Programming Guide 11/2002 Edition

sinumerik

Cycles

SINUMERIK 840D/840Di/810D

SIEMENS

SIEMENS

SINUMERIK 840D/840Di/810D

Cycles

Programming Guide

Valid for

Control	*Software Version*
SINUMERIK 840D	6
SINUMERIK 840DE (export version)	6
SINUMERIK 840D powerline	6
SINUMERIK 840DE powerline	6
SINUMERIK 840Di	2
SINUMERIK 840DiE (export version)	2
SINUMERIK 810D	3
SINUMERIK 810DE (export version)	3
SINUMERIK 810D powerline	6
SINUMERIK 810DE powerline	6

11.2002 Edition

SINUMERIK® Documentation

Printing history

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

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

Status code in the "Remarks" column:

A .... New documentation.
B .... Unrevised edition with new order no.
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.

	Order No.	Remarks
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

This manual is included in the documentation available on CD ROM (**DOCONCD**)

Edition	Order No.	Remarks
11.02	6FC5298-6CA00-0BG3	C

Trademarks

SIMATIC®, SIMATIC HMI®, SIMATIC NET®, SIROTEC®, SINUMERIK®, SIMODRIVE® and SIMODRIVE POSMO® are registered trademarks of Siemens AG. Other names in this publication might be trademarks whose use by a third party for his own purposes may violate the rights of the registered holder.

Further information is available on the Internet under:
http:/www.ad.siemens.de/sinumerik

This document was produced with WinWord V8.0
and Designer V7.0.

Order No. 6FC5 298-6AB40-0BP2
Printed in Germany

Siemens Aktiengesellschaft

Contents

Structure of the manual

The SINUMERIK documentation is organized in three parts:

- General Documentation
- User Documentation
- Manufacturer/Service Documentation.

Target group

This documentation is intended for users of machine tools. This publication provides detailed information that the user requires for operating the SINUMERIK 810D and 840D controls.

Standard scope

This programming guide describes the standard functions. Differences and additions implemented by the machine-tool manufacturer are documented by the machine 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...) can be obtained from your local Siemens branch office.

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

Applicability

This Programming Guide is valid for the following controls:

SINUMERIK 840D	SW 6
SINUMERIK 840DE (export version)	SW 6
SINUMERIK 840Di	SW 2
SINUMERIK 840DiE (export version)	SW 2
SINUMERIK 810D	SW 6
SINUMERIK 810DE (export version)	SW 6

with operator panels OP 010, OP 010C, OP 010S, OP 12 or OP 15 (PCU 20 or PCU 50)

SINUMERIK 840D powerline

From 09.2001 onwards, higher-performance versions of

- SINUMERIK 840D powerline and
- SINUMERIK 840DE powerline

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

SINUMERIK 810D powerline

From 12.2001 onwards, higher-performance versions of

- SINUMERIK 810D powerline and
- SINUMERIK 810DE powerline

will be available. For a list of the available **powerline** modules, please refer to Section 1.1 of the Hardware Description /PHC/.

Structure of descriptions

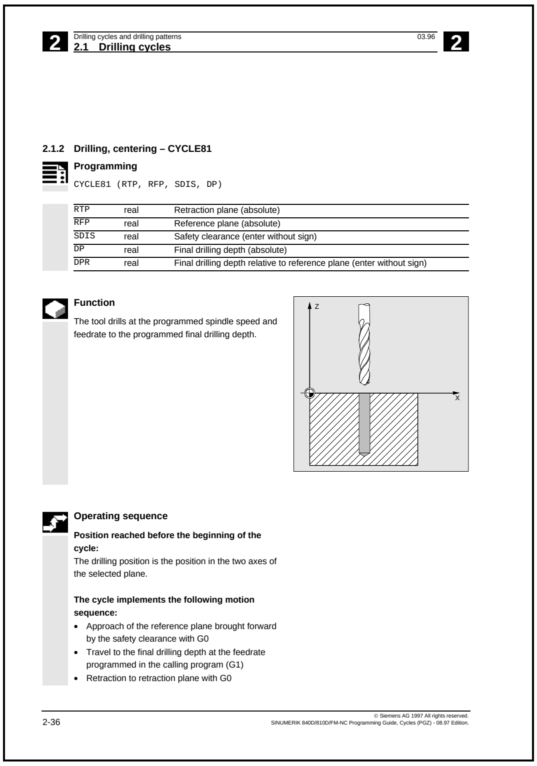
All cycles and program functions were laid out according to the same structure, as far as possible and practicable. The various levels of information have been structured so that you can find the information you are looking for quickly.

1. The function at a glance

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

This information always appears at the beginning of the page.

Note:
In order to keep the documentation succinct we have not provided all the methods or representation of the individual cycles and parameters that are possible in the programming language. Cycles have been programmed in the form in which they most frequently arise on the shop floor.


2 Drilling cycles and drilling patterns 03.96 2
2.1 Drilling cycles

2.1.2 Drilling, centering – CYCLE81

Programming

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

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

Function

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

Operating 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 clearance with G0
- Travel to the final drilling depth at the feedrate programmed in the calling program (G1)
- Retraction to retraction plane with G0

© Siemens AG 1997 All rights reserved.
2-36 SINUMERIK 840D/810D/FM-NC Programming Guide, Cycles (PGZ) - 08.97 Edition.

2. Detailed explanations

In the theoretical sections, you are provided with a detailed description of the following:

What is the cycle used for?

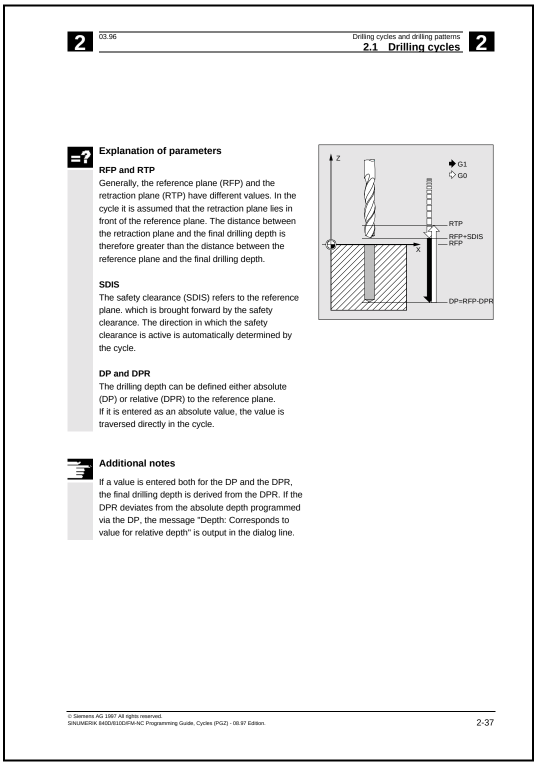
What does the cycle do?

What is the sequence of operations?

What do the parameters do?

What else do you have to look out for?

The theoretical sections provide learning material for the NC beginner. You should work through the manual at least once to get an idea of the scope of the functions and capability of your SINUMERIK control.



2 03.96 Drilling cycles and drilling patterns 2
2.1 Drilling cycles

Explanation of parameters

RFP and RTP
Generally, the reference plane (RFP) and the retraction plane (RTP) have different values. In the cycle it is assumed that the retraction plane lies 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
The safety clearance (SDIS) refers to the reference plane. which is brought forward by the safety clearance. The direction in which the safety clearance is active is automatically determined by the cycle.

DP and DPR
The drilling depth can be defined either absolute (DP) or relative (DPR) to the reference plane.
If it is entered as an absolute value, the value is traversed directly in the cycle.

Additional notes

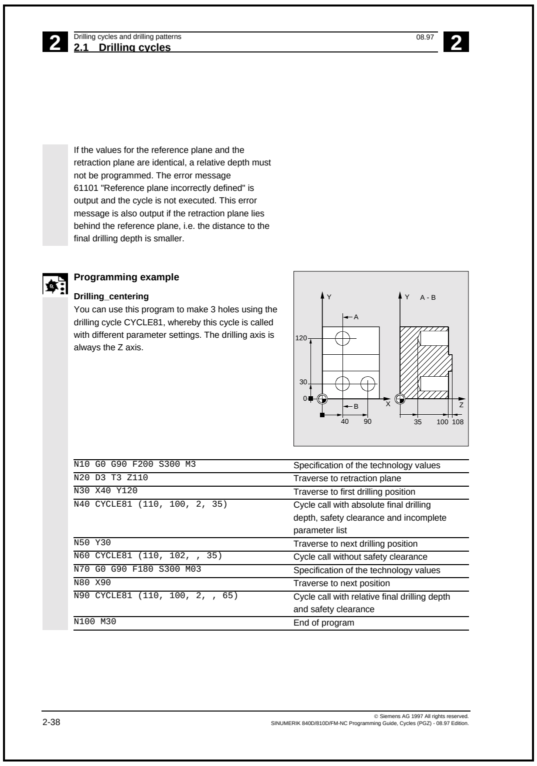
If a value is entered both for the DP and the DPR, the final drilling depth is derived from the DPR. If the DPR deviates from the absolute depth programmed via the DP, the message "Depth: Corresponds to value for relative depth" is output in the dialog line.

© Siemens AG 1997 All rights reserved.
SINUMERIK 840D/810D/FM-NC Programming Guide, Cycles (PGZ) - 08.97 Edition. 2-37

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 and drilling patterns 08.97 2
2.1 Drilling cycles

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

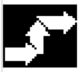
Programming example

Drilling_centering
You can use this program to make 3 holes using the drilling cycle CYCLE81, whereby this cycle is called with different parameter settings. The drilling axis is always the Z axis.

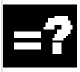
`N10 G0 G90 F200 S300 M3`	Specification of the technology values
`N20 D3 T3 Z110`	Traverse to retraction plane
`N30 X40 Y120`	Traverse to first drilling position
`N40 CYCLE81 (110, 100, 2, 35)`	Cycle call with absolute final drilling depth, safety clearance and incomplete parameter list
`N50 Y30`	Traverse to next drilling position
`N60 CYCLE81 (110, 102, , 35)`	Cycle call without safety clearance
`N70 G0 G90 F180 S300 M03`	Specification of the technology values
`N80 X90`	Traverse to next position
`N90 CYCLE81 (110, 100, 2, , 65)`	Cycle call with relative final drilling depth and safety clearance
`N100 M30`	End of program

2-38 © Siemens AG 1997 All rights reserved.
SINUMERIK 840D/810D/FM-NC Programming Guide, Cycles (PGZ) - 08.97 Edition.

Explanation of symbols



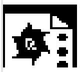
Sequence of operations



Explanation

Function

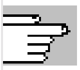
Parameters



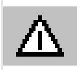
Sample program

Programming

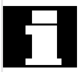
Additional notes



Cross-reference to other documentation or sections



Danger notes and sources of error



Additional notes or background information

Warning notes
The following warning notes with graded degrees of importance are used in this documentation.

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.

Additional equipment
The applications of SIEMENS controls can be expanded by adding special additional devices, equipment and expansion units supplied by SIEMENS.

Personnel
Only **authorized and reliable personnel who have been trained in the use of the equipment** may be allowed to handle the control. Nobody without the necessary training must be allowed to operate the control, even temporarily.

The corresponding **responsibilities** of personnel who set up, operate and maintain the equipment must be clearly **defined** and adherence to these responsibilities **monitored**.

Behavior
Before the control is started up, the personnel who are to work on the control must be thoroughly acquainted with the Operator's Guides. The operating company is also responsible for **constantly monitoring** the overall technical state of the control (noticeable faults and damage, altered service performance).

Servicing
Repairs must be carried out by personnel who are **specially trained and qualified** in the relevant technical subject according to the information supplied in the service and maintenance guide. All relevant safety regulations must be followed.

Note

The following is deemed to be **improper usage** and **exempts the manufacturer from any liability**:

Any application which does not comply with the rules for proper usage described above.

If the control is **not in technically perfect condition** or is operated without due regard for safety regulations and accident prevention instructions given in the Instruction Manual.

If faults that might affect the safety of the equipment are not rectified **before** the control is started up.

Any **modification, bypassing** or **disabling** of items of equipment on the control that are required to ensure fault-free operation, unlimited use and active and passive safety.

Improper usage may result in **unforeseen dangers** to:

- life and limb of personnel
- the control, machine and other assets of the owner and the user.

Notes

General

1.1 General information

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 various standard cycles for the technologies

- Drilling
- Milling
- Turning.

1.2.1 Drilling cycles, drill pattern cycles, milling cycles and turning cycles

You can perform the following cycles with the SINUMERIK 810D and 840D 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

New in SW 5.3 and higher:

CYCLE801	Grid of holes

Milling cycles

LONGHOLE	Milling pattern of elongated holes on a circle
SLOT1	Groove milling pattern arranged 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

New in SW 4 and higher:

POCKET3	Rectangular pocket milling (with any milling tool)
POCKET4	Circular pocket milling (with any milling tool)
CYCLE71	Face milling
CYCLE72	Contour milling

New in SW 5.2 and higher:	
CYCLE73	Pocket milling with islands
CYCLE74	Transfer of pocket edge contour
CYCLE75	Transfer of isolated contour

New in SW 5.3 and higher:	
CYCLE76	Mill a rectangular spigot
CYCLE77	Mill a circular spigot

Turning cycles

CYCLE93	Groove
CYCLE94	Undercut (shape E and F according to DIN)
CYCLE95	Stock removal with relief cut
CYCLE96	Thread undercut (shapes A, B, C and D according to DIN)
CYCLE97	Thread cutting
CYCLE98	Chaining of threads

New in SW 5.1 and higher:	
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 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.

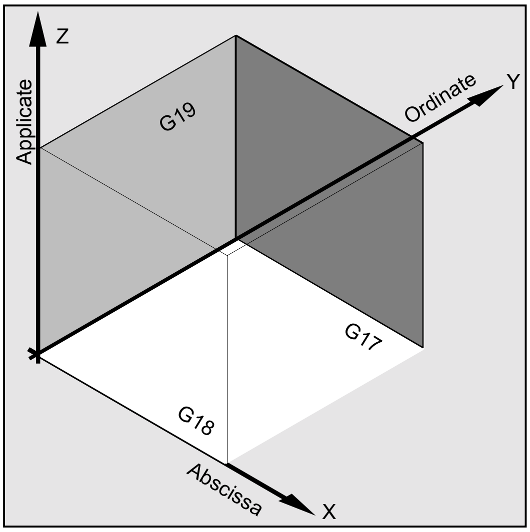
The cycles are supplied on diskette or, for the MMC 102, with the corresponding software release. They are loaded into the parts program memory of the control via the RS-232 interface (see Operator's Guide).

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 every 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 specified when the cycle is called. In the parameter list, you can transfer

- 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 ")". If you wish to leave out parameters in between, a comma, "..., ,..." is used as placeholder.

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 parameter list contains more entries than defined as parameters in the cycle when the cycle is called, the general NC alarm 12340 "Too many parameters" is generated. The cycle is not executed in this case.

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 (to be entered without a sign)
`FFR`	Feedrate
`VARI`	Machining type Values: 0, 1 or 2
`FAL`	Final machining allowance

The cycle is called with 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 before you call the cycle.

Example

`DEF CHAR FORM="E"`	Definition of a parameter, value assignment
`DEF REAL MID=5, FFR, FAL` `DEF INT VARI=1`	Definition of parameters with or without 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 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, alarm 12330 "Parameter type ... incorrect" is issued.

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 the correct assignment of the following parameters or the parameter list must be concluded prematurely with ")".

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 result of which is 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 configurations with an MMC 100.2, the program is executed normally in the NC and the traversing motion is recorded on the screen during the simulation run.

In configurations with an MMC 103, the program is simulated solely in the MMC. For this reason, it is possible to execute cycles without tool data or without prior selection of a tool offset in the MMC with SW 4.4 and higher.
The finished contour is then traversed in the case of cycles which 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 is output 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 (SW 4.3 and higher)

The program editor in the control provides you with programming support to add cycle calls to the program and enter parameters.

In this way, support is provided both for Siemens cycles and user cycles.

Function

The cycle support consists of the three components:

1. Cycle selection
2. Input screen forms for parameter setting
3. Help display per cycle.

It is not absolutely necessary to create help displays when incorporating separate cycles; then, only the input screen forms are displayed for the cycles.

If is also possible to configure the text files of the cycle support as language-independent. In this case, the corresponding text files, located in the MMC, are also required.

A detailed description of the program editor is given in
References: /BA/, "Operator's Guide"

1.4.1 Overview of important files

The following files form the basis for cycle support:

Assignment	File	Application	File type
Cycle selection	cov.com	Standard and user cycles	Text file
Input screen form for parameter setting	sc.com	Standard cycles	Text file
Input screen form for parameter setting	uc.com	User cycles	Text file
Help displays	*.bmp	Standard or user cycles	Bitmap

For MMC 100/MMC 100.2 the help displays must be converted into another format (*.pcx) and linked to produce a loadable file (cst.arj).

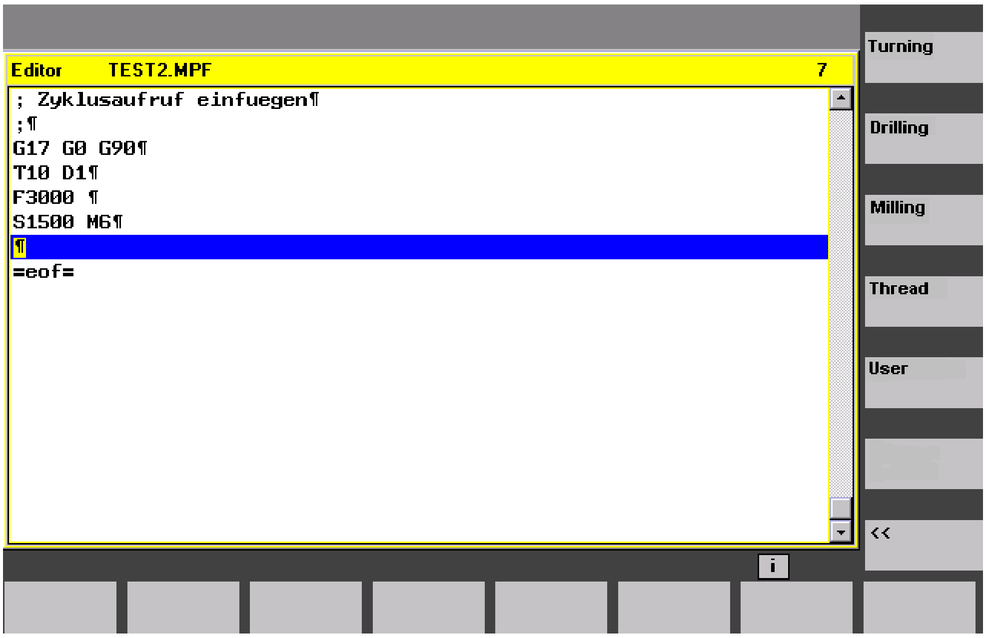
1.4.2 Configuring cycle selection

Function

The cycle selection is configured in the cov.com file:

- The cycle selection is assigned directly to soft keys that are configured in the cov.com file.
- Up to three soft key levels with up to 18 soft keys are supported; this enables the cycles to be classified in subsets, e.g. of one technology.
- If a maximum of six cycles are configured on one of the soft key levels, they all lie on a vertical soft key tree. The 7th and 8th soft keys are reserved for operator functions such as **"Back"** or **"Abort"** or **"OK"**.
  If the corresponding level contains more than six cycles, then the program labels the 7th soft key with **">>"** and switches the vertical soft key over to the 2nd level.
- Only five soft keys are available on the first level, the first soft key is reserved.

Example for cycle selection



Programming

Syntax of the cov.com file (example)

`%_N_COV_COM`	
`;$PATH=/_N_CUS_DIR`	
`;V04.03.01/10.09.97`	
`S2.0.0\Turning\`	
`S3.0.0\Drilling\`	
`S4.0.0\Milling\`	
`S5.0.0\Threads\`	
`S6.0.0\Users\`	
`S3.1.0\Deep hole %ndrilling\C3(CYCLE83)`	Deep hole drilling
`S3.2.0\Boring\`	
`S3.2.1\Boring%n1\C6(CYCLE85)`	Boring 1
`...`	
`M17`	

Explanation of syntax

`Sx.y.z`	Soft key number and level, the decimal point is used to separate the three numbers x denotes the soft key of the 1st level (2 to 18 are possible) y denotes the soft key of the 2nd level (1 to 18 are possible). z denotes the soft key of the 3rd level (1 to 18)
`\text\`	Soft key text, maximum of 2 · 9 characters The separator character is "%n"
`Cxx`	Help display name, a "p" is added to the name of the help display file for cycle support, e.g. Cxxp.bmp
`(Name)`	Cycle name that is written to the program and is present in the input screen form for parameter setting.

After the cycle name, you can write a comment separated from the name by at least one blank.

Special points relating to MMC 102/103

If this file is language-independent, i.e. configured in plain text, the file name must include a language code, e.g.:

- COV_GR.COM for German,
- COV_UK.COM for English,
- COV_ES.COM for Spanish,
- COV_FR.COM for French,
- COV_IT.COM for Italian,

or other codes for different languages.

1.4.3 Configuring input screen forms for parameter assignment

The SC.COM (Siemens cycles) and UC.COM (user cycles) files provide the basis for configuring the input screen form for parameter setting.
The syntax is identical for both files.

Explanation

The following is an example of the cycle header:

```
Name of help display
   |      Cycle name
   |       |         Comments
   |       |             |
//C6 (CYCLE85)  Boring 1
```

`//`	Header detection for a cycle description
`C6`	Name of the help display with a p added (C1 – C28 Siemens Cycles)
`(CYCLE85)`	Name of the cycle. This name is also written to the NC program.
`Boring 1`	Comments (is not evaluated)

Cycle parameterization

`(R/0 2/1/Return plane, absolute)[return plane/RTP]`

Start	`(`
Variable type	`R` REAL `I` INTEGER `C` CHARACTER `S` STRING
Separator	`/`
Value range	Lower limit, blank, upper limit (e.g. `0 2`)
Separator	`/`
Default value	`one value (e. g. 1)`
Separator	`/`
Long text	is output in the dialog line
End	`)`
Start option	`[`
Short text	appears in the parameterization screen form
Separator	`/`
Text in bitmap	Parameter name
End option	`]`

Instead of limiting a value range, it is possible to define individual values by enumeration. These are then selected for input using the toggle button.

```
(I/* 1 2 3 4 11 12 13 14/11/Selecting the
operating mode)[Operating mode / VARI]
```

In order to achieve compatibility with the states of the cycle support for interactive programming of the MMC 102/103, only the section in round brackets is mandatory. The section in square brackets is optional. Values do not have to be provided for lower/upper limit and default, see programming example.

Explanation

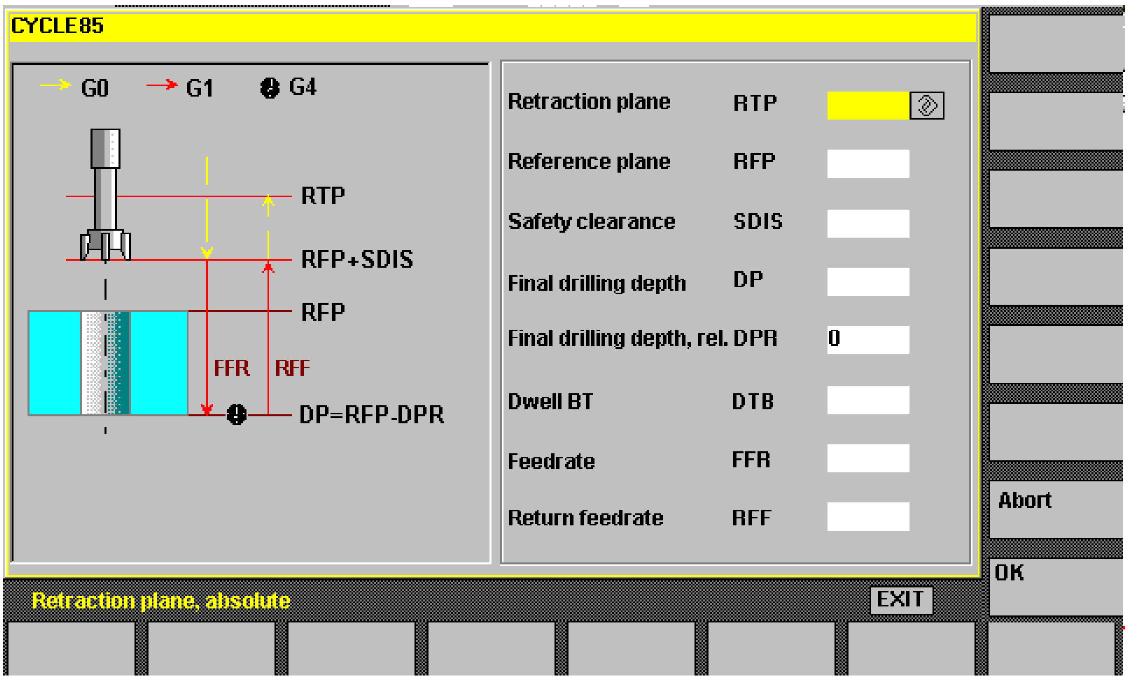
If the section in square brackets is missing, proceed as follows:

Short text=	the first 19 characters of the long text but only up to the first blank from the right or up to the first comma from the left. Shortened texts are marked with an asterisk " * "
Text in bitmap=	is read from the Cxx.awb file

Programming example

Cycle support for the cycle:
corresponds to the COM files SW4
MMC 100/MMC 100.2 and cycle support ASCII
Editor MMC 102/103

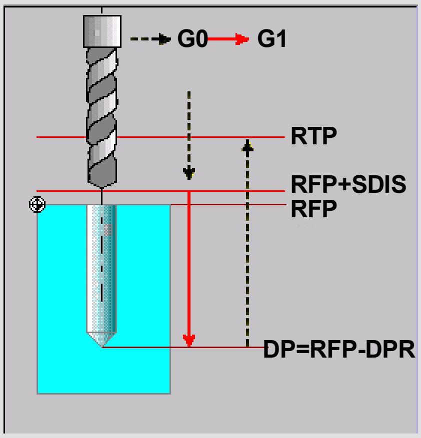
`//C6(CYCLE85)`	Boring 1
`(R///Retraction plane, absolute)[Retraction plane/RTP]`	
`(R///Reference plane, absolute)[Reference plane/RFP]`	
`(R/0 99999//Safety distance, without sign) [safety distance/SDIS]`	
`(R///Final drilling depth, absolute)[Final drilling depth/DP]`	
`(R/0 99999/0/Final drilling depth relative to reference plane)[Final drilling depth rel./,DPR]`	
`(R/0 99999//Dwell at drilling depth)[Dwell BT/DTB]`	
`(R/0.001 999999//Feedrate)[Feedrate/FFR]`	
`(R/0.001 999999//Return feedrate)[Return feedrate/RFF]`	



1.4.4 Configuring help displays

Explanation

Help displays for MMC 100/MMC 100.2
If you wish to modify the standard graphics or create additional graphics, you will need to have a graphic program on your PC. The size of the graphic is limited to max. 272 · 280 pixels. It is recommended that you make all graphics the same size.



Example of cycle parameterization graphics

The MMC uses the PCX format of Zsoft Paintbrush as graphic format. If you do not have a graphic program that can create this format, you can use the Paint Shop Pro program to convert your graphics.

The names of the bitmaps are directly related to file uc.com. If a file is configured there, for example with //C60 (POSITION1), the bitmap must be called C60.bmp for MMC 100.2 or C60p.bmp for MMC 103.

The Paint Shop Pro application is not included on the diskette supplied by Siemens.

Help displays for MMC 102/103
The help displays of the MMC 102/103 are located in the file system under the directory H\DP.DIR\HLP.DIR. You can use the "Copy" function in the Services menu to read data from a floppy disk. To do this, select the destination directory via "Interactive programming" and "DP Help".

1.4.5 Configuring tools (MMC 100 / MMC 100.2 only)

Explanation

For MMC 100 / MMC 100.2, you also require a conversion tool to convert the file format from *.bmp to *.pcx.

This tool is located on the delivery diskette under the path MMC100/MMC100.2\TOOLS.

This enables you to carry out conversion and compression to produce a loadable file for MMC 100 / MMC 100.2.

The PCX files are converted and subsequently compressed into an archive file by means of the tools **PCX_CON.EXE** and **ARJ.EXE**. These tool are contained on the diskette.

The files to be converted must all reside on one path, multiple paths are not supported.

Conversion routine call:

```
makepcx.bat
```

All parameters required have already been stored in this file.

The conversion produces the files *.b00 *.b01 and *.b02. Prior to compression, copy all these files (*.b0* as well as the arj.exe tool into a path and start the following call:

```
arj a cst.arj *.*
```

1.4.6 Loading to the control

Loading with MMC 100 / MMC 100.2

Precondition

The application diskette has already been installed on your PC.

Sequence of operations

- Change to directory **"INSTUTIL"** in your application path and start **"APP_INST.EXE".** The selection menu for software installation is displayed.
- Select menu option **"Modify configuration"**. A further selection menu appears. In this menu select the option **"Add *.* Files ..."**. As the file name enter your graphics file path and file name **"CST.ARJ"** in the input screen form.
- Press the Return key to confirm your input.
- Press **Esc** to return to the main menu where you can transfer your software to the hardware.

Loading to MMC 102/103

Sequence of operations

The help displays for cycle support are located in the directory

Interactive programming\DP help.

They are entered from the diskette in long format using the operations

- "Data Management" and
- "Copy".

1.4.7 Independence of language

Explanation

Cycle support files can also be configured as language-independent.

This is done by replacing all the texts in the cov.com and sc.com files by text numbers. In addition, a text file is also required in the control.

The aluc.txt file with text number range 85000...89899 is reserved for user cycles.

This file is named aluc_(language).com in the MMC 103 and stored in directory DH\MB.DIR (MBDDE alarm texts) in the file system.

Example:

```
//C60 (POSITION1)
(R///$85000)[$85001/XWERT]
(R///$85002) [$85003/YWERT]
(R///$85004) [$85005/ZWERT]
```

Relevant text file:

85000	0	0	Position for the 1st axis of the plane
85001	0	0	Position X
85002	0	0	Position for the 2nd axis of the plane
85003	0	0	Position Y
85004	0	0	Position for the 3rd axis of the plane
85005	0	0	Position Z

Explanation of the syntax:

`$`	Identifier for text numbers
`85000...89899`	Text number for user cycles
`$85000... $...`	Several texts are concatenated

1.4.8 Operating the cycles support function

Explanation

Carry out the steps below to add a cycle call to a program:

- Soft key "Support" in the horizontal soft key bar.
- Soft key "new cycle" (MMC 102/103 only).
- Select the cycle via the vertical soft key bar until the corresponding input screen form appears (the help display appears on the MMC 100 / MMC 100.2 when you press the Info key).
- Enter the parameter value.
- With the MMC 103, it is also possible to input the name of a variable instead of a value in the screen form; the variable name always starts with a letter or an underscore.
- Hit "OK" to confirm (or "Abort" if the input is incorrect).

1.4.9 Integrating user cycles into the MMC 103 simulation function

Explanation

If you wish to simulate user cycles in the MMC 103, the call line for each cycle must be entered in file dpcuscyc.com in directory DH\DP.DIR\SIM.DIR. The call line must be entered there for each cycle.

Programming example

A user cycle named POSITION1 with 3 transfer parameters is loaded to the control for simulation.

```
%_N_POSITION1_SPF
;$PATH=/_N_CUS_DIR
PROC POSITION1 (REAL XWERT, REAL YWERT, REAL ZWERT)
...
M17
```

The following line

`PROC POSITION1 (REAL XWERT, REAL YWERT, REAL ZWERT)`

must then be entered in file dpcuscyc.com.

1.4.10 Typical user cycle configuration

Programming example

1. Change cov.com (menu configuration)
   S6.0.0\User\
   S6.1.0\Position1\
2. Configuring in uc.com (master configuration)

```
//C60 (POSITION1)
(R///$85000)[$85001/XWERT]
(R///$85002) [$85003/YWERT]
(R///$85004) [$85005/ZWERT]
```

3. Configuring in aluc.txt (text file)

85000	0	0	Position for the 1st axis of the plane
85001	0	0	Position X
85002	0	0	Position for the 2nd axis of the plane
85003	0	0	Position Y
85004	0	0	Position for the 3rd axis of the plane
85005	0	0	Position Z

4. Bitmap
   C60.bmp for MMC 100.2
   C60p.bmp in path DH\DP.DIR\HLP.DIR for MMC 103
5. Integrate in simulation MMC 103
   see Subsection 1.4.9

1.5 Cycle support in program editor (SW 5.1 and higher)

As from SW 5.1, the program editor offers an extended cycle support for Siemens and user cycles.

Function

The cycle support offers the following functions:

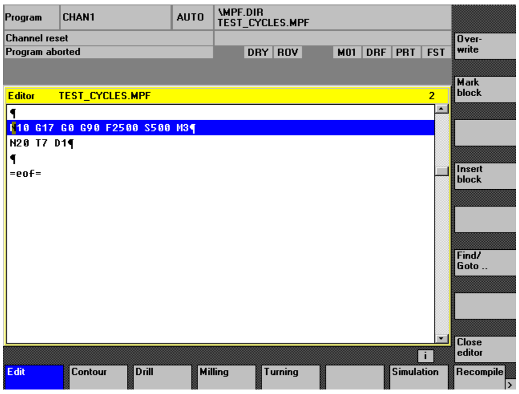
- Cycle selection via soft keys
- Input screen forms for parameter assignment with help displays
- Online help for each parameter (with MMC 103 only)
- Support of contour input.

Retranslatable code is generated from the individual screen forms.

1.5.1 Menus, cycle selection

Explanation

The cycle selection is carried out technology-oriented via 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 form for milling cycles.

Turning — Input screen forms for turning cycles.

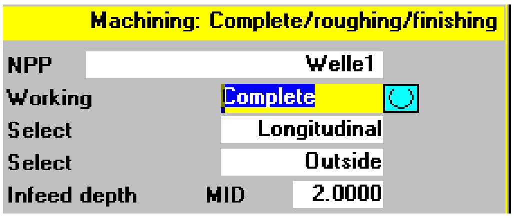
After confirming the screen form input by clicking OK, the technology selection bar is still visible.

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.

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 position.

