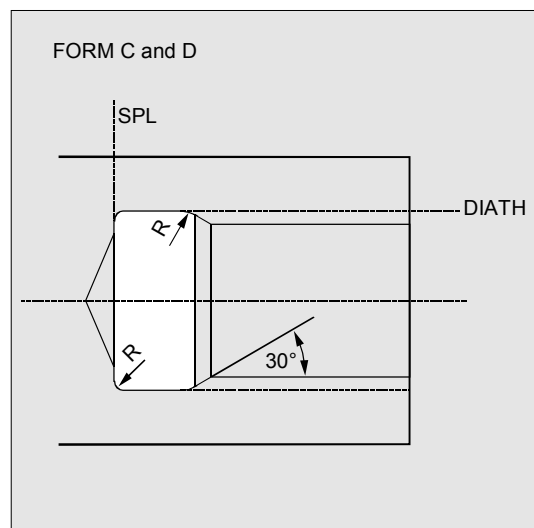
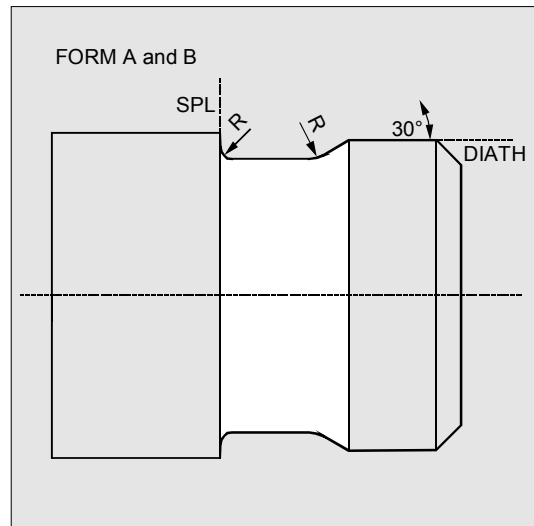
The cycle automatically determines the starting point that is defined by the tool point direction of the active tool and the thread diameter. The position of this starting point in relation to the programmed coordinate values is determined by the tool point position of the active tool.

The cycle monitors the clearance angle of the active tool if forms A and B are being machined. If the cycle detects that the undercut form cannot be machined with the selected tool, the message "Changed undercut form" is output by the control but machining is continued.

**Other information**

You must activate a tool offset before the cycle is called. Otherwise, the following error message is output

61000 "No tool offset active" and the cycle is aborted.

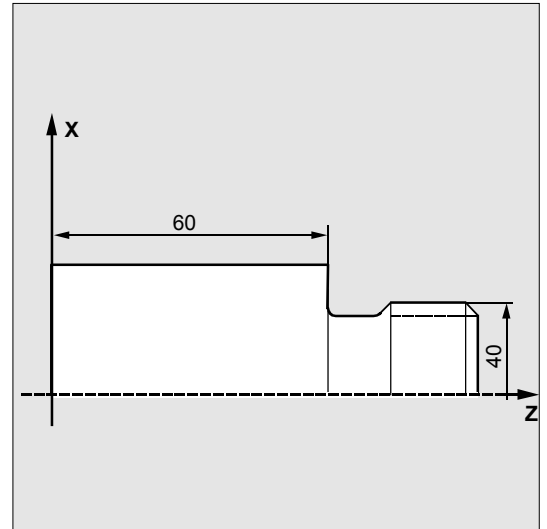




### Programming example

#### Thread undercut\_Form\_A

You can machine a thread undercut of form A with this program.



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

## 4.7 Thread cutting – CYCLE97



### Programming

CYCLE97 (PIT, MPIT, SPL, FPL, DM1, DM2, APP, ROP, TDEP, FAL, IANG, NSP, NRC, NID, VARI, NUMT, \_VRT)



### Parameters

PIT	real	Thread pitch as value (enter without sign)
MPIT	real	Thread pitch as thread size Value range: 3 (for M3) ... 60 (for M60)
SPL	real	Starting point of the thread in the longitudinal axis
FPL	real	End point of the thread in the longitudinal axis
DM1	real	Diameter of the thread at the starting point
DM2	real	Diameter of the thread at the end point
APP	real	Arc-in section (enter without sign)
ROP	real	Arc-out section (enter without sign)
TDEP	real	Thread depth (enter without sign)
FAL	real	Final machining allowance (enter without sign)
IANG	real	Infeed angle Value range: "+" (for flank infeed on flank) "- " (for alternating flank infeed)
NSP	real	Starting point offset for the first thread start (enter without sign)
NRC	int	Number of rough cuts (enter without sign)
NID	int	Number of noncuts (enter without sign)
VARI	int	Definition of the machining type for the thread Value range: 1 ... 4
NUMT	int	Number of thread starts (enter without sign)
_VRT	real	Variable retraction distance based on initial diameter, incremental (enter without sign)

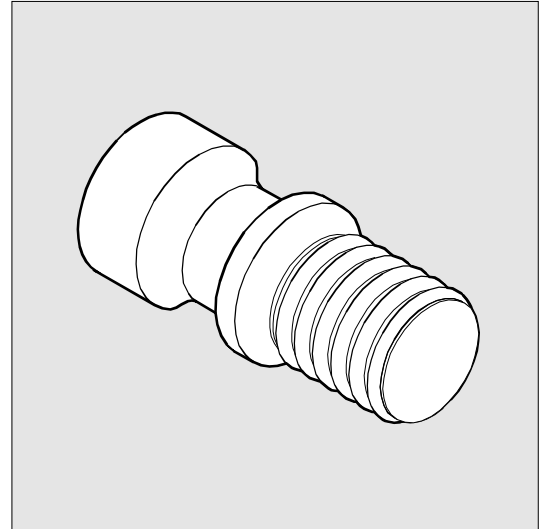


## Function

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. In multiple thread cutting, the thread starts are cut one after the other. Infeed is automatic. You can select either constant infeed per cut or constant cross-section of cut.

A right-hand or left-hand thread is determined by the direction of rotation of the spindle, which is programmed before the cycle call.

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 Guide Fundamentals  
Chapter "Thread cutting with constant pitch,  
G33"



*A speed-controlled spindle with a position measuring system is required to operate this cycle.*



## Sequence

### Position reached before the beginning of the cycle:

The starting position is any position, from which the programmed thread starting point + arc-in section can be approached without collision.

### The cycle implements the following motion sequence:

- Approach the starting point determined by the cycle at the beginning of the arc-in section for the first thread start with G0.
- Infeed for roughing according to the infeed type defined under VARI.
- Thread cutting is repeated according to the number of roughing cuts programmed.
- The final machining allowance is removed in the next cut with G33.
- This cut is repeated according to the number of programmed noncuts.
- The total motion sequence is repeated for each additional thread start.



### Explanation of parameters

#### PIT and MPIT (value and thread size)

The thread pitch is a paraxial value and is entered without a sign. If metric cylindrical threads are being machined, it is also possible to define the thread pitch in the MPIT parameter as a thread size (M3 to M60). These two parameters should be used as alternatives. If they contain conflicting values, the cycle generates alarm 61001 "Thread pitch incorrect" 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. With an inside thread, this corresponds to the tap hole diameter.

#### Connection between SPL, FPL, APP and ROP (starting point, end point, arc-in section and arc-out section)

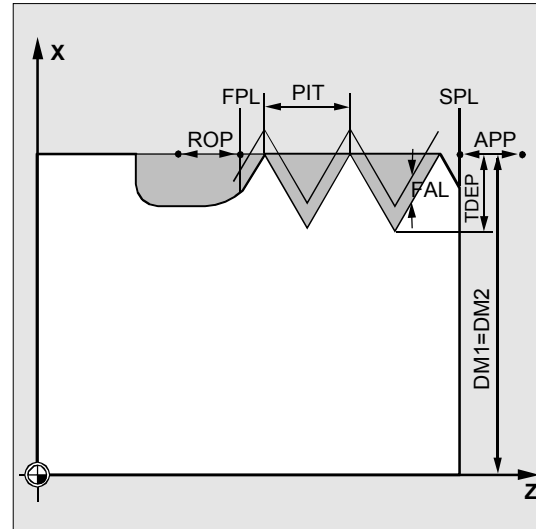
The programmed starting point (SPL) or end point (FPL) represents the original starting point of the thread. However, the starting point used in the cycle is the starting point brought forward by the arc-in section APP and, in the same way, the end point is the programmed end point brought back by the arc-out section ROP. The starting point defined by the cycle always lies 1 mm outside the programmed thread diameter in the facing axis. This retraction plane is automatically generated by the control.

#### Connection between TDEP, FAL, NRC and NID (thread depth, final machining allowance, number of cuts)

The programmed final machining allowance works parallel to the axis and is subtracted from the preset thread depth TDEP; the remainder is divided into roughing cuts.

The cycle automatically calculates the individual current infeed depths, 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. Infeed is then performed with differing values for the infeed depth.

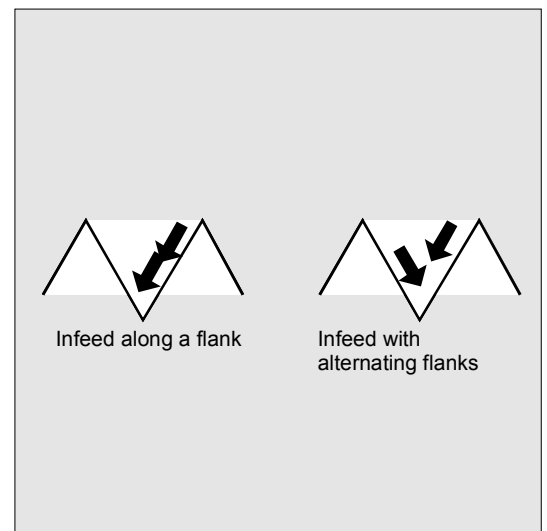
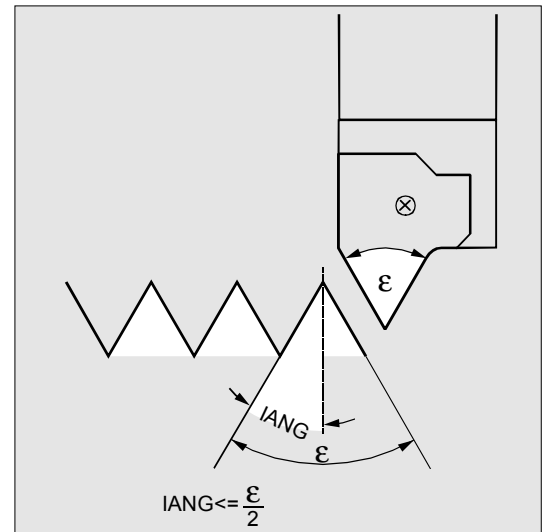


In a second method, the total thread depth is divided into constant infeed depths. The cross-section of cut increases from cut to cut. However, if the values for the thread depth are small, this method can create better cutting conditions.

The final machining allowance, FAL, is removed in one cut after roughing. After this, the noncuts programmed under the NID parameter are executed.

### IANG (infeed angle)

You define the infeed angle with the IANG parameter. If infeed is to be performed at right angles to the cutting direction in the thread, this parameter must be assigned the value zero, i.e., this parameter can also be omitted from the parameter list, as it is then automatically assigned the default value zero. If infeed is to be performed along the flanks, the absolute value of this parameter must be no more than half the flank angle of the tool. The sign entered for this parameter defines how this infeed is performed. If a positive value is entered, infeed is always performed on the same flank, if a negative value is entered, infeed is performed alternately on both flanks. The alternate flanks infeed type can only be used for cylindrical threads. However, if a negative value is assigned to the IANG parameter for a tapered thread, the cycle automatically performs a flank infeed along one flank.



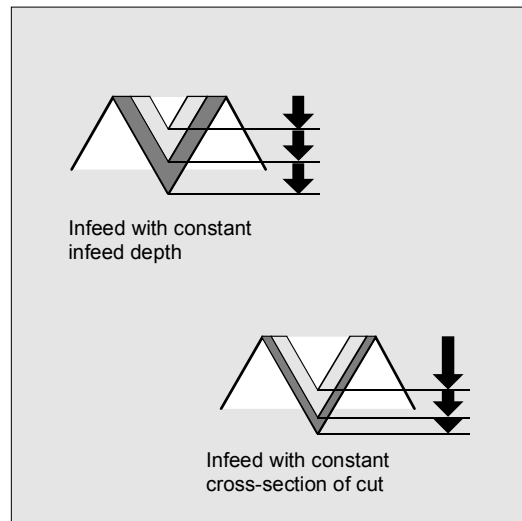
## 4.7 Thread cutting – CYCLE97

### 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 value is a starting point offset. The parameter can be assigned any value between 0.0001 and +359.9999 degrees. If no starting point offset has been entered or the parameter has been omitted from the parameter list, the first thread automatically starts at the zero degrees 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 be a value between 1 and 4 with the following meaning:



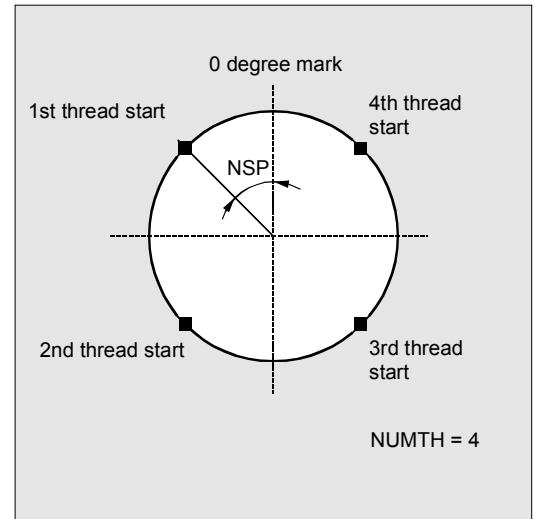
Value	External/internal	Const. infeed/const. cross-section of cut
1	External	Constant infeed
2	Internal	Constant infeed
3	External	Constant cross-section of cut
4	Internal	Constant cross-section of cut

If another value is assigned to the VARI parameter, the cycle is aborted and alarm 61002 "Machining type incorrectly programmed" is output.



**NUMT (number of thread starts)**

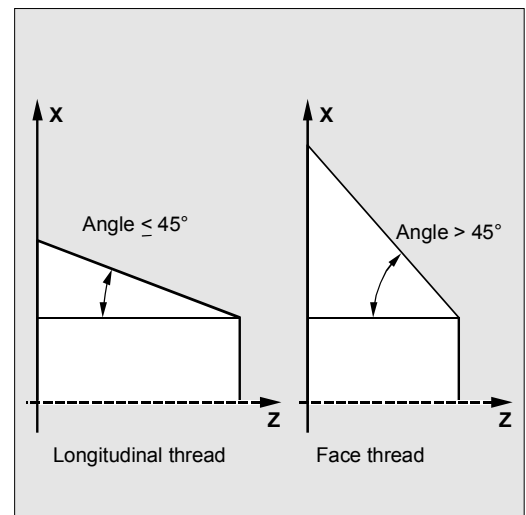
You specify the number of thread starts for a multiple thread with the NUMT parameter. If you require a single thread, either assign the value zero to the parameter or omit it from the parameter list. The thread starts are distributed uniformly around the circumference of the turned part, the first thread start is defined in the NSP parameter. If a multiple-start thread with a non-uniform distribution of thread starts around the circumference is to be machined, the cycle must be called for every thread start and the corresponding starting point offset must be programmed.

**\_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.

**Other information****Difference between a longitudinal thread and a face thread**

The cycle automatically calculates whether a longitudinal or face thread is to be machined. This depends on the angle of the taper, on which the thread is to be machined. If the angle at the taper  $\leq 45$  degrees, the longitudinal axis thread is machined, otherwise it will be the face thread.



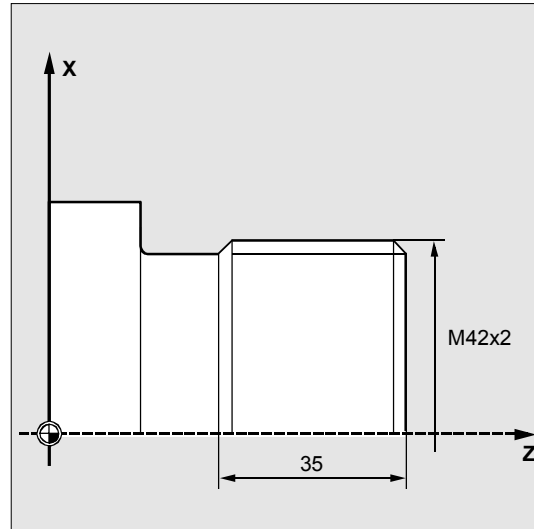
## 4.7 Thread cutting – CYCLE97



## Programming example

## Thread cutting

With this program, you can cut an M42x2 metric outside thread with flank infeed. The infeed is performed with a constant cross-section of cut. Five roughing cuts are made to a thread depth of 1.23 mm without final machining allowance. After machining has been completed, 2 noncuts are performed.