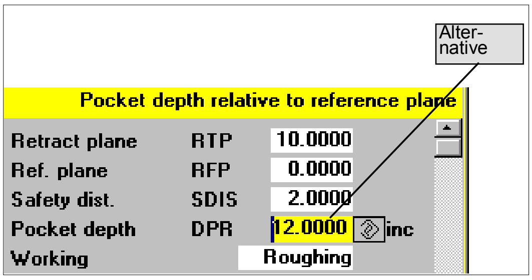
1.5.2 New functions in input screen forms

Function

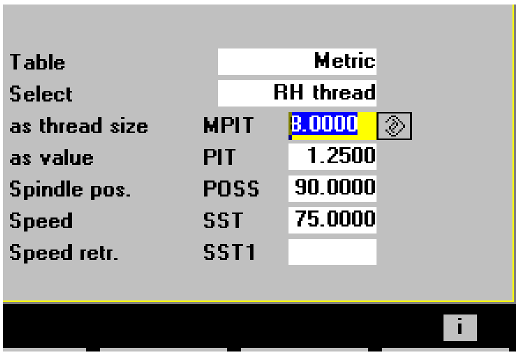


- In many cycles, the processing type may be influenced by means of the VARI parameter. It contains several settings composing one code. These individual settings are divided up into several input fields in the screen forms of the new cycle support. You can switch between the input field with the toggle key.
- The input screen forms are changed dynamically. Only those input fields are displayed that are required for the selected processing type. Unrequired input fields are not displayed. In the example, this is the case with the parameter for the dressing feedrate.
- One input may therefore automatically assign several depending parameters. This is the case with threading which presently 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 input field and 1.137 to the thread depth input field (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 last 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 dressing), only few parameters then have to be changed.

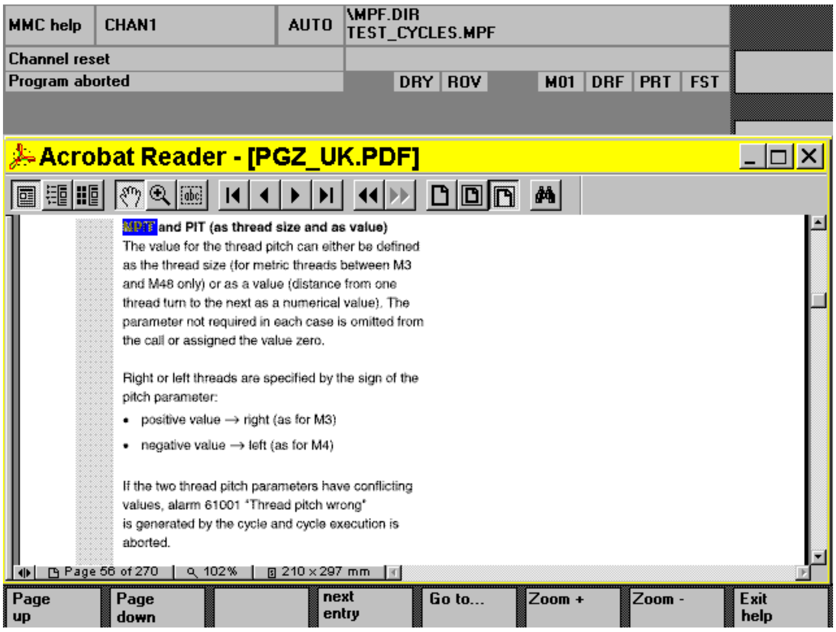
- In screen forms of drilling and milling cycles, certain parameters may be input as absolute or incremental values. The abbreviation ABS for absolute and INC for incremental values is displayed behind the input field. You may switch between them with the "Alternative" soft key. This setting will remain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



- With the MMC 103 you may display additional information on the individual cycle parameters by means of the online help. If the cursor is placed on a parameter and the help icon i is displayed on the bottom right-hand side of the screen, the help function can be activated.



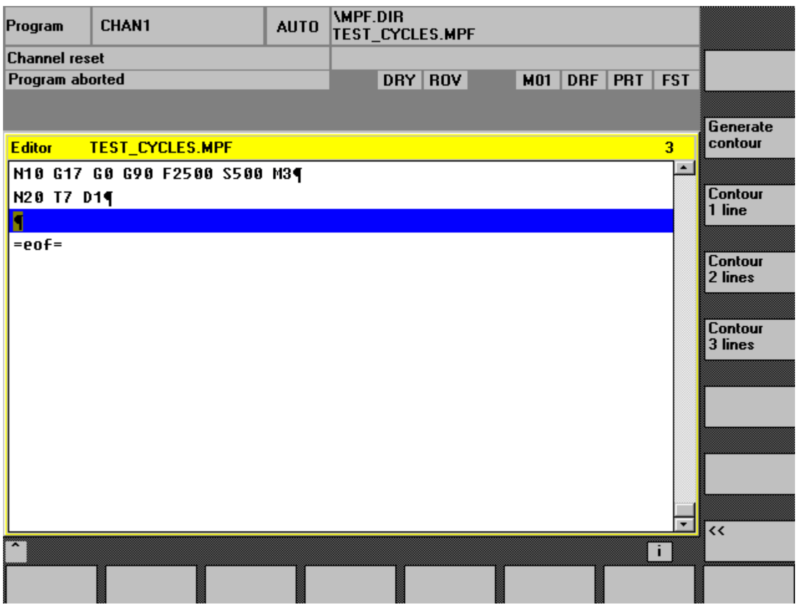
By pressing the info key the parameter explanation is displayed from the Cycle Programming Guide.



Operator commands in the help display

Page up	Paging backward in the documentation.
Page down	Paging forward in the documentation.
next entry	Enables the jump to another piece of text included in the help display.
Go to	Enables the jump to a selected piece of text.
Zoom +	Zoom the text in the help window.
Zoom -	Reduce the text in 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 (refer to References: /OG/, Chapter 6).

Blueprint programming

Contour 1 line

Contour 2 lines

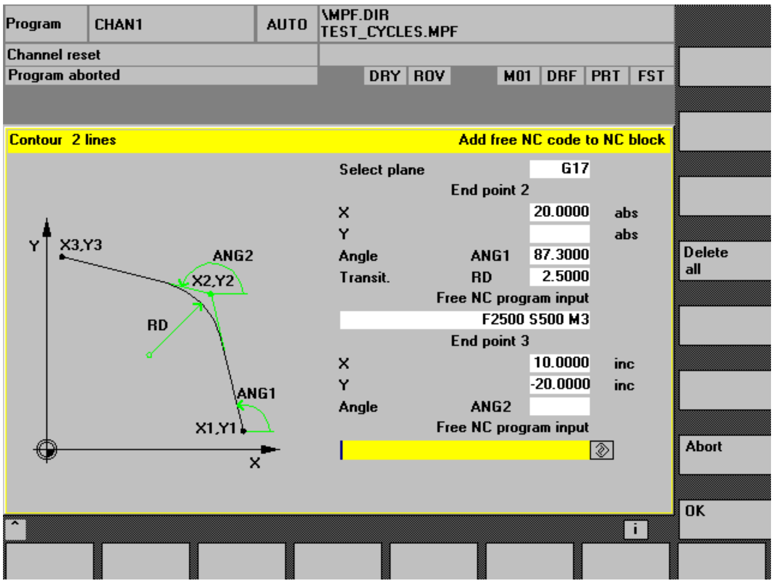
Contur 3 lines

These soft keys support blueprints that are possible with SW 5 or higher.

It consists of one or several straight lines with contour transition elements in-between (radii, chamfers). Each contour element may be preassigned by means of end points or point and angle and supplemented by a free DIN code.

Example

The following DIN code is created from the following input screen form for a 2-straight-line contour definition creates the following DIN code:



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

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

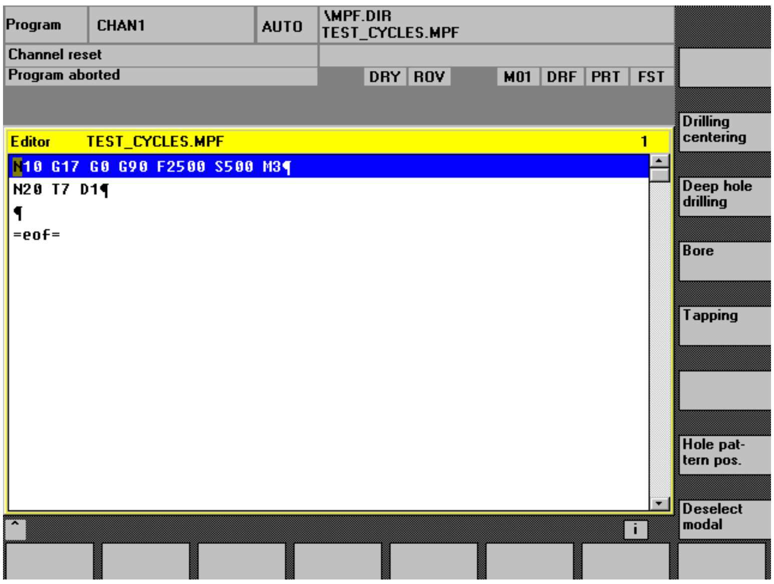
Drilling support

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 |



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

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

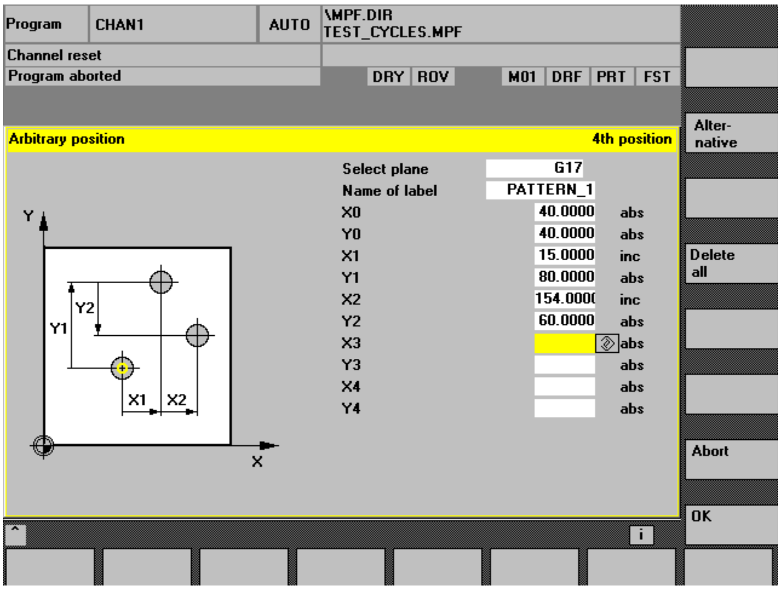
The function of CYCLE87 is covered by the function of CYCLE88 (soft keys "Drilling center." → "Drilling with stop").

Drilling patterns may be repeated if, for example, drilling and tapping are to be executed in succession. Thus, a name is assigned to the drilling pattern which will later be entered in the 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:`	Marker – Name of drilling pattern
`N150 HOLES2(50,50,37,20,20,9)`	Call drilling pattern cycle
`N160 ENDLABEL:`	
`N170 MCALL`	Deselect modal call
`N180 T8 M6`	Change tap
`N190 S400 M3`	
`N200 MCALL CYCLE84(10,0,2,0,30,,3,5,0.8,180,300,500)`	Modal call of tapping cycle
`N210 REPEAT Circle of holes 1`	Repeat drilling pattern
`N220 MCALL`	Deselect modal call

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

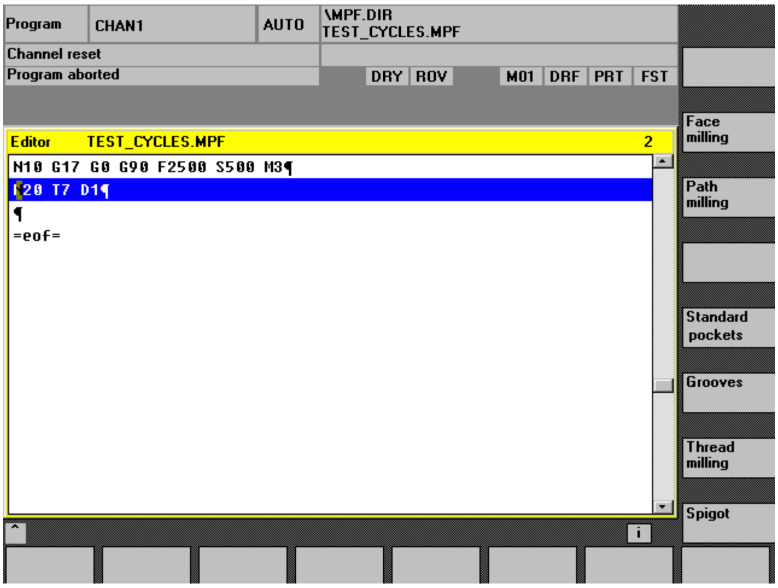


Thus, up to five positions may be programmed in the plane, all values either absolute or incremental (alternate with "Alternat." soft key). The "Delete all" soft key creates an empty screen form.

Milling support

The milling support includes the following selection possibilities:

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



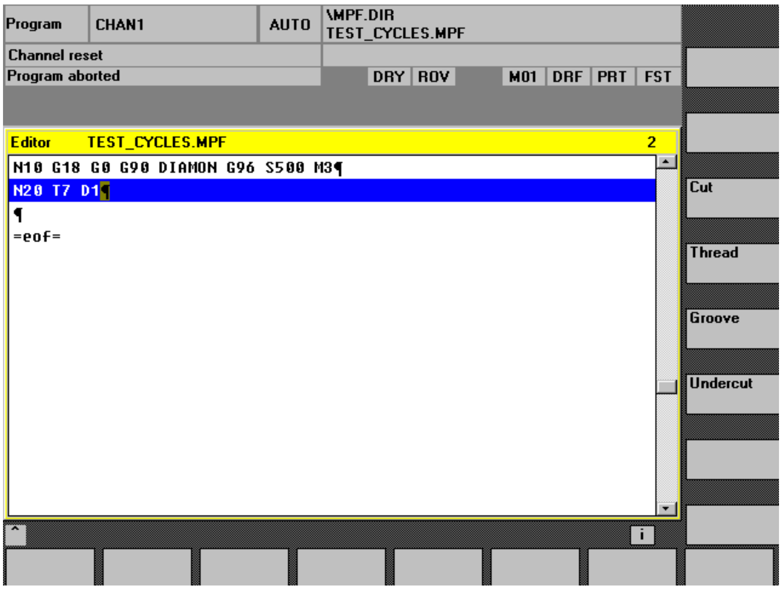
The "Standard pockets", "Grooves" and "Spigots" soft keys each branch into submenus offering a selection of pocket, groove and spigot milling cycles.

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

Turning support

The 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 stored under the "Undercut" soft key.

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

Retranslation

The retranslation of program codes serves to change an existing program with the help of the cycle support. The cursor is set to the line to be changed and the "Retranslation" soft key is pressed.

Thus, the corresponding input screen form which created the program piece is reopened and values may be modified.

Directly entering modifications in the created DIN code may result in the fact that retranslation is no longer possible. 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/, MMC Installation Instructions

BE1 "Expand the Operator Interface"

1.6 Cycle support for user cycles (SW 6.2 and higher)

1.6.1 Overview of important 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 setting	*.com	Standard or user cycles	Text file
Help displays	*.bmp	Standard or user cycles	Bitmap
Online Help (HMI Advanced only)	pgz_<language>.pdf and pgz_<language>.txt	standard cycles only	pdf-file

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

1.6.2 Entry to cycle support

Function

The horizontal soft key HS6 in program editor is designated 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	

A detailed description of the configuration is given in:

References: **/IAM/ HMI/MMC Installation and Start-Up Guide**

BE1 "Expand the Operator Interface"

With HMI Embedded, make the entry in file common.com to activate this soft key as follows:

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

1.6.3 Cycle support configuration

Function

The soft key bars and input screen forms of cycle support can be configured in any files and stored as type *.com in the HMI of the control.
A detailed description of the configuration is given in:

References: **/IAM/ HMI/MMC Installation and Start-Up Guide**

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" by means of RS-232-C). 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 into an empty directory on a PC.
- Copy the separate configuration files *.com 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/, Operator's Guide HMI Embedded**
/IAM/, HMI/MMC Installation & Start-Up Guide

IM2 "Installation and Start-Up Guide HMI Embedded"

1.6.4 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), 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.

1.6.5 Storing bitmaps in data management for HMI Advanced

New paths have been set up In data management (as from 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

In accordance with the current resolution, the search begins in the appropriate directory (for example 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.

1.6.6 Handling bitmaps for HMI Embedded

Function

With HMI Embedded, the bitmaps are incorporated in the HMI software. So far they are grouped together in a package, **cst.arj**, as with MMC100.2. 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 need the tools supplied with the standard cycle software in the \hmi_emb\tools directory:

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

and the script files:

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

File cst.arj contains all the 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 into an empty directory on a PC.
- Create a subdirectory \bmp_file there.
- Copy your own bitmaps *.bmp to this subdirectory \bmp_file.
- 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 that you create will then be in the same directory as the generation tools.

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

References: **/BEM/, Operator's Guide HMI Embedded**
/IAM/, HMI/MMC Installation & Start-Up Guide

IM2 "Installation and Start-Up Guide HMI Embedded"

1.7 Cycle startup (SW 6.2 and higher)

1.7.1 Machine data

The following machine data are used for the cycles. The minimum values for these machine data are given in the table below.

Relevant machine data

MD No.	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 No.	MD name	Value
20240	CUTCOM_MAXNUM_CHECK_BLOCK	4

The machine data files are delivered with these defaults by the machine manufacturer.
It is important to remember that a power ON must be performed if these machine data are changed.

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

1.7.2 Definition files for the cycles GUD7.DEF and SMAC.DEF

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

In SW 6.3 and higher, these definition files GUD7.DEF and SMAC.DEF that are used jointly by a number of cycle packages, will be split. Each package will only have its own definitions. New cycle files GUD7_xxx.DEF and SMAC_xxx.DEF will be introduced, that will be in data management in the definition directory DEF.DIR.
The new files for the standard cycles are:

- GUD7_SC.DEF and
- SMAC_SC.DEF.

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.

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.

1.7.3 New form of delivery for cycles in HMI Advanced

As from 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 infos see:

- the file "siemensd.txt" from the delivery software (standard cycles) or
- for HMI Advanced, F:\dh\cst.dir\HLP.dir\siemensd.txt.

1.8 Special functions for cycles

Function

To provide an overview and for diagnosis of the cycle versions, it will be possible as from SW 6.3 to display and use version screens.
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 the individual packages and cycle files

References: **/BAD/, Operator's Guide HMI Advanced**
/BEM/, Operator's Guide HMI Embedded
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 of all cycles.
In order to display overviews of individual cycle packages, each cycle package must hold a package list of all the associated files.

Package lists

A new file type is introduced for package lists
***.cyp** (for cycle package),
in plain text, cycle package list.
The user can create package lists for his own cycle packages. They must look like this:

Structure of a package list:

First line:	version entry (after the vocabulary word **;VERSION:**) and package name (after the vocabulary word **;PACKAGE:**)
from second line:	list of the files associated with the cycle package with name and type
last line:	M30

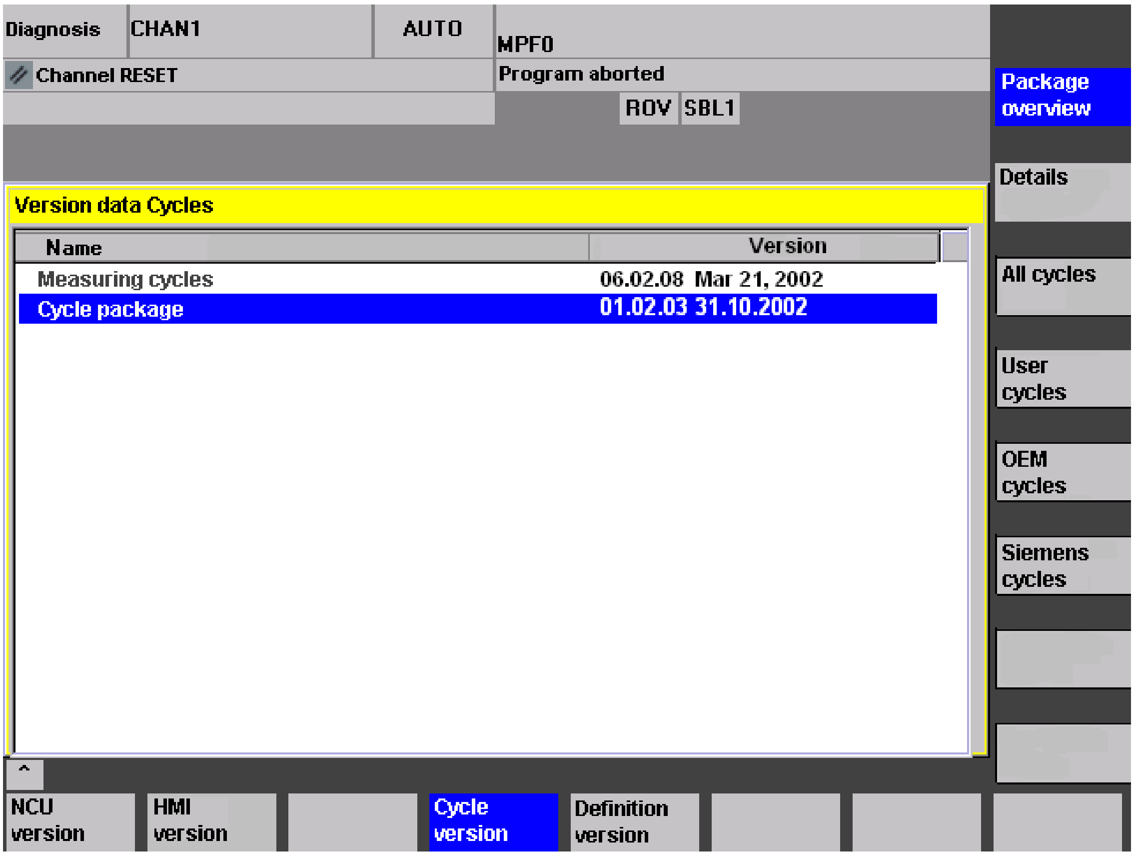
Example:

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

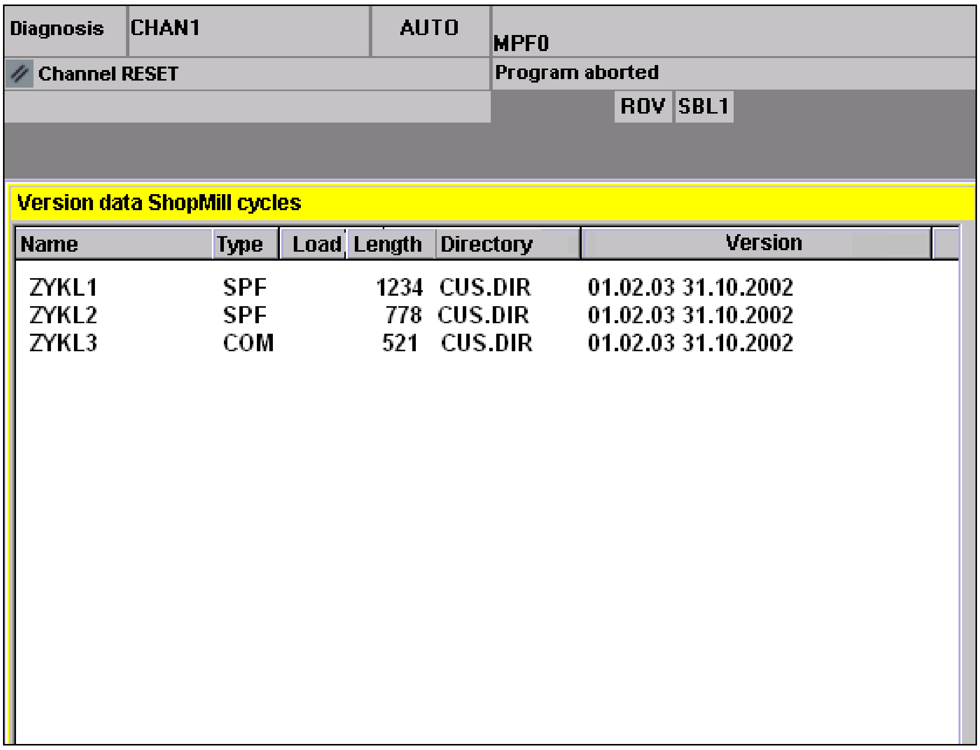
Input in the text file uc.com:

85200 0 0 "cycle package 1"

The following is displayed in the package overview:



The following is displayed in the file overview:



Further notes

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

Version entries in cycles

Just as with package lists, the entry after the vocabulary word ";VERSION:" is classed as the version ID. The version entry has to be in the first ten lines of the cycle, the search does not go any further.

Example:

```
%_N_ZYKL1_SPF
;$PATH=/_N_CUS_DIR
;VERSION: 01.02.03 31.10.2002
;Comments
PROC ZYKL1(REAL PAR1)
...
```

■

Drilling Cycles and Drilling Patterns

2.1 Drilling cycles

The following sections describe how

- drilling cycles and
- drilling pattern cycles

are programmed.

These Sections are intended to guide you in selecting cycles and assigning them with parameters. 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 to familiarize you with the use of cycles.

The sections are structured as follows:

- Programming
- Parameters
- Function
- Sequence of operations
- Explanation of parameters
- Additional notes
- Sample program.

"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 of operations", "Explanation of parameters", "Additional notes" 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:

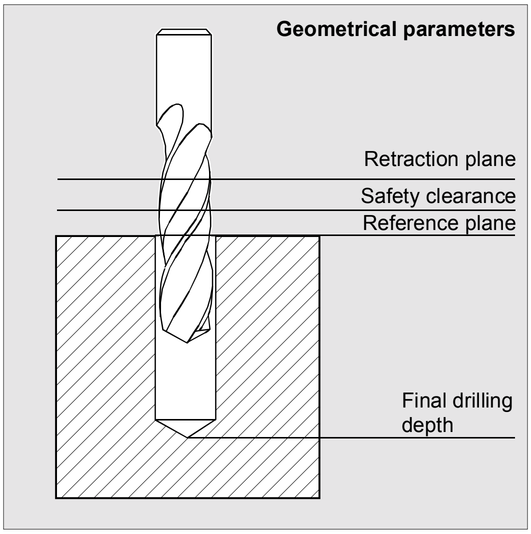
Boring cycle		Special parameterization features
Boring 1 -	`CYCLE85`	Different feedrates for boring and retraction
Boring 2 -	`CYCLE86`	Oriented spindle stop, definition of retraction path, retraction in rapid traverse, definition of spindle direction of rotation
Boring 3 -	`CYCLE87`	Spindle stop M5 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. They are therefore written in each cycle.



2.1.1 Preconditions

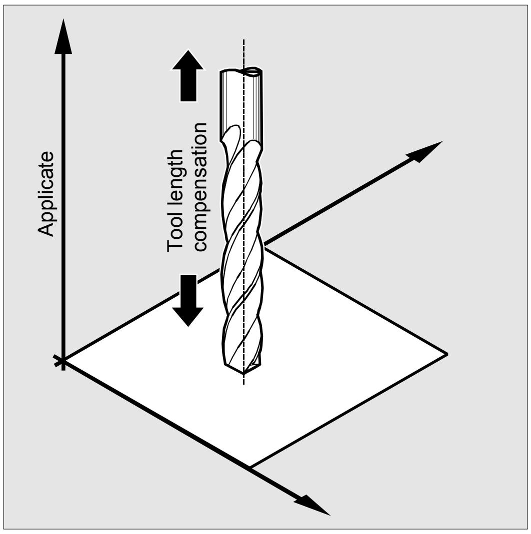
Call and return conditions

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 function and current frame active before the cycle was called remain active beyond the cycle.

Plane definition

In the case of drilling cycles, it is generally assumed that the current workpiece coordinate system in which the machining operation is to be performed is defined by selecting 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 remains active even after the end of the cycle (see also Programming Guide).



Spindle programming

The drilling cycles are written in such a way that the spindle commands always refer to the master spindle 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 must be expressly stated.