DEF REAL MPIT=42, SPL=0, FPL=-35, DM1=42, DM2=42, APP=10, ROP=3, TDEP=1.23, FAL=0, IANG=30, NSP=0 DEF INT NRC=5, NID=2, VARI=3, NUMT=1	;Definition of parameters with ;value assignments
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, -> -> DM2, APP, ROP, TDEP, FAL, IANG, -> -> NSP, NRC, NID, VARI, NUMT)	;Cycle call
N40 G90 G0 X100 Z100	;Approach next position
N50 M30	;End of program

-> Must be programmed in a single block

## 4.8 Thread chaining – CYCLE98



### Programming

CYCLE98 (PO1, DM1, PO2, DM2, PO3, DM3, PO4, DM4, APP, ROP, TDEP, FAL, IANG, NSP, NRC, NID, PP1, PP2, PP3, VARI, NUMT, \_VRT)



### Parameters

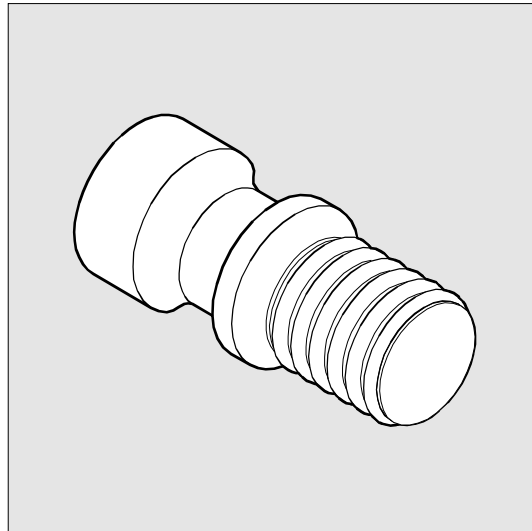
PO1	real	Starting point of the thread in the longitudinal axis
DM1	real	Diameter of the thread at the starting point
PO2	real	First intermediate point in the longitudinal axis
DM2	real	Diameter at the first intermediate point
PO3	real	Second intermediate point
DM3	real	Diameter at the second intermediate point
PO4	real	End point of the thread in the longitudinal axis
DM4	real	Diameter at the end point
APP	real	Arc-in section (enter without sign)
ROP	real	Arc-out section (enter without sign)
TDEP	real	Thread depth (enter without sign)
FAL	real	Final machining allowance (enter without sign)
IANG	real	Infeed angle Value range: "+" (for flank infeed on flank) "-" (for alternating flank infeed)
NSP	real	Starting point offset for the first thread (enter without sign)
NRC	int	Number of rough cuts (enter without sign)
NID	int	Number of noncuts (enter without sign)
PP1	real	Thread pitch 1 as value (enter without sign)
PP2	real	Thread pitch 2 as value (enter without sign)
PP3	real	Thread pitch 3 as value (enter without sign)
VARI	int	Definition of the machining type for the thread Value range: 1 ... 4
NUMT	int	Number of thread starts (enter without sign)
_VRT	real	Variable retraction distance based on initial diameter, incremental (enter without sign)

## 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. In multiple thread cutting, the thread starts are cut one after the other. Infeed is automatic. You can select either constant infeed per cut or constant cross-section of cut. A right-hand or left-hand thread is determined by the direction of rotation of the spindle, which is programmed before the cycle call. 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 Guide Fundamentals  
Chapter "Thread cutting with constant pitch, G33"



### Sequence

#### Position reached before the beginning of the cycle:

The starting position is any position, from which the programmed thread starting point + arc-in section can be approached without collision.

#### The cycle implements the following motion sequence:

- Approach the starting point determined by the cycle at the beginning of the arc-in section for the first thread start with G0.
- Infeed to commence roughing according to the infeed type defined under VARI.
- Thread cutting is repeated according to the number of roughing cuts programmed.
- The final machining allowance is cut in the next cut with G33.
- This cut is repeated according to the number of programmed noncuts.
- The total motion sequence is repeated for each additional thread start.



### Explanation of parameters

#### PO1 and DM1 (starting point and diameter)

You define the original starting point of the thread chain using these parameters. The starting point calculated by the cycle that is initially approached with G0 is the length of the arc-in section in front of the programmed starting point (starting point A in the figure on the preceding page).

#### PO2, DM2 and PO3, DM3 (intermediate point and diameter)

You define two intermediate points in the thread using these parameters.

#### PO4 and DM4 (endpoint and diameter)

The original end point of the thread is programmed under parameters PO4 and DM4.



With an inside thread, DM1...DM4 corresponds to the tap hole diameter.

#### Connection between APP and ROP (arc-in, arc-out sections)

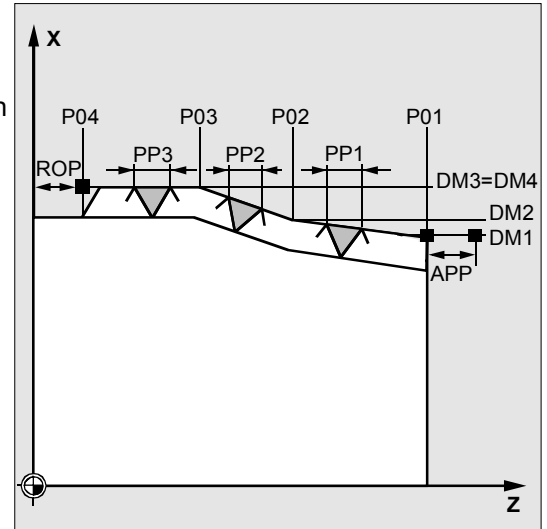
The starting point used in the cycle is the starting point brought forward by the arc-in section APP and, similarly, the end point is the programmed end point brought back by the arc-out section ROP.

The starting point defined by the cycle always lies 1 mm outside the programmed thread diameter in the facing axis. This retraction plane is automatically generated by the control.

#### Connection between TDEP, FAL, NRC and NID (thread depth, final machining allowance, number of rough cuts and noncuts)

The programmed final machining allowance is subtracted from the defined thread depth TDEP and the remainder is divided into rough cuts. The cycle automatically calculates the individual current infeed depths, 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. Infeed is then performed with differing values for the infeed depth.

In a second method, the total thread depth is divided into constant infeed depths. The cross-section of cut increases from cut to cut. However, if the values for the thread depth are small, this method can create better cutting conditions.

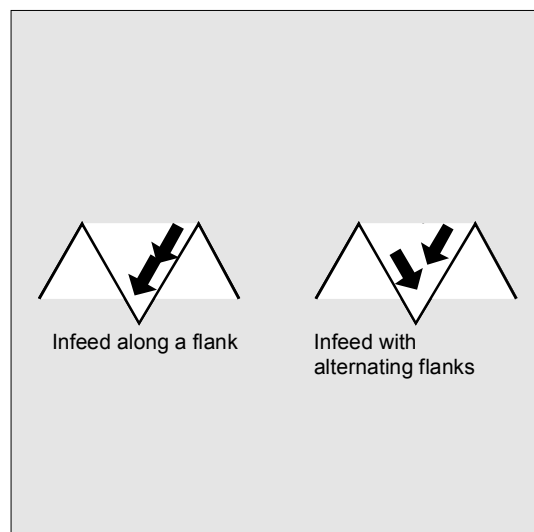
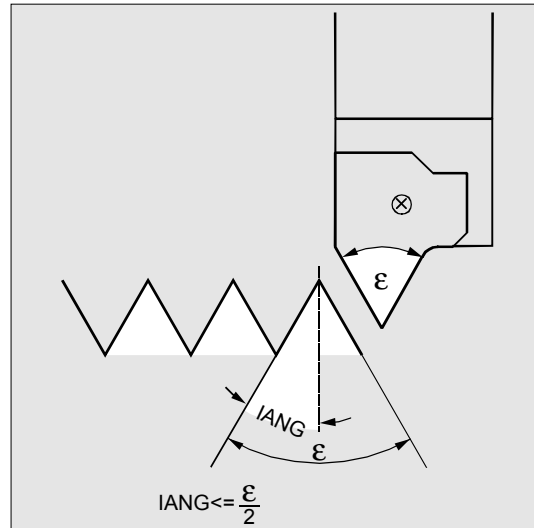


## 4.8 Thread chaining – CYCLE98

The final machining allowance, FAL, is removed in one cut after roughing. After this, the noncuts programmed under the NID parameter are executed.

### IANG (infeed angle)

You define the infeed angle with the IANG parameter. If infeed is to be performed at right angles to the cutting direction in the thread, this parameter must be assigned the value zero, i.e., this parameter can also be omitted from the parameter list, as it is then automatically assigned the default value zero. If infeed is to be performed along the flank, the absolute value of this parameter must be no more than half the flank angle of the tool. The sign entered for this parameter defines how this infeed is performed. If a positive value is entered, infeed is always performed on the same flank, if a negative value is entered, infeed is performed alternately on both flanks. The alternate flanks infeed type can only be used for cylindrical threads. However, if a negative value is assigned to parameter IANG for a tapered thread, the cycle automatically performs a flank infeed along one flank.



### NSP (starting point offset)

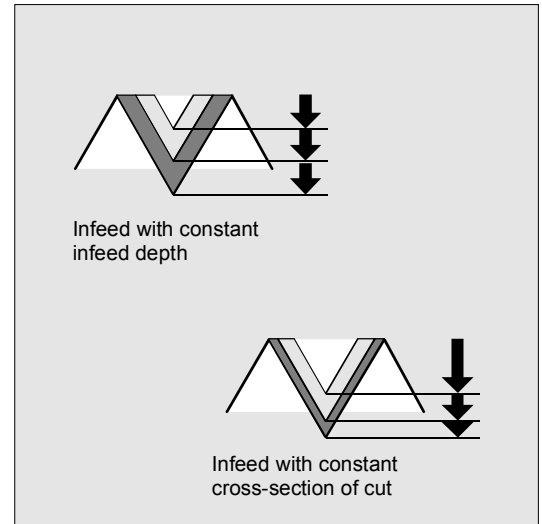
With this parameter, you can program the angular value that defines the point of the first cut of the first thread start on the circumference of the turned part. This value is a starting point offset. The parameter can be assigned any value between 0.0001 and +359.9999 degrees. If no starting point offset has been entered or the parameter has been omitted from the parameter list, the first thread automatically starts at the zero degrees 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 a 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 be a value between 1 and 4 with the following meaning:



Value	External/internal	Const. infeed/const. cross-section of cut
1	External	Constant infeed
2	Internal	Constant infeed
3	External	Constant cross-section of cut
4	Internal	Constant cross-section of cut

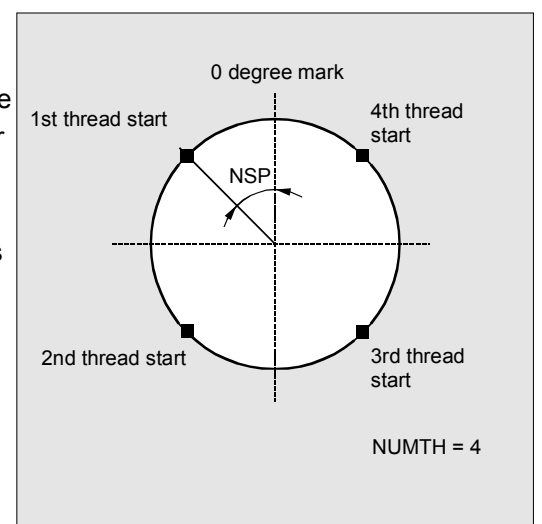
If another value is assigned to the VARI parameter, the cycle is aborted and the following alarm: 61002 "Machining type incorrectly programmed".

**NUMT (number of thread starts)**

You specify the number of thread starts for a multiple thread with the NUMT parameter. If you require a single thread, either assign the value zero to the parameter or omit it from the parameter list.

The thread starts are distributed uniformly around the circumference of the turned part, the first thread start is defined in the NSP parameter.

If a multiple-start thread with a non-uniform distribution of thread starts around the circumference is to be machined, the cycle must be called for every thread start and the corresponding starting point offset must be programmed.



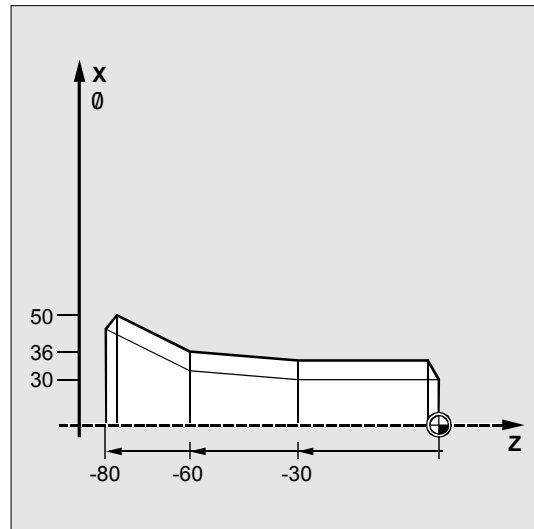
**\_VRT (variable retraction path)**

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.

**Programming example****Thread chain**

With this program, you can produce a chain of threads, starting with a cylindrical thread. Infeed is perpendicular to the thread. Neither a final machining allowance nor a starting point offset have been 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, -> -> 36, -80, 50, 10, 10, 0.92, , , , -> -> 5, 1, 1.5, 2, 2, 3, 1)	;Cycle call
N40 G0 X55	;Traverse axis by axis
N50 Z10	
N60 X40	
N70 M30	;End of program

-> Must be programmed in a single block



## 4.9 Thread recutting



Thread cutting cycles CYCLE97 and CYCLE98 can be used for 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.

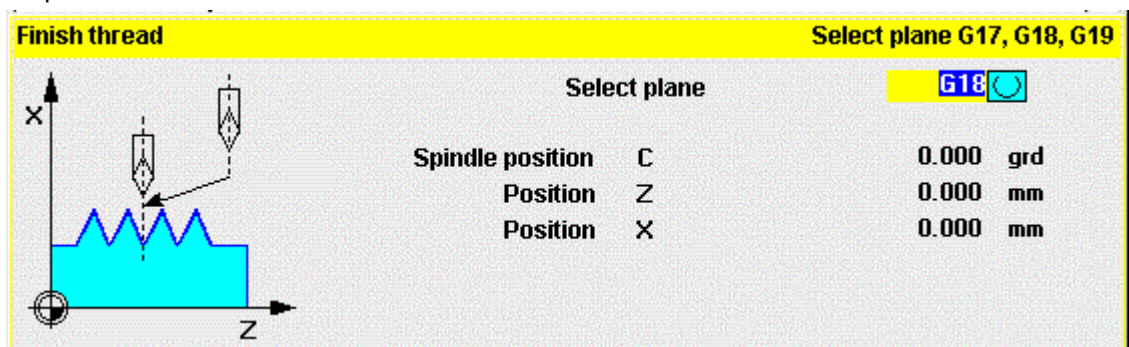
### Prerequisites

The channel, in which the thread recutting program must be executed, is already selected; the relevant axes must already be referenced. The channel is in the Reset state, the spindle is stationary.



### Sequence

- Select JOG in "Machine" operating area.
- Press the "Finish thread" soft key  
→ Open screen form for this function.



- 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).

## 4.9 Thread recutting

- 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.



### Startup

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



The extended stock removal cycle is an option.  
It requires SW 6 in both the NCK and HMI Advanced.