2.1.2 Drilling, centering – CYCLE81

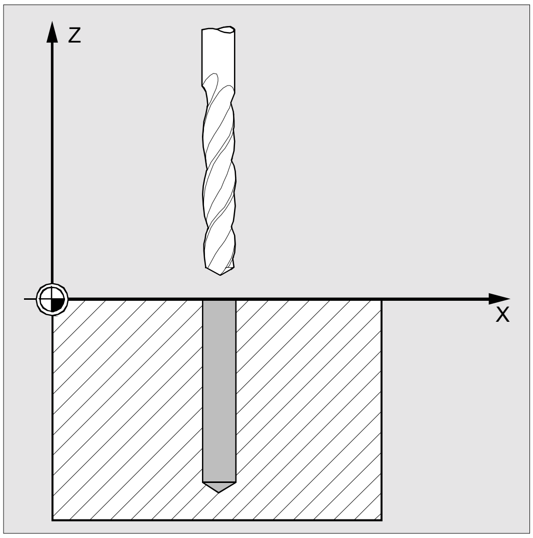
Programming

```
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.



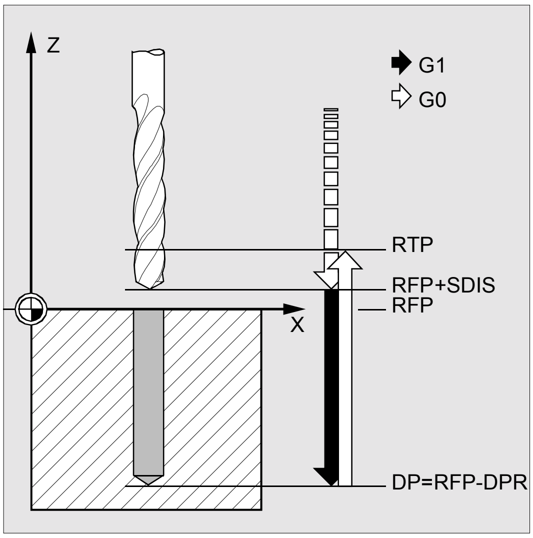
Sequence of operations

Position reached prior to cycle start:
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.

Description of parameters



RFP and RTP (reference plane and retraction plane)
Generally, the reference plane (RFP) and the retraction plane (RTP) have different values. In the cycle it is assumed that the retraction plane lies 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.

Further notes

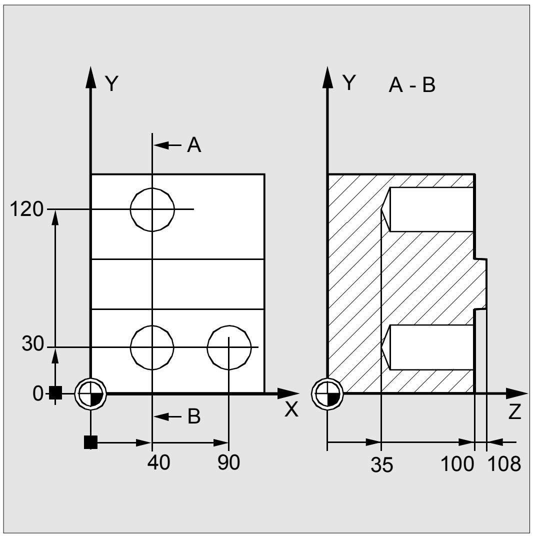
If a value is entered both for the DP and the DPR, the final drilling depth is derived from the DPR. If the DPR deviates from the absolute depth programmed via the DP, the message "Depth: Corresponds to value for relative depth" is output in the dialog line.

If the values for the reference plane and the retraction plane are identical, a relative depth must not be programmed. The error message 61101 "Reference plane incorrectly defined" is output 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 G90 F200 S300 M3`	Specification of technology values
`N20 D1 T3 Z110`	Traverse to retraction plane
`N21 M6`	
`N30 X40 Y120`	Traverse to first drilling position
`N40 CYCLE81 (110, 100, 2, 35)`	Cycle call with absolute final drilling depth, safety distance and incomplete parameter list
`N50 Y30`	Traverse to 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

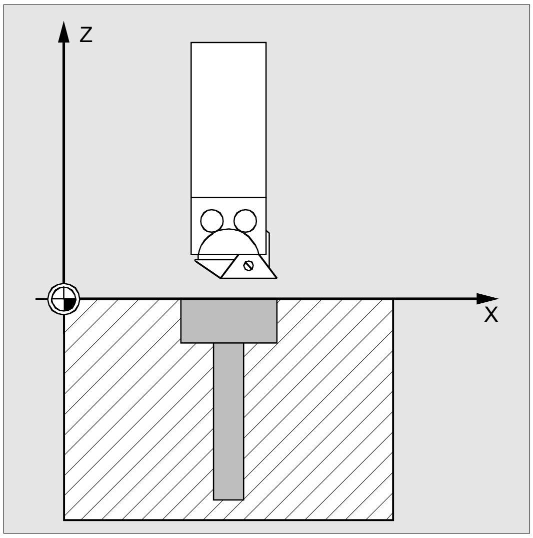
The tool drills at the programmed spindle speed and feedrate to the programmed final drilling depth. A dwell time can be allowed to elapse when the final drilling depth has been reached.

Sequence of operations

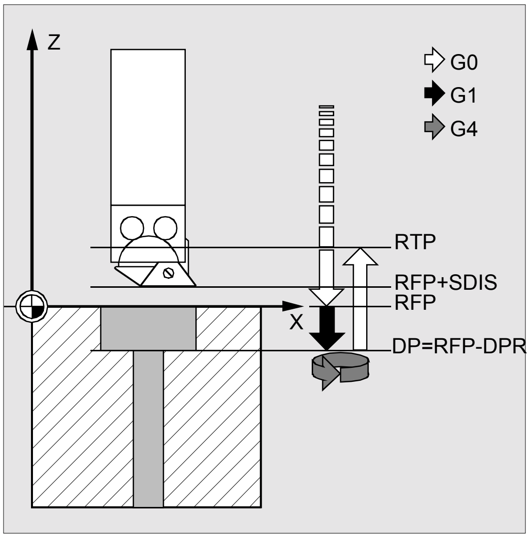
Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

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
- Dwell time at final drilling depth
- Retraction to retraction plane with G0.



Description of parameters

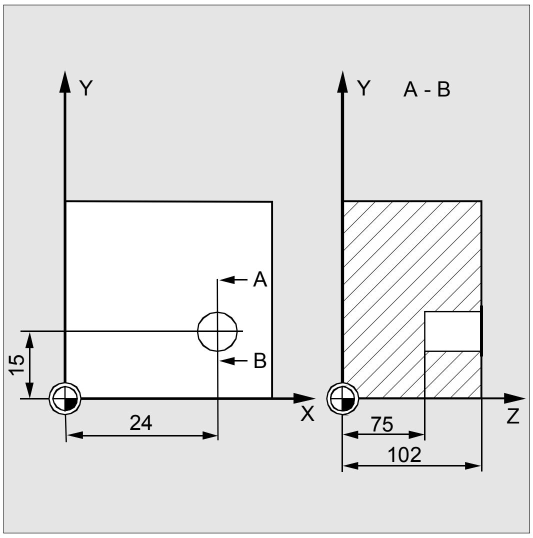


See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, SDIS, DP, DPR

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 27mm 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 4mm.



`N10 G0 G90 F200 S300 M3`	Specification of technology values
`N20 D1 T3 Z110`	Traverse to retraction plane
`N21 M6`	
`N30 X24 Y15`	Traverse to drilling position
`N40 CYCLE82 (110, 102, 4, 75, , 2)`	Cycle call with absolute final drilling depth and safety distance
`N50 M30`	End of program

2.1.4 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 1mm

_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

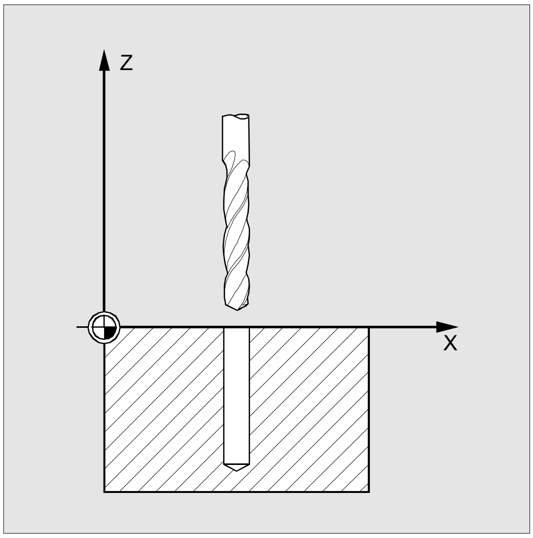
Function

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 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 in each case by 1mm for chip breaking.

Sequence of operations

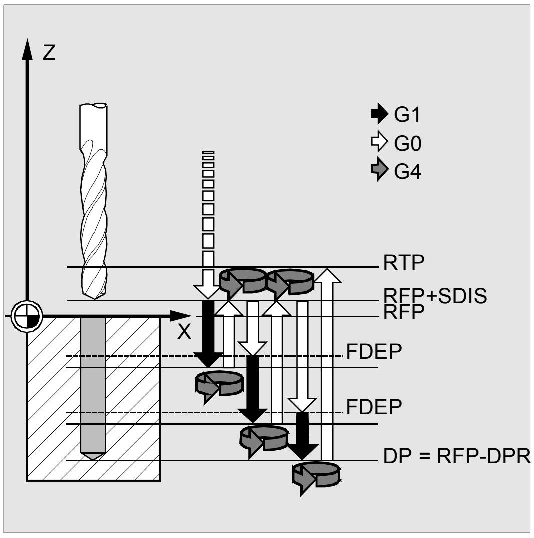
Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

The cycle implements the following motion sequence:



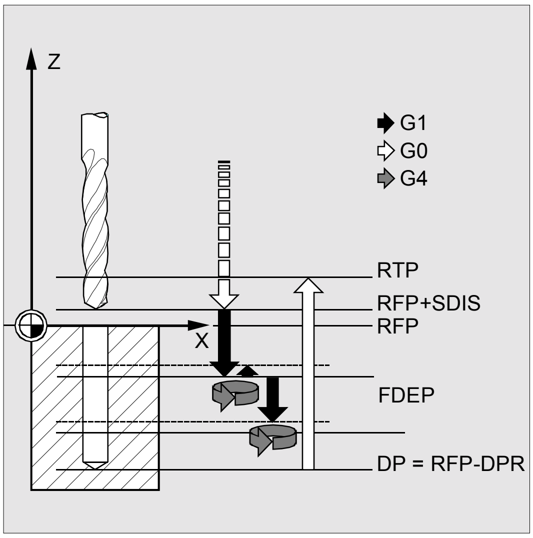
Deep hole drilling with swarf removal (VARI=1):

- 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)
- Dwell time at final drilling depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety distance with G0 for swarf removal
- 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)
- Dwell time at final drilling depth (parameter DTB)
- Retraction by 1 mm from the current drilling depth with G1 and the feedrate programmed in the calling program (for chip breaking)
- 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



Description of parameters

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

FDEP and FDPR (first drilling depth absolute or relative)
The first drilling depth is programmed by either one of these two parameters. The parameter FDPR 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 the swarf to get away 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.
- 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 not executed.

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

-12	corresponds to the first drilling depth
-21	the incremental difference 9 comes from the first drilling depth 12 reduced by the amount of degression 3
-27	previous drilling depth reduced and amount of degression 3
-30, -35, -36	amount of degression
-38, -40	remaining depth divided into two cuts.

DAM<0 (-0.001 to -1) degression factor
The current depth is derived in the cycle as follows:

- In the first step, the depth parameterized with the first drilling depth FDED or FDPR is traversed as long as 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 not executed.

Example:

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

-10	corresponds to the first drilling depth
-18	the incremental difference 8 corresponds to 0.8 *of the first drilling depth
-29.4, -29.52	in each case the previous drilling depth * degression factor
-39.52	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.

VARI (machining mode)

If parameter VARI=0 is set, the drill retracts 1mm after reaching each drilling depth for chip breaking. When VARI=1 (for swarf removal), the drill traverses in each case to the reference plane moved forward by the safety distance.

_AXN (tool axis)

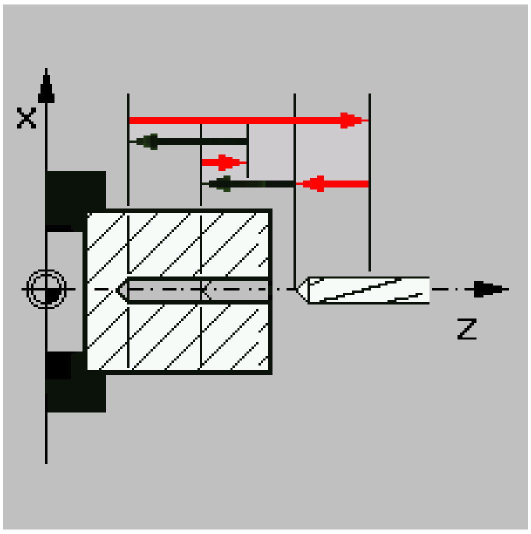
By programming the drilling axis via _AXN, it is possible to omit the switchover from plane G18 to G17 when the deep hole drilling cycle is used on turning machines.

Meaning:

_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 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 in seconds or revolutions.

Value > 0 retraction value
Value = 0 retraction value 1mm

_DTD (dwell time at final drilling depth)

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

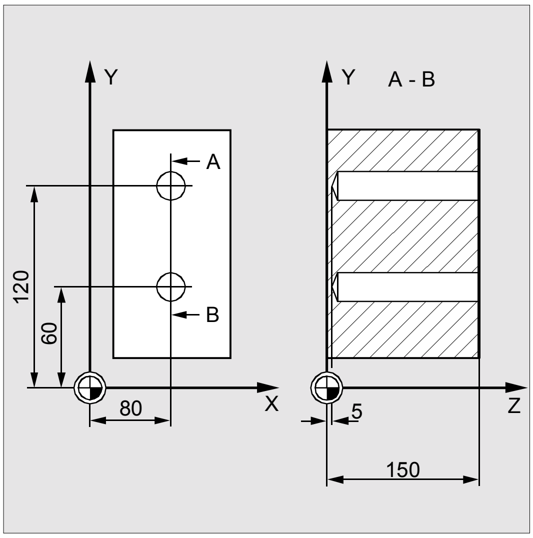
Value > 0 in seconds
Value < 0 in revolutions
Value = 0 dwell time as programmed in DTB

_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

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 zero and machining type chip breaking.

The final drilling depth and the first drilling depth are entered as absolute values. In the second cycle call, a dwell time of 1 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 8mm.

`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`	Traverse to retraction plane
`N30 X80 Y120`	Traverse to first drilling position
`N40 CYCLE83 (RTP, RFP, SDIS, DP, ,->` `-> FDEP, , DAM, , , FRF, VARI, , , _VRT)`	Cycle call, depth parameter with absolute values
`N50 X80 Y60`	Traverse to next drilling position
`N55 DAM=-0.6 FRF=0.5 VARI=1`	Assignment of value

`N60 CYCLE83 (RTP, RFP, SDIS, , DPR, , ->`
`-> FDPR, DAM, DTB, , FRF, VARI, , _MDEP,`
`-> , , _DIS1)`
	Cycle call with relative data for final drilling depth and 1st final drilling depth; the safety distance is 1mm; 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 ... 2000.000mm), the sign determines the direction of rotation in the thread SW 6.2 or higher: If _PTAB=0 or 1: in mm (as earlier SW versions) 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 (SW 6.2 or higher)	int	Tool axis Values: 1 = 1st geometry axis 2 = 2nd geometry axis or else 3rd geometry axis
_PTAB (SW 6.2 or higher)	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

_TECHNO (SW 6.2 or higher)	int	Technological settings UNITS DIGIT: Exact stop behavior Values: 0...and as programmed before cycle call 1...(G601) 2...(G602) 3...(G603) TENS DIGIT: Feedforward control Values: 0...and as programmed before cycle call 1...with feedforward control (FFWON) 2...without feedforward control (FFWOF) HUNDREDS DIGIT: Acceleration Values: 0...and 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: Values: 0...activate spindle mode again (with MCALL) 1...stay in position-control mode (with MCALL)
_VARI (SW 6.2 or higher)	int	Type of machining Values: 0...tapping in one pass 1...deep hole tapping with chip breakage 2...deep hole tapping with stock removal
_DAM (SW 6.2 or higher)	real	Incremental drilling depth Value range: 0 <= max. value
_VRT (SW 6.2 or higher)	real	Variable retraction distance for chip breakage Value range: 0 <= max. value

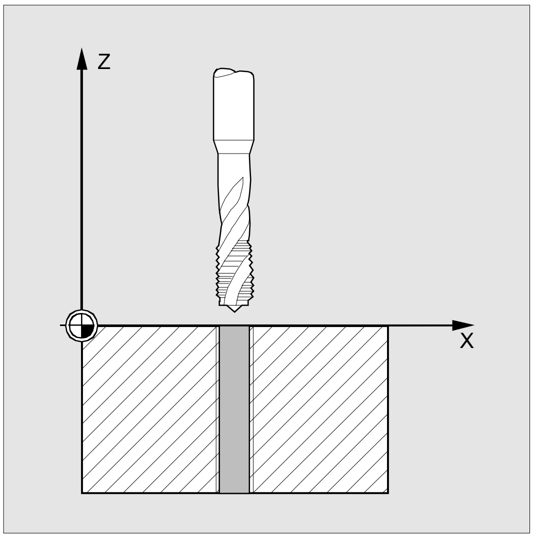
Function

The tool drills at the programmed spindle speed and feedrate to the programmed thread depth.
With cycle CYCLE84 you can perform rigid tapping operations.
In SW 6.2 or higher, 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.

A separate cycle CYCLE840 exists for tapping with compensating chuck (see Subsection 2.1.6).

Sequence of operations

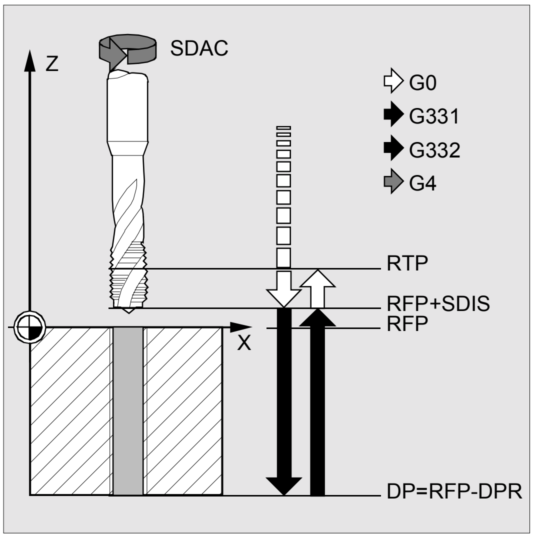


Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

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
- 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.

Description of parameters



See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, SDIS, DP, 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)
Under SDAC you program the direction of rotation after completion of the cycle.
For tapping, the direction is changed automatically by the cycle.

MPIT and PIT (as thread size and as value)

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

POSS (spindle position)

Before tapping starts in the cycle, oriented spindle stop is performed with command SPOS and the spindle is brought into 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.

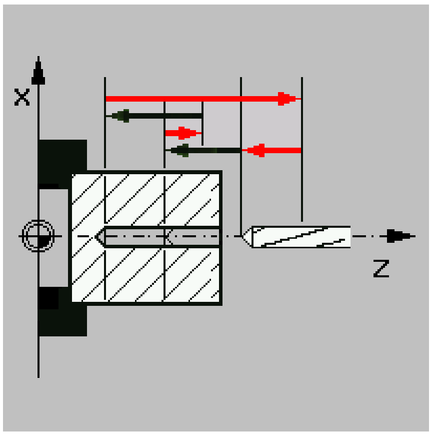
SST1 (retraction speed)

Under SST1 you program the speed for the retraction out of the thread hole in the hole with G332.
If this parameter is assigned the value zero, the retraction movement is performed with the speed programmed under SST.

_AXN (tool axis)

By programming the drilling axis via _AXN, it is possible to omit the switchover from plane G18 to G17 when the deep hole tapping cycle is used on turning machines.
Meaning:
_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 thread pitch PIT)

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

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

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

_TECHNO (technological settings)

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

Possible values are:

Ones 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 feedforward control (FFWON)
- 2=without feedforward 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)

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 breakage (retraction by variable distance from current drilling depth, parameter _VRT, _VARI = 1) and stock 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 < than 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.

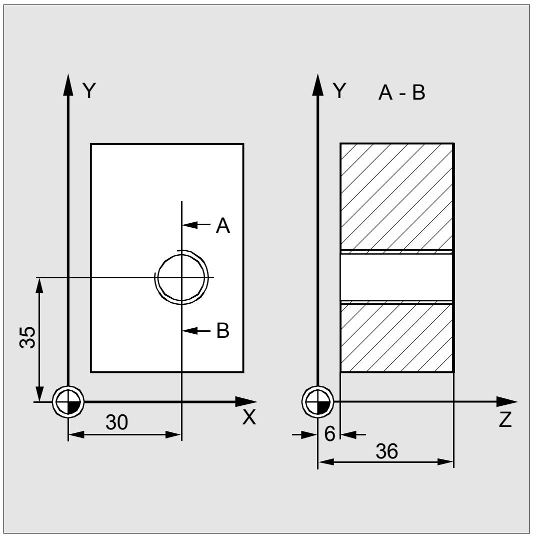
Further notes

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 and Y35 in the XY plane, the tapping axis is the Z axis. No dwell time is programmed. The depth is programmed as a relative value. The parameters for the direction of rotation and the pitch must be assigned values. A metric thread M5 is tapped.



`N10 G0 G90 T4 D1`	Specification of technology values
`N20 G17 X30 Y35 Z40`	Traverse to 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 SW 6.2 or higher: Always operative now 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 (SW 6.2 or higher) 1 = without encoder, first programs the cycle, then the feed (SW 6.2 or higher) 11 = without encoder, calculates the feed in the cycle (SW 6.2 or higher) 20 = with encoder, with dwell time (SW 6.2 or higher)
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 ... 2000.000mm SW 6.2 or higher: If _PTAB=0 or 1: in mm (as earlier SW versions) If _PTAB=2 in thread starts per inch
_AXN (SW 6.2 or higher)	int	Tool axis: Values: 1 = 1st geometry axis 2 = 2nd geometry axis or else 3rd geometry axis
_PTAB (SW 6.2 or higher)	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

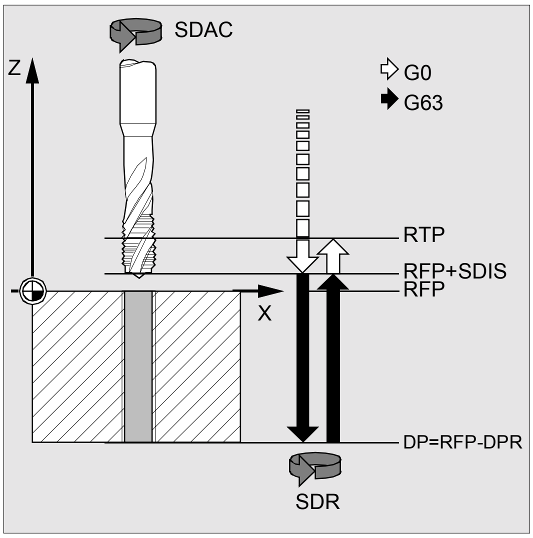
_TECHNO (SW 6.2 or higher)	int	Technological settings UNITS DIGIT: Exact stop behavior Values: 0...and as programmed before cycle call 1...(G601) 2...(G602) 3...(G603) TENS DIGIT: Feedforward control Values: 0...and as programmed before cycle call 1...with feedforward control (FFWON) 2...without feedforward 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.
With this cycle, tapping with compensating chuck can be performed

- without encoder and
- with encoder.

Sequence of operations



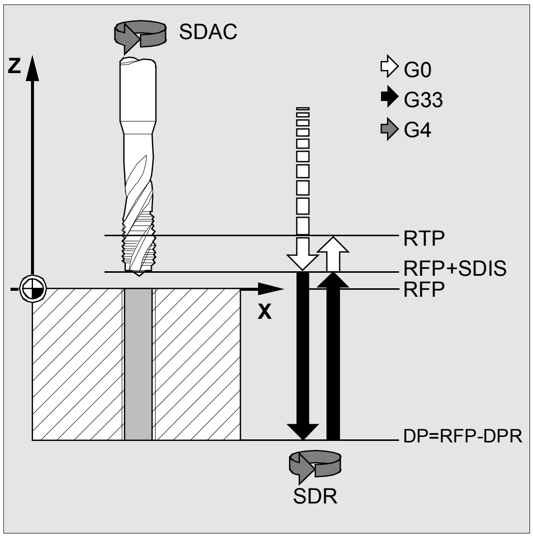
Tapping with compensating chuck without encoder (ENC=1)

Position reached prior to cycle start:
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 without encoder with encoder (ENC=0)

Position reached prior to cycle start:
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
- Dwell time at thread depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety distance with G33
- Retraction to retraction plane with G0

Description of parameters

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

DTB (dwell time)
You program the dwell time in seconds. It works in parameter ENC depending on which 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, for which SDR=0 is set.

SDAC (direction of rotation)
As the cycle can also be called modally (see Section 2.2), it requires a direction of rotation for tapping 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)
If tapping is to be performed without encoder although an encoder exists, parameter ENC must be assigned the value 1.
However, if no encoder exists and the parameter is assigned the value 0, it is ignored in the cycle.

- **Tapping without encoder with input of the lead (SW 6.2 or higher):**
  From **SW 6.2** you can calculate the correlation between feed and speed via the programmed lead within the cycle for tapping without encoder (G63 thread). The speed must be programmed before the cycle is called.
  You can specify the lead either via MPIT (metric thread unit) or via PIT (thread lead as value) just as for tapping without encoder.

The feed is then calculated in the cycle from the lead and the speed. After the cycle has ended, the last programmed feed applies again.
Programming:
ENC=11, programming the lead in MPIT or PIT

- **Tapping with encoder with dwell time (SW 6.2 or higher):**
  As an alternative, you can program a dwell time in the parameter DTB for tapping with encoder (G33 thread) from **SW 6.2**. It applies after tapping and before retraction to the retraction plane RTP, and is required for machines with unfavorable spindle dynamics.
  Programming:
  ENC=20, entering the dwell time in parameter DTB

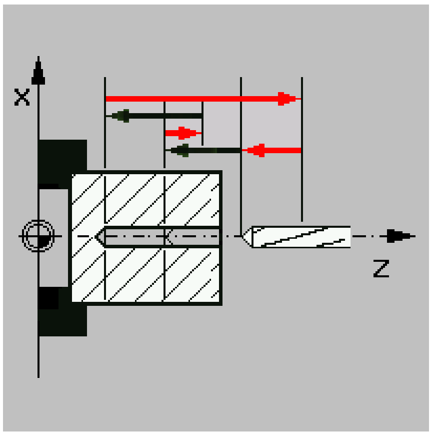
MPIT and PIT (as thread size and as value)

The parameter for the spindle 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 turn 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 wrong" is generated by the cycle and cycle execution is aborted.

_AXN (tool axis)

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

Ones 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 feedforward control (FFWON)
- 2=without feedforward control (FFWOF)

Hundreds digit (brake application point):

- 0=without calculation
- 1=with calculation.

Further notes

Depending on the setting in machine data NUM_ENCS, the cycle selects whether tapping is to performed with or without encoder.

The direction of rotation for the spindle must be programmed with M3 or M4 before the cycle is called.

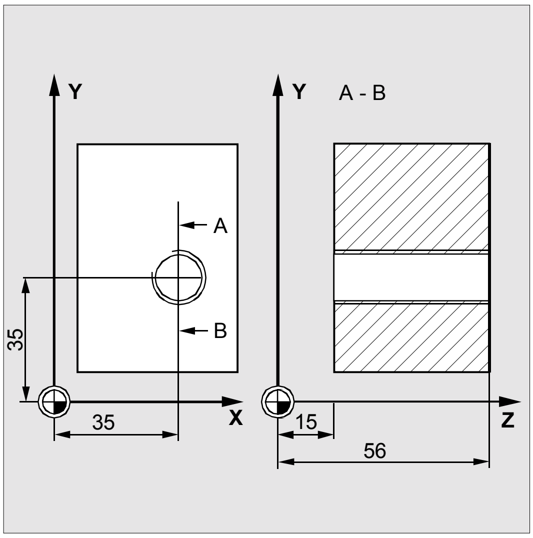
In thread blocks with G63, the values of the feedrate override switch and spindle speed override switch are frozen at 100 percent.

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

Programming example

Thread without encoder

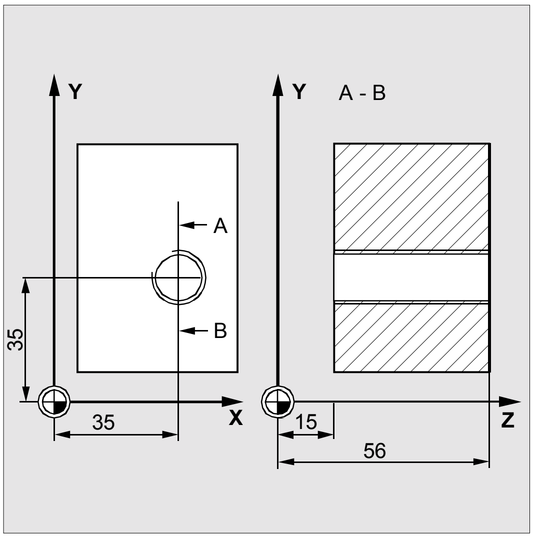
In this program a thread is tapped without encoder at position X35 Y35 in the XY plane, the drilling axis is the Z axis. Parameters SDR and SDAC for the direction of rotation must be assigned, parameter ENC is assigned the value 1, the value for the depth is absolute. Pitch parameter PIT can be omitted.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



`N10 G90 G0 D2 T2 S500 M3`	Specification of technology values
`N20 G17 X35 Y35 Z60`	Traverse to drilling position
`N30 G1 F200`	Specification of path feedrate
`N40 CYCLE840 (59, 56, , 15, , 1, 4, 3, 1)`	Cycle call, dwell time 1 sec, SDR=4, SDAC=3, no safety clearance, parameters MPIT, PIT are not programmed, i.e. the lead 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 boring 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` `DEF REAL PIT=3.5`	Definition of parameters with value assignments
`N10 G90 G0 D2 T2 S500 M4`	Specification of technology values
`N20 G17 X35 Y35 Z60`	Traverse to 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

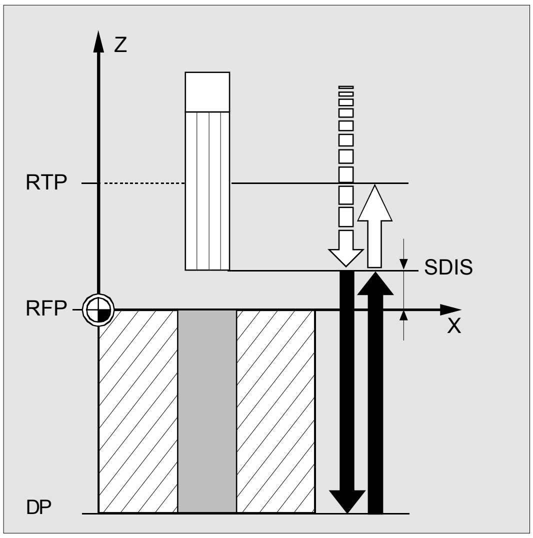
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Final drilling depth (absolute)
DPR	real	Final drilling depth relative to reference plane (enter without sign)
DTB	real	Dwell time at final drilling depth (chip breaking)
FFR	real	Feedrate
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 FFR and RFF respectively.
This cycle can be used to ream drill holes.

Sequence of operations

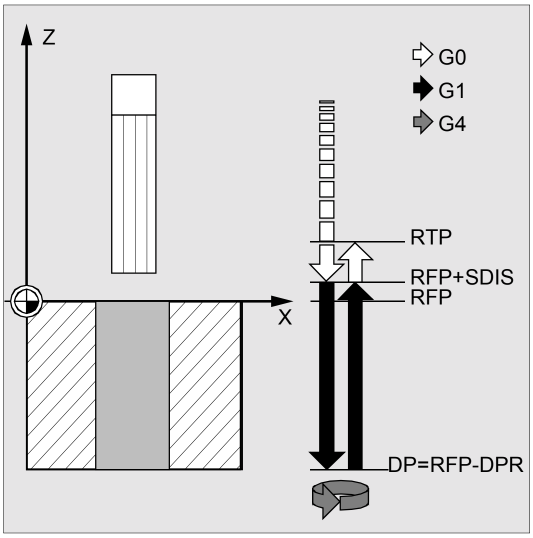
Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.



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
- 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.

Description of parameters



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

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

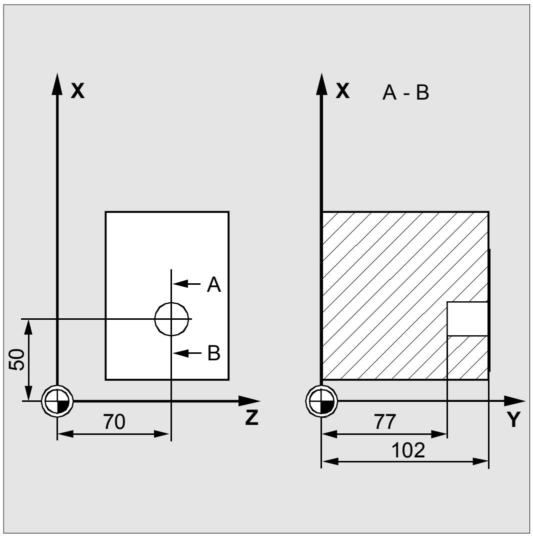
FFR (feedrate)
The feedrate value assigned to FFR is active for boring.

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 with value assignments
`N10 G0 FFR=300 RFF=1.5*FFR S500 M4`	Specification of technology values
`N20 G18 T1 D1 Z70 X50 Y105`	Traverse to 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)
```

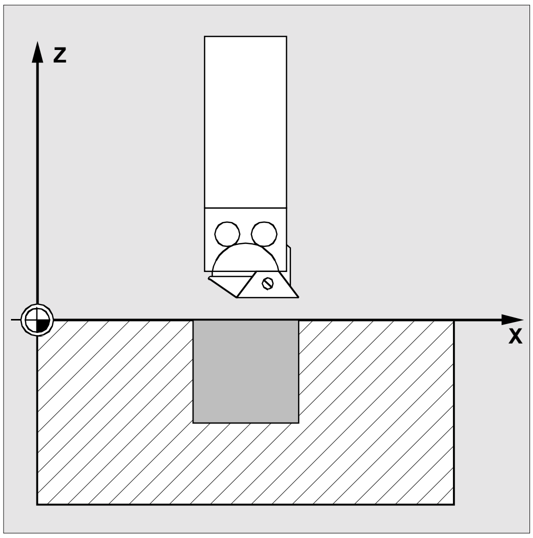
Parameters

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 of operations

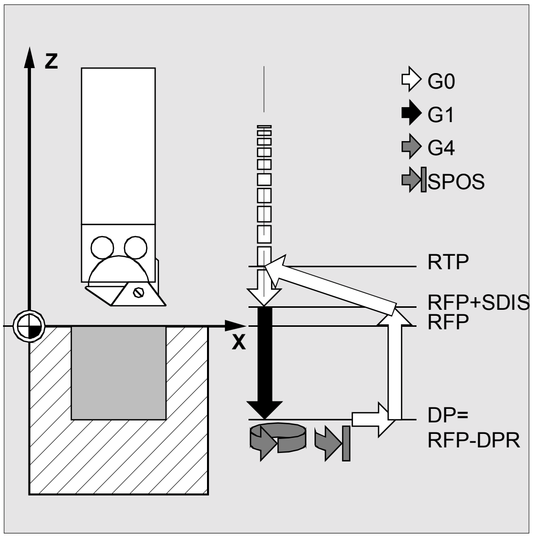


Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

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
- Oriented spindle stop at the spindle position programmed under POSS
- Traverse 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).

Description of parameters



See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, SDIS, DP, 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 retraction movement in the ordinate which is executed after the final drilling 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 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.

Further notes

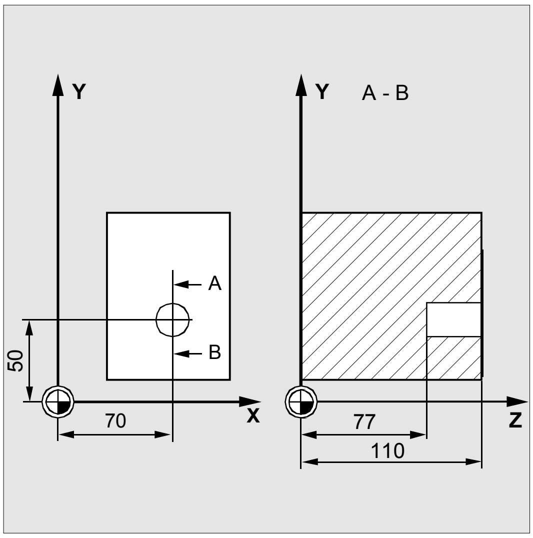
With the SPOS command you can perform an oriented spindle stop of the active master spindle. The angular value is programmed with a transfer parameter.

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 (SW 6.2 or higher)
You can now use the cycle CYCLE86 for turning machines without 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 ZX 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 secs. 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`	Traverse to retraction plane
`N40 X70 Y50`	Traverse to 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

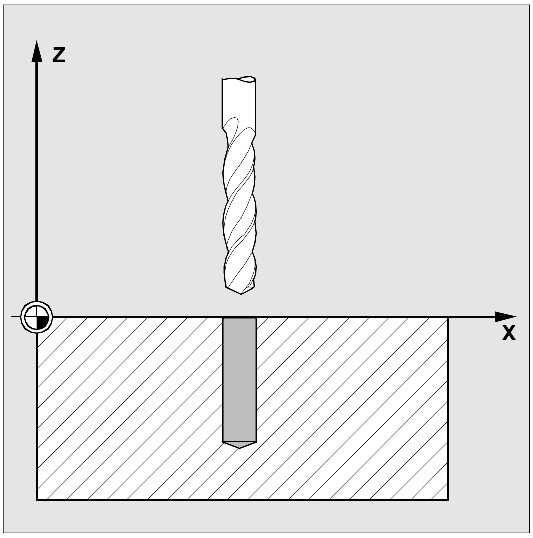
```
CYCLE87 (RTP, RFP, SDIS, DP, DPR, SDIR)
```

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.

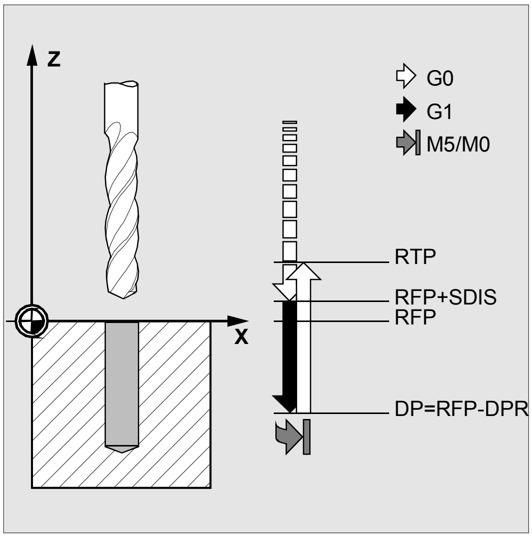


Sequence of operations

Position reached prior to cycle start:
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.



Description of parameters

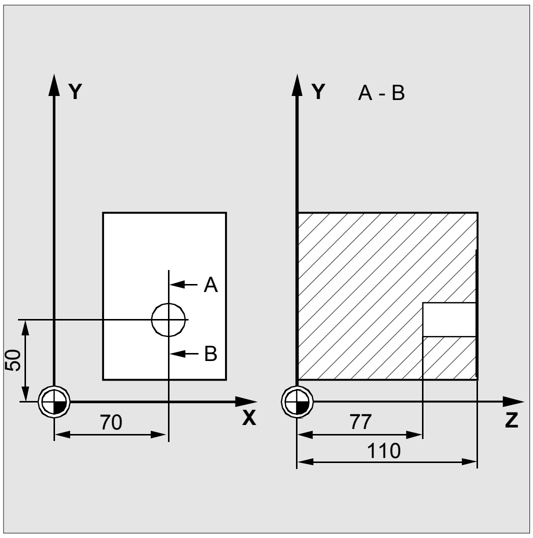
See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, SDIS, DP, DPR

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 ZX plane. the boring axis is the Z axis. The final drilling depth is programmed as an absolute value. The safety distance is 2mm.