### 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)
```



### Parameters

<u>_NP1</u>	string	Name of the contour subroutine for the finished part contour
<u>_NP2</u>	string	Label/block number start of finished part contour, optional (this can be used to define contour sections)
<u>_NP3</u>	string	Label/block number end of finished part contour, optional (this can be used to define contour sections)
<u>_NP4</u>	string	Name of the stock removal program to be generated
<u>_VARI</u>	integer	Type of machining: (enter without sign) UNITS DIGIT: Values: 1...Longitudinal 2...Face 3...Parallel to contour TENS DIGIT: Values: 1...Programmed infeed direction X- 2...Programmed infeed direction X+ 3...Programmed infeed direction Z- 4...Programmed infeed direction Z+ HUNDREDS DIGIT: Values: 1...Roughing 2...Finishing 3...Complete THOUSANDS DIGIT: Values: 1...With rounding 2...Without rounding (liftoff) TEN THOUSANDS DIGIT: Values: 1...Machine relief cuts 2...Do not machine relief cuts HUNDRED THOUSANDS DIGIT: Values: 1...Programmed machining direction X- 2...Programmed machining direction X+ 3...Programmed machining direction Z- 4...Programmed machining direction Z+
<u>_MID</u>	real	Infeed depth (enter without sign)
<u>_FALZ</u>	real	Final machining allowance in the longitudinal axis (enter without sign)

**4.10 Extended stock removal cycle – CYCLE950**

_FALX	real	Final machining allowance in the facing 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	Liftoff distance for roughing, incremental (enter without sign)
_ANGB	real	Liftoff angle for roughing
_SDIS	real	Safety distance 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	int	Absolute or incremental evaluation of the _APZ parameter 90=absolute, 91=incremental
_APX	real	Axial value for defining blank for facing axis
_APXA	int	Absolute or incremental evaluation of the _APX parameter 90=absolute, 91=incremental
_TOL1	real	Blank tolerance



## 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.

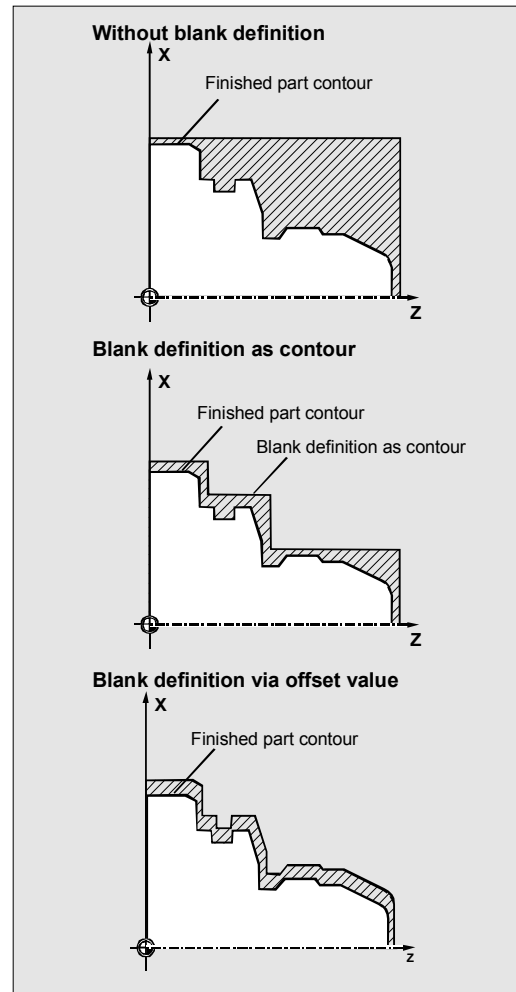
For roughing, the programmed infeed depth is observed precisely; the last two roughing steps are divided equally. Roughing is performed up to the final machining allowance programmed.

Finishing is performed in the same direction as roughing.

The tool radius compensation is automatically selected and deselected by the cycle.

### 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.
- You can specify the contours for stock removal:
  - In a separate program,
  - In the calling main program or
  - As a section of any given 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.



### Sequence

#### Position reached before the beginning of the cycle:

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 path `_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.
- Liftoff and retraction are carried out as paraxial roughing.



### Explanation of 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 programming example 1)
- The contour is defined in the calling program, in which case only \_NP2 and \_NP3 have to be programmed (see programming 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 values are:

**Units digit:**

- 1=Longitudinal
- 2=Face
- 3=Parallel to the contour

**Tens digit:**

- 1=Programmed infeed direction X-
- 2=Programmed infeed direction X+
- 3=Programmed infeed direction Z-
- 4=Programmed infeed direction Z+

**Hundreds digit:**

- 1=Roughing
- 2=Finishing
- 3=Complete

**Thousands digit:**

- 1=With rounding
- 2=Without rounding (liftoff)

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 are retained.

**Ten thousands digit:**

- 1=Machine relief cuts
- 2=Do not machine relief cuts

**Hundred thousands digit:**

- 1=Programmed machining direction X-
- 2=Programmed machining direction X+
- 3=Programmed machining direction Z-
- 4=Programmed machining direction Z+

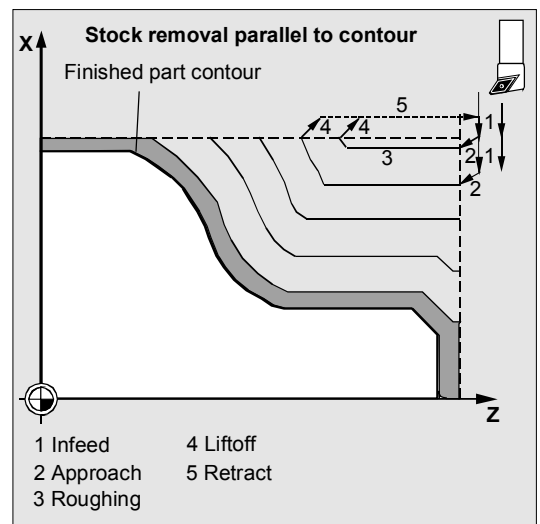
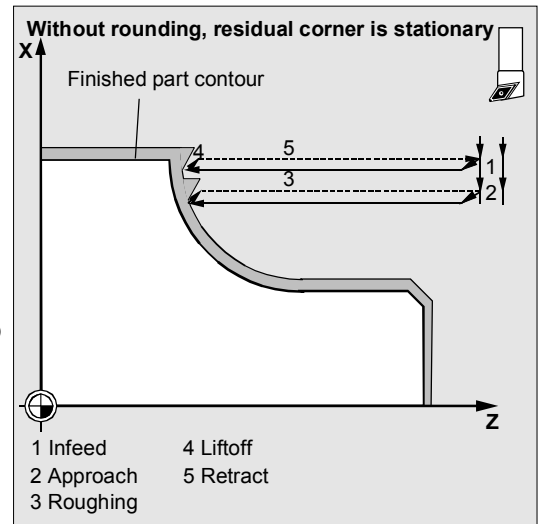
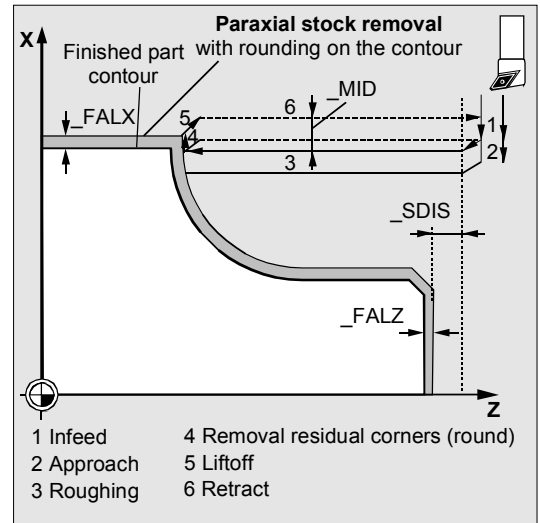
**Example:**

\_VARI=312311 means machining is:

- Longitudinal,
- Infeed direction X- (i.e., external),
- Complete;

The workpiece is not rounded along the contour,

- Relief cuts are machined,
- Machining direction Z-.



## 4.10 Extended stock removal cycle – CYCLE950

### **\_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]`.

`_ZSD[0]=0`: `_MID` 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.

`_ZSD[0]=1`: `_MID` is a radius value

`_ZSD[0]=2`: `_MID` 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 performed to these final machining allowances.

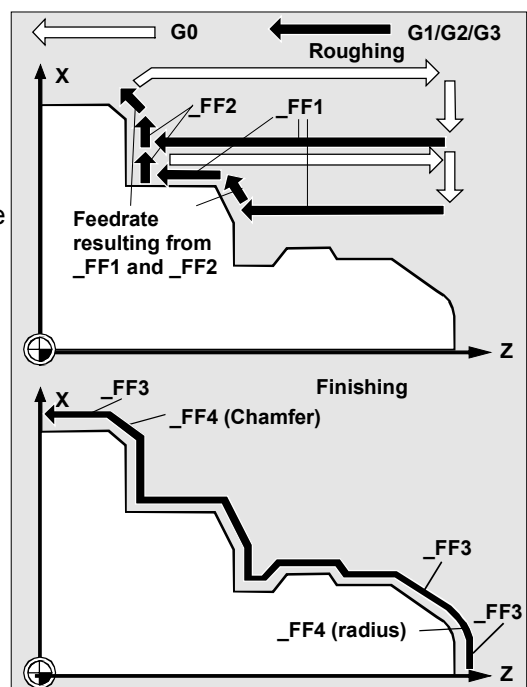
If no final machining allowances are programmed, stock removal is performed up to the end contour during roughing.

### **\_FF1, \_FF2, \_FF3 and FF4 (feedrate)**

Separate feedrates can be specified for roughing and finishing, as shown in the figure opposite. 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 programming example 1 for programming of the parts in the figure opposite)



**\_VRT (lift-off path) and \_ANGB (lift angle)**

The \_VRT parameter can be used to program the amount of liftoff during roughing in both axes.

If \_VRT=0 (parameter not programmed), liftoff is 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 distance)**

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 programming example 3).

An updated blank contour is always generated when a travel program is generated.

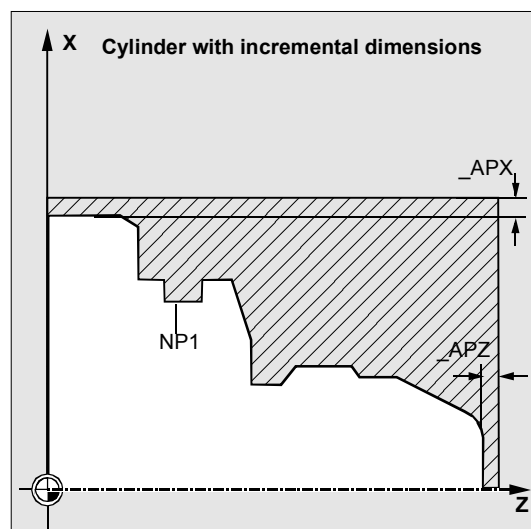
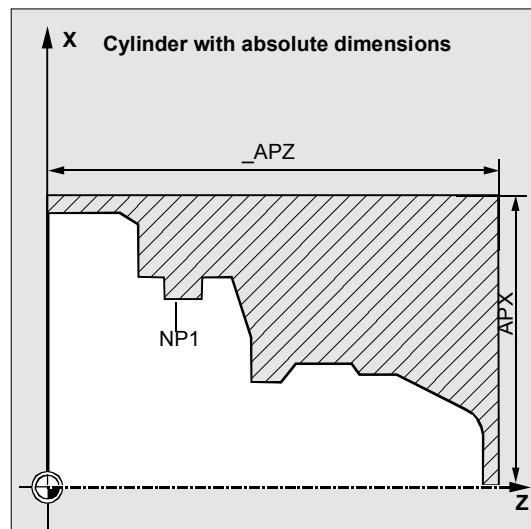
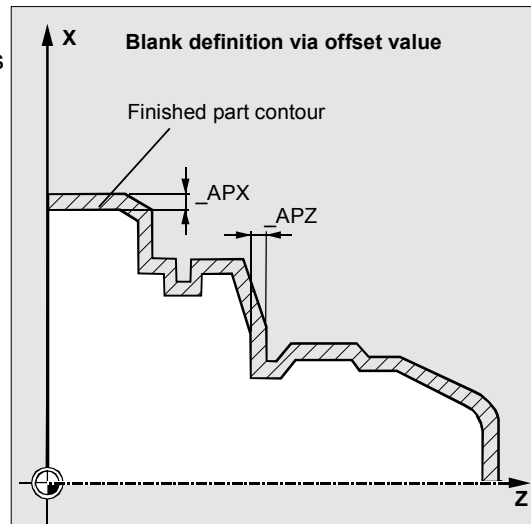
## 4.10 Extended stock removal cycle – CYCLE950

### \_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.

### **Other information**

#### **Contour definition**

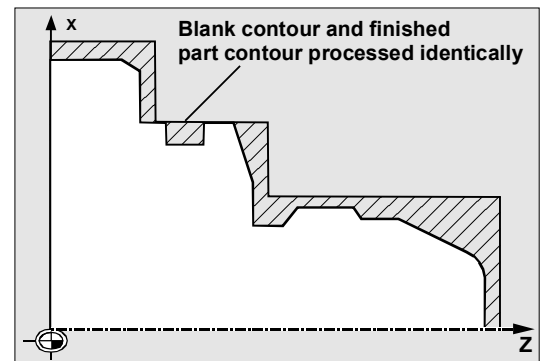
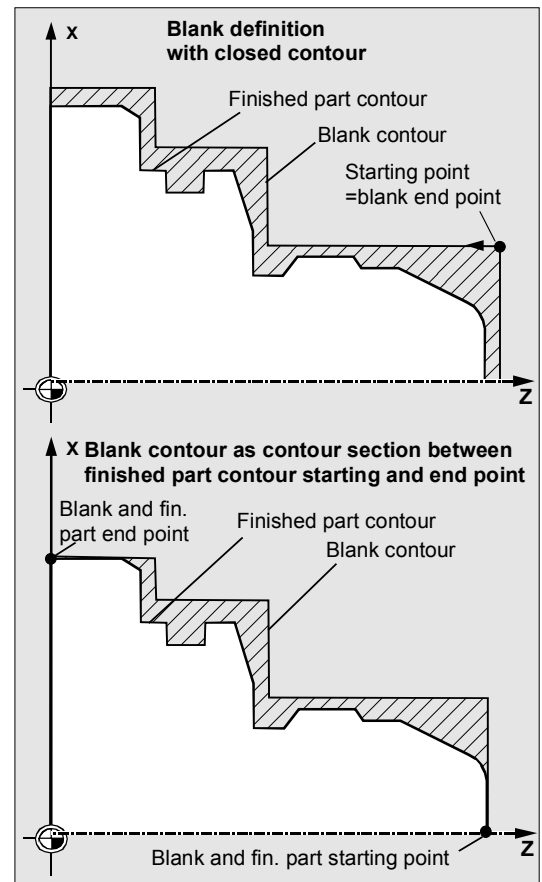
Unlike CYCLE95, one block that contains a link to the current plane is sufficient for contour programming.

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;
- A transfer parameter of the cycle has 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.

### Note on simulation

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.

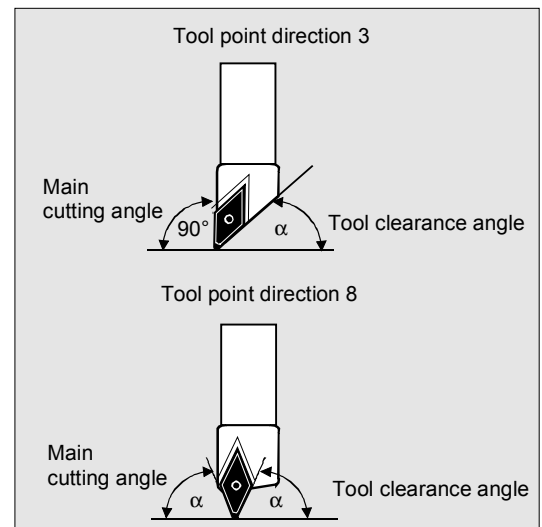
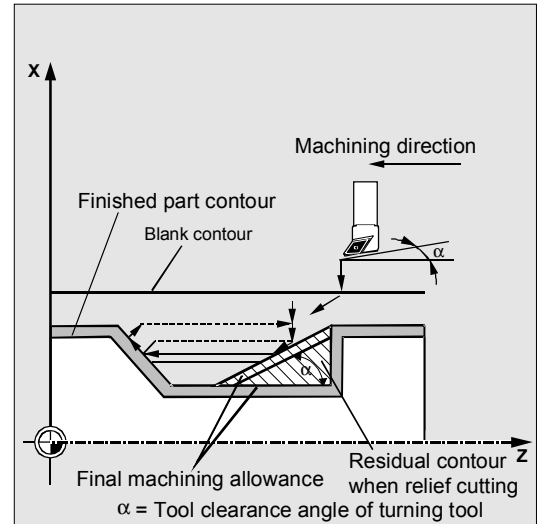
The tool clearance angle must be entered in the tool offset data (parameter 24).

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.



Extended stock removal cannot be performed in m:n configurations.



## 4.10 Extended stock removal cycle – CYCLE950



## Programming example 1

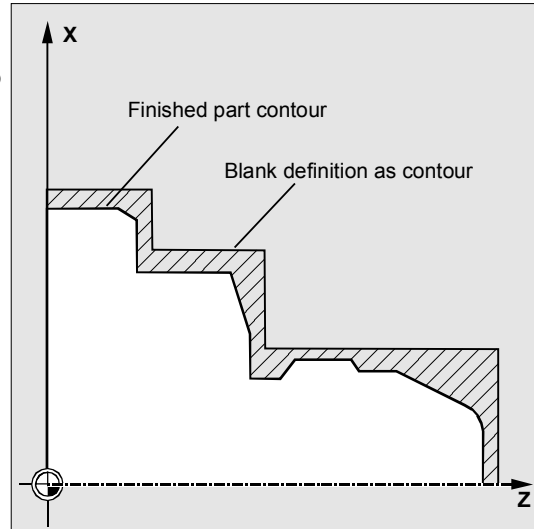
The contour saved in the PART1.MPF program is to be machined from a preformed blank.

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_NE
W_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",,, "Machine_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 MACHINING\_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.

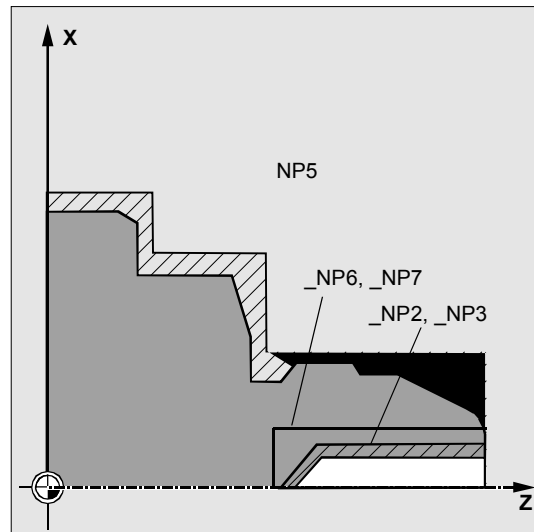


### Programming example 2

A simple inside contour is to be machined on the same part as in programming example 1. A center bore is first made using a diameter 10 drill. The inside contour then is roughed parallel to the contour, since the hole roughly corresponds to the end 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",  
"Machine_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
```



### Programming example 3

The same part from programming example 1 should now 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_WPD
; 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

```

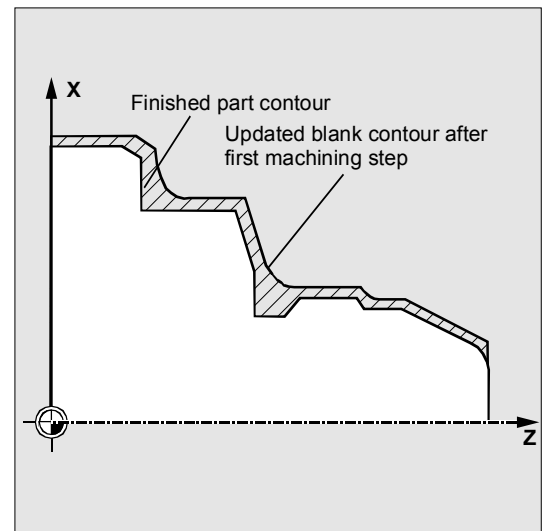
```

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",,,,"Machine_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",,,,"Finish_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 for programming example 1





## Explanation

### Alarm source CYCLE950

Alarm number	Alarm text	Explanation, remedy
61701	"Error in contour description of finished part"	Either none of the _NP1, _NP2 or _NP3 parameters is assigned or error when programming the finished part contour.
61702	"Error in contour description of blank"	Either none of the _NP5, _NP6 or _NP7 parameters is assigned or error when programming the blank contour.
61703	"Internal cycle error while deleting file"	
61704	"Internal cycle error while writing file"	
61705	"Internal cycle error while reading file"	
61706	"Internal cycle error during checksum formation"	
61707	"Internal cycle error during ACTIVATE at MMC"	
61708	"Internal cycle error during READYPROG at MMC"	
61709	"Timeout for contour calculation"	
61720	"Illegal input"	
61721	"Error: unable to determine contour direction"	
61722	"System error"	
61723	"Unable to perform machining"	Use a tool with a larger clearance angle.
61724	"No material available"	
61725	"Out of memory, error in contour generation"	
61726	"Internal error: Out of memory _FILECTRL_INTERNAL_ERROR"	
61727	"Internal error: Out of memory _FILECTRL_EXTERNAL_ERROR"	
61728	"Internal error: Out of memory _ALLOC_P_INTERNAL_ERROR"	
61729	"Internal error: Out of memory _ALLOC_P_EXTERNAL_ERROR"	
61730	"Internal error: Invalid Memory"	
61731	"Internal error: Floating-point exception"	
61732	"Internal error: Invalid instruction"	
61733	"Internal error: Floating_Point_Error"	
61734	"Tool point position not compatible with cutting direction"	

Alarm number	Alarm text	Explanation, remedy
61735	"Finished part lies outside blank contour"	Check definition of blank contour
61736	"Tool insert length < machining depth"	
61737	"Machining_Depth_Of_Cut > Max._Tool_Cutting_Depth"	
61738	"Machining_Cutting_Depth < Min._Tool_Cutting_Depth"	
61739	"Incorrect tool position for this type of machining"	
61740	"Blank must be a closed contour"	Blank contour must be closed, starting point = end point
61741	"Out of memory"	
61742	"Collision during approach, offset not possible"	



## Notes



## Error Messages and Error Handling

5.1	General .....	5-354
5.2	Error handling in the cycles.....	5-354
5.3	Overview of cycle alarms .....	5-355
5.4	Cycle messages.....	5-361

## 5.1 General

If error conditions are detected in the cycles, an alarm is output and execution of the cycle is aborted.