`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`	Traverse to retraction plane
`N40 X70 Y50`	Traverse to drilling position
`N50 CYCLE87 (113, 110, SDIS, DP, , 3)`	Cycle call with programmed spindle direction M3
`N60 M30`	End of program

2.1.10 Boring 4 – CYCLE88

Programming

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

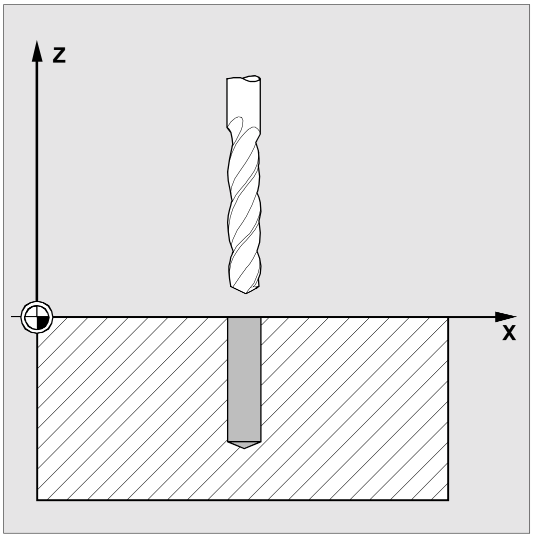
Parameters

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 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 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 of operations



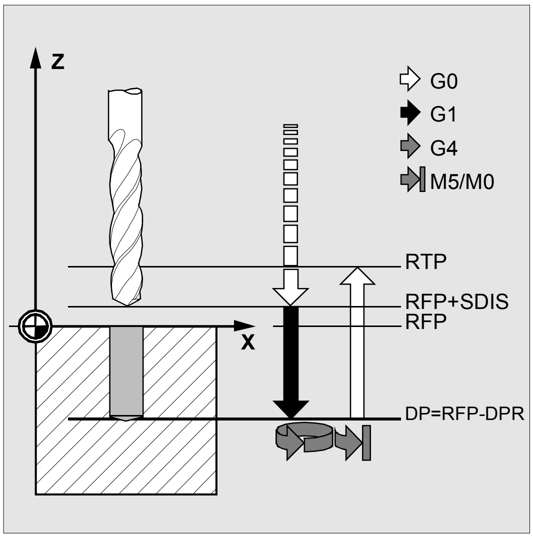
Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

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.

Description of parameters

See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, SDIS, DP, DPR Cycle setting data _ZSD[5] see Section 3.2.

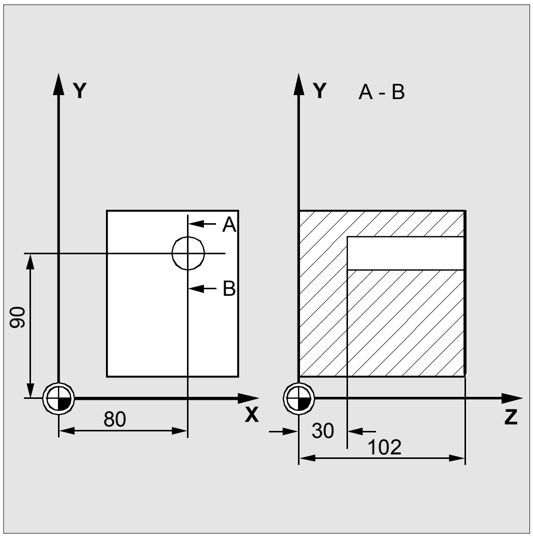


DTB (dwell time)
Parameter DTB is the dwell time at the final drilling depth (chip breaking) in seconds.

SDIR (direction of rotation)
The programmed direction of rotation is active for the movement 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 ZX plane. The boring axis is the Z axis. The safety distance is programmed as 3mm. The final drilling depth is defined as a value relative to the reference plane.
M4 is active in the cycle.



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

-> Must be programmed in a single block

2.1.11 Boring 5 – CYCLE89

Programming

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

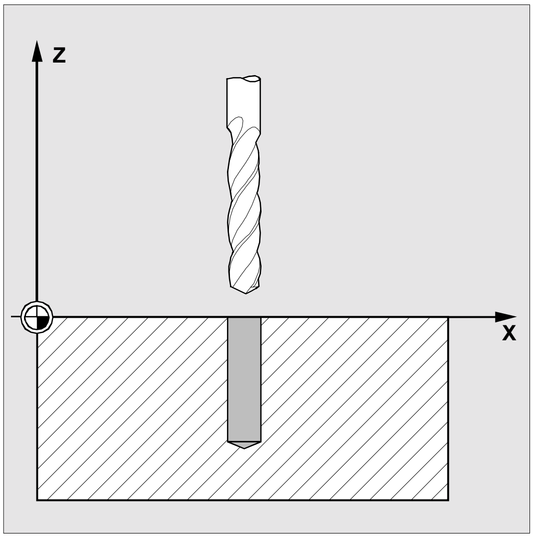
Parameters

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 of operations



Position reached prior to cycle start:
The drilling position is the position in the two axes of the selected plane.

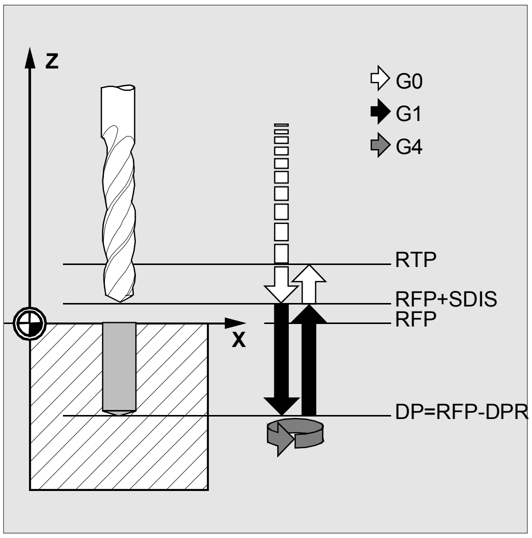
The cycle implements the following motion sequence:

- 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
- Retraction to the reference plane brought forward by the safety distance with G1 and the same feedrate value
- Retraction to retraction plane with G0.

Description of parameters

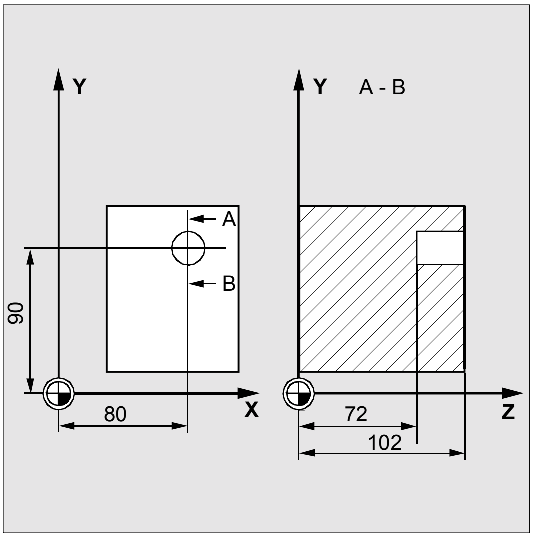
See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, SDIS, DP, DPR

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 5mm 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`	Traverse to 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

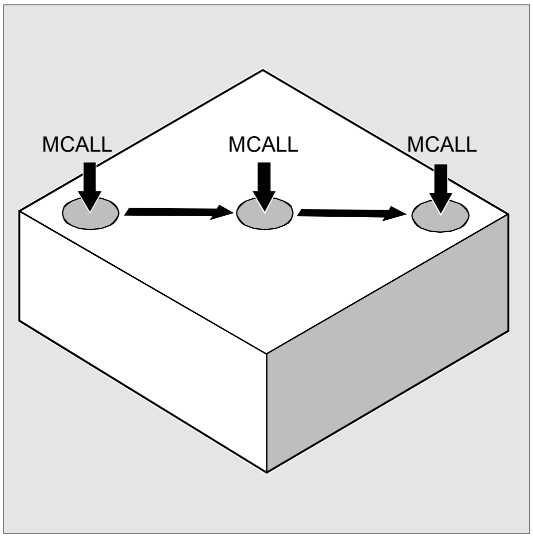
With NC programming it is possible to call any subroutine modally.
This feature is of special importance for drilling cycles.

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, also i.e. maintaining the parameters previous values.
You generate a modal subroutine call by programming the keyword MCALL (modal subroutine call) in front of the subroutine name. This function causes the subroutine to be called and executed automatically after each block that contains traversing movement.
The function is deactivated by programming MCALL without a subroutine name or by a new modal call of another subroutine.

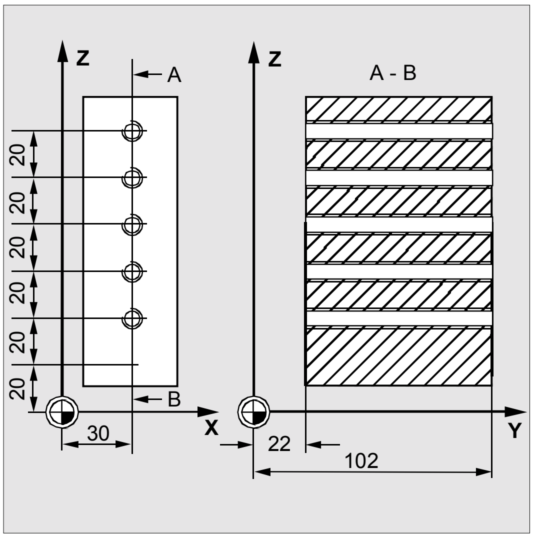
Nesting of modal calls is not permissible, i.e., subroutines that are called modally cannot contain any further modal subroutine calls.

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.

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 20mm. The row of holes starts at Z20 and X30, the first hole in the row being 20mm 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 80mm deep. This is the difference between the reference plane and the final drilling depth.



DEF REAL RFP=102, DP=22, RTP=105, -> -> PIT=4.2, SDIS DEF INT COUNT=1	Definition of parameters with value assignments
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 starting position
N40 MCALL CYCLE81 (RTP, RFP, SDIS, DP)	Modal call of the drilling cycle
N50 MA1: G91 Z20	Traverse to next position (ZX plane) Cycle is executed
N60 COUNT=COUNT+1 N70 IF COUNT<6 GOTOB MA1	Loop for drilling positions along the row of holes
N80 MCALL	Deselect modal call
N90 G90 Y105 Z20	Approach starting position again
N100 COUNT=1	Set counter to zero
N110 ...	Tool change
N120 MCALL CYCLE84 (RTP, RFP, SDIS, -> -> DP , , , 3, , PIT, , 400)	Modal call of tapping cycle
N130 MA2: G91 Z20	Next drilling position
N140 COUNT=COUNT+1 N150 IF COUNT<6 GOTOB MA2	Loop for drilling position of the row of 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

Further notes

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.

In the description of the drilling pattern cycles on the following pages in Section 2.3, the program using these cycles has been adapted and thus simplified. The drilling pattern cycles are based on the call principle.
MCALL DRILLING CYCLE (...)
DRILLING PATTERN (...).

2.3 Drill pattern cycles

The drilling 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 drilling pattern cycle is programmed.

2.3.1 Preconditions

Drilling pattern cycles without drilling cycle call
Drilling pattern cycles can also be used for other applications without the drilling cycle first being called modally because the drilling 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 drilling 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 drilling 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 drilling pattern cycles if they are not expressly declared for a parameter with a description of the response.

2.3.2 Row of holes – HOLES1

Programming

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

Parameters

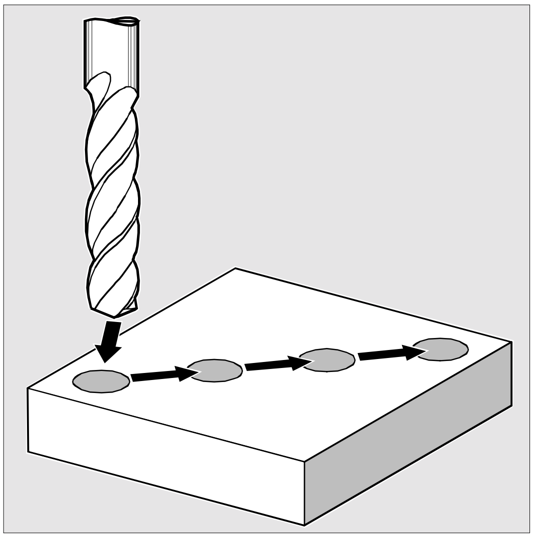
SPCA	real	Abscissa of a reference point on the straight line (absolute)
SPCO	real	Ordinate of this reference point (absolute)
STA1	real	Angle to abscissa Value range –180<STA1<=180 degrees
FDIS	real	Distance between the first hole and the reference point (enter without sign)
DBH	real	Distance between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

Function

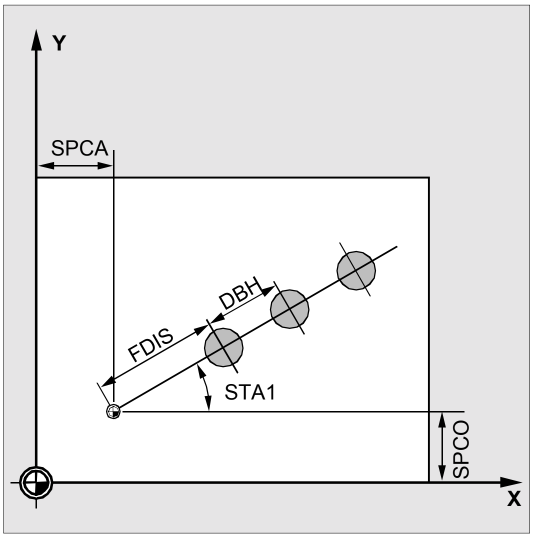
With this cycle you can program a row of holes, i.e. a number of holes that lie along a straight line or a grid of holes. The type of hole is determined by the drilling cycle that has already been called modally.

Sequence of operations

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



Description 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.

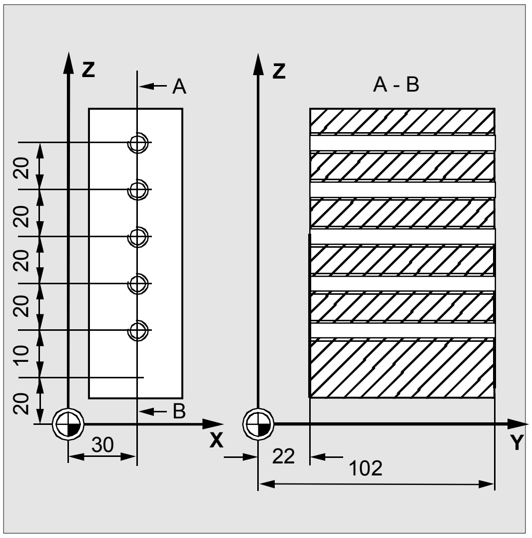
NUM (number)

You determine the number of holes with the parameter NUM.

Programming example

Row of holes

With this program you can machine a row of holes of five tapped holes positioned in parallel to the Z axis on the ZX plane, with a distance between each hole of 20mm. The row of holes starts at Z20 and X30, the first hole in the row being 10mm 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`
`DEF REAL SDIS, FDIS`
`DEF REAL SPCA=30, SPCO=20, STA1=0, ->`
`-> DBH=20`
`DEF INT NUM=5`
	Definition of parameters with value assignments

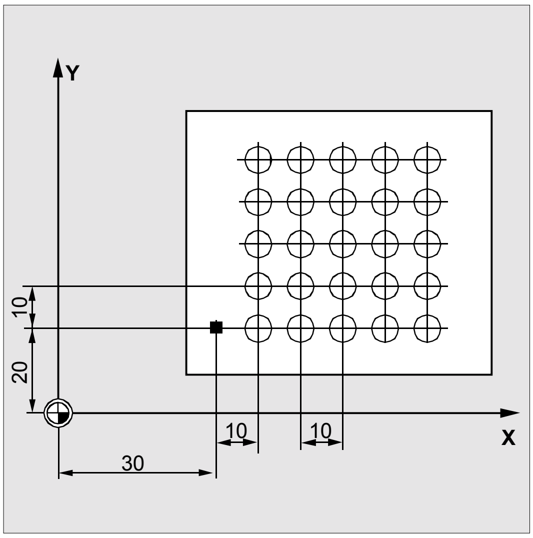
`N10 SDIS=3 FDIS=10`	Value for safety distance and distance of the first hole to the reference point
`N20 G90 F30 S500 M3 D1 T1`	Specification of technology values for the machining section
`N30 G18 G0 Z20 Y105 X30`	Approach starting position
`N40 MCALL CYCLE81 (RTP, RFP, SDIS, DP)`	Modal call of drilling 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
`...`	Tool change
`N70 G90 G0 Z30 Y75 X105`	Traverse to 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 started 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 10mm from one another. The starting point of the grid is X30 Y20.



`DEF REAL RFP=102, DP=75, RTP=105, SDIS=3`
`DEF REAL SPCA=30, SPCO=20, STA1=0, ->`
`-> DBH=10, FDIS=10`
`DEF INT NUM=5, LINENUM=5, COUNT=0`
	Definition of parameters with value assignments

`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 starting position
`N40 MCALL CYCLE81 (RTP, RFP, SDIS, DP)`	Modal call of a drilling cycle
`N50 MARK1: HOLES1 (SPCA, SPCO, STA1, -> -` `> FDIS, DBH, NUM)`	Call of row of holes cycle
`N60 SPCO=SPCO+LINEDIST`	Ordinate of reference point for the next line
`N70 COUNT=COUNT+1` `N80 IF COUNT<LINENUM GOTOB MARK1`	Jump back to MARK1 if the condition is fulfilled
`N90 MCALL`	Deselect modal call
`N100 G90 G0 X=SPCA-10 Y=SPCO Z105`	Approach starting position
`N110 M30`	End of program

-> Must be programmed in a single block

2.3.3 Hole circle – HOLES2

Programming

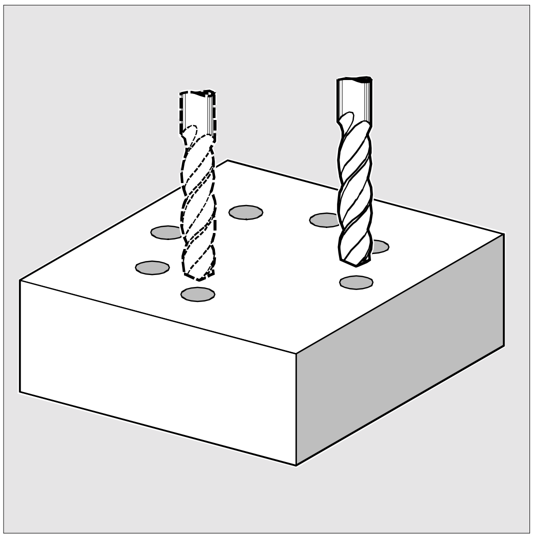
```
HOLES2 (CPA, CPO, RAD, STA1, INDA, NUM)
```

Parameters

`CPA`	real	Center point of circle of holes, abscissa (absolute)
`CPO`	real	Center point of circle of holes, ordinate (absolute)
`RAD`	real	Radius of circle of holes (enter without sign)
`STA1`	real	Initial angle Value range –180<STA1<=180 degrees
`INDA`	real	Indexing angle
`NUM`	int	Number of holes

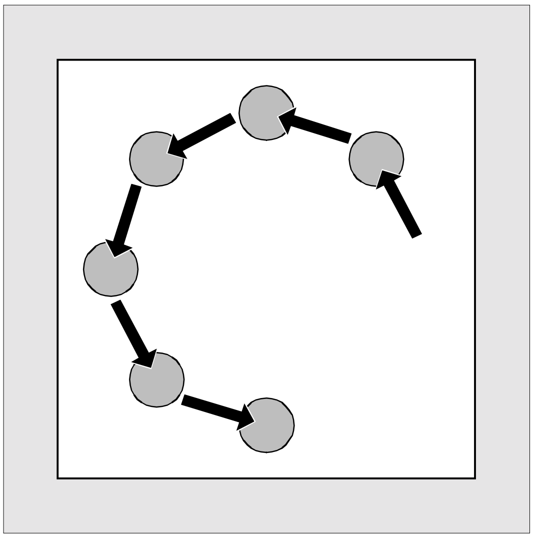
Function

A circle of holes 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.

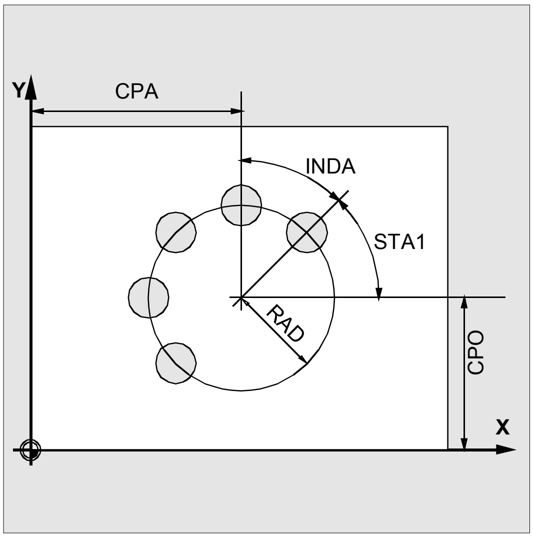


Sequence of operations

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



Description of parameters



CPA, CPO and RAD (center point and radius abscissa, ordinate)
The position of the circle of holes 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.

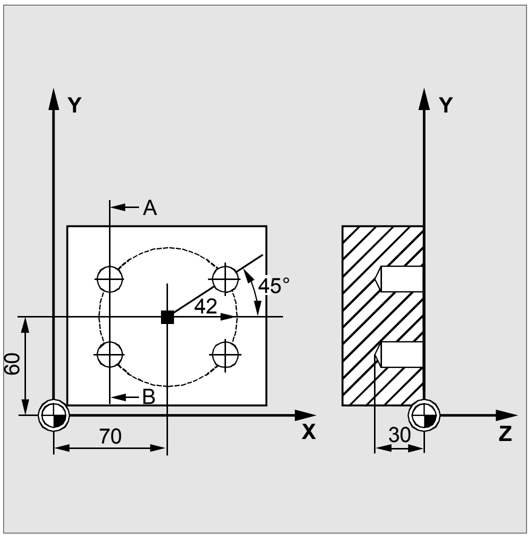
STA1 and INDA (start angle and indexing angle)
The arrangement of the holes in the circle is defined by these parameters.
Parameter STA1 defines the angle of rotation between the positive direction of the abscissa in the coordinate system active before the cycle was called and the first hole. Parameter INDA contains the angle of rotation from one hole to the next.
If parameter INDA is assigned the value zero, the indexing angle is calculated internally from the number of holes which are positioned equally in a circle.

NUM (number)
You determine the number of holes with the parameter NUM.

Programming example

Hole circle

The program uses CYCLE82 4 to produce holes with a depth of 30mm. 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 42mm in the XY plane. The initial angle is 45 degrees.
The safety distance in the drilling axis Z is 2mm.



`DEF REAL CPA=70,CPO=60,RAD=42,STA1=45` `DEF INT NUM=4`	Definition of parameters with value assignments
`N10 G90 F140 S710 M3 D1 T40`	Specification of technology values
`N20 G17 G0 X50 Y45 Z2`	Approach starting position
`N30 MCALL CYCLE82 (2, 0,2, , 30)`	Modal call of drilling cycle, without dwell time, DP is not programmed
`N40 HOLES2 (CPA, CPO, RAD, STA1, , NUM)`	Call of circle of holes 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 (SW 5.3 and higher)

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, abscissa (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

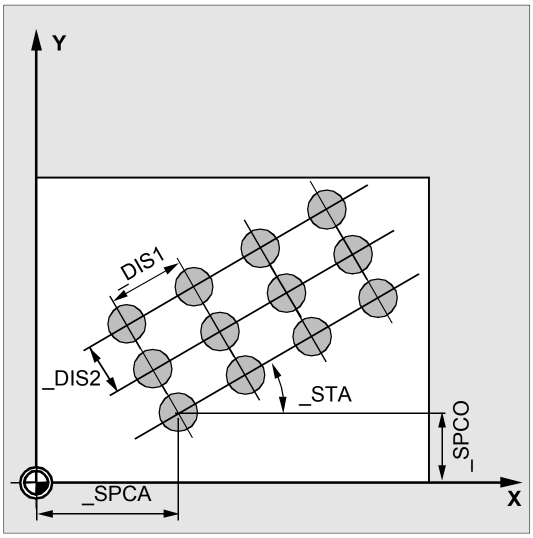
Function

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

Sequence of operations

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.
Starting positions are one of the four possible corner positions in each case.

Description of parameters



_SPCA and _SPCO (reference point abscissa and ordinate)

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

_STA (angle)

The grid of holes can 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)

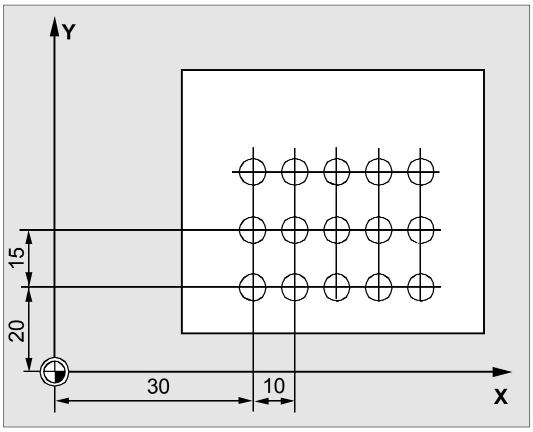
The distances must be entered without sign. To avoid unnecessary empty travel, the dot matrix is machined line by line or column by column based on a comparison of distance measurements.

_NUM1 and _NUM2 (number)

This parameter determines the number of columns or lines.

Programming example

Cycle CYCLE801 is used to machine a dot matrix, consisting of 15 holes 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 information

The following sections describe how milling cycles are programmed.

This section is intended to guide you in selecting cycles and assigning them with parameters. In addition to a detailed description of the function of the individual cycles and the corresponding parameters, you will also find a sample program at the end of each section to familiarize you with the use of cycles.

The Sections are structured as follows:

- **Programming**
- **Parameters**
- **Function**
- **Sequence of operations**
- **Explanation of parameters**
- **Additional notes**
- **Sample program.**

"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 of operations", "Explanation of parameters", "Additional notes" and the "Sample program".

3.2 Preconditions

Programs required in the control

The milling cycles call the programs

- MESSAGE.SPF and
- PITCH.SPF

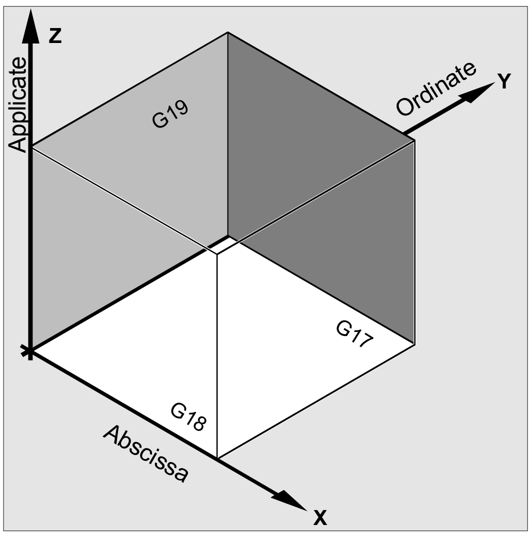
internally as subroutines. Moreover, you need the data block GUD7.DEF and the macro definition file SMAC.DEF.
Load them in the parts program memory of the control unit before executing the milling cycles.

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 right-handed coordinate system.
The G functions and current programmable frame active before the cycle was called remain active beyond the cycle.

Plane definition

In milling cycles, it is generally assumed that the current workpiece coordinate system is defined by selecting plane G17, G18 or G19 and activating a programmable frame (if necessary). The infeed axis is always the 3rd axis of the coordinate system (see also Programming Guide).



Spindle programming

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 with the command SETMS (see also Programming Guide).

Machining status messages

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

The following messages might be displayed:

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

A few parameters of milling cycles (SW 4 and higher) and their behavior can be modified by cycle settings.

The cycle setting data are defined in data block GUD7.DEF.

The following new cycle setting data are introduced:

_ZSD[x]	Value	Meaning	Cycles affected
_ZSD[1]	0	Depth computation in the new cycles is made between the reference plane + safety distance and depth (_RFP + _SDIS - _DP)	POCKET1 to POCKET4, LONGHOLE, CYCLE71, SLOT1, CYCLE72, SLOT2
	1	Depth computation is made without including safety distance	
_ZSD[2]	0	Dimension of rectangular pocket or rectangular spigot from the center point	POCKET3 CYCLE76
	1	Dimension of rectangular pocket or rectangular spigot from a corner	
_ZSD[5]	0	Execute at drilling depth M5 M0	CYCLE88
	1	Execute at drilling depth M5	

3.3 Thread cutting – CYCLE90

Programming

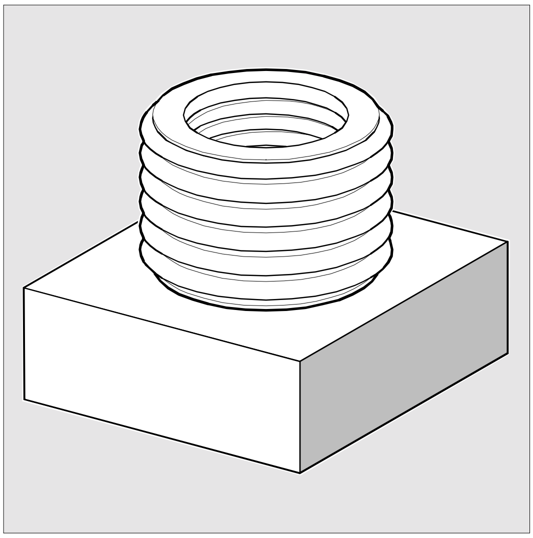
```
CYCLE90 (RTP, RFP, SDIS, DP, DPR, DIATH, KDIAM, PIT, FFR, CDIR, TYPTH,
CPA, CPO)
```

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 ... 2000.000mm
FFR	real	Feedrate for thread milling (enter without sign)
CDIR	int	Direction of rotation for thread milling Value: 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 geometrical axes of the current plane which you define before calling the cycle are involved in this movement.

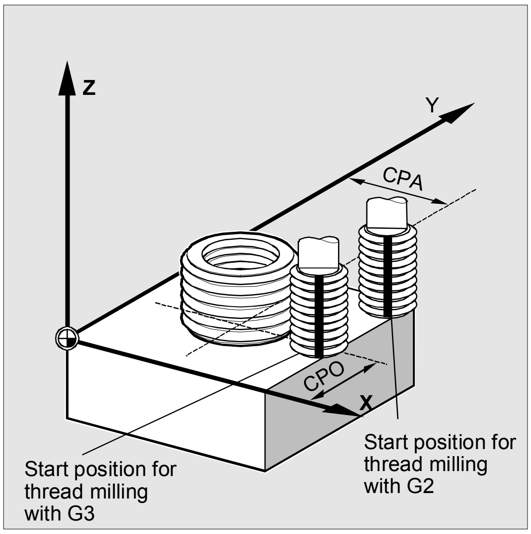


The programmed feedrate F depends on the axis grouping defined in the FGROUP instruction before the cycle call
(see Programming Guide).

Sequence of operations

Outside threads

Position reached prior to cycle start:
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
- Movement to the thread diameter along a circular path in the direction G2/G3 opposite to that 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 prior to cycle start:
This can be any position from which the starting position on 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
- 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, the depth data must be programmed as absolute values and before cycle start, the machine must be positioned on the retraction plane or one position behind the retraction plane.

Programming example

(thread from bottom to top)

A thread must be cut starting from –20 up to 0 with a 3mm 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 at least a depth of –21.5 (half pitch in excess).

Overshoot in the thread longitudinal direction

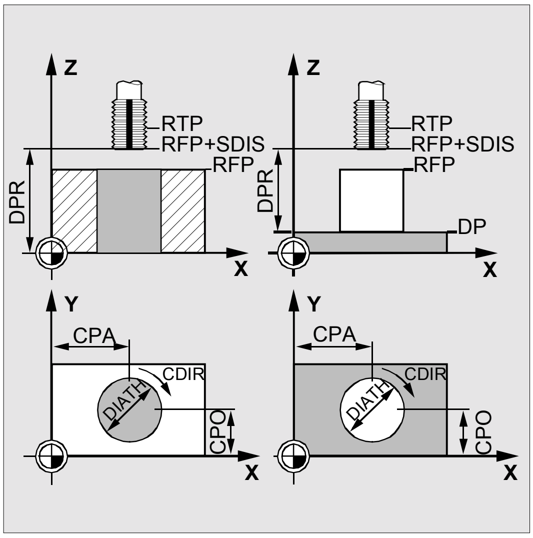
For thread milling, the travel-in and travel-out movements occur along all three axes concerned. This means that the travel-out movement includes a further step in the vertical axis, beyond the programmed thread depth.
The overshoot is calculated:

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

Δz	Overshoot, internal
p	Thread pitch
WR	Tool radius
DIATH	Outside diameter of the thread
RDIFF	Radius difference for travel-out

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

Description of parameters



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

DIATH, KDIAM and PIT (nominal diameter, core diameter and thread pitch)
With these parameters you define the thread data such as, nominal diameter, core diameter and pitch. 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 message
"Wrong milling direction, G3 will be generated" is output.
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.

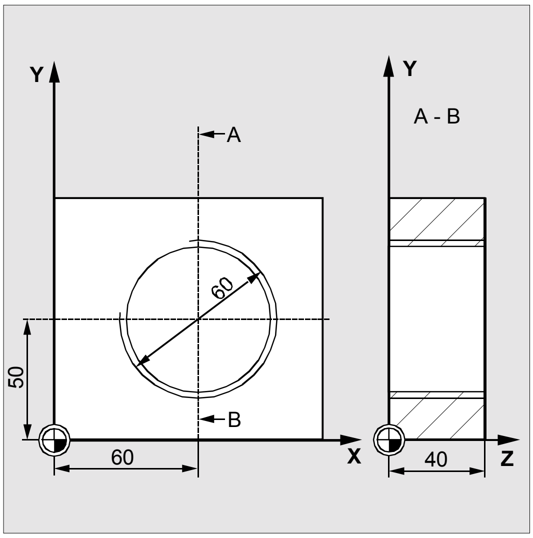
CPA and CPO (center point)
With these parameters you define the center point of the hole or spigot on which the thread is to be machined.

Further notes

The milling cutter radius is taken into account by the cycle. A tool offset must therefore be programmed before the cycle is called. Otherwise alarm 61000 "No tool offset active" is output 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.

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, ->`
`-> DPR=40, DIATH=60, KDIAM=50`
`DEF REAL PIT=2, FFR=500, CPA=60,CPO=50`
`DEF INT CDIR=2, TYPTH=0`
	Definition of variables with value assignment

`N10 G90 G0 G17 X0 Y0 Z80 S200 M3`	Approach starting position
`N20 T5 D1`	Specification of technology values

`N30 CYCLE90 (RTP, RFP, SDIS, ->`
`-> DPR, DIATH, KDIAM, PIT, FFR, CDIR,`
`TYPTH, CPA, CPO)`
	Cycle call

`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)
```

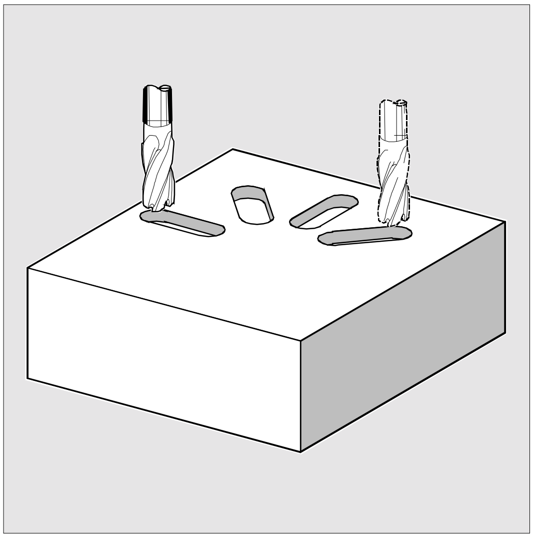
Parameters

`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`	int	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 most optimum path. 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.

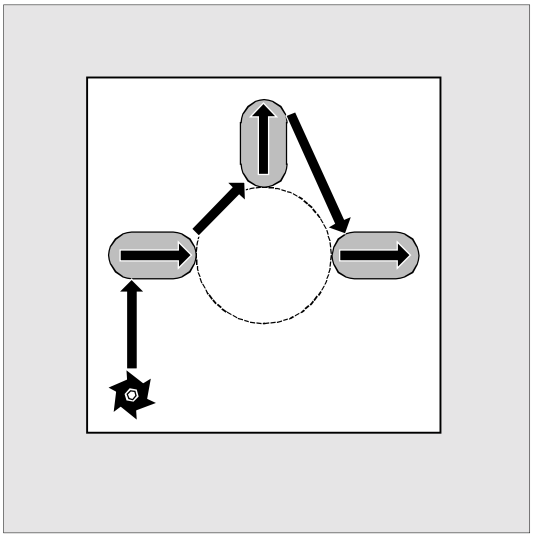


Sequence of operations

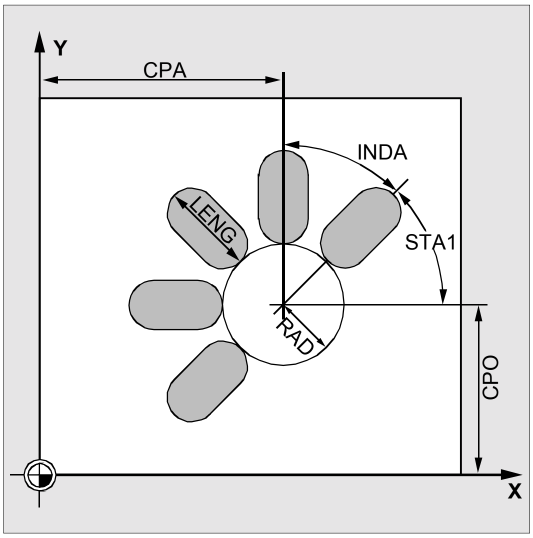
Position reached prior to cycle start:
The starting position can be any position from which each of the elongated holes can be approached without collision.

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.



Description of parameters



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

DP and DPR (elongated hole depth)
The elongated hole 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 us determined with the parameter NUM.

LENG (elongated hole length)
The elongated hole length is programmed under LENG. If it is detected during the cycle run 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 0.5x 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 moved 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.

STA1 and INDA (start 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.

Further notes

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

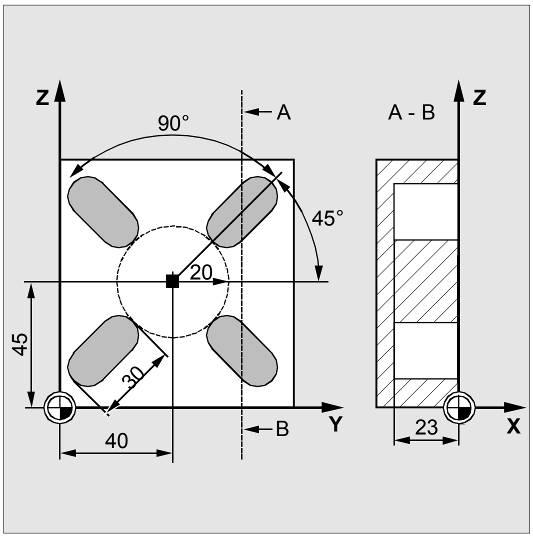
If incorrect values are assigned to the parameters that determine the arrangement and size of the elongated holes and thus cause mutual contour violation of the elongated holes, the cycle is not started. The cycle is aborted after the error message 61104 "Contour violation of slots/elongated holes" is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

During the cycle, the workpiece coordinate system is shifted and rotated. The values in the workpiece coordinate system are displayed on the actual value display as if the longitudinal axis of the elongated hole being machined were 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 it was before the cycle was called.

Programming example

Machining elongated holes

With this program you can machine four elongated holes 30mm in length and with a relative depth of 23mm (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 20mm in the YZ plane. The initial angle is 45 degrees, the indexing angle is 90 degrees. The maximum infeed depth is 6mm, the safety distance is 1mm.



`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	int	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	int	Milling direction for machining the slot Value: 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	int	Machining type (enter without sign) UNITS DIGIT: Value: 0...Complete machining 1...Roughing 2...Finishing TENS DIGIT: Value: 0...Perpendicular with G0 1...Perpendicular with G1 3...Oscillation with G1
MIDF	real	Maximum infeed depth for finishing
FFP2	real	Feedrate for finishing

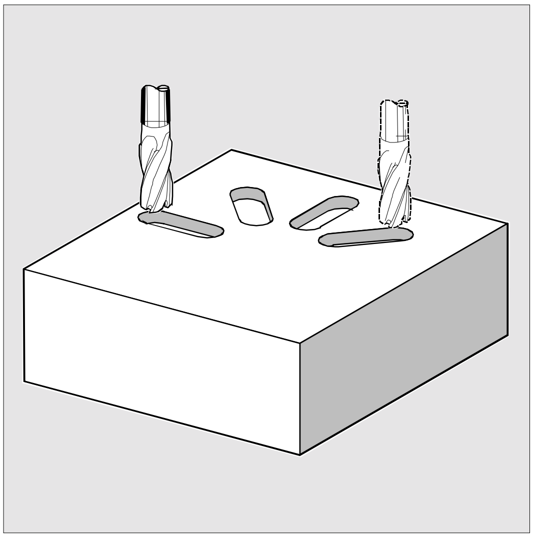
SSF	real	Speed for finishing
_FALD	real	Final machining allowance on the base of slot
_STA2	real	Maximum insertion angle for oscillation movement

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

Function

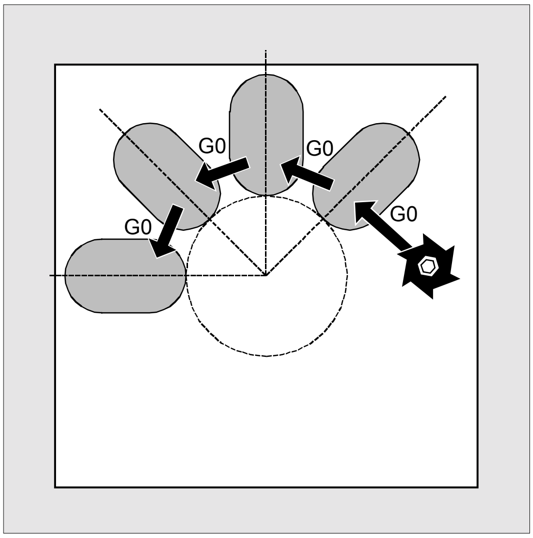
Cycle SLOT1 is a combined roughing-finishing cycle.

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.



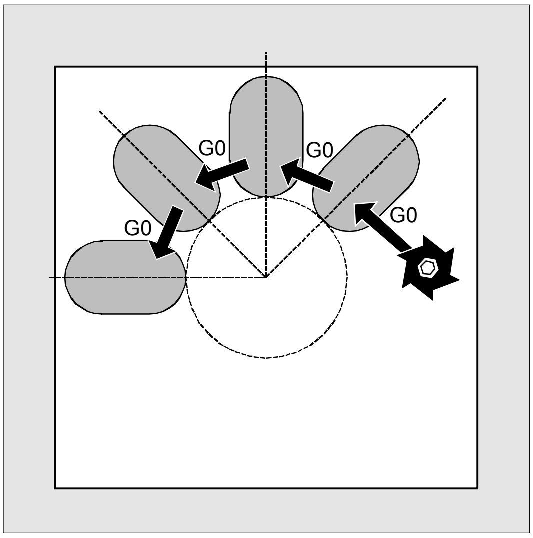
Sequence of operations

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.

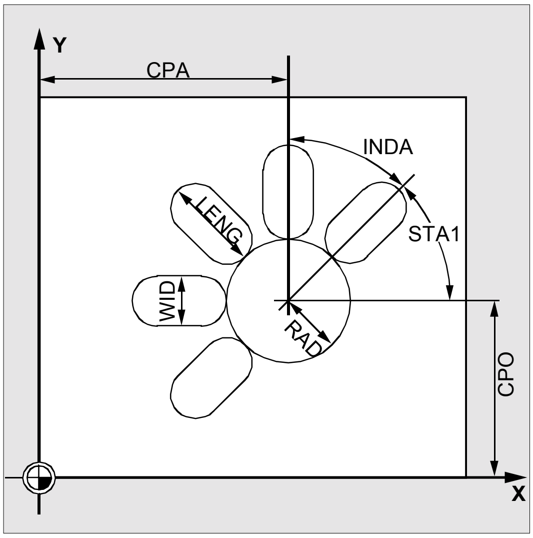


The cycle implements the following motion sequence:

- Travel to 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 feed value FFD.
  - Solid machining of the slot to the final machining allowance on slot base and slot edge at feed value FFP1.
    Subsequent finishing at feed 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 so 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.
- At the end of machining the last slot the tool moves to the retraction plane with G0 and the cycle ends.



Description of parameters



See Section 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, 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" will be 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 circle of holes 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 (start 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 arranged equally around the circle.

FFD and FFP1 (feedrate depth and plane)

Feedrate FFD is operative for vertical infeed to the machining plane with G1 and for insertion with oscillation 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 0.5x 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.

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

CDIR (milling direction)

You define the slot machining direction in this parameter.

Under parameter _CDIR the mill direction

- direct "2 for G2" and "3 for G3" or
- alternatively "up-cut milling" or "down-cut milling"

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

Up-cut milling	**Down-cut milling**
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 allowed 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.
  - Removing stock for the remaining final machining allowance takes place at the spindle speed preset by SSF and feed FFP2. The depth infeed takes place with vertical indeed via MIDF. If MIDF=0, the infeed is equal to the final depth.
    If FFP2 is not programmed, feed FFP1 acts. Similarly, if there is no message from SSF, i.e. the speed programmed before cycle start is active.
- 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) is already machined to a remaining final machining allowance and that it is only necessary to machine the final machining allowance. If FFP2 and SSF are not programmed, the feedrate FFP1 or the speed programmed before the cycle call is active. Depth infeed takes place with via MIDF with paraxial infeed.
  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 the parameter VARI, the cycle aborts after output of the alarm 61102 "Operating mode not defined correctly".

Cutter diameter=slot width (WID)

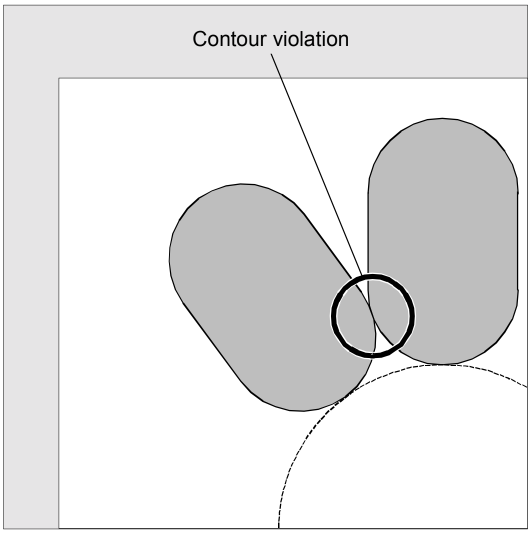
- With complete machining, finishing machining only takes place on the base.
- In machining mode VARI=32 positioning is paraxial in Z with G1 followed by finish-cutting (infeed possible via MIDF).

_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 oscillation motion.

- **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.
- **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 has reached 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.

Further notes

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 and thus cause 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".

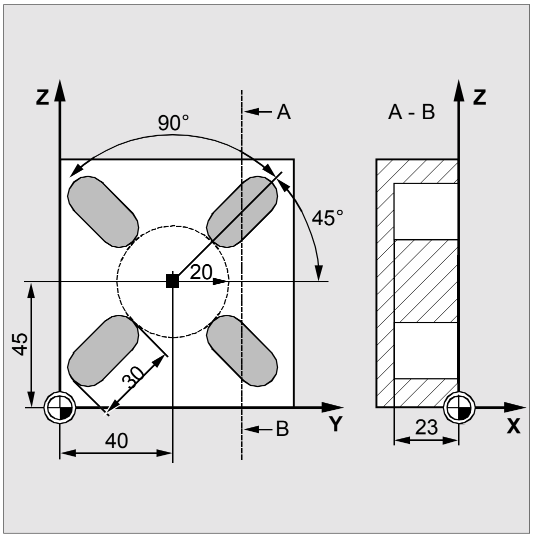
During the cycle, the workpiece coordinate system is shifted and rotated. The values in the workpiece coordinate system displayed on the actual value display are such that the longitudinal axis of the slot that has just been machined corresponds to the first axis of the current machining plane.

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 30mm, width 15mm and depth 23mm. The safety distance is 1mm, the final machining allowance is 0.5mm, the milling direction is G2, the maximum infeed in the depth is 10mm.
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 starting position
N30 SLOT1 (5, 0, 1, -23, , 4, 30, 15, -> ->40, 45, 20, 45, 90, 100, 320, 10, -> ->2, 0.5, 30, 10, 400, 1200, 0.6, 5)	Cycle call, parameters VARI, MIDF, FFP2 and SSF omitted
N40 M30	End of program

-> Must be programmed in a single block

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)
```

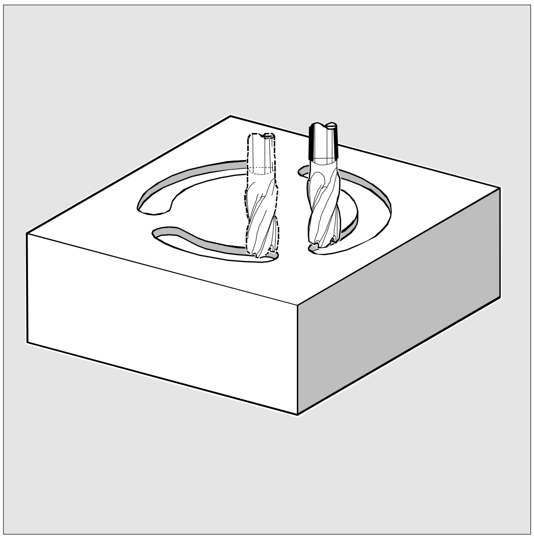
Parameters

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	int	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	int	Milling direction for machining the circumferential slot Value: 2 (for G2) 3 (for G3)
FAL	real	Final machining allowance on slot edge (enter without sign)
VARI	int	Type of machining UNITS DIGIT: Values: 0=Complete machining 1=Roughing 2=Finishing TENS DIGIT (SW 6.3 and higher) Value: 0=positioning from slot to slot in a straight line with G0 1=positioning from slot to slot on 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.

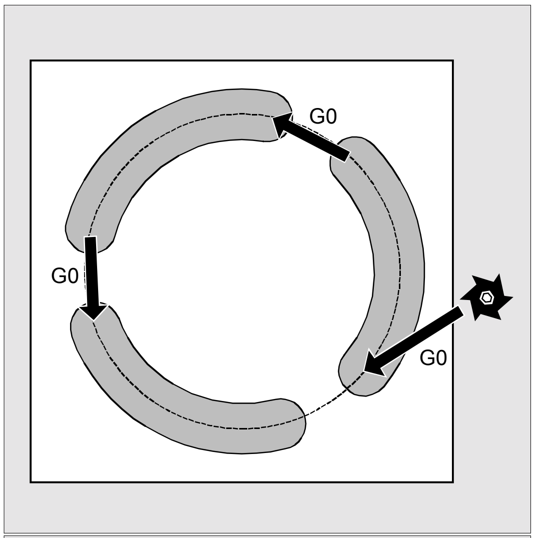


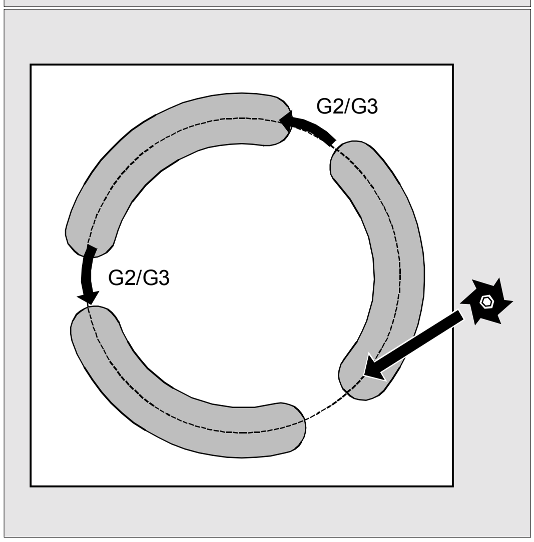
Sequence of operations

Position reached prior to cycle start:
The starting position can be any position from which each of the slots can be approached without collision.

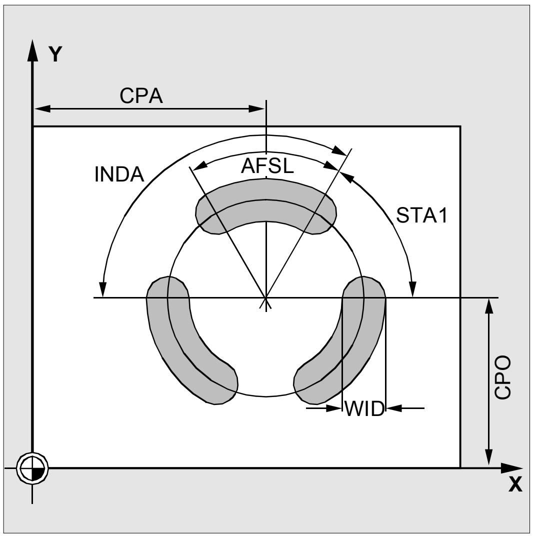
The cycle implements the following motion sequence:

- Travel to 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 reached in the machining plane specified in the display to the retraction plane with G0 and the cycle is terminated.





Description of parameters



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

See Section 3.5 (SLOT1) for a description of parameters DP, DPR, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, 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 with 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 (start 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 arranged equally around the circle.

New machining types in SW 6.3 and higher:

Finishing the edge only (VARI=x3)

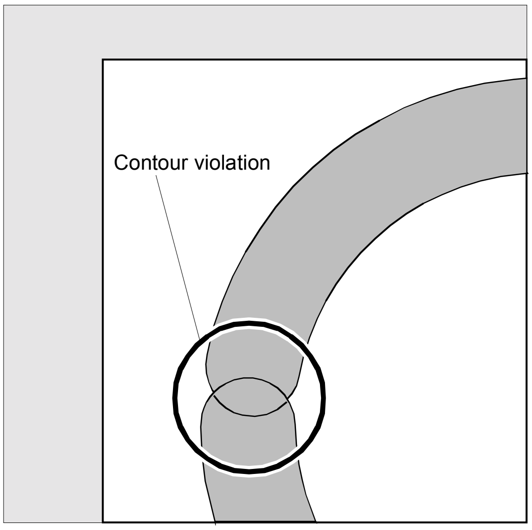
- There is a new selection "Finishing the edge". This is the only machining type where the diameter of the cutter is allowed to be less than half the slot width. There is no checking of whether it is sufficiently large to execute the final machining allowance FAL.
- A number of depth infeeds are possible. 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)

- Especially 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, that does not allow one slot to be positioned directly 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.

Further notes

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

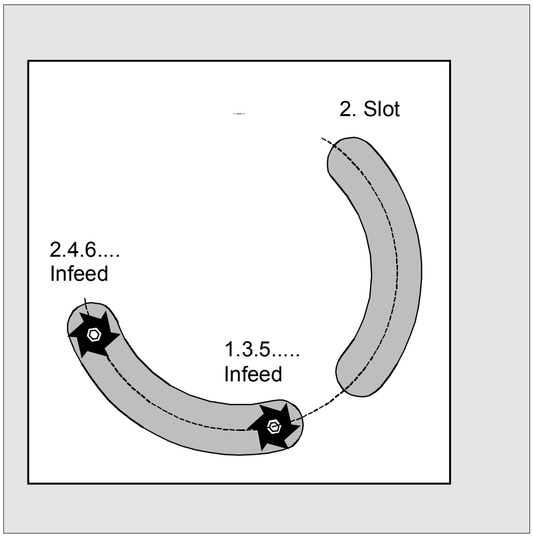


If incorrect values are assigned to the parameters that determine the arrangement and size of the slots and thus cause mutual contour violation of the slots, the cycle is not started.
The cycle is aborted after the error message 61104 "Contour violation of slots/elongated holes" is output.

During the cycle, the workpiece coordinate system is shifted and rotated. The actual-value display in the workpiece coordinate system is always displayed such that the circumferential slot currently being machined on the 1st axis of the current processing level starts and the zero point of the workpiece coordinate system lies in the center of the circle. When the cycle is completed, the workpiece coordinate system is again in the same position as it was before the cycle was called.

Special case: Slot width = cutter diameter

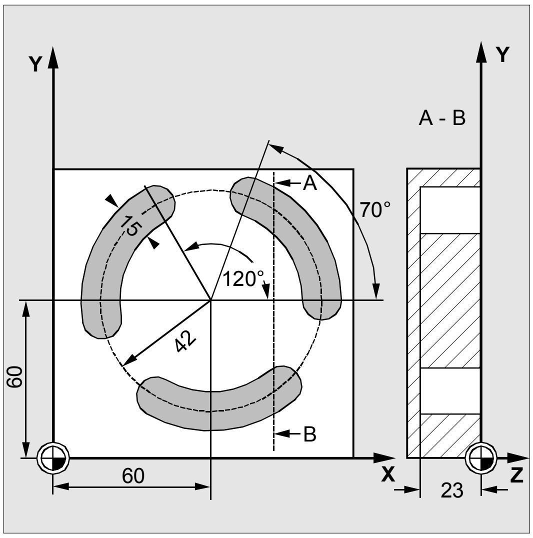
- The machining case of slot width = cutter diameter is allowed for roughing and finishing. This machining case occurs if
  slot width WID – 2 * final machining allowance FAL = cutter diameter.
- The traversing strategy is then as for the LONGHOLE cycle, i.e. depth infeed takes place alternately at the reversal points, see graphic.



Programming example

Slots2

With this program you can machine three circumferential slots arranged on a circle whose center point is X60 Y60 and radius 42mm in the XY plane. The circumferential slots have the following dimensions: Width 15mm, angle for slot length 70 degrees, depth 23mm. The initial angle is 0 degrees, the indexing angle is 120 degrees. The slot contours are machined to a final machining allowance of 0.5mm, the safety distance in infeed axis Z is 2mm, the maximum depth infeed is 6mm. 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

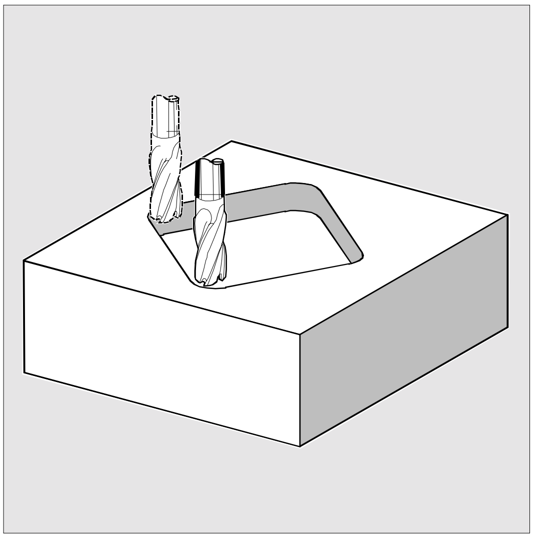
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Pocket depth (absolute)
DPR	real	Pocket depth relative to the reference plane (enter without sign)
LENG	real	Pocket length (enter without sign)
WID	real	Pocket width (enter 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<=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	int	Milling direction for machining the pocket Value: 2 (for G2) 3 (for G3)
FAL	real	Final machining allowance on pocket edge (enter without sign)
VARI	int	Type of machining Value: 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).

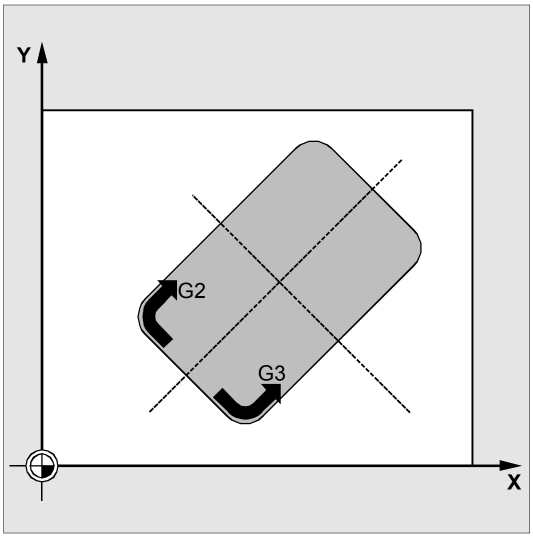
The pocket milling cycle POCKET3 can be performed with any tool.

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 of operations



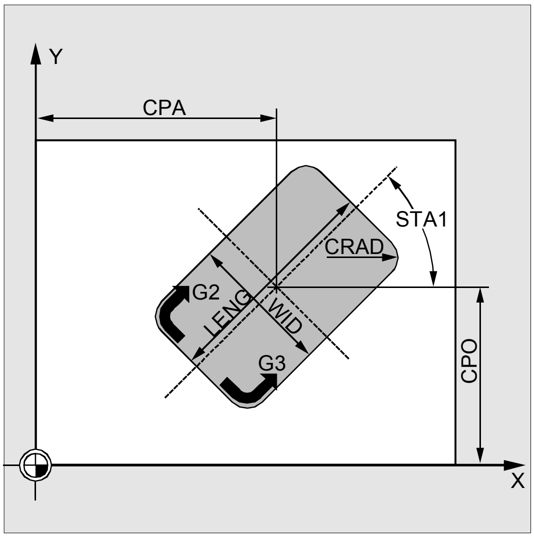
Position reached prior to cycle start:
This can be any position from which the starting position on the center point of the pocket at the retraction plane level can be approached without collision.

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, with G0 the reference plane brought forward by the safety distance is approached. 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 allowance along the contour at 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.

Description of parameters



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

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

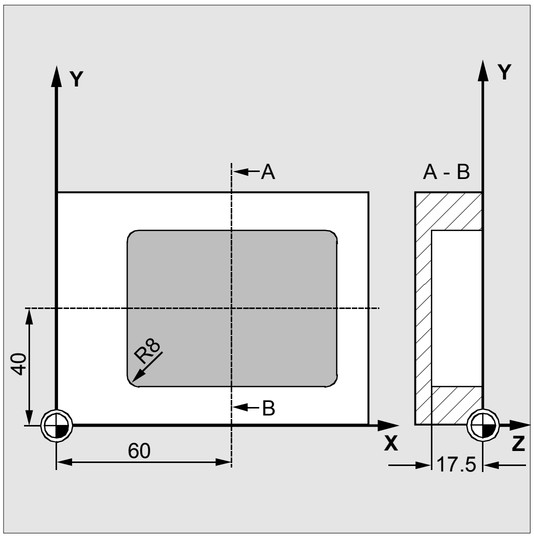
STA1 (angle)
STA1 defines the angle between the positive abscissa and the longitudinal axis of the pocket.

Further notes

A tool offset must be activated before the cycle is called. Otherwise the cycle is aborted and alarm 61000 "No tool offset active" is output.
A new 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 60mm long, 40mm wide, 17.5mm deep (difference between the reference plane and the base of the pocket) and which has a corner radius of 8mm in the XY plane. The angle to the X axis is 0 degrees. The final machining allowance of the pocket edges is 0.75mm, the safety distance in the Z axis, which is added to the reference plane, is 0.5mm. The center point of the pocket lies at X60 and Y40, the maximum depth infeed is 4mm.
Only roughing is to be performed.



`DEF REAL LENG, WID, DPR, CRAD` `DEF INT VARI`	Definition of variables
`N10 LENG=60 WID=40 DPR=17.5 CRAD=8` `N20 VARI=1`	Value assignments
`N30 G90 S600 M4`	Specification of technology values
`N35 T20 D2`	
`N37 M6`	
`N40 G17 G0 X60 Y40 Z5`	Approach starting position

`N50 POCKET1 (5, 0, 0.5, , DPR, ->`
`-> LENG, WID, CRAD, 60, 40, 0, ->`
`-> 120, 300, 4, 2, 0.75, VARI)`
	Cycle call
	Parameters MIDF, FFP2 and SSF are 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

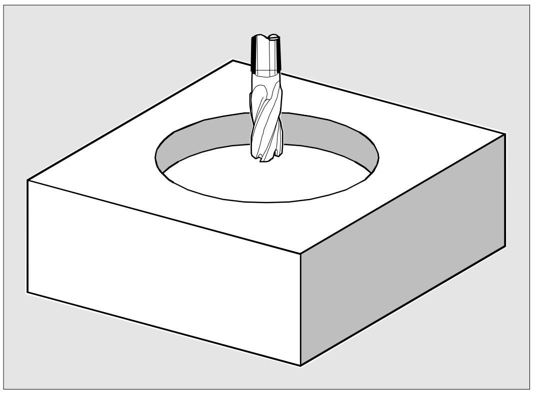
RTP	real	Retraction plane (absolute)
RFP	real	Reference plane (absolute)
SDIS	real	Safety distance (enter without sign)
DP	real	Pocket depth (absolute)
DPR	real	Pocket depth relative to the reference plane (enter without sign)
PRAD	real	Pocket radius (enter without sign)
CPA	real	Pocket center point, 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	int	Milling direction for machining the pocket Value: 2 (for G2) 3 (for G3)
FAL	real	Final machining allowance on pocket edge (enter without sign)
VARI	int	Type of machining Value: 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.

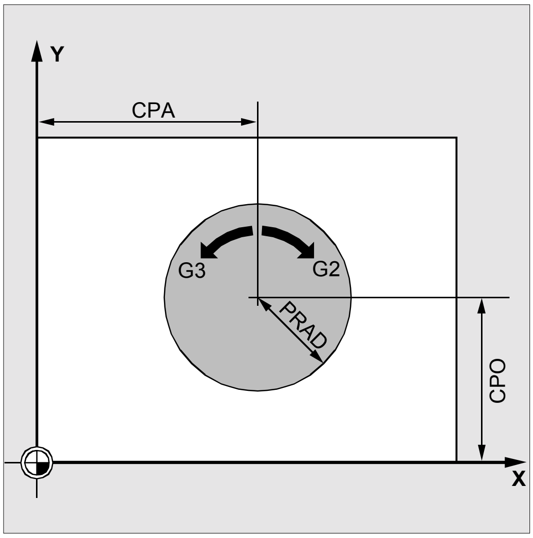


Sequence of operations

Position reached prior to cycle start:
This can be any position from which the starting position on 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, with G0 the reference plane brought forward by the safety distance is approached. 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.
  - Final machining 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.



Description of parameters

See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, SDIS.
See Section 3.7 for a description of parameters DP, DPR.

See Section 3.5 (SLOT1) for a description of parameters FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, 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 "Milling cutter radius too large" is output.

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

Further notes

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

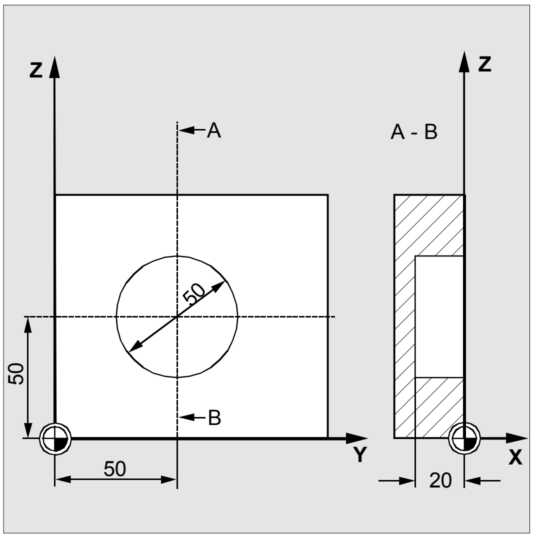
The depth infeed is always made in the pocket center point. It can be useful to drill there beforehand.