The cycles continue to output messages in the dialog line of the control. These messages do not interrupt execution.



For more information on errors and required responses, as well as messages output in the control's dialog line, please refer to the section for the relevant cycle.

## 5.2 Error handling in the cycles

If error conditions are detected in the cycles, an alarm is output and processing is aborted.

Alarms with numbers between 61000 and 62999 are output in the cycles. This range is again subdivided according to alarm responses and delete criteria.

The error text displayed with the alarm number provides an explanation of the cause of the error.

Alarm number	Delete criterion	Alarm response
61000 ... 61999	NC_RESET	Block preprocessing in the NC is aborted
62000 ... 62999	Backspace key	Program execution is not interrupted; display only

### 5.3 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 Drilling, drilling pattern and milling cycle alarms
- X=6 Turning cycle alarms

The table below lists the errors that occur in the cycles, when they occur and how to eliminate them.

Alarm number	Alarm text	Source	Explanation, remedy
61000	"No tool offset active"	LONGHOLE SLOT1 SLOT2 POCKET1 to POCKET4 CYCLE71 CYCLE72 CYCLE90 CYCLE93 to CYCLE96	D offset must be programmed before the cycle call.
61001	"Thread pitch incorrectly defined"	CYCLE84 CYCLE840 CYCLE96 CYCLE97	Check parameters for thread size and check pitch information (contradict each other).
61002	"Machining type incorrectly defined"	SLOT1 SLOT2 POCKET1 to POCKET4 CYCLE71 CYCLE72 CYCLE76 CYCLE77 CYCLE93 CYCLE95 CYCLE97 CYCLE98	The value assigned to the VARI parameter for the machining type is incorrect and must be altered.

## 5.3 Overview of cycle alarms

Alarm number	Alarm text	Source	Explanation, remedy
61003	"No feedrate programmed in cycle"	CYCLE71 CYCLE72	The feedrate parameter has been incorrectly set and must be altered.
61005	"3rd geometry axis not available"	CYCLE86	With an application on the turning machine with no Y axis in the G18 plane.
61009	"Active tool number = 0"	CYCLE71 CYCLE72	No tool (T) has been programmed before the cycle call.
61010	"Final machining allowance too large"	CYCLE72	The final machining allowance on the base is greater than the total depth and must be reduced.
61011	"Scaling not allowed"	CYCLE71 CYCLE72	A scale factor is currently active that is not permissible for this cycle.
61012	"Different scales in one plane"	CYCLE76 CYCLE77	
61014	"Retraction plane is exceeded"	CYCLE72	Check the RTP parameter.
61019	"Parameter incorrectly set"	CYCLE83 CYCLE60	Check the parameter value.
61101	"Reference plane incorrectly defined"	CYCLE71 CYCLE72 CYCLE81 to CYCLE90 CYCLE840 SLOT1 SLOT2 POCKET1 to POCKET4 LONGHOLE	If they are relative depth values, different values must be entered for the reference plane and the retraction plane or an absolute value must be entered for the depth.
61102	"No spindle direction programmed"	CYCLE86 CYCLE87 CYCLE88 CYCLE840 POCKET3 POCKET4	The SDIR parameter (or SDR in CYCLE840) must be programmed.
61103	"Number of holes is zero"	HOLES1 HOLES2	No value has been programmed for the number of holes.
61104	"Contour violation of slots"	SLOT1 SLOT2 LONGHOLE	Incorrect parameterization of the milling pattern in the parameters that define the position of the slots/elongated holes on the circle, and their shape.
61105	"Cutter radius too large"	SLOT1 SLOT2 POCKET1 to POCKET4 LONGHOLE CYCLE90	The diameter of the milling cutter being used is too large for the figure that is to be machined; either a tool with a smaller radius must be used or the contour must be changed.

Alarm number	Alarm text	Source	Explanation, remedy
61106	"Number or spacing of circle elements"	HOLES2 LONGHOLE SLOT1 SLOT2	Incorrect parameterization of NUM or INDA, the circular elements cannot be arranged in a full circle.
61107	"First drilling depth incorrectly defined"	CYCLE83	First drilling depth is inverted in relation to total drilling depth.
61108	"No valid settings for parameters _RAD1 and _DP1"	POCKET3 POCKET4	The _RAD1 and _DP parameters, which define the path for depth infeed, have been incorrectly set.
61109	"Parameter _CDIR incorrectly defined"	POCKET3 POCKET4	The value of the _CDIR parameter for milling direction has been incorrectly set and must be altered.
61110	"Finishing allowance on base > infeed depth"	POCKET3 POCKET4	The final machining allowance on the base has been set to a higher value than the maximum depth infeed; either reduce final machining allowance or increase depth infeed.
61111	"Infeed width > tool diameter"	CYCLE71 POCKET3 POCKET4	The programmed infeed width is greater than the diameter of the active tool and must be reduced.
61112	"Tool radius negative"	CYCLE72 CYCLE76 CYCLE77 CYCLE90	The radius of the active tool is negative, the setting must be changed to a positive value.
61113	"Parameter _CRAD for corner radius too large"	POCKET3	The _CRAD parameter for the corner radius has been set too high and must be reduced.
61114	"Direction of machining G41/G42 incorrectly defined"	CYCLE72	The machining direction of the cutter radius compensation G41/G42 has been incorrectly set.
61115	"Contour approach or return mode (straight line/circle/plane/ space) incorrectly defined"	CYCLE72	The contour approach or return mode has been incorrectly defined; check the _AS1 or _AS2 parameters.
61116	"Approach or return path=0"	CYCLE72	The approach or return path is set to zero and must be increased; check the _LP1 or _LP2 parameters.
61117	"Active tool radius <= 0"	CYCLE71 POCKET3 POCKET4	The radius of the active tool is negative or zero and must be altered.
61118	"Length or width = 0"	CYCLE71	The length or width of the milling surface is not permissible; check the _LENG and _WID parameters.

## 5.3 Overview of cycle alarms

Alarm number	Alarm text	Source	Explanation, remedy
61124	"Infeed width is not programmed"	CYCLE71	A value for the <code>_MIDA</code> infeed width must always be programmed for active simulation without a tool.
61125	"Technology selection in the <code>_TECHNO</code> parameter incorrectly defined"	CYCLE84 CYCLE840	Check the <code>_TECHNO</code> parameter.
61126	"Thread length too short"	CYCLE840	Program lower spindle speed/raise reference plane.
61127	"Speed ratio of tapping axis incorrectly defined (machine data)"	CYCLE84 CYCLE840	Check machine data 31050 and 31060 in the appropriate gear stage of the tapping axis.
61128	"Plunge angle=0 for insertion with oscillation or helix"	SLOT1	Check the <code>_STA2</code> parameter.
61180	"No name assigned to swivel data record even though machine data <code>\$MN_MM_NUM_TOOL_CARRIER &gt; 1</code> "	CYCLE800	No name assigned to swivel data record even though several swivel data records exist ( <code>\$MN_MM_NUM_TOOL_CARRIER &gt; 0</code> ) or no swivel data record defined ( <code>\$MN_MM_NUM_TOOL_CARRIER = 0</code> ).
61181	"NCK software version too old (no <code>TOOLCARRIER</code> functionality)"	CYCLE800	<code>TOOLCARRIER</code> functionality with NCU 6.3xx and higher
61182	"Name of swivel data record unknown"	CYCLE800	See Swivel cycle startup CYCLE800 → Kinematics name (swivel data record).
61183	"Retraction mode <code>GUD7_TC_FR</code> outside value range 0..2"	CYCLE800	See Swivel cycle startup CYCLE800 → Retraction; 1st transfer parameter <code>CYCLE800(x,...)</code> is faulty >2.
61184	"No solution can be found with current angle inputs"	CYCLE800	
61185	"Angle ranges declared for rotary axes missing or incorrect (min.>max.)"	CYCLE800	Check startup of swivel cycle CYCLE800.
61186	"Invalid rotary axis vectors → check startup of swivel cycle CYCLE800"	CYCLE800	Startup of swivel cycle CYCLE800: missing or incorrect entry for rotary axis vector V1 or V2.
61187	"Block search mode impermissible → select block search with contour calculation"	CYCLE800	Select block search with contour calculation.

Alarm number	Alarm text	Source	Explanation, remedy
61188	"No axis name declared for 1 <sup>st</sup> rotary axis → check startup of swivel cycle CYCLE800"	CYCLE800	Startup of swivel cycle CYCLE800: missing entry under rotary axis 1 identifier.
61191	"5-axis transformation not set up"	CYCLE832	
61192	"Second 5-axis transformation not set up"	CYCLE832	
61193	"Compressor option not set up"	CYCLE832	
61194	"Spline interpolation option not set up"	CYCLE832	
61200	"Too many elements in machining block"	CYCLE76 CYCLE77	Revise machining block, delete elements if necessary.
61213	"Circle radius too small"	CYCLE77	Enter a larger value for the circle radius.
61215	"Blank dimension incorrectly programmed"	CYCLE76 CYCLE77	Check dimensions of blank spigot. The blank spigot must be larger than the finished spigot.
61601	"Finished part diameter too small"	CYCLE94 CYCLE96	A finished part diameter has been programmed.
61602	"Tool width incorrectly defined"	CYCLE93	Grooving tool is larger than the programmed groove width.
61603	"Groove shape incorrectly defined"	CYCLE93	<ul style="list-style-type: none"> <li>• Radii/chamfers at the groove base do not match the groove width.</li> <li>• Face groove on a contour element running parallel to the longitudinal axis is not possible.</li> </ul>
61604	"Active tool is in violation of programmed contour"	CYCLE95	Contour violation in relief cut elements as a result of the clearance angle of the tool being used, i.e., use a different tool or check the contour subroutine.
61605	"Contour incorrectly programmed"	CYCLE76 CYCLE77 CYCLE95	Illegal relief cut element detected.
61606	"Error during contour preparation"	CYCLE95	An error was detected during contour preparation, this alarm is always output with NCK alarm 10930 ... 10934, 15800 or 15810.
61607	"Starting point incorrectly programmed"	CYCLE95	The starting point reached before the cycle was called does not lie outside the rectangle described by the contour subroutine.
61608	"Wrong tool point position programmed"	CYCLE94 CYCLE96	A tool point position of between 1 and 4 that matches the undercut form must be programmed.

## 5.3 Overview of cycle alarms

Alarm number	Alarm text	Source	Explanation, remedy
61609	"Form incorrectly programmed"	CYCLE94 CYCLE96 LONGHOLE POCKET3 SLOT1	Check parameter for undercut form or shape of slot or pocket.
61610	"No infeed depth programmed"	CYCLE76 CYCLE77 CYCLE96	
61611	"No intersection found"	CYCLE95	No intersection could be calculated with the contour. Check contour programming or change infeed depth.
61612	"Thread cannot be recut"	CYCLE97 CYCLE98	
61613	"Undercut position incorrectly defined"	CYCLE94 CYCLE96	Check value in the _VARI parameter.
61803	"Programmed axis does not exist"	CYCLE83 CYCLE84 CYCLE840	Check the _AXN parameter.
61807	"Incorrect spindle direction programmed (active)"	CYCLE840	Check the SDR and SDAC parameters.
61905	"Cutter radius too small"	SLOT2	The diameter of the cutter being used is too small, residual material remains in the slot; use a tool with a larger radius.
62100	"No drilling cycle active"	HOLES1 HOLES2	No drilling cycle has been modally called prior to the drilling pattern cycle.
62101	"Mill direction incorrect – G3 will be generated"	POCKET3 POCKET4 SLOT2	Check value in the CDIR parameter.
62105	"Number of columns or rows is zero"	CYCLE800	
62180	"Set rotary axes x.x [deg]"	CYCLE800	Angles to be set for manual rotary axes
62181	"Set rotary axes x.x [deg]"	CYCLE800	Angle to be set for manual rotary axis



## 5.4 Cycle messages

The cycles output messages in the dialog line of the control. These messages do not interrupt execution. They provide information about specific cycle behavior and how machining is progressing, and are usually displayed for the duration of the machining operation or until the end of the cycle. The following messages can be displayed:

Message text	Source
"Depth: According to value for relative depth"	CYCLE81...CYCLE89, CYCLE840
"Machining elongated hole"	LONGHOLE
"Machining slot"	SLOT1
"Machining circumferential slot"	SLOT2
"Incorrect mill direction: G3 will be generated"	SLOT1, SLOT2, POCKET1, POCKET2, CYCLE90
"Changed form of the undercut"	CYCLE94, CYCLE96
"1st drilling depth: According to value for relative depth"	CYCLE83
"Caution: Final machining allowance $\geq$ tool diameter!"	POCKET1, POCKET2
"Thread start: - machine as longitudinal thread"	CYCLE97, CYCLE98
"Thread start: - machine as 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

■

## Notes

## Appendix

A	Abbreviations .....	A-364
B	Definition of Terms .....	A-373
C	References .....	A-393
D	Identifiers .....	A-405
E	Index .....	I-409
F	Commands, identifiers .....	I-411

**A Abbreviations**

<b>AS</b>	Automation System
<b>ASCII</b>	American Standard Code for Information Interchange
<b>ASIC</b>	Application-Specific Integrated Circuit: User switching circuit
<b>ASUB</b>	Asynchronous SUBroutine
<b>AuxF</b>	Auxiliary Function
<b>AV</b>	Job planning
<b>BA</b>	Operating mode
<b>BAG</b>	Mode group
<b>BB</b>	Ready
<b>BCD</b>	Binary-Coded Decimals: Decimal numbers coded in binary format
<b>BCS</b>	Basic Coordinate System
<b>BIN</b>	BINary files
<b>BIOS</b>	Basic Input Output System
<b>BOT</b>	BoOT files: Boot files for SIMODRIVE 611D
<b>BP</b>	Basic Program
<b>C Bus</b>	Communications Bus
<b>CAD</b>	Computer-Aided Design
<b>CAM</b>	Computer-Aided Manufacturing
<b>CNC</b>	Computerized Numerical Control
<b>COM</b>	COMmunication

<b>COR</b>	COordinate Rotation
<b>CP</b>	Communications Processor
<b>CPU</b>	Central Processing Unit
<b>CR</b>	Carriage Return
<b>CRC</b>	Cutter Radius Compensation
<b>CRT</b>	Cathode Ray Tube: Picture tube
<b>CSB</b>	Central Service Board: PLC module
<b>CSF</b>	Control System Flowchart (programming method for PLC)
<b>CTS</b>	Clear To Send: Signal from serial data interfaces
<b>CUTOM</b>	CUTter radius cOMPensation: Tool radius compensation
<b>DAC</b>	Digital-to-Analog Converter
<b>DB</b>	Data Block in the PLC
<b>DBB</b>	Data Block Byte in the PLC
<b>DBW</b>	Data Block Word in the PLC
<b>DBX</b>	Data block bit in the PLC
<b>DC</b>	Direct Control: Movement of the rotary axis across the shortest path to the absolute position within one revolution
<b>DCD</b>	Data Carrier Detect
<b>DCE</b>	Data Communication Equipment
<b>DDE</b>	Dynamic Data Exchange
<b>DIN</b>	Deutsche Industrie Norm (German Industry Standard)
<b>DIO</b>	Data Input/Output: Data transfer display
<b>DIR</b>	DIRectory
<b>DLL</b>	Dynamic Link Library

<b>DOS</b>	Disk Operating System
<b>DPM</b>	Dual-Port Memory
<b>DPR</b>	Dual-Port RAM
<b>DRAM</b>	Dynamic Random Access Memory
<b>DRF</b>	Differential Resolver Function (handwheel)
<b>DRY</b>	DRY run: Dry run feedrate
<b>DSB</b>	Decoding Single Block
<b>DTE</b>	Data Terminal Equipment
<b>DW</b>	Data Word
<b>EIA code</b>	Special punched tape code, number of holes per character always odd
<b>ENC</b>	ENCoder: Actual-value sensor
<b>EPROM</b>	Erasable Programmable Read-Only Memory
<b>ERROR</b>	ERROR from printer
<b>FB</b>	Function Block
<b>FBS</b>	Slimline screen
<b>FC</b>	Function Call: Function block on the PLC
<b>FDB</b>	Product database
<b>FDD</b>	Floppy Disk Drive
<b>FDD</b>	Feed Drive
<b>FEPROM</b>	Flash-EPROM: Read and write memory
<b>FIFO</b>	First In First Out: memory, which works without address specification and whose data are read in the order in which they were stored.
<b>FIPO</b>	Fine InterPOLator

<b>FM</b>	Function Module
<b>FPU</b>	Floating Point Unit
<b>FRA</b>	FRAME block
<b>FRAME</b>	Data block (FRAME)
<b>FST</b>	Feed STop
<b>GUD</b>	Global User Data
<b>HD</b>	Hard Disk
<b>HEX</b>	HEXadecimal number
<b>HHU</b>	HandHeld Unit
<b>HMS</b>	High-resolution Measuring System
<b>HW</b>	HardWare
<b>I</b>	Input
<b>I/O</b>	Input/Output
<b>I/RF</b>	Incoming/Regenerative Feedback unit (power supply) on the SIMODRIVE 611(D)
<b>IBN</b>	Startup
<b>IC (GD)</b>	Implicit Communication (Global Data)
<b>ICA</b>	Interpolatory CompensAtion
<b>IF</b>	Pulse enable of drive module
<b>IM</b>	Interface Module
<b>IMR</b>	Interface Module Receive: Interface module for receiving data
<b>IMS</b>	Interface Module Send: Interface module for transmitting data
<b>INC</b>	Increment
<b>INI</b>	INItializing data

<b>IPO</b>	InterPOLator
<b>IS</b>	Interface Signal
<b>ISA</b>	Industry Standard Architecture
<b>ISO</b>	International Organization for Standardization
<b>ISO code</b>	Special punched tape code, number of holes per character always even
<b>JOG</b>	JOGging: Setup mode
<b>K1 .. K4</b>	Channel 1 to channel 4
<b>K<sub>ij</sub></b>	Transmission ratio
<b>K<sub>v</sub></b>	Servo gain factor
<b>LAD</b>	LADder logic (programming method for PLC)
<b>LCD</b>	Liquid Crystal Display
<b>LEC</b>	Leadscrew Error Compensation
<b>LED</b>	Light Emitting Diode
<b>LF</b>	Line Feed
<b>LR</b>	Position controller
<b>LUD</b>	Local User Data
<b>MB</b>	MegaByte
<b>MC</b>	Measuring Circuit
<b>MCP</b>	Machine Control Panel
<b>MCS</b>	Machine Coordinate System
<b>MD</b>	Machine Data
<b>MDA</b>	Manual Data Automatic: Manual input
<b>MMC</b>	Man-Machine Communication: User interface on numerical control systems for operator control, programming and simulation



<b>MPF</b>	Main Program File: NC part program (main program)
<b>MPI</b>	MultiPort Interface
<b>MS</b>	MicroSoft (software manufacturer)
<b>MSD</b>	Main Spindle Drive
<b>NC</b>	Numerical Control
<b>NCK</b>	Numerical Control Kernel: NC kernel with block preparation, travel range, etc.
<b>NCU</b>	Numerical Control Unit: NCK hardware unit
<b>NRK</b>	Numeric Robotic Kernel: name of NCK operating system
<b>NURBS</b>	Non-Uniform Rational B-Spline
<b>OB</b>	Organization Block in the PLC
<b>OC&amp;M</b>	Operator Control and Monitoring
<b>OEM</b>	Original Equipment Manufacturer
<b>OI</b>	Operator Interface
<b>OP</b>	Operator Panel: Operating setup
<b>OPI</b>	Operator Panel Interface: Interface for connection to the operator panel
<b>OPT</b>	OPTions
<b>Order No.</b>	Machine-readable product designation
<b>OSI</b>	Open Systems Interconnection: Standardization for computer communication
<b>P bus</b>	I/O (Peripherals) bus
<b>PC</b>	Personal Computer
<b>PCIN</b>	Name of SW for data exchange with the control

<b>PCMCIA</b>	Personal Computer Memory Card International Association plug-in memory normalization board
<b>PG</b>	Programming device
<b>PLC</b>	Programmable Logic Controller
<b>PMS</b>	Position Measuring System
<b>POS</b>	POSitioning
<b>Q</b>	Output
<b>RAM</b>	Random Access Memory: program memory that can be read and written to
<b>REF</b>	REFerence point approach function
<b>REPOS</b>	REPOSition function
<b>RISC</b>	Reduced Instruction Set Computer: Type of processor with small instruction set and ability to process instructions at high speed
<b>ROV</b>	Rapid OVerride
<b>RPA</b>	R-Parameter Active: Memory area in NCK for R-NCK for R parameter numbers
<b>RPY</b>	Roll Pitch Yaw: Type of rotation of a coordinate system
<b>RS-232-C</b>	Serial interface (definition of the exchange lines between DTE and DCE)
<b>RTS</b>	Request To Send: RTS, control signal of serial data interfaces
<b>SBL</b>	Single BLock
<b>SD</b>	Setting Data
<b>SDB</b>	System Data Block
<b>SEA</b>	Setting data active: Identifier (file type) for setting data
<b>SFB</b>	System Function Block
<b>SFC</b>	System Function Call

<b>SK</b>	Soft Key
<b>SKP</b>	SKiP: Skip block
<b>SM</b>	Stepper Motor
<b>SPF</b>	Sub Program File: Subroutine
<b>SR</b>	SubRoutine
<b>SRAM</b>	Static RAM (non-volatile)
<b>SSI</b>	Serial Synchronous Interface
<b>STL</b>	STatement List
<b>SW</b>	SoftWare
<b>SYF</b>	SYstem Files
<b>T</b>	Tool
<b>TC</b>	Tool Change
<b>TEA</b>	Testing data active: Identifier for machine data
<b>TLC</b>	Tool Length Compensation
<b>TNRC</b>	Tool Nose Radius Compensation
<b>TO</b>	Tool Offset
<b>TOA</b>	Tool Offset Active: Identifier (file type) for tool offsets
<b>TRANSMIT</b>	TRANSform Milling Into Turning: Coordinate conversion on turning machines for milling operations
<b>TRC</b>	Tool Radius Compensation
<b>UFR</b>	User FRame: Zero offset
<b>WCS</b>	Workpiece Coordinate System
<b>WOP</b>	Workshop-Oriented Programming
<b>WPD</b>	WorkPiece Directory

<b>ZO</b>	Zero Offset
<b>ZOA</b>	Zero Offset Active: Identifier (file type) for zero offset data
<b>μC</b>	Microcontroller

**B Definition of Terms**

Important terms are listed in alphabetical order. The "->" symbol precedes terms, which are explained in a separate entry in this list.

**A****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 keyswitch settings, which can be evaluated via the PLC.

**Alarms**

All -> messages and alarms are displayed on the operator panel in plain text with date and time as well as the appropriate symbol for the delete criterion. Alarms and messages are displayed separately.

1. Alarms and messages in the parts program  
Alarms and messages can be displayed directly from the parts program in plain text.
2. Alarms and messages from PLC  
Alarms and messages relating to the machine can be displayed directly from the PLC program in plain text.  
No additional function block packages are required for this purpose.

**Approach fixed point**

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.

**Approach machine fixed point**

Approach motion towards one of the predefined -> fixed machine points.

**Auxiliary functions**

Auxiliary functions can be used to pass -> parameters to the -> PLC in -> parts programs, triggering responses there, which are defined by the machine manufacturer.

**B**

<b>Backlash compensation</b>	Compensation of a mechanical machine backlash, e.g., backlash on leadscrews. The backlash compensation can be entered separately for each axis.
<b>Backup battery</b>	The backup battery provides non-volatile storage for the -> user program in the -> CPU and ensures that defined data areas and flags, timers and counters are retentive.
<b>Blank</b>	Workpiece as it is before the part is machined.
<b>Block</b>	A section of a -> parts program terminated with a line feed. A distinction is made between -> main blocks and -> subblocks.
<b>Block search</b>	The block search function allows any point in the parts program to be selected, at which machining must start or be continued. The function is provided for the purpose of testing parts programs or continuing machining after an interruption.
<b>Booting</b>	Loading the system program after power ON.
<b>C</b>	
<b>Channel</b>	A channel can execute a -> parts program independently of the other channels. A channel exclusively controls the axes and spindles assigned to it. Part programs runs of various channels can be coordinated by -> synchronization.
<b>Channel structure</b>	The channel structure makes it possible to process the -> programs of individual channels simultaneously and asynchronously.
<b>Circular interpolation</b>	The -> tool is required to travel in a circle between defined points on the contour at a specified feedrate while machining the workpiece.
<b>CNC</b>	➔ NC
<b>CNC high-level language</b>	The high-level language offers: -> User variables, -> Predefined user variables, -> System variables, -> Indirect programming, -> Arithmetic and trigonometric functions, -> Comparison operations and logic operations, -> Program branches and jumps, -> Program coordination (SINUMERIK 840D), -> Macros.

<b>CNC programming language</b>	The CNC programming language is based on DIN 66025 with high-level language expansions. The -> CNC high-level language and programming support the definition of macros (sequenced statements), for example.
<b>COM</b>	NC control component for the implementation and coordination of communication.
<b>Compensation axis</b>	Axis having a setpoint or actual value modified by the compensation value.
<b>Compensation table</b>	Table containing interpolation points It provides the compensation values of the compensation axis for selected positions on the basic axis.
<b>Compensation value</b>	Difference between the axis position measured by the position sensor and the desired, programmed axis position.
<b>Connecting cables</b>	Connecting cables are pre-assembled or user-assembled 2-wire cables with a connector at each end. They are used to connect the -> CPU to a -> programming device or to other CPUs via the -> multipoint interface (MPI).
<b>Contour</b>	Outline of a -> workpiece.
<b>Contour monitoring</b>	The following error is monitored within a definable tolerance band as a measure of contour accuracy. Overloading of the drive, for example, may result in an unacceptably large following error. In such cases, an alarm is output and the axes are stopped.
<b>Coordinate system</b>	See -> machine coordinate system -> workpiece coordinate system.
<b>CPU</b>	Central Processor Unit, -> PLC
<b>Cycle</b>	Protected subroutine for executing a repeated machining operation on the -> workpiece
<b>Cycle support</b>	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.

**D****Data Block**

1. A data unit on the -> PLC, which can be accessed by -> HIGHSTEP programs.
2. A data unit on 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 transmitting and receiving CNC user data, e.g., parts programs, tool offsets, etc., via the serial interface. The PCIN program can run under 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-diagnosis program as well as test functions for servicing purposes: status, alarm and service displays.

**Digital input/output  
module**

Digital modules are signal transducers 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 zero 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).

**E****Editor**

The editor makes it possible to create, modify, 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.



<b>Exact stop</b>	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.
<b>Exact stop limit</b>	When all path axes reach their exact stop limits, the control responds as if it had reached its precise destination point. The -> parts program continues execution at the next block.
<b>External zero offset</b>	A zero offset specified by the -> PLC.
<b>F</b>	
<b>Feedforward control, dynamic</b>	Contour inaccuracies resulting from following errors can be almost completely eliminated by the dynamic, acceleration-dependent feedforward control function. Feedforward control ensures an excellent degree of machining accuracy even at high tool path velocities. The function can be selected and deselected axis-specifically in the parts program.
<b>Feedrate override</b>	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.
<b>Finished-part contour</b>	Contour of the finished workpiece. See also -> Blank.
<b>Fixed machine point</b>	A point defined uniquely by the machine tool, such as the reference point.
<b>Frame</b>	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.
<b>G</b>	
<b>General reset</b>	The following -> CPU memories are erased by a general reset operation: <ul style="list-style-type: none"><li>• -&gt; Working memory</li><li>• -&gt; Read/write area of the -&gt; load memory</li><li>• -&gt; System memory</li><li>• -&gt; Backup memory</li></ul>

<b>Geometry</b>	Description of a -> workpiece in the -> workpiece coordinate system.
<b>Geometry axis</b>	Geometry axes are used to describe a 2- or 3-dimensional range in the workpiece coordinate system.
<b>Global main program/subroutine</b>	Each global main program/subroutine 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.
<b>Ground</b>	"Ground" is the term applied to all the electrically inactive, interconnected parts of a piece of equipment, which cannot carry any hazardous contact voltage even in the event of a fault.
<b>H</b>	
<b>Helical interpolation</b>	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: <ol style="list-style-type: none"><li>1. Circular movement in one plane</li><li>2. Linear movement perpendicular to this plane.</li></ol>
<b>High-speed digital inputs/outputs</b>	High-speed CNC program routines (interrupt routines), for example, can be started via the digital inputs. High-speed, program-driven switching functions can be initiated via the digital CNC outputs (SINUMERIK 840D).
<b>HIGHSTEP</b>	Combination of the programming features for the -> PLC in the S7-300/400 range.
<b>I</b>	
<b>I/O module</b>	I/O modules create the link between the CPU and the process. I/O modules are: <ul style="list-style-type: none"><li>• -&gt;Digital input/output modules</li><li>• -&gt;Analog input/output modules</li><li>• -&gt;Simulator modules</li></ul>

<b>Identifier</b>	In accordance with DIN 66025, identifiers (names) for variables (calculation variables, system variables, user variables), for subroutines, for vocabulary words and words can contain several 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.
<b>Inch system</b>	Measurement system, which defines distances in inches and fractions of inches.
<b>Inclined-surface machining</b>	Drilling and milling operations on workpiece surfaces that do not lie in the coordinate planes of the machine can be performed easily using the "inclined-surface machining" function.
<b>Increment</b>	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.
<b>Incremental dimension</b>	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.
<b>Initialization block</b>	Initialization blocks are special -> program blocks. They contain value assignments that are performed before program execution. Initialization blocks are used primarily for initializing predefined data or global user data.
<b>Initialization files</b>	An initialization file can be created for each -> workpiece. The various variable value instructions, which apply exclusively to one workpiece, can be stored in it.
<b>Intermediate blocks</b>	Movements with selected tool offset (G41/G42) can be interrupted by a limited number of intermediate blocks (blocks without axis motions in the offset plane). When such blocks are used, the tool offset can still be calculated correctly. The permissible number of intermediate blocks read in advance by the control can be set via system parameters.
<b>Interpolator</b>	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**

Interpolative compensation provides a means of compensating for leadscrew errors (LEC) and measuring-system errors (MSEC) resulting from the production process.

**Interrupt routine**

Interrupt routines are special -> subroutines, which can be started by events (external signals) in the machining process. The parts program block being processed is aborted and the axis position at the instant of interruption is stored automatically.

**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).

**J****JOG**

Control operating mode (setup operation): The machine can be set up in the JOG mode. 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).

**K****Keyswitch**

1. **S7-300**: The keyswitch is the mode selector switch on the -> CPU. The keyswitch is operated by a removable key.
2. **840D/FM-NC**: The keyswitch on the -> machine control panel has 4 positions that are assigned functions by the control's operating system. Further, the keyswitch has three differently colored keys, which can be removed in the specified positions.

**K<sub>ü</sub>**

Transmission ratio

**K<sub>v</sub>**

Servo gain factor, a control variable in a control loop.

**L****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.

<b>Leadscrew error compensation</b>	Compensation for the mechanical inaccuracies of a leadscrew participating in the feed. The control uses stored deviation values for the compensation.
<b>Limit speed</b>	Minimum/maximum (spindle) speed: The maximum speed of a spindle can be limited by values defined in the machine data, the -> PLC or -> setting data.
<b>Linear axis</b>	The linear axis is an axis, which, in contrast to a rotary axis, describes a straight line.
<b>Linear interpolation</b>	The tool travels along a straight line to the destination point while machining the workpiece.
<b>Load memory</b>	With CPU 314 of the -> PLC, the load memory is the equivalent of the -> working memory.
<b>Look Ahead</b>	The <b>Look Ahead</b> function is a means of optimizing the machining velocity by looking ahead over a parameterizable number of traversing blocks.
<b>Look Ahead for contour violations</b>	The control detects and reports the following types of collision: 1. Path is shorter than tool radius. 2. Width of inside corner is less than the tool diameter.
<b>M</b>	
<b>Machine</b>	Operating area of the control.
<b>Machine axes</b>	Physically existent axes on the machine tool.
<b>Machine control panel</b>	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.
<b>Machine coordinate system</b>	A coordinate system, which is related to the axes of the machine tool.
<b>Machine zero</b>	A fixed point on the machine tool, which can be referenced by all (derived) measuring systems.
<b>Macro techniques</b>	Grouping a set of instructions under a single identifier. The identifier in the program refers to the collected sequence of instructions.

<b>Main block</b>	A block prefixed by ":", containing all the parameters required to start execution of a -> parts program.
<b>Main program</b>	-> Parts program identified by a number or name, in which other main programs, subroutines or -> cycles may be called.
<b>MDA</b>	Control operating mode: Manual Data Automatic: In the MDA mode, individual program blocks or block sequences with no reference to a main program or subroutine can be input and executed immediately afterwards through actuation of the NC start key.
<b>Measuring circuit</b>	<ul style="list-style-type: none"><li>• 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.</li><li>• 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.</li></ul>
<b>Messages</b>	All messages programmed in the parts program and -> alarms detected by the system are displayed on the operator panel in plain text with date and time, as well as the appropriate symbol for the delete criterion. Alarms and messages are displayed separately.
<b>Metric system</b>	Standardized system of units: for lengths in millimeters (mm), meters (m), etc.
<b>Mirroring</b>	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.
<b>Module</b>	"Module" is the term given to any files required for creating and processing programs.
<b>Mounting rail</b>	This rail is used to mount the S7-300 system modules.
<b>Multipoint interface</b>	<p>The multipoint interface (MPI) is a 9-pin sub-D port. A parameterizable number of devices can be connected to an MPI for the purpose of communicating with one another:</p> <ul style="list-style-type: none"><li>• Programming devices</li><li>• MMI (HMI) systems</li><li>• Other automation systems</li></ul> <p>The "Multipoint Interface MPI" parameter block of the CPU contains the -&gt; parameters, which define the properties of the multipoint interface.</p>

**N****NC**

Numerical Control: NC incorporates all the components of the machine tool control system: -> 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**

A network is the interconnection of several S7-300s and other terminal devices, such as a programming device, by means of -> connecting cables. The networked devices interchange data via the network.

**Node number**

The node number is the "contact address" of a -> CPU or -> programming device or another intelligent I/O module, if these devices exchange data with one another via a -> network. The node number is assigned to the CPU or the programming device using the -> "S7 Configuration" S7-Tool.

**NRK**

Numeric Robotic Kernel (operating system of the -> NCK)

**NURBS**

Motion control and path interpolation are implemented internally in the control on the basis of NURBS (Non-Uniform Rational B Splines). A standard procedure is thus available (SINUMERIK 840D) as an internal control function for all modes of interpolation.

**O****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.

**Orientable tool retraction**

RETTOOL: 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.

<b>Oriented spindle stop</b>	Stops the workpiece spindle with a specified orientation angle, e.g., to perform an additional machining operation at a specific position.
<b>Override</b>	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.
<b>P</b>	
<b>Parameters</b>	<ol style="list-style-type: none"> <li>1. The <b>S7-300</b> uses two types of parameter: <ul style="list-style-type: none"> <li>- STEP 7 instruction parameter A STEP 7 instruction parameter is the address of the operand to be machined or a constant.</li> <li>- -&gt; Parameter block parameter A parameter block parameter defines a module's behavior.</li> </ul> </li> <li>2. <b>840D</b>: <ul style="list-style-type: none"> <li>- Control operating area</li> <li>- Arithmetic parameters can be set or requested as required by the programmer of the -&gt; parts program for any purpose in that program.</li> </ul> </li> </ol>
<b>Parts program</b>	<p>A sequence of instructions to the NC control, which combine to produce a specific -&gt; workpiece.</p> <p>Likewise, performing a certain machining operation on a specific -&gt; blank.</p>
<b>Parts program management</b>	The parts program management function 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.
<b>PG</b>	Programming device
<b>PLC</b>	<p>Programmable Logic Control: -&gt; Programmable logic control</p> <p>Component of the -&gt; NC control: Programmable controller for processing the control logic of the machine tool.</p>
<b>PLC</b>	➔ Programmable Logic Controller



<b>PLC program memory</b>	<ul style="list-style-type: none"><li>• 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 KB user memory for this purpose.</li><li>• 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.</li></ul>
<b>PLC programming</b>	The PLC is programmed with <b>STEP 7</b> software. STEP 7 programming software is based on the standard <b>WINDOWS</b> operating system and incorporates the functionality of STEP5 programming with innovative expansions and developments.
<b>Polar coordinates</b>	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.
<b>Polynomial interpolation</b>	Polynomial interpolation provides a means of generating a very wide range of curves, including <b>straight-line, parabolic and exponential functions</b> (SINUMERIK 840D).
<b>Positioning axis</b>	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 the -> path axes.
<b>Power On</b>	Control is switched off and then switched on again.
<b>Preset</b>	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.
<b>Program</b>	<ol style="list-style-type: none"><li>1. Operating area of the control.</li><li>2. Sequence of instructions to the control.</li></ol>
<b>Program block</b>	Program blocks contain the main programs and subroutines of the -> parts programs.
<b>Programmable frames</b>	Programmable -> frames can be used to dynamically define new coordinate system starting points 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 controllers, whose functions are stored as a program in the control unit. The design and wiring of the unit are not, therefore, dependent on the control functions. Programmable logic controllers have the same structure as a computer, i.e., they consist of a CPU with memory, input/output modules and an internal bus system. The I/Os and programming language are selected according to the requirements of the control technology involved.