A new 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, 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,->`
`-> PRAD=25, FFD=100, FFP1, MID=6`
`N10 FFP1=FFD*2`
	Definition of variables with value assignment

`N20 G19 G90 G0 S650 M3`	Specification of technology values
`N25 T10 D1`	
`N27 M6`	
`N30 Y50 Z50`	Approach starting position
`N40 POCKET2 (RTP, RFP, , DP, , PRAD, ->` `-> 50, 50, FFD, FFP1, MID, 3, )`	Cycle call 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 in 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

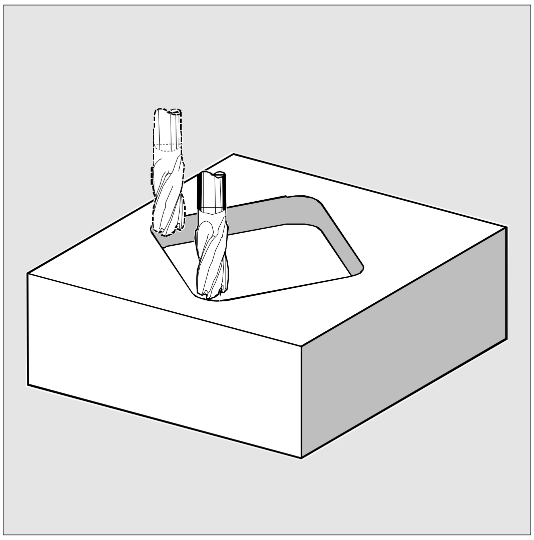
The following input parameters are always required:

_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° ≤ _STA < 180°
_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	int	Milling direction: (enter without sign) Value: 0...Down-cut milling (as spindle rotation) 1...Up-cut milling 2...with G2 (independent of spindle direction) 3...with G3
_VARI	int	Type of machining: (enter without sign) UNITS DIGIT: Value: 1...Roughing 2...Finishing

		TENS DIGIT: Value: 0...Perpendicular to pocket center with G0 1...Perpendicular to pocket center with G1 2...Along a helix 3...Oscillating 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 oscillating motion
_DP1	real	Insertion depth per 360° revolution on insertion along helical path

Function

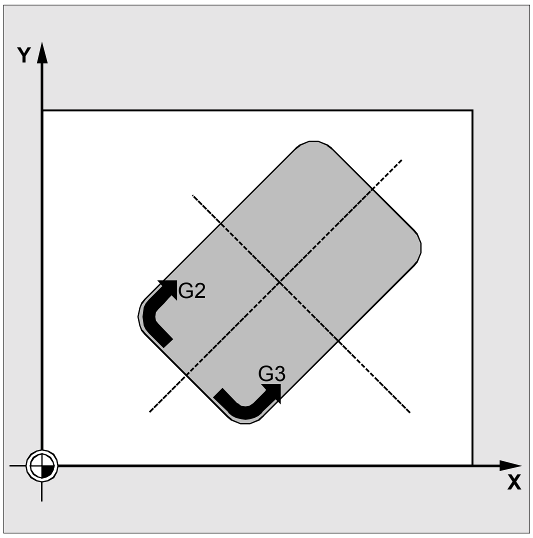
The cycle can be applied to roughing and finishing. For finishing, a face cutter is needed.
The depth infeed will always start at the pocket center point and be performed vertically from there; thus predrill can be suitably performed in this position.



New functions compared to POCKET1:

- The milling direction can be defined with a G instruction (G2/G3) or up-cut milling or down-cut from the spindle direction
- For solid machining, the maximum infeed width in the plane is programmable
- Final machining allowance for the pocket base
- Three different insertion strategies:
  - Vertically at the pocket center point
  - Along a helical path around the pocket center
  - Oscillating around the pocket central axis
- Shorter 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 of operations



Position reached prior to cycle start:
This can be any position from which the starting position on the center point of the pocket at the retraction plane level can be approached without collision.

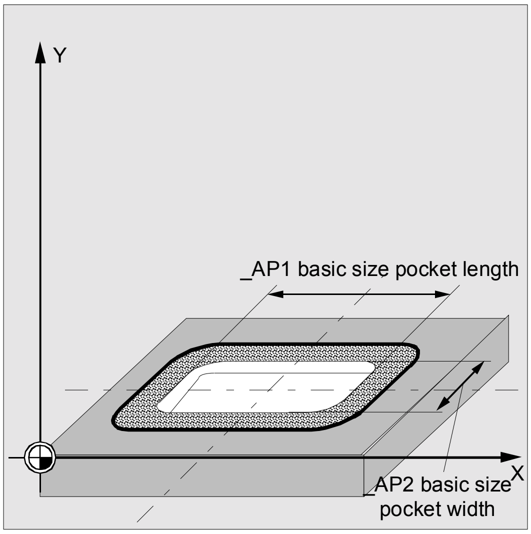
Motion sequence when roughing (VARI=X1):
With G0, the pocket center point is approached at the retraction plane level and then, from this position, with G0 the reference plane brought forward by the safety distance is approached. Pocket machining is then performed according to the selected insertion strategy and considering 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 (≤ 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.
  Then pocket solid machining starts in this plane and continues until reaching the final machining allowance.
  The starting point of the helical path described is on the pocket longitudinal axis in the "plus direction" and reached 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 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.

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 size 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.
Depth infeed to account for workpiece sizes is carried out according to the programmed type (helical path, oscillating, vertical). If the cycle recognizes that by means of the blank contour and the radius of the active tool there is enough room in the pocket center, infeed takes place as long as possible vertically downwards to the pocket center in order to avoid time-consuming approach paths in the open.
The pocket is solid machined beginning from the top and proceeding in the downward direction.

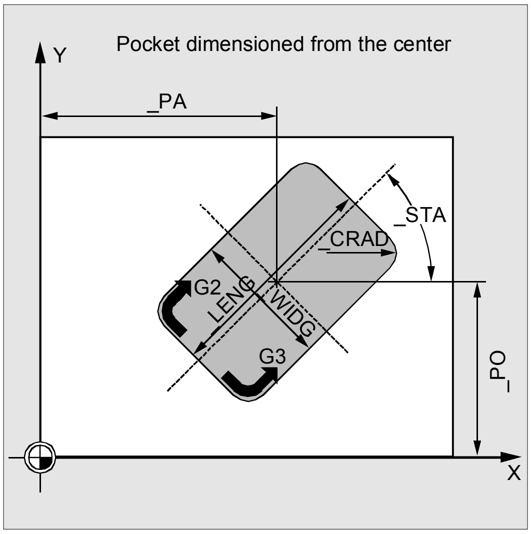
Motion sequence when finishing (VARI=X2)

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

- Finishing on 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 2mm or, if "less room" is available, equals the difference between the corner radius and the mill radius.
  If the final machining allowance on the edge is larger than 2 mm, the approach radius is increased accordingly.
  The depth infeed is performed with G0 in the open towards the pocket center and the starting point of the approach path is also reached with G0.
- Finishing on the base
  During finishing on the base, the machine performs G0 towards the pocket center until reaching a distance equal to pocket depth + final machining allowance + safety distance. From this point onwards, 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.

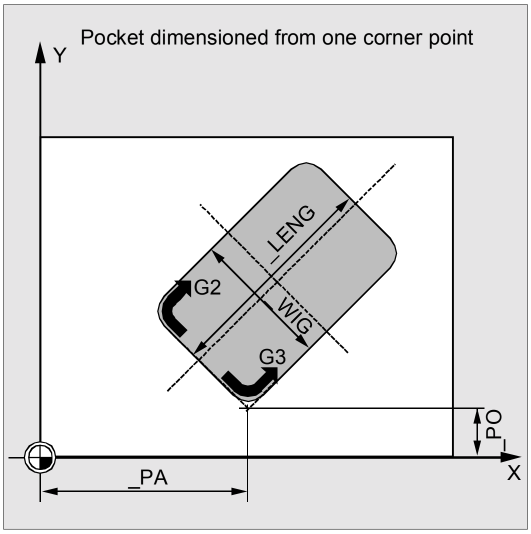
Description of parameters

See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters _RTP, _RFP, _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 one 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 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.



_PA, _PO (reference point)

The center 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 must be entered with sign (_LENG, _WID), thus completely 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 ≥ tool diameter, the pocket will not necessarily be machined completely. The message

"Caution: Final machining allowance ≥ tool diameter"

is output but the cycle is continued.

_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).

_FFD and _FFP1 (infeed 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 (milling direction)

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

Under parameter _CDIR the mill direction

- direct "2 for G2" and "3 for G3" or
- alternatively "up-cut milling" or "down-cut milling"

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

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

_VARI (machining mode)
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 percent of the mill diameter as maximum infeed width.

Further notes

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 horizontal and vertical planes.

_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 oscillation.

_DP1 (insertion depth)
With the parameter _DP1 you define the infeed depth for insertion on the helical path.

Further notes

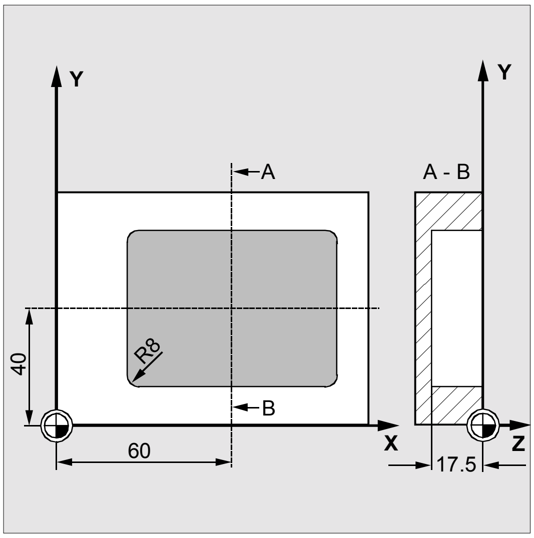
A tool offset must be activated before the cycle is called. Otherwise the cycle is aborted and alarm 61000 "No tool offset active" is output.

A new 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 60mm long, 40mm wide, 17.5mm deep in the XY plane, and which has a corner radius of 8mm. The angle in relation to the X axis is 0 degrees. The final machining allowance of the pocket edges is 0.75mm, 0.2mm at the base, the safety distance in the Z axis, which is added to the reference plane, is 0.5mm. The center point of the pocket lies at X60 and Y40, the maximum depth infeed is 4mm.
Up-cut milling uses the spindle rotation direction as 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 starting 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.

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:

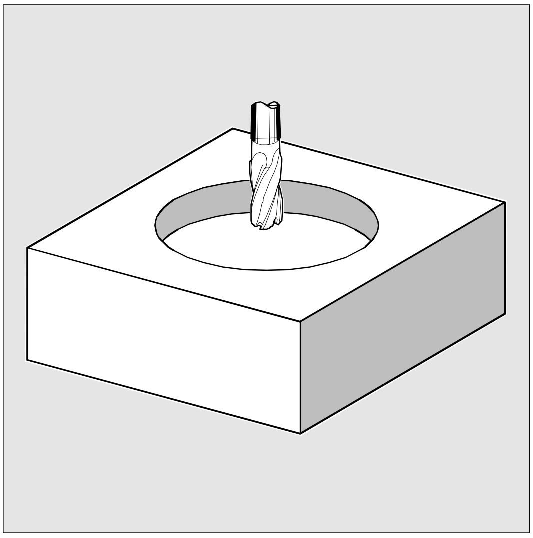
_RTP	real	Retraction plane (absolute)
_RFP	real	Reference plane (absolute)
_SDIS	real	Safety distance (to be added to the reference plane, enter without sign)
_DP	real	Pocket depth (absolute)
_PRAD	real	Pocket radius
_PA	real	Pocket center point, abscissa (absolute)
_PO	real	Pocket center point, ordinate (absolute)
_MID	real	Maximum infeed depth (enter without sign)
_FAL	real	Final machining allowance on pocket edge (enter without sign)
_FALD	real	Final allowance at base (enter without sign)
_FFP1	real	Feedrate for surface machining
_FFD	real	Feedrate for depth infeed
_CDIR	int	Milling direction: (enter without sign) Value: 0...Down-cut milling (as spindle rotation) 1...Up-cut milling 2...with G2 (independent of spindle direction) 3...with G3
_VARI	int	Type of machining: (enter without sign) UNITS DIGIT: Value: 1...Roughing 2...Finishing TENS DIGIT: Value: 0...Perpendicular to the pocket center with G0 1...Perpendicular to the pocket center with G1 2...Along a helix

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 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

Function

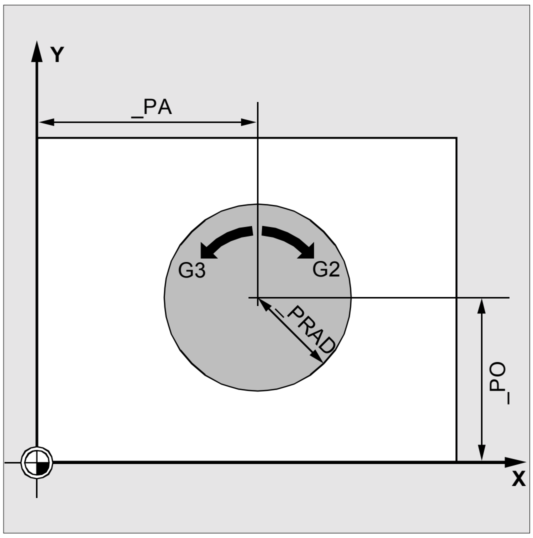
With this cycle you can machine circular pockets in the machining plane.
For finishing, a face cutter is needed.
The depth infeed will always start at the pocket center point and be performed vertically from there; thus predrill can be suitably performed in this position.



New functions compared to POCKET2:

- The milling direction can be defined with a G instruction (G2/G3) or up-cut milling or down-cut from the spindle direction
- For solid machining, the maximum infeed width in the plane is programmable
- 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
- Shorter 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 of operations



Position reached prior to cycle start:
This can be any position from which the starting position on 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, with G0 the reference plane brought forward by the safety distance is approached. Pocket machining is then performed according to the selected insertion strategy and considering the programmed base size.

Insertion strategies:
see Section 3.9 (POCKET3)

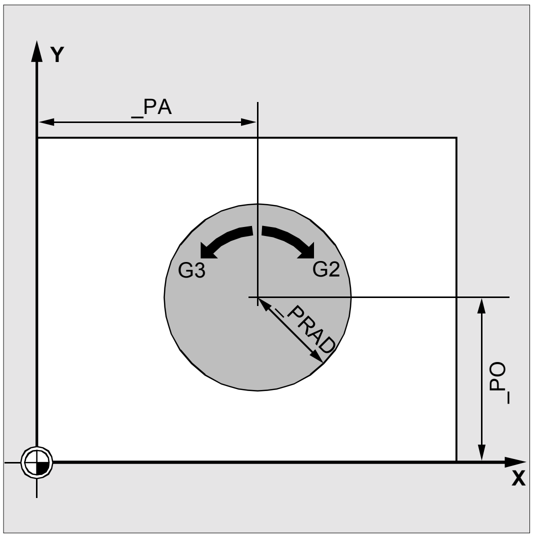
Accounting for blank dimensions
During solid machining, it is possible to take blank dimensions (for example, in the machining of precast workpieces) into account.
For circular pockets, the basic size _AP1 at the edge is also circular (with a smaller radius than the pocket radius).

For additional explanations see Section 3.9 (POCKET3)

Motion sequence when finishing (VARI=X2):
Finishing is performed in sequence from the edge until reaching the final machining allowance on the base, then the base is finished. If one of the final machining allowances is equal to zero, this part of the finishing process is skipped.

- Finishing on the edge
  While finishing on the edge, the pocket is only machined once.
  For finishing on the edge the path includes one fourth of circle which reaches the pocket radius. The radius of this path is less or equal to 2mm or, if "less room" is available, equals the difference between the pocket radius and the mill radius.
  The depth infeed is performed with G0 in the open towards the pocket center and the starting point of the approach path is also reached with G0.
- Finishing on the base
  During finishing on the base, the machine performs G0 towards the pocket center until reaching a distance equal to pocket depth + final machining allowance + safety distance. From this point onwards, 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.

Description of parameters



See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters _RTP, _RFP, _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, _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 "Milling 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 mode)
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.

Rough cutting only edge with POCKET4 (SW 5.3 and higher)
POCKET4 (circular pocket) can now machine only the edge when rough-cutting.
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:
Condition: 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
→_AD≥19.75	rough dimension for depth must be greater or equal to pocket depth incremental minus depth allowance, i.e. 21-1.25=19.75

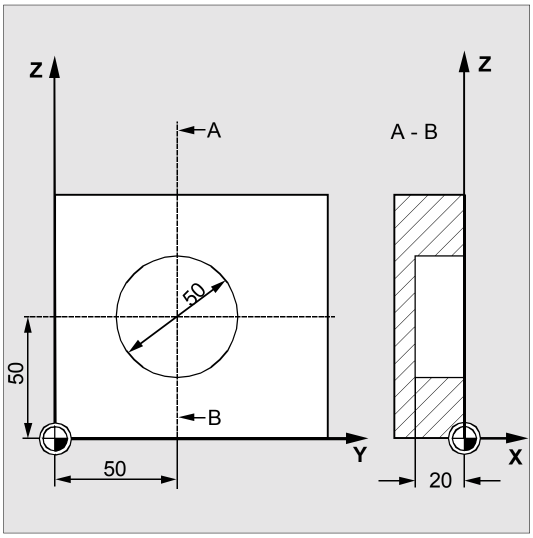
Further notes

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

A new 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 down-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 starting position
`N30 Pocket4(3, 0, 0, -20, 25, 50, -> 50, 6, 0, 0, 200, 100, 1, 21, 0, -> 0, 0, 2, 3)`	Cycle call
`N40 M30`	End of program

-> Must be programmed in a single block

3.11 Face milling – CYCLE71

The cycle CYCLE71 is available in SW 4 and higher.

Programming

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

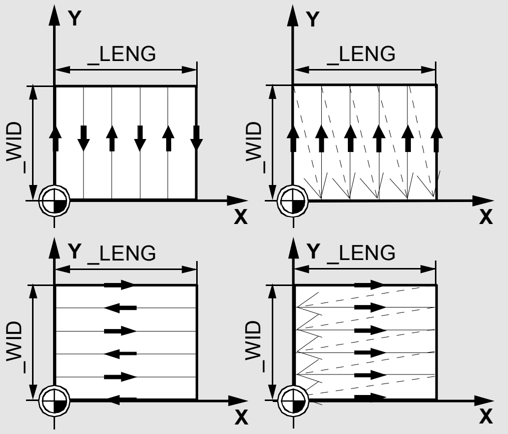
Parameters

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 plus/minus sign.
_WID	real	Rectangle length along the 2nd axis, incremental. The corner from which dimensions are measured is given by the plus/minus 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° ≤ _STA < 180°
_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 the roughing mode, _FALD refers to the remaining material on the surface.
_FFP1	real	Feedrate for surface machining
_VARI	int	Type of machining: (enter without sign) UNITS DIGIT: Value: 1...Roughing 2...Finishing TENS DIGIT: Value: 1...Parallel to the abscissa, in one direction 2...Parallel to the ordinate, in one direction 3...Parallel to the abscissa, with changing direction 4...Parallel to the ordinate, with changing direction
_FDP1	real	Overrun travel in direction of plane infeed (incr., 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.



Possible machining strategies for face milling

Sequence of operations

Position reached prior to cycle start:
This can be any position from which the starting position on the infeed point at the retraction plane level can be reached without collision.

The cycle implements the following motion sequence:

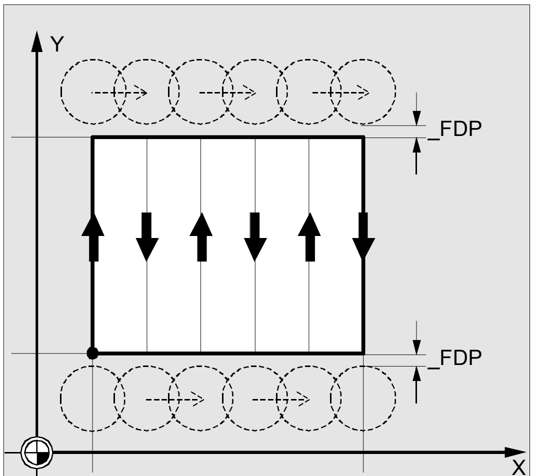
- G0 is applied to approach the infeed point on the current position plane. The reference plane, shifted forward by the safety distance, is then also approached with G0 from this position. Then, also with G0, infeed to machining plane. G0 is possible, since infeed occurs in the open.
  There are several roughing strategies (paraxial in one direction or back and forth).
- Motion sequence when roughing (VARI=X1):
  Roughing is possible on several planes according to the programmed values _DP, _MID and _FALD. Machining will be 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 stock removal on 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, and thus no width exceeds the maximum possible value. The tool center point thus does not always travel exactly to the edge (only if _MIDA = mill 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 less than _MIDA. The other paths for width infeed are calculated internally so as to produce a uniform path width (<=_MIDA).

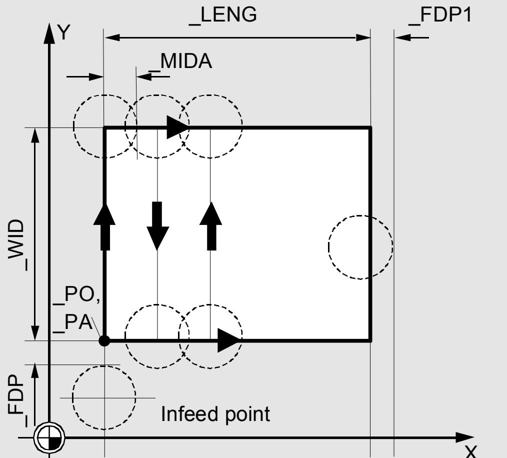
- Motion sequence when finishing (VARI=X2):
  When finishing, the surface is once milled in the plane. The final machining allowance for roughing must also be selected so that the remaining depth can be machined in one pass with the finishing tool.
  After each surface milling pass in the plane, the 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 reached at rapid traverse.
  Roughing in one directions stops when reaching the calculated infeed depth + safety distance. 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.



Milling during finishing in one direction (type of machining 22)

Description of parameters

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



Roughing with _MIDA larger than mill radius (Type of machining 41)

_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.

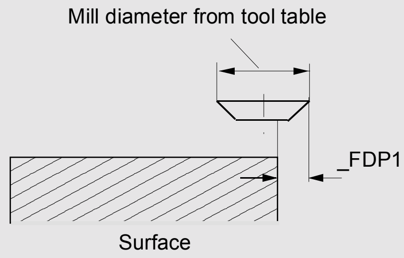
_LENG, _WID (length)
With parameters _LENG and _WID you determine the length and width of the rectangle in the plane. The sign determines the position of the rectangle relative to _PA and _PO.

_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 percent of the mill diameter as 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

By means of this parameter an overrun travel in the direction of the plane infeed may be defined (_MIDA) allowing the deviation between the current cutter radius and the cutting edge (e.g. tool nose radius or inclined cutting tips) to be compensated. The last cutter center point path therefore 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 risk of collision.

_VARI (machining mode)

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 changing direction
- 4=Parallel to the ordinate, with changing direction.

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

Further notes

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

Programming example

Face milling

Parameters for cycle call:

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

`%_N_TSTCYC71_MPF` `;$PATH=/_N_MPF_DIR`	Program for face milling with CYCLE71
`;*`	
`$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 starting position
`;`	
`CYCLE71(  10,  0,   2,-11, 100, 100, ->` `-> 60, 40, 10, 6, 10, 5, 0, 4000, 31, 2)`	Cycle call
`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 (not for FM-NC).

Programming

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

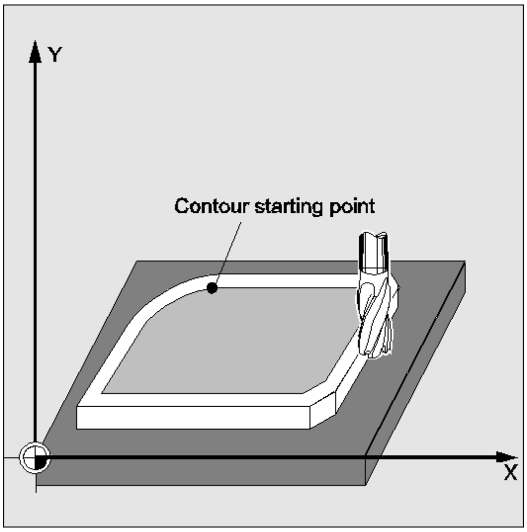
Parameters

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	int	Type of machining: (enter without sign) UNITS DIGIT: Value: 1...Roughing 2...Finishing TENS DIGIT: Value: 0...Intermediate paths with G0 1...Intermediate paths with G1 HUNDREDS DIGIT: Value: 0...Return at end of contour to _RTP 1...Return at end of contour to _RFP + _SDIS 2...Return at end of contour to _SDIS 3...No return to end of contour
_RL	int	Contouring is centric, on right or left (with G40, G41 or G42, enter without sign) Value: 40...G40 (approach and return, straight line only) 41...G41 42...G42

_AS1	int	Specification of approach direction/path: (enter without sign) UNITS DIGIT: Value: 1...Straight tangential line 2...Quadrant 3...Semicircle TENS DIGIT: Value: 0...Approach to the contour in the plane 1...Approach to the contour along a spatial path
_LP1	real	Length of the approach travel (along a straight line) or radius of the mill center path of the arc of approach (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	int	Specification of return direction/path: (enter without sign) UNITS DIGIT: Value: 1...Straight tangential line 2...Quadrant 3...Semicircle TENS DIGIT: Value: 0...Return to the contour in the plane 1...Return to the contour along 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)

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 (center, on left or right of contour).

The contour must be programmed in the direction of milling and must be in the same plane. In addition, its 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 considering a final machining allowance if necessary at several depths until reaching the final machining allowance) and finishing (single-pass of final contour, if necessary at several depths)
- Flexible approach and retraction to/from the contour either tangentially or radially (quadrant or semicircle)
- Programmable depth infeed
- Intermediate motions either with rapid traverse or at feedrate.

The requirement for executing a cycle is an NC SW 4.3. or higher that includes the function "Soft approach and return".

Sequence of operations

Position reached prior to cycle start:

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 parameter.

- Travel to starting point for initial cut with G0/G1 (and _FF3). This point is calculated internally 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 path compensation is activated in this block.
- Depth infeed to first or next machining depth plus programmed safety distance DISCL with G0/G1. The first processing depth is given by
  - the overall depth,
  - the final machining allowance and
  - the maximum possible depth infeed.
- Approach the contour perpendicular to the feed depth and approach in the plane then at the feedrate programmed for surface machining, or programmed under _FAD for 3D machining corresponding to the programming for soft approach.
- Milling along the contour with G40/G41/G42.
- Soft retraction from the contour with G1 and still with the feedrate for surface machining by lift DISCL.
- 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 reaching the final machining allowance in depth.

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

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

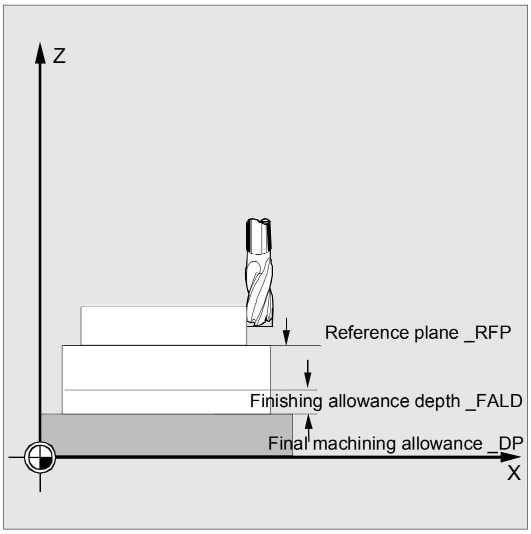
At the end of the cycle, the tool is positioned at the contour retraction point at the retraction plane level.

Contour programming

For programming the contour, please note the following:

- In the subroutine no programmable frame (TRANS, ROT, SCALE, MIRROR) may be selected before the first programmed position.
- The first block of the contour subroutine is a straight line block containing G90, G0 and defines the contour start.
- The cutter radius compensation is selected and deselected from the upper level cycle; then the contour subroutine has no G40, G41, G42 programmed.

Description of parameters



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

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

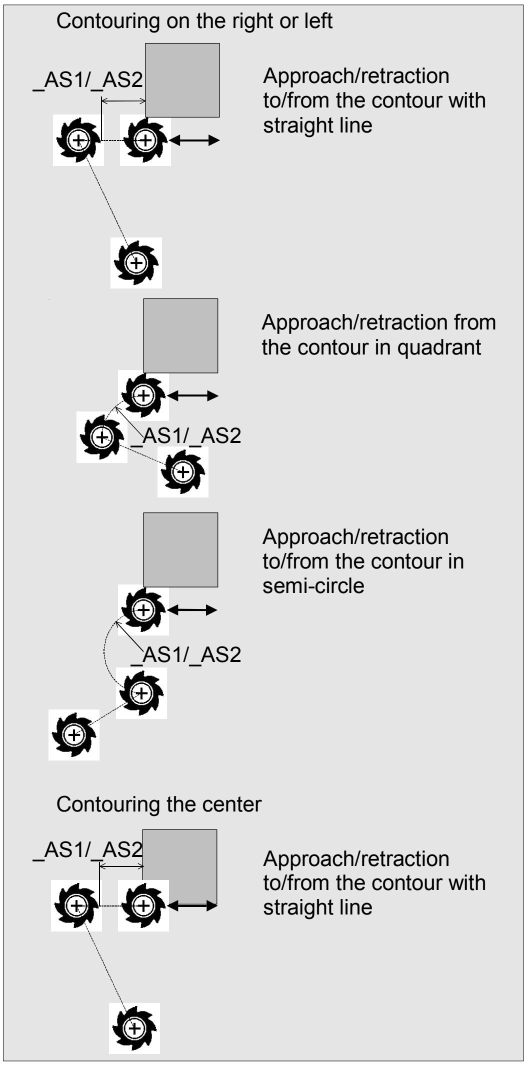
In SW 5.2 and higher, the milling contour can also be a section of the calling routine or from any other program. The section is identified by start or end labels or by block numbers. In this case, the program name and labels/block number are identified by an ":".

Examples:

_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 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 for the section in _KNAME must be adjusted if the program is modified and subsequently renumbered.

_VARI (machining mode)
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. See "Parameter CYCLE72" for possible settings.

_AS1, _AS2 (direction of approach/approach travel, direction of retraction/retraction travel)
With the parameter _AS1 you can specify the approach travel and with _AS2 the retraction travel. For possible values, see "Parameter CYCLE72". If _AS2 is not programmed, then the behavior programmed for the approach path will 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), 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 retract path=0."

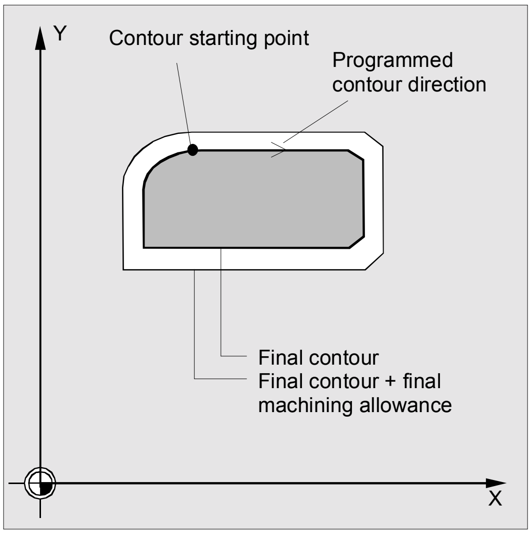
When G40 is programmed, the approach or retract path corresponds to the distance between the tool center point and the starting or end point of the contour.

_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 feed (G01). If no feedrate is programmed, the intermediate motions are carried out with surface feed for G01.

Further notes

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

Programming example 1



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

Parameters for cycle call:

- Retraction plane 250mm
- Reference plane 200
- Safety distance 3mm
- Depth 175mm
- Maximum depth infeed 10
- Final machining allowance in depth 1.5mm
- Feedrate depth infeed 400 mm/min
- Final machining allowance in the plane 1mm
- Feedrate in the plane 800 mm/min
- Machining: Roughing up to the finishing allowance, intermediate travel with G1, during the intermediate motions, return along 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 20mm radius
- Retraction feedrate 1000 mm/min

`%_N_RANDKONTUR1_MPF` `;$PATH=/_N_MPF_DIR`	Program for re-milling a contour 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 starting position

`N30 CYCLE72 ( "MYKONTUR", 250, 200, ->`
`-> 3, 175, 10,1, 1.5, 800, 400, 111, ->`
`-> 41, 2, 20, 1000, 2, 20)`
	Cycle call

`N90 X100 Y200`	
`N95 M02`	End of program

-> Must be programmed in a single block

`%_N_MYKONTUR_SPF` `;$PATH=/_N_SPF_DIR`	Subroutine for contour milling (for example)
`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 (SW 5.2 and higher)

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 ->`
`CYCLE72 ( "START:END", 250, 200, ->`
`-> 3, 175, 10,1, 1.5, 800, 400, 11, ->`
`-> 41, 2, 20, 1000, 2, 20)`
	Approach starting position, cycle call

`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 (SW 5.3 and higher)

Programming

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

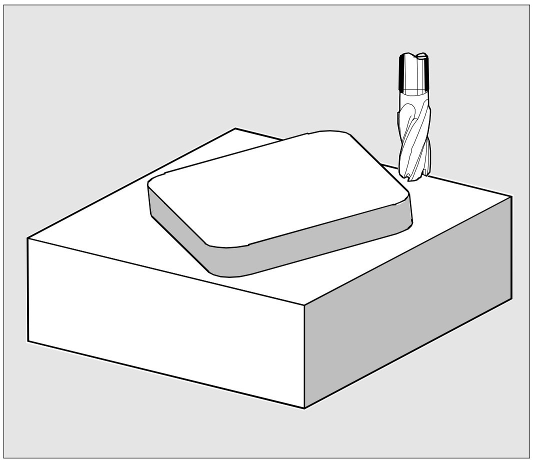
Parameters

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`	int	Milling direction: (enter without sign) Value: 0...Upcut milling 1...Downcut milling 2 with G2 (irrespective of spindle direction) 3...with G3
`_VARI`	int	Type of machining: Value: 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

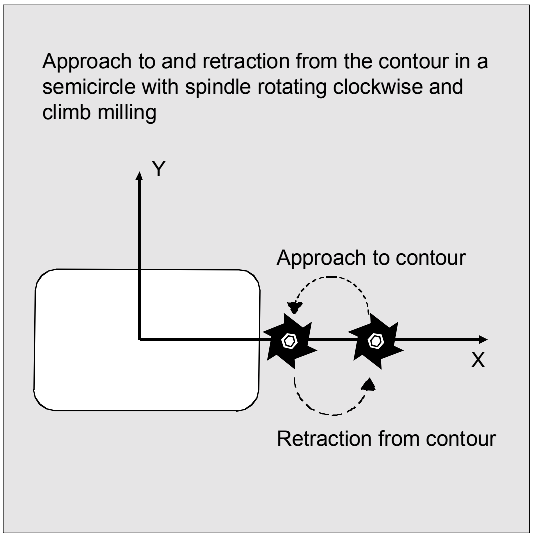
With this cycle you can machine rectangular spigots in the machining plane. For finishing, a face cutter is needed. Depth infeed is always performed in the position reached prior to semicircular positioning on the contour.



Sequence of operations

Position reached prior to cycle start:
The starting point is a position in the positive range of the abscissa with integrated approach semicircle and allowance for programmed, abscissa-related blank dimension.

Sequence of motions for roughing (_VARI=1)
Approach to and exit from contour:
The retraction plane (_RTP) is approached in rapid traverse so that the tool can be positioned from there on the starting point in the machining plane. The starting point is defined as being 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 milling 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 in to the next machining depth. The contour is then approached again along a semicircle and the spigot circumnavigated once. This process is repeated until the programmed spigot depth is 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.

Sequence of motions for finishing (VARI=X2)

Depending on the setting of parameters _FAL and _FALD, a finishing operation is performed on the spigot surface or base or both. The approach strategy matches the motions in the plane executed for roughing operations.