**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 area within a -> working area, which the tool tip is not permitted to enter.

**Q****Quadrant error compensation**

Contour errors at quadrant transitions caused by frictional fluctuations on guideways can be largely eliminated by means of quadrant error compensation. A circularity test is performed to parameterize the quadrant error compensation function.

**R****R parameters**

Arithmetic parameters can be set or requested by the programmer of the -> parts program for any purpose in that program.

**Rapid retraction 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 retracted quickly from the workpiece contour currently being machined. The retraction angle and the distance retracted can also be parameterized. An interrupt routine can be executed after the rapid retraction (SINUMERIK FM-NC, 840D).

**Rapid traverse**

The highest speed of an axis. It is used, for example, to move the tool from rest to the -> workpiece contour or retract the tool from the contour.

<b>Reference point</b>	A point on the machine tool, to which the measurement system of the -> machine axis refers.
<b>Reference point approach</b>	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.
<b>REPOS</b>	<ol style="list-style-type: none"><li>1. Reapproach contour, triggered by operator REPOS allows the tool to be returned to the interrupt position by means of the direction keys.</li><li>2. Programmed contour reapproach A selection of approach strategies are available in the form of program commands: Approach point of interruption, approach start of block, approach end of block, approach a point on the path between start of block and interruption.</li></ol>
<b>Retentive data</b>	Data areas in data blocks and timers, counters, and markers are retentive if their content is not lost on restart or power-off.
<b>Rigid tapping</b>	Rigid tapping can be drilled with the help of this function. When the rigid tapping function is used, interpolation of the spindle acting as a rotary axis and the drilling axis ensures that threads are cut exactly to the final drilling depth, e.g., tapped blind hole (requirement: spindle is operating in axis mode).
<b>Rotary axis</b>	Rotary axes apply a workpiece or tool rotation to a defined angular position.
<b>Rotary axis, turning continuously</b>	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.
<b>Rotation</b>	Component of a -> frame, which defines a rotation of the coordinate system through a specific angle.
<b>Rounding axis</b>	Rounding axes apply a workpiece or tool rotation to an angular position corresponding to an indexing grid. On reaching a grid, the rounding axis is "in position".

**S****S7 Configuration**

S7 Configuration is a tool for parameterizing modules. S7 Configuration is used to set a variety of -> parameter blocks of the -> CPU and the I/O modules on the -> programming device. These parameters are uploaded to the CPU.

**S7-300 bus**

The S7-300 bus is a serial data bus, which supplies modules with the appropriate voltage and, via which they exchange data with one another. The modules are connected by -> bus connectors.

**Safety functions**

The control incorporates monitors, which are active at all times and, which are designed to detect malfunctions in the -> CNC, the programmable controller (-> PLC) and the machine at an early stage, in order to minimize the risk of damage to the tool, workpiece or machine. 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.

**Scaling**

Component of a -> frame, which causes axis-specific alterations in the scale.

**Serial RS-232C interface**

For the purpose of data input and output,

- The MMC100 MMC module includes a serial RS232C interface
- The MMC101 and MMC102 MMC modules have two RS232C interfaces

available. Machining programs and manufacturer and user data can be imported and stored via these interfaces.

**Services**

Operating area of the control.

**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 quantities can be simulated via operating 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.

<b>Software limit switch</b>	Software limit switches define the limits of the travel range of an axis and prevent the slide contacting the hardware limit switches. Two pairs of values can be assigned per axis and activated separately via the -> PLC.
<b>Spindles</b>	The spindle functionality is a two-level construct: <ol style="list-style-type: none"><li>1. Spindles: Analog/digital speed or position controlled spindle drives (SINUMERIK 840D)</li><li>2. Auxiliary spindles: Speed-controlled spindle drives "Aux. spindle" function package, e.g., for power tools.</li></ol>
<b>Spline interpolation</b>	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.
<b>Standard cycles</b>	Standard cycles are provided for machining operations, which are frequently repeated: <ul style="list-style-type: none"><li>• Cycles for drilling/milling applications</li><li>• Cycles for turning applications (SINUMERIK FM-NC)</li></ul> 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.
<b>Subblock</b>	Block prefixed by "N" containing information for a machining step, such as a position parameter.
<b>Subroutine</b>	A sequence of instructions of a -> parts program, which can be called repeatedly with various defining parameters. The subroutine is called from a main program. Every subroutine can be protected against unauthorized read-out and display. -> Cycles are a type of subroutine.
<b>Synchronization</b>	Instructions in -> parts programs for coordination of the operations in different -> channels at specific machining points.
<b>Synchronized actions</b>	<ol style="list-style-type: none"><li>1. Auxiliary function output While a workpiece is being machined, technological functions (-&gt; auxiliary functions) can be output from the CNC program to the PLC. These auxiliary functions control, for example, supplementary devices on the machine tool, such as the sleeve, gripper, chuck, etc.</li><li>2. High-speed auxiliary function output For time-critical switching functions, the acknowledgement times for the -&gt; auxiliary functions can be minimized and unnecessary stops in the machining process can be avoided.</li></ol>

<b>Synchronized axes</b>	Synchronized axes take the same time to traverse their path as the geometry axes take for their path.
<b>System memory</b>	The system memory is a memory in the CPU, in which the following data are stored: <ul style="list-style-type: none"><li>• Data required by the operating system</li><li>• The timer, counter and marker operands.</li></ul>
<b>System variables</b>	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.
<b>T</b>	
<b>Teach In</b>	<b>Teach In</b> 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.
<b>Text editor</b>	-> Editor
<b>Tool</b>	A software tool for entering and changing the -> parameters of a parameter block. Tools include: <ul style="list-style-type: none"><li>• -&gt; S7 Configuration</li><li>• S7-TOP</li><li>• S7-Info</li></ul>
<b>Tool</b>	A part used on the machine tool for machining. Examples include turning tools, mills, drills, laser beams, etc.
<b>Tool nose radius compensation</b>	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.
<b>Tool offset</b>	By programming a <b>T function</b> (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.

<b>Tool radius compensation</b>	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).
<b>Transformation</b>	Programming in a Cartesian coordinate system, execution in a non-Cartesian coordinate system (e.g., with machine axes as rotary axes).
<b>Travel range</b>	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).
<b>U</b>	
<b>User-defined variable</b>	Users can define variables in the -> parts program or data block (global user data) for their own use. A definition contains a data type specification and the variable name. See also -> System variable.
<b>V</b>	
<b>Variable definition</b>	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.
<b>Velocity control</b>	In order to achieve an acceptable traverse velocity on very short traverse movements, predictive evaluation over several blocks (-> Look Ahead) can be set.
<b>Vocabulary words</b>	Words with a specific notation, which have a defined meaning in the programming language for -> parts programs.
<b>W</b>	
<b>Workpiece</b>	Part to be made/machined by the machine tool.
<b>Workpiece contour</b>	Setpoint contour of the -> workpiece to be made/machined.
<b>Workpiece coordinate system</b>	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.
<b>Workpiece zero</b>	The workpiece zero is the starting point for the -> workpiece coordinate system. It is defined by the distance from the machine zero.

**Z****Zero offset**

Specification of a new reference point for a coordinate system through reference to an existing origin 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 zero 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 the offsets, which define the position of the workpiece zero, it is possible to superimpose an external zero offset

- By means of a handwheel (DRF offset) or
- The PLC.

**3. Programmable**

It is possible to program zero offsets for all path and positioning axes by means of the TRANS statement.



**C References**

<b>/BU/</b>	<b>General documentation</b> SINUMERIK & SIMODRIVE, Automation Systems for Machine Tools Catalog NC 60 Order number: E86060-K4460-A101-A9-7600
<b>/IKPI/</b>	Industrial Communication and Field Devices Catalog IK PI Order number: E86060-K6710-A101-B2-7600
<b>/ST7/</b>	SIMATIC Products for Totally Integrated Automation and Micro Automation Catalog ST 70 Order number: E86060-K4670-A111-A8-7600
<b>/ZI/</b>	MOTION-CONNECT Connections & System Components for SIMATIC, SINUMERIK, MASTERDRIVES, and SIMOTION Catalog NC Z Order number: E86060-K4490-A001-B1-7600
<b>/CD1/</b>	<b>Electronic documentation</b> The SINUMERIK System (03.04 Edition) <b>DOC ON CD</b> (includes all SINUMERIK 840D/840Di/810D/802 and SIMODRIVE publications) Order number: 6FC5298-7CA00-0BG0

### User documentation

<b>/AUK/</b>	SINUMERIK 840D/810D <b>AutoTurn</b> Short Operating Guide Order number: 6FC5298-4AA30-0BP2	(09.99 Edition)
<b>/AUP/</b>	SINUMERIK 840D/810D <b>AutoTurn Graphic Programming System</b> Operator's Guide Programming/Setup Order number: 6FC5298-4AA40-0BP3	(02.02 Edition)
<b>/BA/</b>	SINUMERIK 840D/810D <b>MMC</b> Operator's Guide Order number: 6FC5298-6AA00-0BP0	(10.00 Edition)
<b>/BAD/</b>	SINUMERIK 840D/840Di/810D <b>HMI Advanced</b> Operator's Guide Order number: 6FC5298-6AF00-0BP3	(03.04 Edition)
<b>/BAH/</b>	SINUMERIK 840D/840Di/810D <b>HT 6</b> Operator's Guide Order number: 6FC5298-0AD60-0BP3	(03.04 Edition)
<b>/BAK/</b>	SINUMERIK 840D/840Di/810D <b>Short Operating Guide</b> Order number: 6FC5298-6AA10-0BP0	(02.01 Edition)
<b>/BAM/</b>	SINUMERIK 810D/840D Operating/Programming <b>ManualTurn</b> Order number: 6FC5298-6AD00-0BP0	(08.02 Edition)
<b>/BAS/</b>	SINUMERIK 840D/840Di/810D Operating/Programming <b>ShopMill</b> Order number: 6FC5298-6AD10-0BP3	(10.04 Edition)
<b>/BAT/</b>	SINUMERIK 840D/810D Operating/Programming <b>ShopTurn</b> Order number: 6FC5298-6AD50-0BP2	(06.03 Edition)
<b>/BEM/</b>	SINUMERIK 840D/810D <b>HMI Embedded</b> Operator's Guide Order number: 6FC5298-6AC00-0BP3	(03.04 Edition)
<b>/BNM/</b>	SINUMERIK 840D/840Di/810D <b>Measuring Cycles</b> User's Guide Order number: 6FC5298-7AA70-0BP1	(10.04 Edition)
<b>/BTDI/</b>	SINUMERIK 840D/840Di/810D Motion Control Information System (MCIS) <b>Tool Data Information</b> User's Guide Order number: 6FC5297-6AE01-0BP1	(06.04 Edition)
<b>/CAD/</b>	SINUMERIK 840D/840Di/810D <b>CAD Reader</b> Operator's Guide Order number: (part of the online help)	(03.02 Edition)

<b>/DA/</b>	SINUMERIK 840D/840Di/810D <b>Diagnostics Guide</b> Order number: 6FC5298-6AA20-0BP4	(03.04 Edition)
<b>/KAM/</b>	SINUMERIK 840D/810D <b>ManualTurn</b> Short Guide Order number: 6FC5298-5AD40-0BP0	(04.01 Edition)
<b>/KAS/</b>	SINUMERIK 840D/810D <b>ShopMill</b> Short Guide Order number: 6FC5298-5AD30-0BP0	(04.01 Edition)
<b>/KAT/</b>	SINUMERIK 840D/810D <b>ShopTurn</b> Short Guide Order number: 6FC5298-6AF20-0BP0	(07.01 Edition)
<b>/PG/</b>	SINUMERIK 840D/840Di/810D Programming Guide <b>Fundamentals</b> Order number: 6FC5298-7AB00-0BP0	(03.04 Edition)
<b>/PGA/</b>	SINUMERIK 840D/840Di/810D Programming Guide <b>Advanced</b> Order number: 6FC5298-7AB10-0BP0	(03.04 Edition)
<b>/PGA1/</b>	SINUMERIK 840D/840Di/810D <b>System Variables</b> Lists Manual Order number: 6FC5298-7AE10-0BP0	(03.04 Edition)
<b>/PGK/</b>	SINUMERIK 840D/840Di/810D <b>Programming</b> Short Guide Order number: 6FC5298-7AB30-0BP0	(03.04 Edition)
<b>/PGM/</b>	SINUMERIK 840D/840Di/810D <b>ISO Milling</b> Programming Guide Order number: 6FC5298-6AC20-0BP2	(11.02 Edition)
<b>/PGT/</b>	SINUMERIK 840D/840Di/810D <b>ISO Turning</b> Programming Guide Order number: 6FC5298-6AC10-0BP2	(11.02 Edition)
<b>/PGZ/</b>	SINUMERIK 840D/840Di/810D <b>Cycles</b> Programming Guide Order number: 6FC5298-7AB40-0BP1	(10.04 Edition)
<b>/PI /</b>	PCIN 4.4 Software for Data Transfer to/from <b>MMC Modules</b> Order number: 6FX2060-4AA00-4XB0 (English, German, French) Order from: WK Fürth	
<b>/SYI/</b>	SINUMERIK 840Di <b>System Overview</b> Order number: 6FC5298-6AE40-0BP0	(02.01 Edition)

**Manufacturer/service documentation****a) Lists**

**/LIS/** SINUMERIK 840D/840Di/810D (03.04 Edition)  
SIMODRIVE 611D

**Lists**

Order number: 6FC5297-7AB70-0BP0

**b) Hardware**

**/ASAL/** SIMODRIVE (10.03 Edition)  
**Asynchronous Motors** Planning Guide, General  
Order number: 6SN1197-0AC62-0BP0

**/APH2/** SIMODRIVE (10.03 Edition)  
**Asynchronous Motors 1PH2** Planning Guide  
Order number: 6SN1197-0AC63-0BP0

**/APH4/** SIMODRIVE (10.03 Edition)  
**Asynchronous Motors 1PH4** Planning Guide  
Order number: 6SN1197-0AC64-0BP0

**/APH7S/** SIMODRIVE (04.04 Edition)  
**Asynchronous Motors 1PH7** Planning Guide  
Order number: 6SN1197-0AC65-0BP0

**/APL6/** MASTERDRIVES VC/MC (03.04 Edition)  
**Asynchronous Motors 1PL6** Planning Guide  
Order number: 6SN1197-0AC67-0BP0

**/BH/** SINUMERIK 840D/840Di/810D (04.04 Edition)  
**Operator Components** Manual  
Order number: 6FC5297-7AA50-0BP0

**/BHA/** SIMODRIVE Sensor (03.04 Edition)  
**Absolute Encoder with Profibus DP** User's Guide (HW)  
Order number: 6SN1197-0AB10-0YP3

**/EMV/** SINUMERIK, SIROTEC, SIMODRIVE (03.04 Edition)  
**EMC Installation Guideline** Planning Guide  
Order number: 6FC5297-0AD30-0BP2

The up-to-date declaration of conformity can be viewed on the Internet at

<http://www4.ad.siemens.de>

Please enter the ID number 15257461 in the "Search" field (top right) and click on "go".

**/GHA/** SINUMERIK/SIMOTION (02.03 Edition)  
**ADI4 - Analog Drive Interface for 4 Axes**  
Equipment Manual  
Order number: 6FC5297-0BA01-0BP1

<b>/PFK6/</b>	SIMODRIVE 611, SIMOVERT MASTERDRIVES <b>AC Servo Motors 1FK6</b> Planning Guide Order number: 6SN1197-0AD05-0BP0	(05.03 Edition)
<b>/PFK7/</b>	SIMODRIVE 611, SIMOVERT MASTERDRIVES <b>AC Servo Motors 1FK7</b> Planning Guide Order number: 6SN1197-0AD06-0BP0	(01.03 Edition)
<b>PFS6/</b>	SIMOVERT MASTERDRIVES <b>AC Servo Motors 1FS6</b> Planning Guide Order number: 6SN1197-0AD08-0BP0	(07.03 Edition)
<b>/PFT5/</b>	SIMODRIVE <b>AC Servo Motors 1FT5</b> Planning Guide Order number: 6SN1197-0AD01-0BP0	(02.04 Edition)
<b>/PFT6/</b>	SIMODRIVE 611, SIMOVERT MASTERDRIVES <b>Synchronous Servo Motors 1FT6</b> Planning Guide Order number: 6SN1197-0AD02-0BP0	(01.04 Edition)
<b>/PFU/</b>	SINAMICS, SIMOVERT MASTERDRIVES, MICROMASTER <b>SIEMOSYN Motors 1FU8</b> Order number: 6SN1197-0AC80-0BP0	(09.03 Edition)
<b>/PHC/</b>	SINUMERIK 810D <b>Configuring Manual (HW)</b> Order number: 6FC5297-6AD10-0BP1	(11.02 Edition)
<b>/PHD/</b>	SINUMERIK 840D <b>Configuring Manual (HW)</b> Order number: 6FC5297-6AC10-0BP3	(11.03 Edition)
<b>/PJAL/</b>	SIMODRIVE 611, SIMOVERT MASTERDRIVES MC <b>Synchronous Servo Motors</b> Planning Guide, <b>General for 1FT / 1FK Motors</b> Order number: 6SN1197-0AD07-0BP1	(01.04 Edition)
<b>/PJFE/</b>	SIMODRIVE <b>Built-In Synchronous Motors 1FE1</b> Planning Guide Three-Phase AC Motors for Main Spindle Drives Order number: 6SN1197-0AC00-0BP4	(02.03 Edition)
<b>/PJF1/</b>	SIMODRIVE <b>Built-In Synchronous Motors 1FE1 051.-1FE1 147.</b> Installation Guide Three-Phase AC Motors for Main Spindle Drives Order number: 610.43000.02	(12.02 Edition)
<b>/PJLM/</b>	SIMODRIVE <b>Linear Motors 1FN1, 1FN3</b> Planning Guide ALL General Information on the Linear Motor 1FN1 Three-Phase AC Linear Motor 1FN1 1FN3 Three-Phase AC Linear Motor 1FN3 CON Connections Order number: 6SN1197-0AB70-0BP3	(06.02 Edition)

/PJM/	SIMODRIVE 611, SIMOVERT MASTERDRIVES (11.00 Edition) <b>Servo Motors</b> Planning Guide Three-Phase Motors for Feed and Main Spindle Drives Order number: 6SN1197-0AA20-0BP4
/PJM2/	SIMODRIVE (07.03 Edition) <b>Servo Motors</b> Planning Guide Three-Phase Servo Motors for Feed and Main Spindle Drives Order number: 6SN1197-0AC20-0BP0
/PJTM/	SIMODRIVE (11.03 Edition) <b>Built-In Torque Motors 1FW6</b> Planning Guide Order number: 6SN1197-0AD00-0BP2
/PKTM/	SIMODRIVE, SIMOVERT MASTERDRIVES (03.04 Edition) <b>Complete Torque Motors 1FW3</b> Planning Guide Order number: 6SN1197-0AC70-0BP0
/PJU/	SIMODRIVE 611 (10.04 Edition) <b>Converters</b> Planning Guide Order number: 6SN1197-0AA00-0BP7
/PMH/	SIMODRIVE Sensor (07.02 Edition) <b>Hollow-Shaft Measuring System SIMAG H</b> Planning/Installation Guide Order number: 6SN1197-0AB30-0BP1
/PMH2/	SIMODRIVE Sensor (03.04 Edition) <b>Hollow-Shaft Measuring System SIMAG H2</b> Planning/Installation Guide Order number: 6SN1197-0AB31-0BP1
/PMHS/	SIMODRIVE (12.00 Edition) <b>Measuring System for Main Spindle Drives</b> <b>Gear Wheel Encoder SIZAG2</b> Installation Guide Order number: 6SN1197-0AB00-0YP3
/PMS/	SIMODRIVE (03.04 Edition) Planning Guide <b>ECO Motor Spindle for Main Spindle Drives</b> Order number: 6SN1197-0AD04-0BP1
PPH/	SIMODRIVE (12.01 Edition) <b>1PH2/1PH4/1PH7 Motors</b> Planning Guide Three-Phase Asynchronous Motors for Main Spindle Drives Order number: 6SN1197-0AC60-0BP0
/PPM/	SIMODRIVE (11.01 Edition) Hollow-Shaft Motors for Main Spindle Drives <b>1PM4 and 1PM6</b> Planning Guide Order number: 6SN1197-0AD03-0BP0

## c) Software

## /FB1/

SINUMERIK 840D/840Di/810D/FM-NC (03.04 Edition)

**Basic Machine (Part 1)** Description of Functions

(the various manuals are listed below)

Order number: 6FC5297-7AC20-0BP0

A2	Various Interface Signals
A3	Axis Monitoring, Protection Zones
B1	Continuous Path Mode, Exact Stop and Look Ahead
B2	Acceleration
D1	Diagnostic Tools
D2	Interactive Programming
F1	Travel to Fixed Stop
G2	Velocities, Setpoint/Actual Value Systems, Closed-Loop Control
H2	Output of Auxiliary Functions to PLC
K1	Mode Group, Channels, Program Operation Mode
K2	Axis Types, Coordinate Systems, Frames, Actual-Value System for Workpiece, External Zero Offset
K4	Communication
N2	EMERGENCY STOP
P1	Traverse Axes
P3	Basic PLC Program
R1	Reference Point Approach
S1	Spindles
V1	Feeds
W1	Tool Offset

## /FB2/

SINUMERIK 840D/840Di/810D (03.04 Edition)

**Extended Functions (Part 2)** Description of Functions

including FM-NC: Turning, Stepper Motor

(the various manuals are listed below)

Order number: 6FC5297-7AC30-0BP0

A4	Digital and Analog NCK I/Os
B3	Several Operator Panels and NCUs
B4	Operation via PG/PC
F3	Remote Diagnostics
H1	Jog with/without Handwheel
K3	Compensations
K5	Mode Groups, Channels, Axis Replacement
L1	FM-NC Local Bus
M1	Kinematic Transformation
M5	Measurements
N3	Software Cams, Position Switching Signals
N4	Punching and Nibbling
P2	Positioning Axes
P5	Oscillation
R2	Rotary Axes
S3	Synchronous Spindles
S5	Synchronized Actions (SW 3 and lower, higher /FBSY/)
S6	Stepper Motor Control
S7	Memory Configuration
T1	Indexing Axes
W3	Tool Change
W4	Grinding

- /FB3/** SINUMERIK 840D/840Di/810D (08.04 Edition)  
**Special Functions (Part 3)** Description of Functions  
 (the various manuals are listed below)  
 Order number: 6FC5297-7AC80-0BP1  
 F2 3-Axis to 5-Axis Transformation  
 G1 Gantry Axes  
 G3 Cycle Times  
 K6 Contour Tunnel Monitoring  
 M3 Coupled Axes and ESR  
 S8 Constant Workpiece Speed for Centerless Grinding  
 S9 Setpoint Switching (S9)  
 T3 Tangential Control  
 TE0 Installation and Activation of Compile Cycles  
 TE1 Clearance Control  
 TE2 Analog Axis  
 TE3 Speed/Torque Coupling Master-Slave  
 TE4 Transformation Package Handling  
 TE5 Setpoint Exchange  
 TE6 MCS Coupling  
 TE7 Retrace Support  
 TE8 Unlocked Path-Synchronous Switching Signal Output  
 V2 Preprocessing  
 W5 3D Tool Radius Compensation
- /FBA/** SIMODRIVE 611D/SINUMERIK 840D/810D (03.04 Edition)  
**Drive Functions** Description of Functions  
 (the various sections are listed below)  
 Order number: 6SN1197-0AA80-1BP1  
 DB1 Operational Messages/Alarm Responses  
 DD1 Diagnostic Functions  
 DD2 Speed Control Loop  
 DE1 Extended Drive Functions  
 DF1 Enable Commands  
 DG1 Encoder Parameterization  
 DL1 Linear Motor MD  
 DM1 Calculation of Motor/Power Section Parameters and Controller Data  
 DS1 Current Control Loop  
 DÜ1 Monitors/Limitations
- /FBAN/** SINUMERIK 840D/SIMODRIVE 611 digital (02.00 Edition)  
**ANA Module** Description of Functions  
 Order number: 6SN1197-0AB80-0BP0
- /FBD/** SINUMERIK 840D (07.99 Edition)  
**Digitizing** Description of Functions  
 Order number: 6FC5297-4AC50-0BP0  
 DI1 Startup  
 DI2 Scanning with Tactile Sensors (scancad scan)  
 DI3 Scanning with Lasers (scancad laser)  
 DI4 Milling Program Generation (scancad mill)
- /FBDM/** SINUMERIK 840D/840Di/810D (09.03 Edition)  
 NC Program Management Description of Functions  
 DNC Machines  
 Order number: 6FC5297-1AE81-0BP0



<b>/FBDN/</b>	SINUMERIK 840D/840Di/810D (03.03 Edition) Motion Control Information System (MCIS) <b>NC Program Management DNC</b> Description of Functions Order number: 6FC5297-1AE80-0BP0 DN1 DNC Plant/DNC Cell DN2 DNC IFC SINUMERIK, NC Data Transfer via Network
<b>/FBFA/</b>	SINUMERIK 840D/840Di/810D (11.02 Edition) <b>ISO Dialects for SINUMERIK</b> Description of Functions Order number: 6FC5297-6AE10-0BP3
<b>/FBFE/</b>	SINUMERIK 840D/810D (04.03 Edition) <b>Remote Diagnostics</b> Description of Functions Order number: 6FC5297-0AF00-0BP2 FE1 ReachOut Remote Diagnosis FE3 pcAnywhere Remote Diagnosis
<b>/FBH/</b>	SINUMERIK 840D/840Di/810D (02.04 Edition) <b>HMI Programming Package</b> Order number: (included with the software)  Part 1 User's Guide Part 2 Description of Functions
<b>/FBH1/</b>	SINUMERIK 840D/840Di/810D (03.03 Edition) <b>HMI Programming Package</b> <b>ProTool/Pro Option SINUMERIK</b> Order number: (included with the software)
<b>/FBHL/</b>	SINUMERIK 840D/SIMODRIVE 611 digital (10.03 Edition) <b>HLA Module</b> Description of Functions Order number: 6SN1197-0AB60-0BP3
<b>/FBIC/</b>	SINUMERIK 840D/840Di/810D (06.03 Edition) Motion Control Information System (MCIS) <b>TDI Ident Connection</b> Description of Functions Order number: 6FC5297-1AE60-0BP0
<b>/FBMA/</b>	SINUMERIK 840D/810D (08.02 Edition) <b>ManualTurn</b> Description of Functions Order number: 6FC5297-6AD50-0BP0
<b>/FBO/</b>	SINUMERIK 840D/810D (09.01 Edition) Configuring of <b>OP 030 Operator Interface</b> Description of Functions (the various sections are listed below) Order number: 6FC5297-6AC40-0BP0 BA Operator's Guide EU Development Environment (Configuring Package) PSE Introduction to Configuring the Operator Interface Screen Kit: Software Update and Configuration
<b>/FBP/</b>	SINUMERIK 840D (03.96 Edition) <b>C-PLC Programming</b> Description of Functions Order number: 6FC5297-3AB60-0BP0

<b>/FBR/</b>	SINUMERIK 840D/840Di/810D (01.04 Edition) <b>Computer Link (SinCOM)</b> Description of Functions Order number: 6FC5297-6AD60-0BP1 HCI Interface for Host Computer NPL PLC/NCK Interface
<b>/FBSI/</b>	SINUMERIK 840D/SIMODRIVE (11.03 Edition) SINUMERIK <b>Safety Integrated</b> Description of Functions Order number: 6FC5297-6AB80-0BP2
<b>/FBSP/</b>	SINUMERIK 840D/840Di/810D (11.03 Edition) <b>ShopMill</b> Description of Functions Order number: 6FC5297-6AD80-0BP2
<b>/FBST/</b>	SIMATIC (01.01 Edition) <b>FM STEPDRIVE/SIMOSTEP</b> Description of Functions Order number: 6SN1197-0AA70-0YP4
<b>/FBSY/</b>	SINUMERIK 840D/810D (03.04 Edition) <b>Synchronized Actions</b> Description of Functions Order number: 6FC5297-7AD40-0BP2
<b>/FBT/</b>	SINUMERIK 840D/810D (03.04 Edition) <b>ShopTurn</b> Description of Functions Order number: 6FC5297-6AD70-0BP2
<b>/FBTC/</b>	SINUMERIK 840D/810D (01.02 Edition) IT Solutions <b>Tool Data Communication SinTDC</b> Description of Functions Order number: 6FC5297-5AF30-0BP0
<b>/FBTD/</b>	SINUMERIK 840D/810D (02.01 Edition) IT Solutions <b>Tool Information Systems (SinTDI)</b> Description of Functions with Online Help Order number: 6FC5297-6AE00-0BP0
<b>/FBTP/</b>	SINUMERIK 840D/840Di/810D (01.03 Edition) Motion Control Information System (MCIS) <b>Preventive Maintenance TPM</b> Description of Functions Order number: Document is included with the software
<b>/FBU/</b>	SIMODRIVE 611 universal/universal E (10.04 Edition) <b>Closed-Loop Control Component for Speed Control and Positioning</b> Description of Functions Order number: 6SN1197-0AB20-1BP1
<b>/FBU2/</b>	SIMODRIVE 611 <b>universal</b> (04.02 Edition) Installation Guide (included with every SIMODRIVE 611 universal)
<b>/FBW/</b>	SINUMERIK 840D/810D (03.04 Edition) <b>Tool Management</b> Description of Functions Order number: 6FC5297-6AC60-0BP1
<b>/HBA/</b>	SINUMERIK 840D/840Di/810D (03.02 Edition) Manual <b>@Event</b> Order number: 6AU1900-0CL20-0BA0

<b>/HBI/</b>	SINUMERIK 840Di <b>SINUMERIK 840Di</b> Manual Order number: 6FC5297-7AE60-0BP0	(03.04 Edition)
<b>/INC/</b>	SINUMERIK 840D/840Di/810D <b>Commissioning Tool SINUMERIK SinuCOM NC</b> System Overview Order number: (part of the Online Help for the Startup Tool)	(06.03 Edition)
<b>/PJE/</b>	SINUMERIK 840D/810D <b>HMI Embedded Configuring Package</b> Description of Functions Software Update, Configuration, Installation Order number: 6FC5297-6EA10-0BP0 (the PS Configuring Syntax document is supplied with the software and is available as a pdf file)	(08.01 Edition)
<b>/POS1/</b>	SIMODRIVE <b>POSMO A</b> User's Guide <b>Decentralized Positioning Motor on PROFIBUS DP</b> Order number: 6SN2197-0AA00-0BP7	(08.04 Edition)
<b>/POS2/</b>	SIMODRIVE <b>POSMO A</b> Installation Guide (included with every POSMO A)	(05.03 Edition)
<b>/POS3/</b>	SIMODRIVE POSMO SI/CD/CA <b>Distributed Servo Drive Systems</b> User's Guide Order number: 6SN2197-0AA20-0BP7	(10.04 Edition)
<b>/POS4/</b>	SIMODRIVE <b>POSMO SI</b> Installation Guide (included with every POSMO SI)	(08.03 Edition)
<b>/POS5/</b>	SIMODRIVE <b>POSMO CD/CA</b> Installation Guide (included with every POSMO CD/CA)	(08.03 Edition)
<b>/S7H/</b>	SIMATIC S7-300 <b>Technological Functions</b> Installation Manual Order number: 6ES7398-8AA03-8AA0 - Reference Manual: CPU Data (HW Description) - Reference Manual: Module Data	(2002 Edition)
<b>/S7HT/</b>	SIMATIC S7-300 <b>STEP 7 Manual, Fundamentals, V. 3.1</b> Order number: 6ES7810-4CA02-8BA0	(03.97 Edition)
<b>/S7HR/</b>	SIMATIC S7-300 <b>STEP 7 Manual, Reference Manuals, V. 3.1</b> Order number: 6ES7810-4CA02-8BR0	(03.97 Edition)
<b>/S7S/</b>	SIMATIC S7-300 <b>FM 353 for Stepper Drive</b> Positioning Module Order together with configuring package	(04.02 Edition)
<b>/S7L/</b>	SIMATIC S7-300 <b>FM 354 for Servo Drive</b> Positioning Module Order together with configuring package	(04.02 Edition)
<b>/S7M/</b>	SIMATIC S7-300 <b>FM 357.2 Multimodule</b> for Servo and Stepper Drives Order together with configuring package	(01.03 Edition)

<b>/SP/</b>	SIMODRIVE 611-A/611-D <b>SimoPro 3.1</b> Program for Configuring Machine Tool Drives Order number: 6SC6111-6PC00-0BA□ Order from: WK Fürth
<b>d) Startup</b>	
<b>/BS/</b>	SIMODRIVE 611 analog (10.00 Edition) <b>Startup Software for Main Spindle and Asynchronous Motor Modules Version 3.20</b> Description Order number: 6SN1197-0AA30-0BP1
<b>/IAA/</b>	SIMODRIVE 611 analog (10.00 Edition) <b>Installation and Start-U Guide</b> Order number: 6SN1197-0AA60-0BP6
<b>/IAC/</b>	SINUMERIK 810D (11.02 Edition) <b>Installation and Start-Up Guide</b> (including description of SIMODRIVE 611D start-up software) Order number: 6FC5297-6AD20-0BP1
<b>/IAD/</b>	SINUMERIK 840D/SIMODRIVE 611D (03.04 Edition) <b>Installation and Startup Guide</b> (including description of SIMODRIVE 611D start-up software) Order number: 6FC5297-7AB10-0BP0
<b>/IAM/</b>	SINUMERIK 840D/840Di/810D (03.04 Edition) <b>HMI Installation and Start-Up Guide</b> Order number: 6FC5297-6AE20-0BP3 AE1 Updates/Supplements BE1 Expanding the Operator Interface HE1 Online Help IM2 Starting Up HMI Embedded IM4 Starting Up HMI Advanced TX1 Creating Foreign Language Texts

**D Identifiers**

List of input/output variables for cycles

<i>Name</i>	<i>Meaning in English</i>
<b>AD</b>	Allowance depth
<b>AFSL</b>	Angle for slot length
<b>ANG1, ANG2</b>	Flank angle
<b>ANGB</b>	Liftoff angle for roughing
<b>AP1</b>	Unfinished dimension in plane
<b>AP2</b>	Unfinished dimension in plane
<b>APP</b>	Approach path
<b>APX</b>	Axial value for defining blank for facing axis
<b>APXA</b>	Absolute or incremental evaluation of parameter APX
<b>APZ</b>	Axial value for defining blank for longitudinal axis
<b>APZA</b>	Absolute or incremental evaluation of parameter APZ
<b>AS1, AS2</b>	Direction of approach/approach travel
<b>AXN</b>	Tool axis
<b>BNAME</b>	Name for program of drill positions
<b>CDIR</b>	Circle direction
<b>CPA</b>	Center point, abscissa
<b>CPO</b>	Center point, ordinate
<b>CRAD</b>	Corner radius
<b>DAM</b>	Degression value, Path for roughing interrupt
<b>DBH</b>	Distance between holes
<b>DIAG</b>	Groove depth
<b>DIATH</b>	Diameter of thread
<b>DIS1</b>	Distance
<b>DIS1</b>	Distance between columns
<b>DIS2</b>	Number of lines, Distance between rows
<b>DM1 ...DM4</b>	Diameter
<b>DP</b>	Depth
<b>DP1</b>	First depth
<b>DPR</b>	Depth, relative
<b>DT</b>	Dwell time
<b>DTB</b>	Dwell time at bottom
<b>DTD</b>	Dwell time at depth
<b>DTS</b>	Dwell time at starting point
<b>ENC</b>	Tapping with/without encoder
<b>FAL</b>	Finish allowance
<b>FAL1</b>	Finish allowance on groove base
<b>FAL2</b>	Finish allowance on flanks
<b>FALD</b>	Finish allowance depth
<b>FALX</b>	Finish allowance, x axis
<b>FALZ</b>	Finish allowance, z axis