Description of parameters

See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, SDIS, DP, DPR.
See Section 3.9 for a description of parameters _MID, _FAL, _FALD, _FFP1, _FFD.
See Section 3.2 for cycle setting data _ZSD[2].

_LENG, _WID and _CRAD (spigot length, spigot width and corner radius)

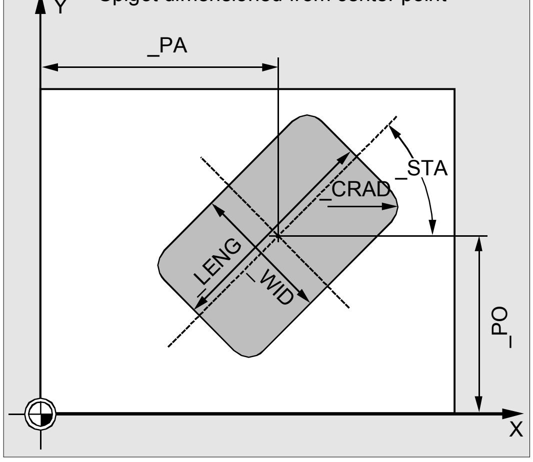
The shape of a spigot in the plane is determined with parameters _LENG, _WID and _CRAD.
The spigot can be dimensioned from the center or from one corner point. When dimensioning from a corner point, use _LENG and _WID with sign.
The absolute length value (_LENG) always refers to the abscissa (with a plane angle of zero degrees).

_PA, _PO (reference point)

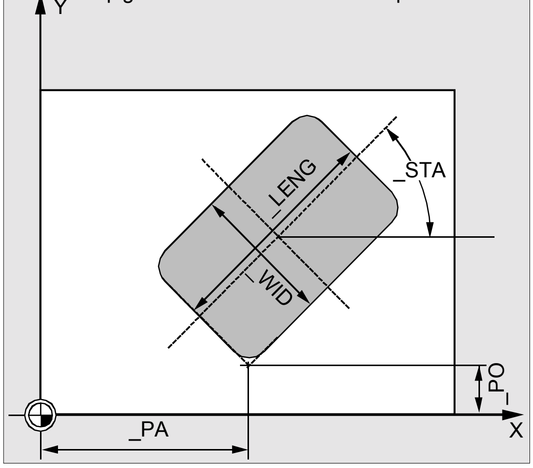
Parameters _PA and _PO are set to define the reference point of the spigot in abscissa and ordinate. This is either the spigot center point or a corner point. The value of this parameter depends on cycle setting data bit _ZSD[2]:

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

When the spigot is dimensioned from a corner, the length and width parameters must be entered with sign (_LENG, _WID) so that a unique position for the spigot is defined.



Spigot dimensioned from center point



Spigot dimensioned from corner point

_STA (angle)

_STA specifies the angle between the 1st axis of the plane (abscissa) and the longitudinal axis of the spigot.

_CDIR (milling direction)

The machining direction of the spigot is defined in this parameter.

Under parameter _CDIR the mill direction

- direct "2 for G2" and "3 for G3" or
- alternatively "up-cut milling" or "down-cut milling"

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

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

_VARI (machining mode)

You can define the type of machining with parameter _VARI.

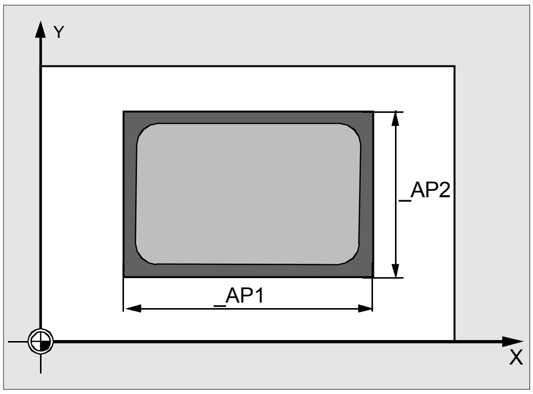
Possible values are:

- 1=Roughing
- 2=Finishing.

_AP1, _AP2 (blank dimensions)

Blank dimensions (e.g. in the case of precast workpieces) can be taken into account in machining of the spigot.

The basic size for the length and width (_AP1 and _AP2) are programmed without sign and their symmetrical positions around the spigot center computed in the cycle. The internally calculated radius of the approach semicircle is dependent on this dimension.

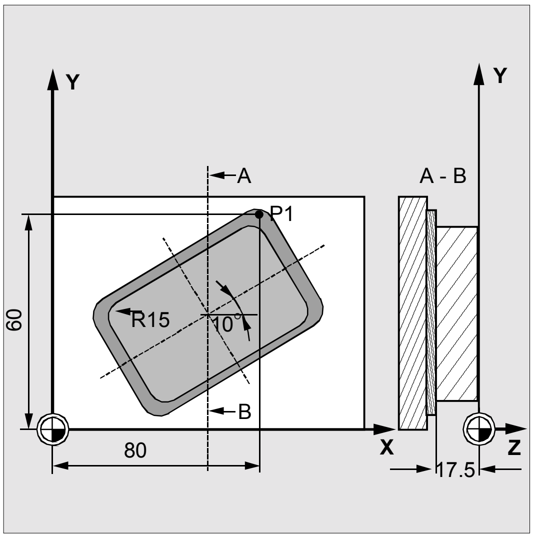


Further notes

A tool offset must be activated before the cycle is called. The cycle is otherwise aborted with alarm 61009 "Active tool number=0".
A new 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

Spigots
This program allows you to machine a spigot that is 60mm long, 40mm wide, 15mm deep in the XY plane and with a corner radius of 15mm. 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 to define a unique position for the spigot. The spigot is premachined with an allowance of 80mm in its length and 50mm in its width.



`N10 G90 G0 G17 X100 Y100 T20 D1 S3000 M3`	Specification of technology values
`N11 M6`	
`N20 _ZSD[2]=1`	Dimensioning of spigot referred to corners

`N30 CYCLE76 (10, 0, 2, -17.5, , -60, ->`
`-> -40, 15, 80, 60, 10, 11, , , 900, ->`
`-> 800, 0, 1, 80, 50)`
	Cycle call

`N40 M30`	End of program

-> Must be programmed in a single block

3.14 Milling circular spigots – CYCLE77 (SW 5.3 and higher)

Programming

CYCLE77 (_RTP, _RFP, _SDIS, _DP, _DPR, _PRAD, _PA, _PO, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _AP1)

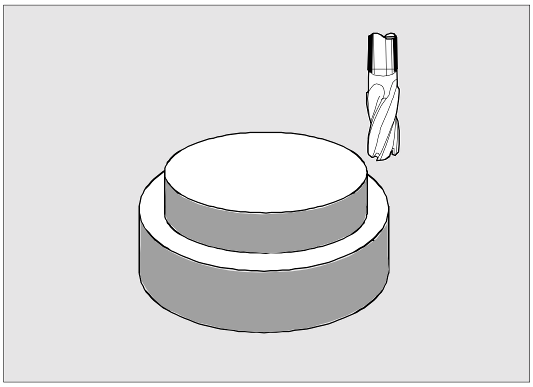
Parameters

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	int	Milling direction: (enter without sign) Value: 0...Upcut milling 1...Downcut milling 2 with G2 (irrespective of spindle direction) 3...with G3
_VARI	int	Type of machining Value: 1...Roughing to final machining allowance 2...Finishing (allowance X/Y/Z=0)
_AP1	real	Diameter of blank spigot

Function

With this cycle you can machine circular spigots in the machining plane. For finishing, a face cutter is needed. Depth infeed is always performed in the position reached prior to semicircular positioning on the contour.

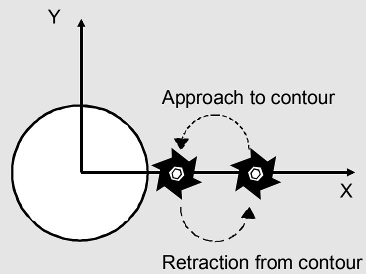


Sequence of operations

Position reached prior to cycle start:
The starting point is a position in the positive range of the abscissa with integrated approach semicircle and allowance for programmed blank dimension.

Sequence of motions for roughing (_VARI=1)
Approach to and exit from contour:
The retraction plane (_RTP) is approached in rapid traverse so that the tool can be positioned from there on the starting point in the machining plane. The starting point is defined as being 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 milling 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 in to the next machining depth.
The contour is then approached again along a semicircle and the spigot circumnavigated once. This process is repeated until the programmed spigot depth is reached.
The tool then approaches the retraction plane (_RTP) in rapid traverse.

Approach to and retraction from the contour in a semicircle with spindle rotating clockwise and climb milling



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.

Sequence of motions for finishing (_VARI=2)

Depending on the setting of parameters _FAL and _FALD, a finishing operation is performed on the spigot surface or base or both. The approach strategy matches the motions in the plane executed for roughing operations.

Description of parameters

See Subsection 2.1.2. (Drilling, Centering – CYCLE81) for a description of parameters RTP, RFP, SDIS, DP, DPR.
See Section 3.9 for a description of parameters _MID, _FAL, _FALD, _FFP1, _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 (milling direction)

The machining direction of the spigot is defined in this parameter.
Under parameter _CDIR the mill direction

- direct "2 for G2" and "3 for G3" or
- alternatively "up-cut milling" or "down-cut milling" can be programmed. Up-cut milling or down-cut milling is determined within the cycle via the spindle direction activated prior to the cycle call.

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

_VARI (machining mode)

You can define the type of machining with parameter _VARI.
Possible values are:

- 1=Roughing
- 2=Finishing.

_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.

Further notes

A tool offset must be activated before the cycle is called. The cycle is otherwise aborted with alarm 61009 "Active tool number=0".

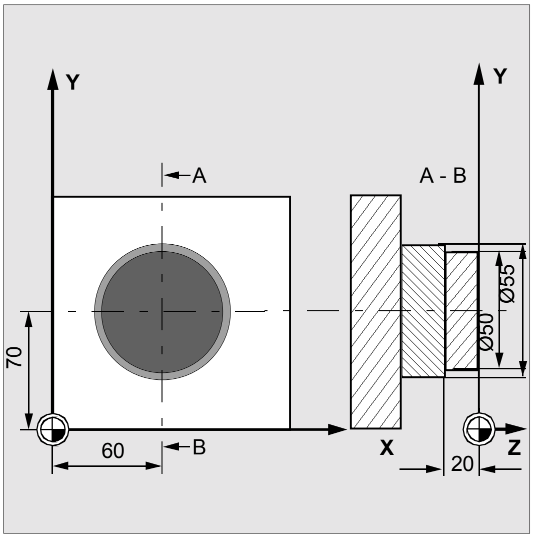
A new workpiece coordinate system that influences the actual value display is used in the cycle. The zero point of this coordinate system lies on the pocket center point.

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 55mm and a maximum infeed of 10mm per cut. Enter a final machining allowance for finishing the spigot surface. The entire spigot is machined in a down-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, ->` `-> 70, 10, 0.5, 0, 900, 800, 1, 1, 55)`	Roughing cycle call
`N30 D1 T2 M6`	Tool change
`N40 S2400 M3`	Specification of technology values
`N50 CYCLE77 (10, 0, 3, -20, , 50, 60, ->` `-> 70, 10, 0, 0, 800, 800, 1, 2, 55)`	Finishing cycle call
`N40 M30`	End of program

-> Must be programmed in a single block

3.15 Pocket milling with islands – CYCLE73, CYCLE74, CYCLE75 (SW 5.2 and higher)

Pocket milling with islands is an option and requires SW 5.2 in both the NCK and MMC 103.

Precondition

To use the pocket milling cycle with islands, you need specific machine data settings.

References: for current infos see:

- the file "siemensd.txt" from the delivery software (standard cycles) or
- for 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, 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( ) ;Machine pocket

3.15.1 Transfer pocket edge contour – CYCLE74

Pocket milling with islands is an option and requires SW 5.2 in both the NCK and MMC 103.

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 done 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.

So 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","MARKER_START", "MARKER_END")

The program name is 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 and requires SW 5.2 in both the NCK and MMC 103.

Programming

CYCLE75 (_KNAME, _LSANF, _LSEND)

Parameters

_KNAME	string	Name of contour subroutine of island contour
_LSANF	string	Block number/label identifying start of contour definition
_LSEND	string	Block number/label identifying end of contour definition

Function

Cycle CYCLE75 transfers island contours to the pocket milling cycle CYCLE73. The cycle is called once for each island contour. It need not be called if no island contours are programmed.

The transferred parameter values are written to the temporary file opened by CYCLE74.

Description 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. 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 stand at a starting position of the programmed contour element before CYCLE73 is invoked.

The necessary programs must always be stored in one directory (workpiece or parts program). It is permissible to use the subroutine memory for pocket edge or island contours.

Workpiece-related geometric dimensional data may be programmed in either metric or inches. Switching between these units of measurement within individual contour programs will 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.

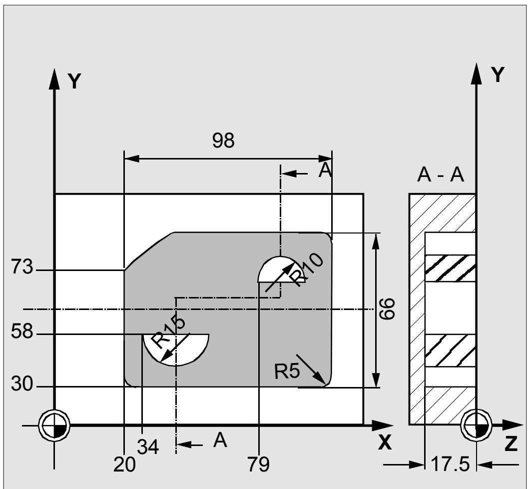
When the pocket machining program is calculated, only the geometries in the plane are taken into account.

If other axes or functions (T.., D.., S.. M.. etc.) are programmed in contour sections, they are skipped when the contour is prepared internally in the cycle.

All 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 must always be programmed again with the offsets calculated into the coordinates.

Programming example

Sample program 1.mpf (pocket with islands)



All radii on R5 corners

```
%_N_SAMPLE1_MPF
;$PATH=/_N_MPF_DIR
; Example_1: Pocket with islands
;
```

`$TC_DP1[5,1]=120 $TC_DP6[5,1]=6 $TC_DP3[5,1]=111`	;Tool offset mill T5 D1
`$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 _ENDEDGE: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 contours`	
`SAMPLE_CONT:`	
`CYCLE74 ("SAMPLE1","_EDGE","_ENDEDGE")`	;Transfer edge contour
`CYCLE75 ("SAMPLE1","_ISLAND1","_ENDISLAND1")`	;Transfer island contour 1
`CYCLE75 ("SAMPLE1","_ISLAND2","_ENDISLAND2")`	;Transfer island contour 2
`ENDLABEL:`	
`M30`	

3.15.4 Pocket milling with islands – CYCLE73

Pocket milling with islands is an option and requires SW 5.2 in both the NCK and MMC 103.

Programming

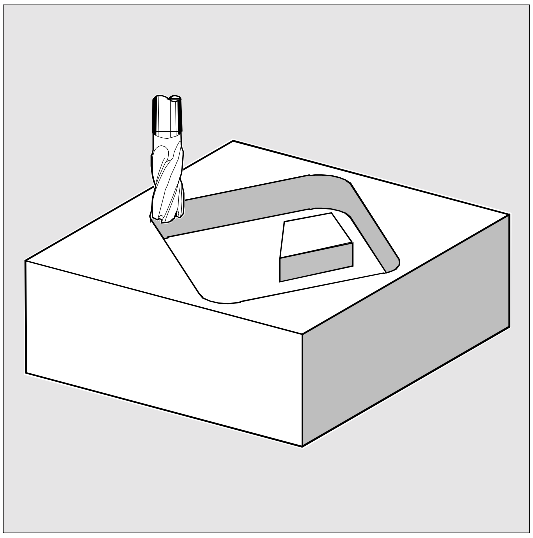
CYCLE73 (_VARI, _BNAME, _PNAME, _TN, _RTP, _RFP, _SDIS, _DP, _DPR, _MID, _MIDA, _FAL, _FALD, _FFP1, _FFD, _CDIR, _PA, _PO, _RAD, _DP1)

Parameters

_VARI	int	Type of machining: (enter without sign) UNITS POSITION (select machining): Value: 1...Rough cut (remove stock) from solid material 2...Rough cut residual material 3...Finish edge 4...Finish base 5...Rough drill TENS DIGIT (select insertion strategy): Value: 1...Perpendicular with G1 2...Along a helix 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 stock removal 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 depth in the plane (enter without sign)
_FAL	real	Final machining allowance in the plane (enter without sign)
_FALD	real	Final machining allowance on base (enter without sign)

_FFP1	real	Feedrate for surface machining
_FFD	real	Feedrate for depth infeed
_CDIR	int	Milling direction for machining the pocket: (enter without sign) Value: 0...Down-cut milling (as spindle rotation) 1...Downcut milling 2...with G2 (irrespective of spindle direction) 3...with G3
_PA	real	Start point in first axis (only with manual selection of start point)
_PO	real	Start point in second axis (only with manual selection of start point)
_RAD	real	Radius center-point path on insertion along helical path or max. insertion angle for oscillating insertion motion
_DP1	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:

- Rough drill
- 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 finishing dimensions. 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.

Rough drill

Depending on the milling tool used, it may be necessary to drill before solid machining the workpiece. The cycle automatically calculates the rough drilling positions as a function of the solid machining operation to be performed afterwards. The drilling cycle called modally beforehand is executed at each of these positions. Rough drilling can be executed in a number of technological machining operations (e.g. 1. centering, 2. drilling).

Rough drilling sequence

In the first machining section of the rough drilling operation, a REPEAT command must be used after a modal call for 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 as well as the programs for solid machining and drilling.
Parameter _VARI is the only one to define all solid machining parameters and it must always be programmed for this reason.
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

Rough drill, with solid machining

`ACCEPTANCE4_CONT:`	;Marker with name for beginning of pocket ;contour
`CYCLE74("EDGEA01", ,)`	;Definition of contour for pocket edge
`CYCLE75("ISL11A01", ,)`	;Definition of contour for 1st island
`CYCLE75("ISL1A01", ,)`	
`CYCLE75("ISL2A01", ,)`	
`CYCLE75("ISL3A01", ,)`	
`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`	;Branch to Solid machine pocket
`ACCEPTANCE4_MACH:`	;Start of section Generate programs
`;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 section Generate programs
`T3 M6`	
`D1 M3 S2000`	
`;REPEAT ACCEPTANCE4_CONT ENDLABEL`	;Required only if there is more than one pocket ;contour
`CYCLE73(1011,"ACCEPTANCE4_DRILL","ACCEPTANCE4_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. 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 feed _FFP1. The machining direction corresponds to the setting in _CDIR. The pocket can be split if the ratio between mill diameter and clearance between islands or between islands and edge contours is not ideal. For this purpose, the cycle calculates additional start points for mill 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 return 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 of motions for finishing (_VARI=XXX3)

- The pocket and island contours are circumnavigated once each 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 respectively of the finishing operation are each 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 the same as for solid machining.
- Parameters _FAL, _FALD and _VARI=XXX4 must be assigned for simultaneous finishing in the plane and on the base.

Description of parameters

_VARI (machining mode)

You can define the type of machining with parameter _VARI. Possible values are:

Units digit:

- 1=Rough cut (solid machine) from solid material
- 2=Rough cut residual material
- 3=Finish edge
- 4=Finish base
- 5=Rough drill.

When "Rough cut 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 mill diameter, then setting "2" can be selected to machine them afterwards with a smaller milling tool. To do this, cycle CYCLE73 must be called again.

Tens digit:

- 1=Vertical 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 (these may be several revolutions on the helical path) has been calculated, a full circle is made to remove the slope on insertion.
Then pocket solid machining starts in this plane and continues until reaching the final machining allowance.

- **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.

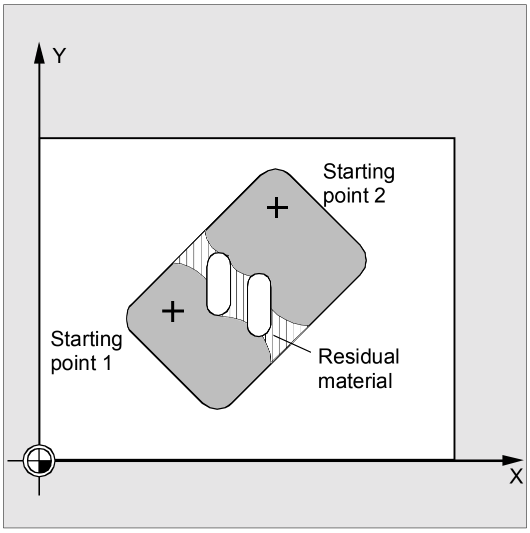
Hundreds digit: (_VARI=X1XX)

- 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.
Notice: Manually specified start positions must not be too close to the island surface. Manually specified start positions are not monitored internally.
If the pocket has to be split as a result of the island position and the mill diameter used, then 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 rough drill 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 corresponding 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 needed only 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 cutting 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 are generally set to different values. In the cycle it is assumed that the retraction plane lies 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 (absolute pocket depth) and
_DPR (incremental pocket depth)
The pocket depth can be specified as either an absolute value (_DP) or relative 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 percent of the mill radius as the maximum infeed width.
If an infeed width of more than 80 percent of the mill 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 ≥ 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.

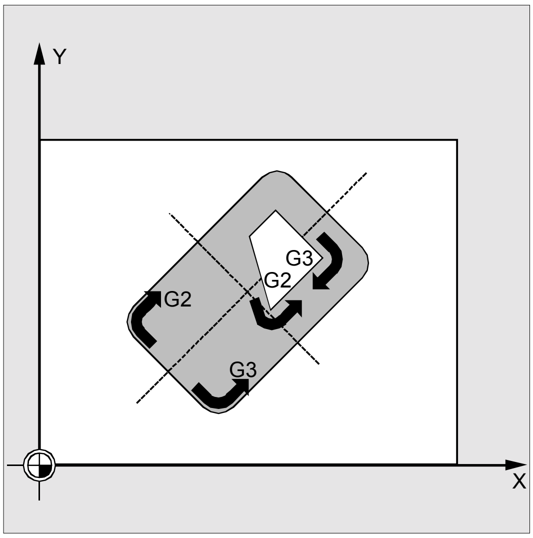
_FFD and _FFP1 (feedrate for depth infeed and surface machining)
Feedrate _FFD is used for insertion into the material. Feedrate FFP1 is used for all movements in the plane traversed at feedrate when machining.

_CDIR (milling direction)
The value for the machining direction of the pocket is defined in this parameter.
Under parameter _CDIR the mill direction

- direct "2 for G2" and "3 for G3" or
- alternatively "up-cut milling" or "down-cut milling"

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

Up-cut milling	Down-cut milling
M3 → G3	M3 → G2
M4 → G2	M4 → 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. It must be noted 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 (referred 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.

Further notes

Name for 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 then repeated later with the REPEAT instruction.
When programs are written using the cycles support function, a name for the pocket machining program is therefore entered in the respective screen forms. The name length is restricted to 8 characters.
In sample program 2, this is, for example "ACCEPTANCE4".
The T number contains the milling tool for all machining technologies. When residual material is machined more than once, the tool used beforehand 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 optimum timing, the calculation is carried out in the MMC.

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 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;
- In the case of different technologies, such as solid machining and residual material, with machining programs generated in different ways.

Program storage in the file system
If the contours for CYCLE73 are programmed outside the program that makes the call, the following applies for the search in the file system of the control:

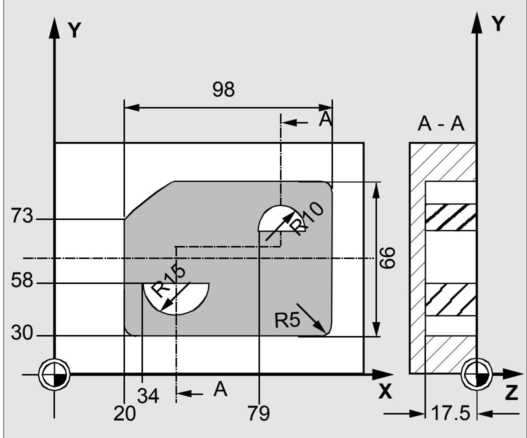
- If the calling program is stored in a workpiece directory, then the programs containing the edge or island contours must be stored in the same workpiece directory;
- If the invoked program is located in the "Parts Program" directory (MPF.DIR), a search is also made for the programs there.

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 flow into 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)



All radii on R5 corners

```
%_N_SAMPLE1_MPF
;$PATH=/_N_WKS_DIR/_N_CC73BEI1_WPD
;Example_1: Pocket with islands
;Solid machine and finish
```

`$TC_DP1[5,1]=120 $TC_DP3[5,1]=111 $TC_DP6[5,1]=4`	;Tool offset mill T5 D1
`$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`	
`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 _ENDEDGE: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`	
`;`	

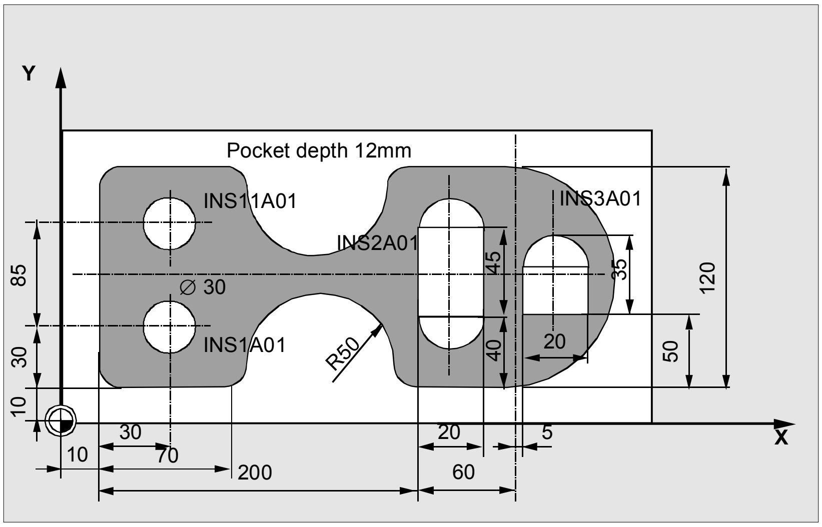
```
;Programming contours
_MACHINE:
SAMPLE1_CONT:
CYCLE74 ("","_EDGE","_ENDEDGE")
CYCLE75 ("","_ISL1","_ENDISL1")
CYCLE75 ("","_ISL2","_ENDISL2")
ENDLABEL:

;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 rough drilled to ensure optimum insertion of the milling tool.

- Center for rough drilling
- Rough drill
- Solid machine pocket with islands, mill radius 12mm
- Solid machine residual material, mill radius 6mm
- Finish pocket, mill radius 5mm.

Sketch of machining operation



Machining program:

```
%_N_SAMPLE2_MPF
;$PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
; Example_2: Pocket with islands
; 2*rough drill, 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("ISL11A01",,)
CYCLE75("ISL1A01",,)
CYCLE75("ISL2A01",,)
CYCLE75("ISL3A01",,)
ENDLABEL:

;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
```

Edge contour programming example 2:

```
%_N_EDGEA01_MPF
;$PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
;Ste 17.05.99
;Edge contour programming example 2

N5 G0 G90 X260 Y0
N7 G3 X260 Y120 CR=60
N8 G1 X170 RND=15
N9 G2 X70 Y120 CR=50
N10 G1 X0 RND=15
N11 Y0 RND=15
N35 X70 RND=15
N40 G2 X170 Y0 CR=50
N45 G1 X260 Y0
N50 M30
```

Island contour sample program 2

```
%_N_ISL1A01_MPF
;$PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
;Ste 18.06.99
;Island contour sample program 2

N5 G90 G0 X30 Y15
N10 G91 G3 X0 Y30 CR=15
N12 X0 Y-30 CR=15
N15 M30

%_N_ISL11A01_MPF
;$PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
;Ste 18.06.99
;Island contour sample program 2

N5 G90 G0 X30 Y70
N10 G91 G3 X0 Y30 CR=15
N12 X0 Y-30 CR=15
N15 M30

%_N_ISL2A01_MPF
;$PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
;Ste 18.06.99
;Island contour sample program 2

N5 G90 G0 X200 Y40
N10 G3 X220 Y40 CR=10
N15 G1 Y85
N20 G3 X200 Y85 CR=10
N25 G1 Y40
N30 M30
```

```
%_N_ISL3A01_MPF
;$PATH=/_N_WKS_DIR/_N_CC73BEI2_WPD
;Ste 18.06.99
;Island contour sample program 2