<b>FDEP</b>	First depth
<b>FDIS</b>	First distance
<b>FDP1</b>	Overrun path in direction to plane
<b>FDPR</b>	First depth, relative
<b>FF1</b>	Feedrate for roughing
<b>FF2</b>	Feedrate for insertion
<b>FF3</b>	Feedrate for finishing
<b>FF4</b>	Feedrate for contour transition elements
<b>FFCP</b> (SW 6.3 and higher)	Feedrate for circular positioning
<b>FFD</b>	Feedrate for depth
<b>FFP1</b>	Feedrate surface
<b>FFP2</b>	Feedrate for finishing
<b>FFR</b>	Feedrate
<b>FORM</b>	Definition of form
<b>FPL</b>	Final point along longitudinal axis
<b>FRF</b>	Feedrate factor
<b>IANG</b>	Infeed angle
<b>IDEP</b>	Infeed depth
<b>INDA</b>	Incremental angle
<b>KDIAM</b>	Internal diameter of thread
<b>KNAME</b>	Name of the contour subroutine
<b>LENG</b>	Elongated hole length, pocket length
<b>LP1</b>	Length of approach travel, radius
<b>LP2</b>	Length of return travel, radius
<b>LSANF</b>	Block number/label identifying start of contour definition
<b>LSEND</b>	Block number/label identifying end of contour definition
<b>MDEP</b>	Minimum depth
<b>MID</b>	Maximum infeed depth
<b>MIDA</b>	Maximum infeed width
<b>MIDF</b>	Maximum infeed depth for finishing
<b>MPIT</b>	Thread lead as thread size
<b>NID</b>	Number of noncuts
<b>NP1 ... NP8</b>	Name/Label ...
<b>NPP</b>	Name of parts program
<b>NRC</b>	Number of roughing cuts
<b>NSP</b>	Start point offset first thread
<b>NUM</b>	Number of holes
<b>NUM1</b>	Number of columns
<b>NUM2</b>	Number of lines
<b>NUMT</b>	Number of threads
<b>PA</b>	Reference point, abscissa
<b>PIT</b>	Thread lead
<b>PNAME</b>	Name for pocket milling machining program

<b>PO</b>	Reference point, ordinate
<b>PO1 ... PO4</b>	Point in longitudinal axis
<b>POSS</b>	Position for oriented spindle stop
<b>PP1 ... PP3</b>	Thread pitch 1...3 as value
<b>PRAD</b>	Pocket radius
<b>RAD</b>	Radius
<b>RAD1</b>	Radius
<b>RCI1, RCI2</b>	Radius/chamfer inside
<b>RCO1, RCO2</b>	Radius/chamfer outside
<b>RFF</b>	Retract feed
<b>RFP</b>	Reference plane
<b>RL</b>	Bypass contour
<b>ROP</b>	Run out path
<b>RPA</b>	Retract path, abscissa
<b>RPAP</b>	Retract path, applicate
<b>RPO</b>	Retract path, ordinate
<b>RTP</b>	Retract plane
<b>SDAC</b>	Spindle direction after cycle
<b>SDIR</b>	Spindle direction
<b>SDIS</b>	Safety distance
<b>SDR</b>	Spindle direction for retraction
<b>SPCA</b>	Reference point, abscissa
<b>SPCO</b>	Reference point, ordinate
<b>SPD</b>	Starting point in the facing axis
<b>SPL</b>	Starting point along longitudinal axis
<b>SSF</b>	Speed for finishing
<b>SST</b>	Speed for tapping
<b>SST1</b>	Speed for retraction
<b>STA, STA1</b>	Angle
<b>STA2</b>	Insertion angle
<b>TDEP</b>	Thread depth
<b>TN</b>	Name of stock removal tool
<b>TOL1</b>	Blank tolerance
<b>TYPTH</b>	Type of thread
<b>VARI</b>	Working
<b>VRT</b>	Variable return path
<b>WID</b>	(Pocket) width
<b>WIDG</b>	Groove width

## Notes



**E Index****A**

Absolute drilling depth 2-52, 3-119, 3-124, 3-138,  
3-160, 3-198

Axis assignment 1-21

**B**

Behavior when quantity parameter is zero 2-94

Blank 4-334

Blank updating 4-343

Blueprint programming 1-30

Boring 2-49

Boring 1 2-78

Boring 2 2-81

Boring 3 2-85

Boring 4 2-87

Boring 5 2-89

Boring cycle 2-49

**C**

Call 1-21, 2-50

Call conditions 1-21

Centering 2-51

Circumferential slot - SLOT2 3-130

Contour 1 - line 1-30

Contour 2 - line 1-30

Contour 3 - line 1-30

Contour definition 4-306, 4-341

Contour monitoring 4-284, 4-309

Contour programming 4-336

CONTPRON 4-307

Cycle alarms 5-355

Cycle auxiliary subroutines 1-20

Cycle call 1-23

Cycle setting data, milling 3-109

Cycle setting data, Turning 4-284

Cycle support for user cycles 1-35

Cycle support in program editor 1-27

CYCLE60 (SW 6.4 and higher) 3-270

CYCLE71 3-158

CYCLE72 3-164

CYCLE73 3-183, 3-189

CYCLE74 3-183, 3-184

CYCLE75 3-183, 3-186

CYCLE76 3-174

CYCLE77 3-179

CYCLE800 3-211

CYCLE801 2-102

CYCLE81 2-51

CYCLE82 2-54

CYCLE8250 2-54

CYCLE83 2-56

CYCLE832 (SW 6.3 and higher) 3-257

CYCLE84 2-63

CYCLE840 2-70

CYCLE8450 2-63

CYCLE85 2-78

CYCLE8550 2-78

CYCLE86 2-81

CYCLE87 2-85

CYCLE88 2-87

CYCLE8850 2-87

CYCLE89 2-89

CYCLE90 3-110

CYCLE93 4-286

CYCLE94 4-295

CYCLE95 4-299

CYCLE950 4-331

CYCLE96 4-312

CYCLE97 4-316

CYCLE98 4-323

**D**

Deep-hole drilling 2-56

Deep-hole drilling with chip breaking 2-58

Deep-hole drilling with swarf removal 2-57

Definition files for cycles 1-41

Dot matrix 2-102

Drill pattern cycles 1-19, 2-94

Drill pattern cycles without drilling cycle call 2-94

Drilling 2-51

Drilling cycles 1-19, 2-48

Drilling, counterboring 2-54

**E**

Elongated holes on a circle - LONGHOLE 3-117

Engraving cycle – CYCLE60 (SW 6.4 and higher)  
3-270

Extended stock removal cycle - CYCLE950 4-331

**F**

Face milling 3-158

Face thread 4-321

FGROUP 3-110

Form of delivery for cycles in HMI Advanced 1-43

Free contour programming 1-30

- G**  
Geometrical parameters 2-49  
Grooving cycle - CYCLE93 4-286
- H**  
High-speed settings – CYCLE832 (with Cycles SW 6.3 and higher) 3-257  
Hole circle 2-99  
HOLES1 2-95  
HOLES2 2-99
- I**  
Inside threads 3-112
- L**  
LONGHOLE 3-117  
Longitudinal thread 4-321
- M**  
Machine data 1-40  
Machining parameters 2-49  
Machining plane 1-21  
Machine data 1-44  
MCALL 2-91  
Messages 1-22, 5-361  
Milling circular pockets - POCKET2 3-140  
Milling circular pockets - POCKET4 3-153  
Milling circular spigots - CYCLE77 3-179  
Milling Cycles 1-19  
Milling rectangular pockets - POCKET1 3-136  
Milling rectangular pockets - POCKET3 3-144  
Milling rectangular spigots - CYCLE76 3-174  
Modal call 2-91
- O**  
Outside threads 3-111  
Overview of cycle alarms 5-355  
Overview of cycles 1-18
- P**  
Parallel-contour 4-333  
Parameter list 1-23  
Path milling 3-164  
Plane definition 1-21  
Plausibility checks 2-94  
Pocket milling with islands 3-183  
Pocket milling with islands - CYCLE73 3-189  
POCKET1 3-136  
POCKET150 3-136  
POCKET2 3-140  
POCKET3 3-144  
POCKET4 3-153  
POCKET450 3-153
- R**  
Reference plane 2-52, 3-198  
Relative drilling depth 2-52, 3-119, 3-124, 3-138, 3-160, 3-198  
Residual material 4-334  
Retraction plane 2-52, 3-198  
Return conditions 1-21  
Rigid tapping 2-63  
Row of holes 2-95
- S**  
Safety distance 2-52, 3-198  
SETMS 3-109  
Simulation of cycles 1-26  
Simulation without tool 1-26  
SLOT1 3-122  
SLOT150 3-122  
SLOT2 3-130  
Slots on a circle - SLOT1 3-122  
Spindle handling 4-283  
SPOS 2-65, 2-66  
Starting point 4-309  
Stock removal cycle - CYCLE95 4-299  
Swiveling – CYCLE800 3-211  
Swiveling in JOG mode (Cycles SW 6.5 and higher) 3-223
- T**  
Tapping with compensating chuck 2-70  
Tapping with compensating chuck with encoder 2-72  
Tapping with compensating chuck without encoder 2-71  
Thread chaining - CYCLE98 4-323  
Thread cutting - CYCLE97 4-316  
Thread milling 3-110  
Thread recutting 4-329  
Thread undercut - CYCLE96 4-312  
Tool clearance angle 4-284  
Transfer island contour - CYCLE75 3-186  
Transfer pocket edge contour - CYCLE74 3-184  
Turning cycles 1-20
- U**  
Undercut cycle - CYCLE94 4-295  
Upgrading cycles in HMI Advanced 1-43

**F Commands, identifiers****C**

CYCLE60 (SW 6.4 and higher) 3-270  
CYCLE71 3-158  
CYCLE72 3-164  
CYCLE73 3-189  
CYCLE74 3-184  
CYCLE75 3-186  
CYCLE76 3-174  
CYCLE77 3-179  
CYCLE800 3-211  
CYCLE801 2-102  
CYCLE81 2-51  
CYCLE82 2-54  
CYCLE83 2-56  
CYCLE832 (SW 6.3 and higher) 3-257  
CYCLE84 2-63  
CYCLE840 2-70  
CYCLE85 2-78  
CYCLE86 2-81  
CYCLE87 2-85  
CYCLE88 2-87  
CYCLE89 2-89  
CYCLE90 3-110  
CYCLE93 4-286  
CYCLE94 4-295  
CYCLE95 4-299  
CYCLE950 4-331  
CYCLE96 4-312  
CYCLE97 4-316  
CYCLE98 4-323

**H**

HOLES1 2-95  
HOLES2 2-99

**L**

LONGHOLE 3-117

**P**

POCKET1 3-136  
POCKET2 3-140  
POCKET3 3-144  
POCKET4 3-153

**S**

SLOT1 3-122  
SLOT2 3-130



## Notes

To  
 SIEMENS AG  
 A&D MC BMS  
 P.O. Box 3180  
 D-91050 Erlangen, Germany  
 Phone +49 (0)180 / 5050 - 222 [Hotline]  
 Fax +49 (0) 9131 / 98 - 2176 [Documentation]  
 E-mail: motioncontrol.docu@erlf.siemens.de

Suggestions
Corrections
For Publication/Manual: SINUMERIK 840D/840Di/810D Cycles User Documentation
Programming Guide
Order no.: 6FC5298-7AB40-0BP1 10.04 Edition
Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.

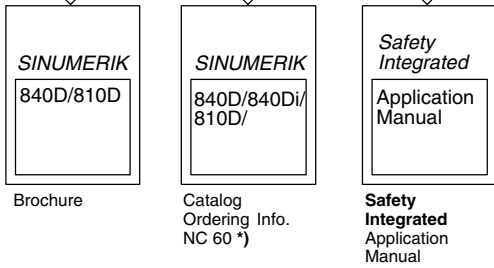
<b>From</b>
Name
Company/Dept.
Address:
Zip code:                      Town:
Phone:                              /
Fax:                                      /

**Suggestions and/or corrections**

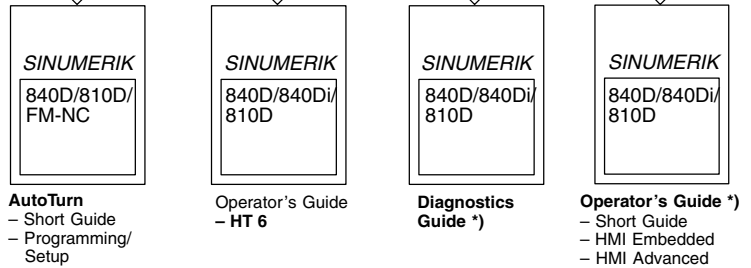


# Overview of SINUMERIK 840D/840Di/810D Documentation (10.2004)

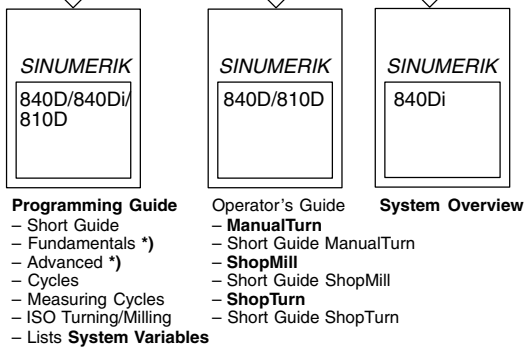
## General Documentation



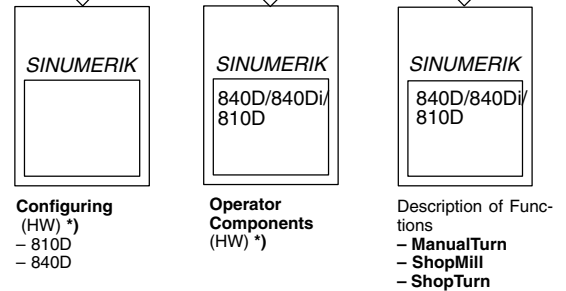
## User Documentation



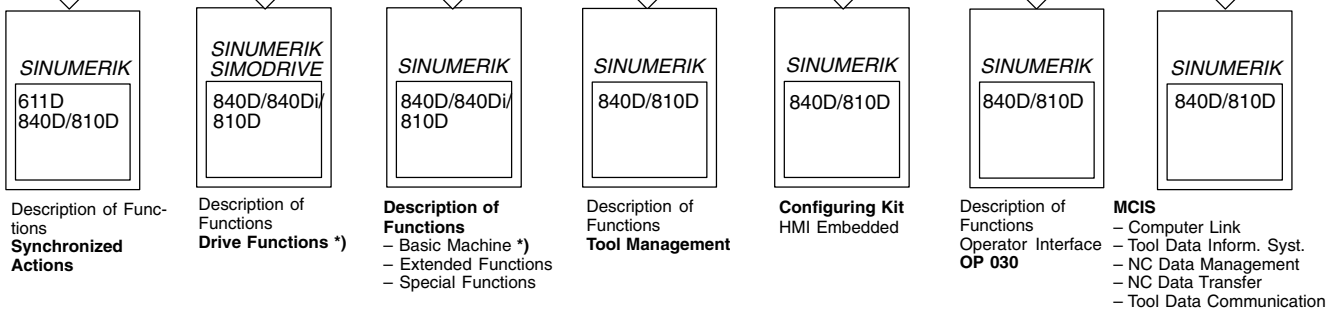
## User Documentation



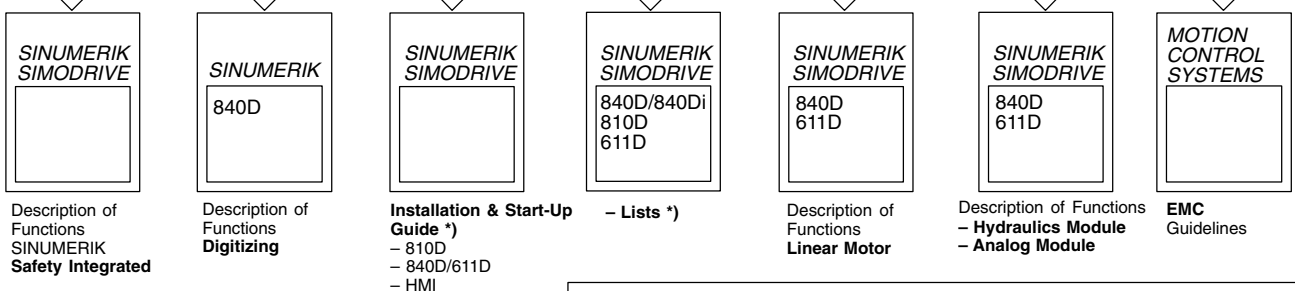
## Manufacturer/Service Documentation



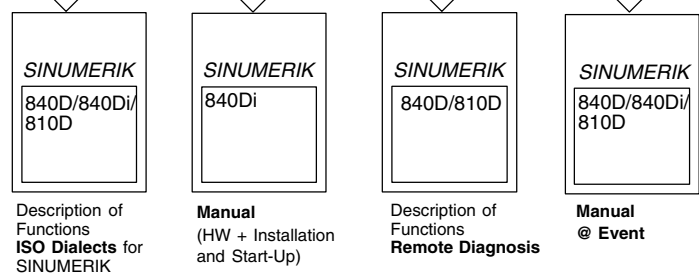
## Manufacturer/Service Documentation



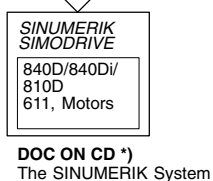
## Manufacturer/Service Documentation



## Manufacturer/Service Documentation



## Electronic Documentation



\*) These documents are a minimum requirement

**Siemens AG**

Automation & Drives

Motion Control Systems

P. O. Box 3180, D – 91050 Erlangen  
Germany

[www.siemens.com/motioncontrol](http://www.siemens.com/motioncontrol)

© Siemens AG, 2004  
Subject to change without prior notice  
Order No. 6FC5298-7AB40-0BP1

Printed in Germany