N5 G0 G90 X265 Y50
N10 G1 G91 X20
N15 Y25
N20 G3 X-20 I-10
N25 G1 Y-25
N30 M30
```

Programming example 3

Machining task:
Shows 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 performed complete on both pockets before the next tool is used.

- Rough drill
- 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:

;Machining contours pocket 2
GOTOF ENDLABEL
SAMPLE2_CONT:
CYCLE74("EDGEA01",,)
CYCLE75("ISL11A01",,)
CYCLE75("ISL1A01",,)
CYCLE75("ISL2A01",,)
CYCLE75("ISL3A01",,)
ENDLABEL:

;Drill
T2 M6
D1 M3 F6000 S4000
MCALL CYCLE81(10,0,1,-8,)
REPEAT POCKET1_MACH POCKET1_MACH_END
MCALL

G55
MCALL CYCLE81(10,0,1,-8,)
REPEAT SAMPLE2_MACH SAMPLE2_MACH_END
MCALL

;Removing pocket 1
T3 M6
G54 D1 M3 S3300
GOTOF POCKET1_MACH_END
POCKET1_MACH:
REPEAT POCKET1_CONT ENDLABEL
CYCLE73(1025,"POCKET1_DRILL","POCKET1_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)

;Removing 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)
```

```
;Removing 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	"Several insertion points required"	
61912	"No path generated"	
61913	"No residual material generated"	
61914	"Programmed helix violates contour"	
61915	"Approach/liftoff motion violates contour"	
61916	"Ramp path too short"	
61917	"Residual corners might be left with less than 50 percent overlap"	
61918	"Cutter radius too large for residual material"	
61980	"Error in island contour"	
61981	"Error in 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	"Not D1 programmed as active tool cutting edge"	

3.16 Swiveling – CYCLE800 (SW 6.2 and higher)

The Swiveling cycle is not an option, but is available for the HMI with SW 06.02 and NC SW 6.3 (CCU SW 4.3) and higher.
The functions

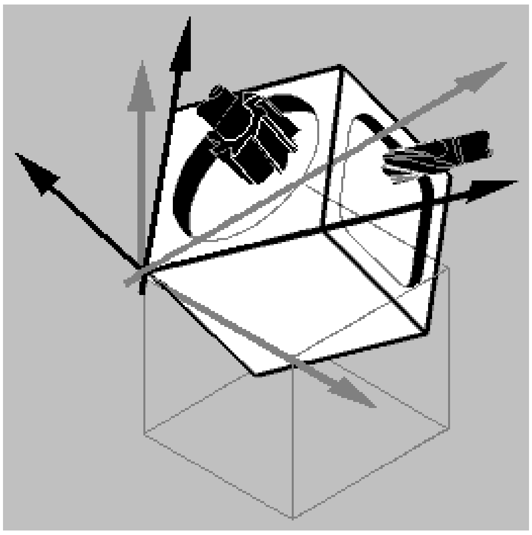
- 3/2 axes inclined machining and
- orientatable toolholder

are available in the basic version.

References: Description of Functions 840D/840Di/810D
/W1/ "Tool Offset"
/R2/ "Rotary Axes"
/K2/ "System Frames" (SW 6.1 and higher)

Function

The cycle is used during milling 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 workholder (swivel table) → type P
3. Mixed kinematics from 1 and 2 → type M

Important

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 system frames (swivel frame), tool reference (PARTFRAME), toolholder (TOOLFRAME) and workpiece reference (WPFRAME) (see HMI → parameters, active ZO).

The current machining plane (G17, G18, G19) is taken into account by the swivel cycle.

Swiveling onto 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 coordinate system X, Y, Z 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 via machine data. Any type of machining operation can be performed on the swivel plane, e.g. through calling standard or measuring cycles.

Note about calling the 5-axis transformation

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).

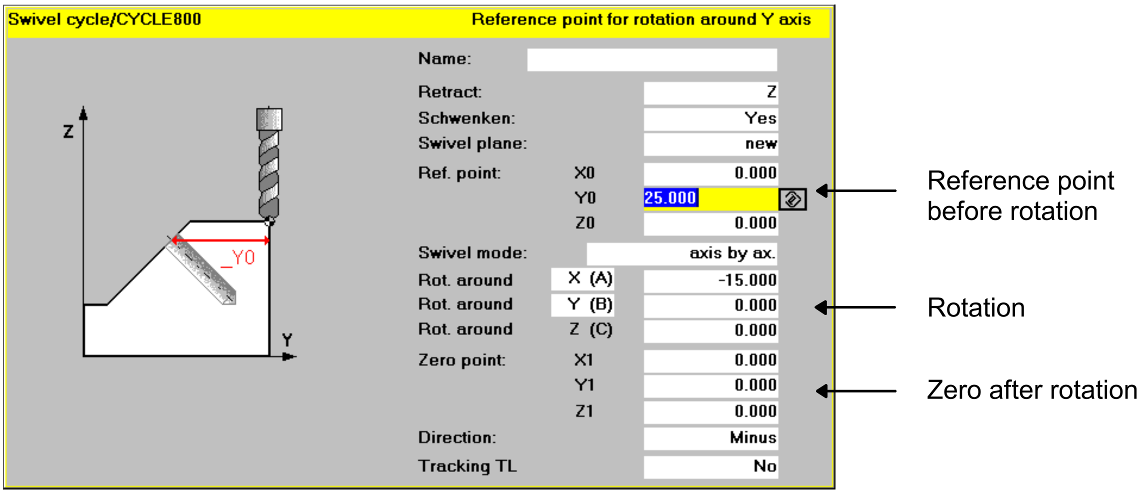
Example (machine with swivel table)

`G54`	
`T="MILL_10mm"`	
`M6`	
`CYCLE800(1,"",0,57,0,40,0,-45,0,0,0,0,0,-1)`	;Swivel cycle
`CYCLE71(50,24,2,0,0,0,80,60,0,4,10,5,0,2000,31,5)`	;Face mill
`TCARR=0`	;swivel data record deselection
`PAROTOF`	
`TOROTOF`	;(machine kinematic type "T" and "M" only)
`TRAORI`	
`G54`	;Calculation of new zero offset
`EXTCALL "WALZ"`	;5-axis machining program with direction vectors ;(A3, B3, C3)
`M2`	

3.16.1 Operation, parameter assignment, input screen form

Description of parameters

Interactive screen form CYCLE800 on standard interface



_TC (name of swivel data record)

The existing swivel data records (see Start-Up of CYCLE800) can be selected (toggled).

Each swivel data record has a name. No name needs to be declared if there is only one swivel data record.
"0" → swivel data record deselected.

_FR (retraction)

- Do not retract
- Traverse axis Z
- Traverse axis Z, XY (only when CYCLE800 is active in the start-up menu).

The retraction positions can be entered in the start-up menu display.
The retraction positions are approached absolutely. If a different retraction sequence or incremental positioning is desired, the process can be modified accordingly in user cycle TOOLCARR during start-up.

Note

When programming with standard cycles and 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 violation), as 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.

_DIR (swivel, direction)

- **Swivel yes**
  Rotary axes are positioned or the operator can turn the manual rotary axes.
- **Swivel no (calculation only)**
  If rotary axes are not to be traversed after activating the swivel cycle, the "Swivel no" selection applies. Application: Auxiliary swivel levels according to workpiece drawing
- **Direction minus/plus**
  Reference to rotary axis 1 or 2 when direction of travel is selected in the swivel cycle interactive screen form. Due to the angular range of the rotary axes of machine kinematics, the NCU calculates two possible solutions. One of these solutions is normally appropriate technologically. The rotary axis (1st or 2nd rotary axis) to which the two solutions are to refer, is selected in the CYCLE800 start-up menu. The choice between the two possible solutions to be traversed is made in the input screen form for the swivel cycle.

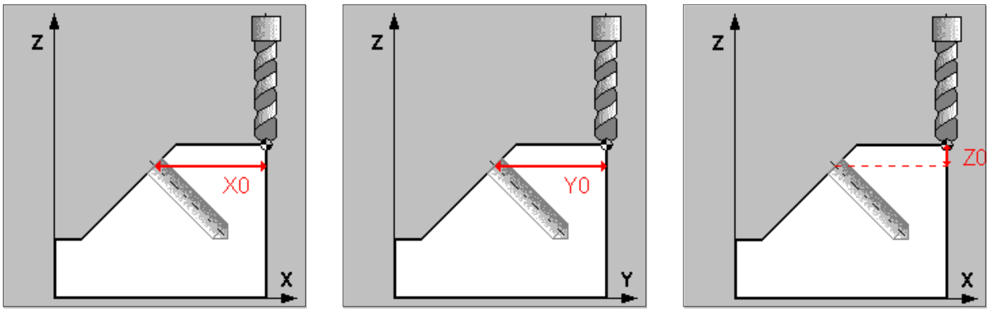
_ST (swivel plane)

- **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 to ensure that a swivel frame from another program is not active.
- **Additive**
  The swivel frame is added to the swivel frame of 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.

The following help displays relate to machining plane G17 (tool axis Z).

X0, Y0, Z0 (reference points prior to rotation)

Reference points



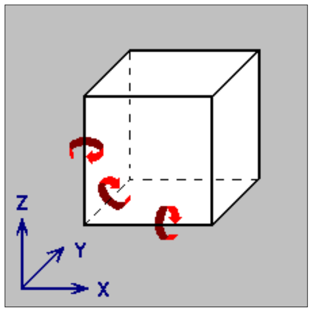
_MODE (swivel mode)

This parameter defines the swivel mode for the axis.

- Axial
- Projection angle[1)]
- Solid angle[1)]

Swivel mode always refers to the coordinate system of the workpiece and is therefore not dependent on the machine.

Axial

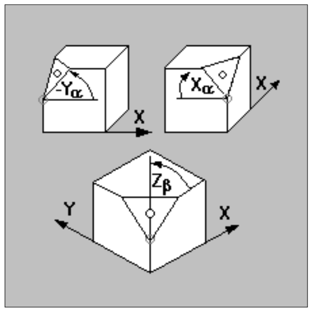


You can set which swivel modes are available in the CYCLE800 start-up menu.

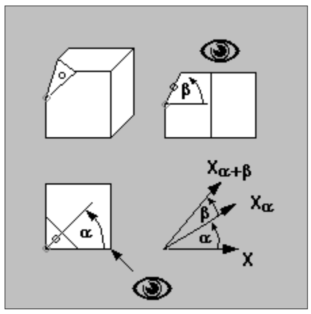
- With the axial swiveling variant, the tool is rotated about the individual axes in succession, with each rotation starting from the previous rotation. The axis sequence can be selected freely.
- When swiveling using the angle of projection, the angle value of the swiveled surface is projected onto the 1st two axes of the coordinate system. The 3rd rotation starts from the previous rotation. The axis sequence can be selected freely.
- With the solid angle variant, the tool is rotated first about the Z axis and then the Z axis. The second rotation starts from the first.

The positive direction of each rotation for the different swivel variants is shown in the help displays.

Projection angle



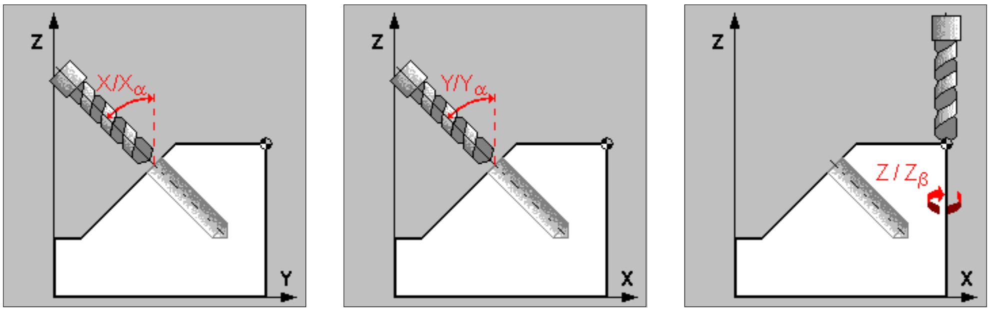
Solid angle



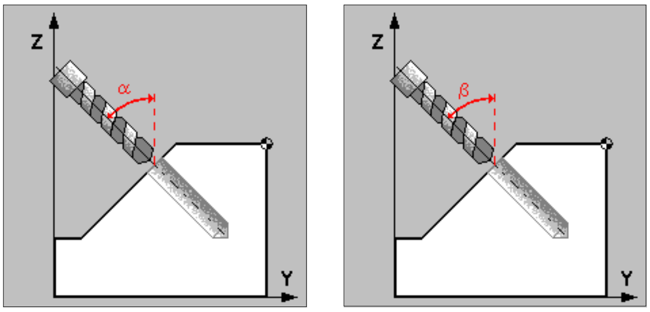
1) Only available if machine manufacturer is selected in the start-up screen form.

A, B, C (rotations)

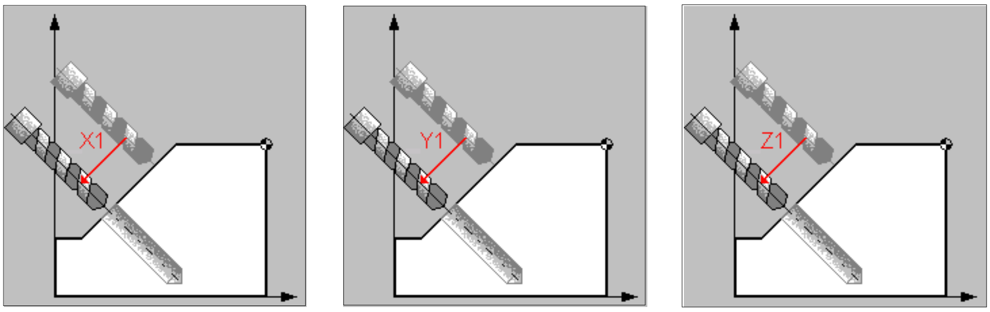
- Rotations (axial, projection angle)



- Rotation (solid angle)



X1, Y1, Z1 (zero point after rotation)



Correction tool (_TC_N_WZ)

- **yes / non**
  Display can be deactivated via the CYCLE800 start-up menu.
- **Yes:** When swiveling onto a machining plane, the rotary axes can be corrected to prevent risk of collision.
  Preconditions:
  1. TRAORI option is required.
  2. The machine manufacturer has adapted user cycle TOOLCARR.spf appropriately.

3.16.2 Operating instructions

- If the rotary axes of machine kinematics are defined as the manual axes (CYCLE800 start-up 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 key WCS is active on the machine control panel. This ensures that the geometry axes are traversed and not the machine axes.
- It is possible (without operator support) 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]).

3.16.3 Parameters

Programming

CYCLE800(_FR, _TC, _ST, _MODE, _XO, _YO, _ZO, _A, _B, _C, _X1, _Y1, _Z1, _DIR)

Parameters

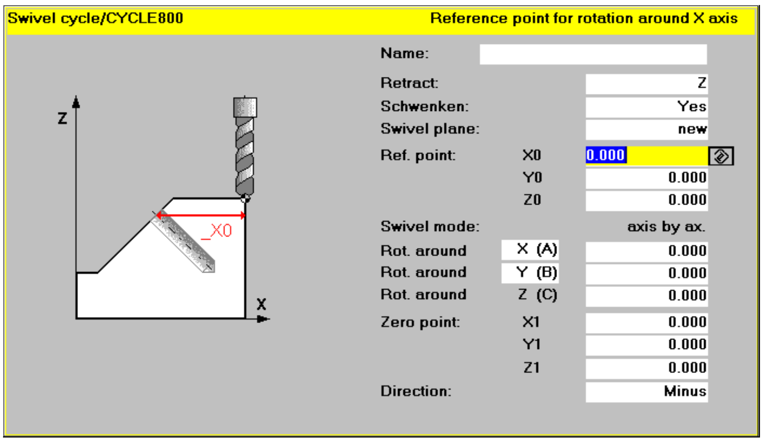
_FR	int	Retraction Value: 0...Do not retract 1...Retract axis Z (default)[1] 2...Retract axis Z, X, Y[1] 1) can be adjusted in user cycle TOOLCARR
_TC	String[20]	Name of swivel data record " Swivel data record 1 (default) "HEAD1" Name of swivel data record "0" Deselection of data record
_ST	int	Swivel plane UNITS DIGIT: Value: 0... new 1... additive TENS DIGIT[2]: Value: 0x...Do not correct tool tip 1x... Correct tool tip HUNDREDS DIGIT: Value: reserved 2) Precondition: Option TRAORI must be available
_MODE	int	Swivel mode Value: 0x...axial (default) 4x...Solid angle 8x...Projection angle Evaluation of angle: 7 6 5 4 3 2 1 0 Bits 1 0 – Angle of rotation 1: 01: Rotation around 1st axis 10: Rotation around 2nd axis 11: Rotation around 3rd axis Bits 3 2 – Angle of rotation 2: 01: Rotation around 1st axis 10: Rotation around 2nd axis 11: Rotation around 3rd axis Bits 5 4 – Angle of rotation 3: 01: Rotation around 1st axis 10: Rotation around 2nd axis 11: Rotation around 3rd axis Bits 7 6: 00: Swivel angle per axis (_A, _B, _C) 01: Solid angle (_A, _B) 10: Angle of projection (_A, _B, _C) Note Bits 0 to 5 have no meaning for solid angle

`_X0, _Y0, _Z0`	real	Reference point prior to rotation
`_A`	real	1. Axis angle (axial swivel mode) 2. Angle of rotation in XY plane about the Z axis (swivel mode 'solid angle') 3. Axis angle (swivel mode 'projection angle') sequence of axes
`_B`	real	1. Axis angle (axial swivel mode) 2. Angle of rotation in space about the Y axis (swivel mode 'solid angle')
`_C`	real	Axis angle (swivel modes 'axial', 'projection angle')
`_X1, _Y1, _Z1`	real	Zero point after rotation
`_DIR`	int	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. Value: -1 (minus)...lower rotary axis value (default) +1 (plus)...higher rotary axis value 0...no movement of rotary axes (calculation only)

Programming example 1

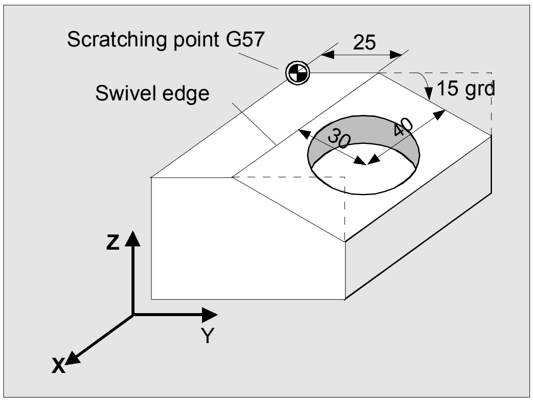
Set swivel plane ZERO

```
%_N_SWIVEL_0_SPF
;$PATH=/_N_WCS_DIR/_N_HAA_SWIVEL_WPD
G54
CYCLE800(1,"",0,57,0,0,0,0,0,0,0,0,0,-1)
M2
```

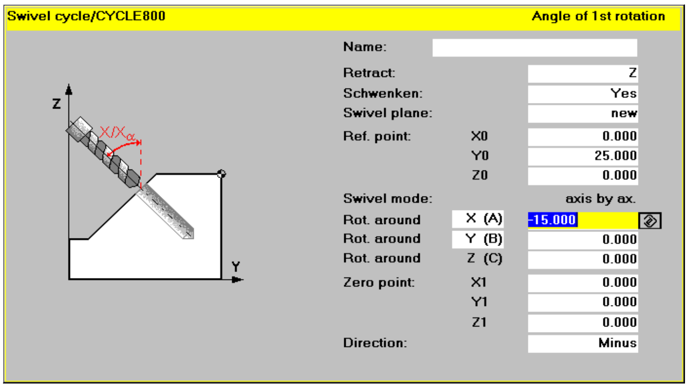


Programming example 2

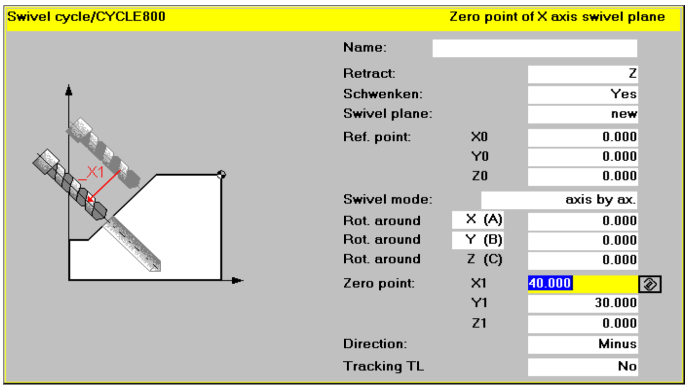
Face milling and milling a circular pocket on a machining plane swiveled through 15 degrees




```
%_N_SWIVEL_CIRCULARPOCKET_SPF
;$PATH=/_N_WCS_DIR/_N_HAA_SWIVEL_WPD
N12 T="MILL_26mm"
N14 M6
N16 G57
N18 CYCLE800(1,"",0,57,0,0,0,0,0,0,0,0,0,1)
N20 M3 S5000
N22 CYCLE71(50,2,2,0,0,0,80,60,0,4,15,5,0,2000,31,5)            ;Face milling
N24 CYCLE800(1,"",0,57,0,25,0,-15,0,0,0,0,0,-1)
```




```
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)
```



```
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 Starting up CYCLE800

When CYCLE800 is started up, the data (swivel data record) is set in the tool data $TC_CARR1...40. This is grouped to form Swivel start-up menus.
→ Operating area "Start-up"; Soft key "Swivel cycle"

References: for current infos see:

- the file "siemensd.txt" from the delivery software (standard cycles) or
- for HMI Advanced, F:\dh\cst.dir\HLP.dir\siemensd.txt.

The following cycles must be loaded:

- CYCLE800.SPF, CYCPE_SC.SPF (standard cycles)
- TOOLCARR-SPF (user cycle)
- PROG_EVENT.SPF (manufacturer cycle).

The GUD variables _TC_FR to _TC_NUM (GUD7.def) must be activated.

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)
  → is machine data that must not be changed
- Machine data with a variable value assignment (V)
  → is machine data where the default value can be set
  to a higher or lower value.

MD No.	MD identifiers	Value	Comments	Can be changed
10602	$MN_FRAME_GEOAX_CHANGE_MODE	1	1)	V
11450	$MN_SEARCH_RUN_MODE	Bit 1=1	1)	G
18088	$MN_MM_NUM_TOOL_CARRIER	n>0	n → Number of swivel data records 1)	G
20108	$MC_PROG_EVENT_MASK	0	-	V
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
20126	$MC_TOOL_CARRIER_RESET_VALUE	0...n	is described in CYCLE800	V
20150	$MC_GCODE_RESET_VALUES[41]	1	TCOABS 1)	G
20150	$MC_GCODE_RESET_VALUES[51]	2	PAROT 1)	G
20150	$MC_GCODE_RESET_VALUES[52]	1	TOROT 1) (for kinematics type T and M only)	V

1) For note on machine data, see next page

MD No.	MD identifiers	Value	Comments	Can be changed
20152	$MC_GCODE_RESET_MODE[41]	0	(default)[1)]	G
20152	$MC_GCODE_RESET_MODE[51]	0	(default)[1)]	V
20152	$MC_GCODE_RESET_MODE[52]	0	(default)[1)]	V
20180	$MC_TOCARR_ROT_ANGLE_INCR[0]	0	(default)[1)]	G
20180	$MC_TOCARR_ROT_ANGLE_INCR[1]	0	(default)[1)]	G
20182	$MC_TOCARR_ROT_ANGLE_OFFSET[0]	0	(default)[1)]	G
20182	$MC_TOCARR_ROT_ANGLE_OFFSET[1]	0	(default)[1)]	G
20184	$MC_TOCARR_BASE_FRAME_NUMBER	-1	(default)[1)]	G
22530	$MC_TOCARR_CHANGE_M_CODE	0	[1)]	V
24006	$MC_CHSFRAME_RESET_MASK	Bit 4=1	if system frame WPFRAME is to remain active after a reset	V
24008	$MC_CHSFRAME_POWERON_MASK	Bit 4, 3, 2=1	if system frames PAROT, TOROT, WPFRAME are to be cleared on power ON	V
28082	$MC_MM_SYSTEM_FRAME_MASK	Bit 4, 3, 2=1	Setting up system frames	G
30455	MISC_FUNCTION_MASK	Bit 2, 0=1	for rotary axes defined as modulo axes[1)]	V

Setting data

SD no.	SD identifier	Value	Comments	Can be changed
42980	$SC_TOFRAME_MODE	3	kinematics type "T" and "M" only	V

Changes to the relevant machine data result in reorganization of the buffered memory (data loss!). Series start-up must be performed after setting the MD and before NCK reset.

References: /IAM/ HMI/MMC Installation and Start-Up Guide

1) For note on machine data, see this page and the next

Note on MD 10602:
If after TRAORI, the ZO is not to be reprogrammed. e.g. for tool correction.

Note on MD 11450/MD 20108:
Activate PROGEVENT after block search

Note about 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 20150/MD 20152:
Equals 0, if there is no plane changeover + SWIVEL (G17-G19).

Note on MD 20180/MD 20182:
The swivel data record with the Hirth tooth system is described in the CYCLE800 rotary axes start-up screen.

Note about MD 22530:
If more than one swivel data record is 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 special functions, the PLC can, for example, limit or invert the spindle speed.

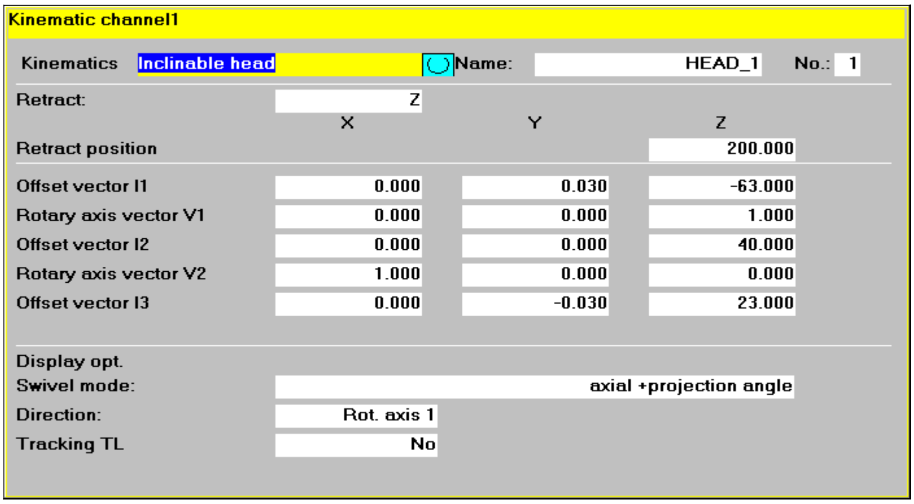
Note on MD 30455
This allows the axis to traverse for G90 with DC (shortest path); see user cycle TOOLCARR.spf

Soft key [Save data record]
Current swivel data record is saved as the parts program. The parts program corresponds to the name of the swivel data record.

Soft key [Delete data record]
Current swivel data record is deleted.

The CYCLE800 start-up process is supported by the following menu displays:

Start-up of kinematic chain



A swivel data record must be created for each swivel head, swivel table or each swivel head/table combination.
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 parameters $TC_CARR1[n] to $TC_CARR40[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 more than one swivel data record is declared in each NC channel, then a name is assigned to each record. No name need 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 +/-).

Only characters valid for NC programming must appear in the name!

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 start-up engineer determines whether the options 'Retract axis Z' and 'Retract axes Z,X,Y' are possible in the input menu for the swivel cycle.
The mode of retraction can be modified in user cycle TOOLCARR.spf (marker _M41, _M40) if necessary. If user cycle TOOLCARR.spf is not modified, the retraction position is approached as an absolute machine position.

⚠ Please note the following when moving 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 positions in the kinematic chain are measured by the machine manufacturer; they are always relevant with respect to a swivel head table (swivel data record). Offset vectors I1 to I4 refer to the **non-swiveled state of the rotary axes**.
The machine kinematics used need not be fully implemented from the control viewpoint. But then 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 with or without measuring systems are possible and can be used with "plain machines".

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 vector meanings are as follows:

References: Description of Functions
840D/840Di/810D
/W1/ "Tool Offset
(3/2 axis inclined machining)

- Offset vector I1
  → Distance between reference point of toolholder and rotary axis 1
- Offset vector I2
  → Distance between rotary axis 1 and rotary axis 2
- Offset vector I3
  → Distance between rotary axis 2 and tool reference point
- Offset vector I4
  → Distance between rotary axis 2 and table reference point
- Rotary axis vector V1
  → Direction of rotary axis 1
- Rotary axis vector V2
  → Direction of rotary axis 2.

The offset vectors must not necessarily point to the pivot point of the rotary axes, but it is important that they are directed towards a point in the direction of rotation.
The plus/minus sign of the rotary axis vectors is determined by the direction of rotation of the particular rotary axis around the corresponding machine axis.
→ see start-up examples.

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 (Tens digits)

- 0:
  0: Axial
  1: Axial + angle of projection
  2: Axial + angle of projection + solid angle
- 1: Rotary axis 1
  0: automatic
  1: manual
- 2: Rotary axis 2
  0: automatic
  1: manual
- 3: Selection axis feed direction
  0: no
  1: Reference to rotary axis 1
  2: Reference to rotary axis 2
- 4: Correct tool tip
  0: no
  1: yes
- 5: reserved
- 6: Retraction mode
  0: Z axis
  1: Z axis or ZXY axis
- 7: Automatic compensation with Hirth tooth system
  0: no
  1: 1st rotary axis
  2: 2nd rotary axis
  3: 1st and 2nd rotary axis
- 8: Swivel data record change / tool change 1)

0:	no	2)
1:	manual	automatic
2:	automatic	automatic
3:	no	2)
4:	manual	manual
5:	automatic	manual

1) Relevant for ShopMill/ShopTurn only.
2) If a swivel data record change has not been agreed, the setting "Tool change automatic/manual" is not relevant.

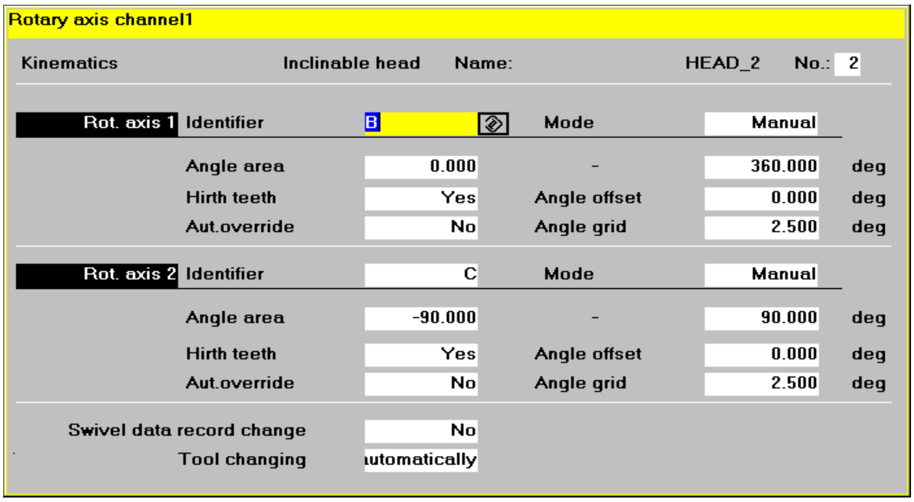
The following display options affect the input screen form for the swivel cycle:

- **Swivel mode**
  → **axial**
  → **axial and projection angle**
  → **axial and projection angle and solid angle**
  Example:
  Selection of swivel mode during start-up:
  Axial, angle of projection
  In this case, only 'axial' or 'projection angle' is included in the input menu. Programming of a solid angle is not desired or possible in this example.

- **Direction**
  → **Rotary axis 1**
  → **Rotary axis 2**
  → **no**
  When direction of travel is selected in the swivel cycle interactive screen form, reference to **rotary axis** 1 or 2.
  Because of the angular range of **rotary axes** of machine kinematics, the NCU calculates 2 possible solutions. One of these solutions is normally appropriate technologically. The selection as to which **rotary axis** the two solutions must refer to is made in the start-up menu. The selection of the solution to be applied is made in the interactive screen form for the swivel cycle.
  With "no", the direction parameter is not displayed in the operating screen.

- **Correct tool**
  → **no**
  → **yes**
  "Correct tool" display in the swivel cycle interactive screen form. The function correct tool the option 5-axis transformation (TRAORI).
  In the user cycle, query TOOLCARR.spf Variable GUD7 _TC_N_WZ.

Starting up rotary axis parameters



Input of data relevant to the swivel cycle for rotary axes 1 and 2.

Name/kinematics→ see CYCLE800 "Kinematics" start-up menu

Identifiers

$TC_CARR35[n] rotary axis 1
$TC_CARR36[n] rotary axis 2
Axis identifiers of rotary axes. 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 axes of the NCU are known, the same axis identifiers of the relevant NC rotary axes must be selected (see automatic mode).
If the axes of the NCU are not known, no known identifiers can be used.

Mode

$TC_CARR37[n] see display options

- **Automatic**
  NC rotary axes are automatically moved to the appropriate swivel angle
- **Manual**
  Rotary axes are moved manually by the operator to the appropriate position. On the "plainest machines" with manually adjustable **rotary axes** (measuring system: Steel gauge), the axis identifier need not be registered with the NCU.

Even mixed machine kinematics
(e.g. 1st rotary axis automatic, 2nd rotary axis manual),
and "incomplete" machine kinematics
(e.g. 1st rotary axis rotates around axis X) are allowed.
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 must not be the software end position range of the particular rotary axis.
With modulo axes, enter a traversing range of between **0 and 360 degrees**.

Hirth tooth system
$TC_CARR26[n]... $TC_CARR29[n]

- **no**
  The fields below are concealed.
- **yes**
  → **Angular offset** of Hirth tooth system at beginning of gearing.
  → **Angular grid** of Hirth tooth system
  → **Automatic correction yes/no**
  The tool might exit the angular grid set for the Hirth tooth system if the swivel head jams. If this happens, the swiveled frame will need to be recalculated (TCOABS) with the current angular values (of the Hirth tooth system). This functionality is calculated by the **Automatic correction (yes)** setting in the swivel cycle.

Swivel data record change (only relevant for ShopMill/ShopTurn)

- **no**
- **automatic**
- **manual**

Tool change (only relevant for ShopMill/ShopTurn)

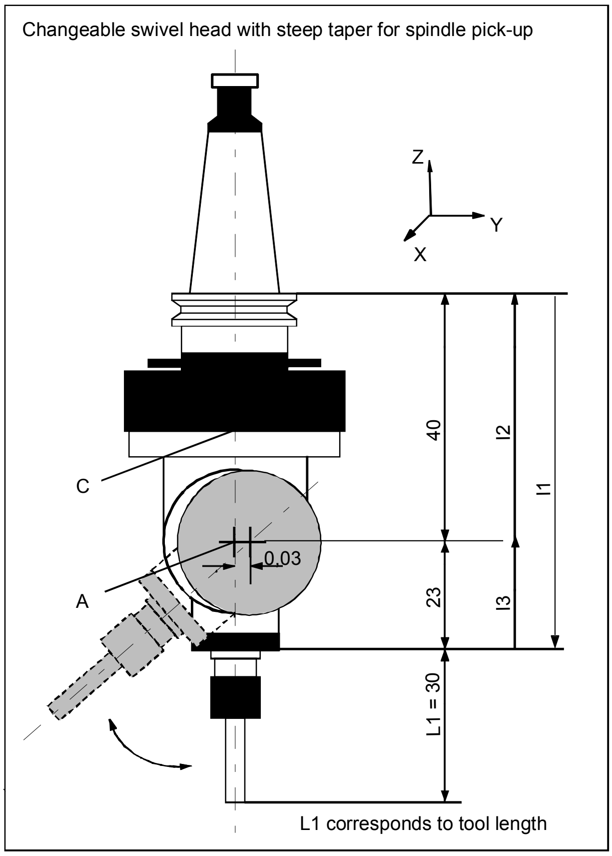
- **automatic**
- **manual**

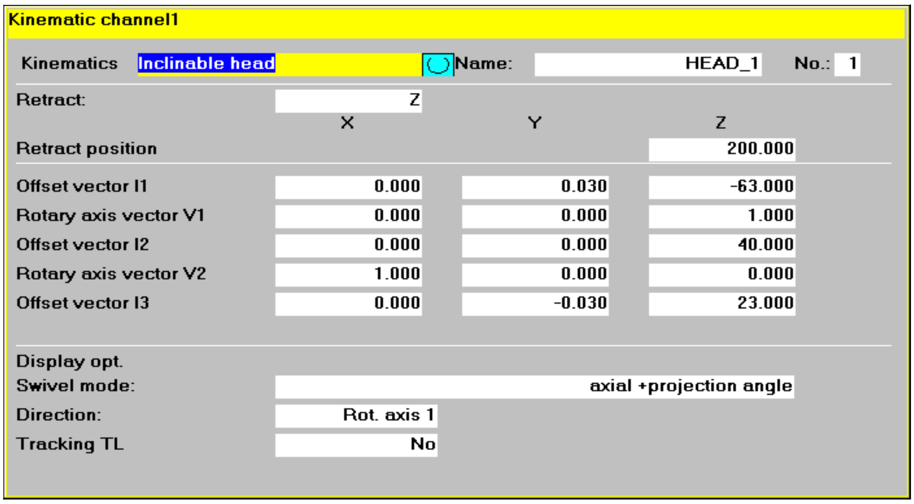
"Tool change" display for kinematic type T and M only

Start-up 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
(drawing not true-to-scale)





Kinematic channel1

Kinematics: Inclinable head — Name: HEAD_1 — No.: 1

Retract: Z

	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

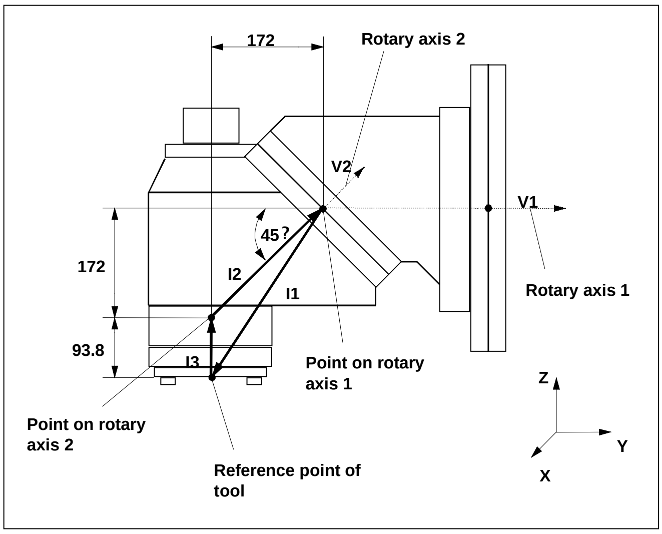
Display opt.
Swivel mode: axial +projection angle
Direction: Rot. axis 1
Tracking TL: No

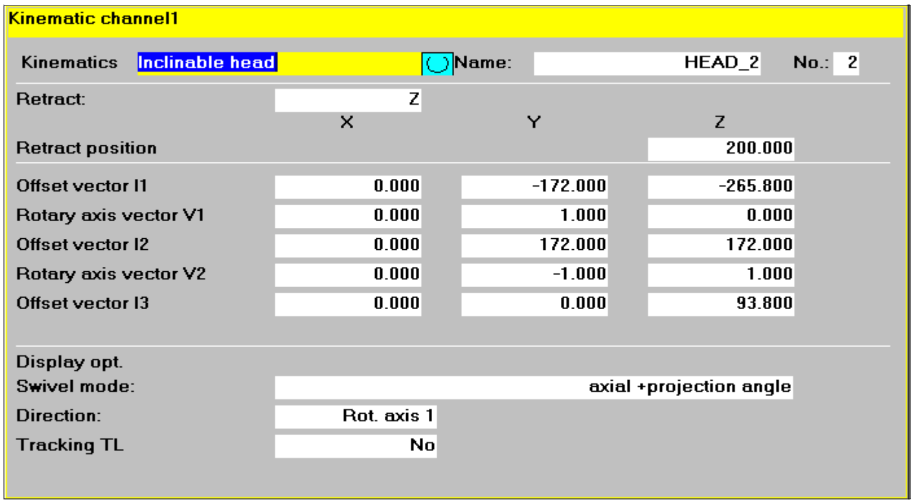
Example 2: Swivel head 2 "HEAD_2"

Offset vector I1: Distance between point on **rotary axis** 1 and the reference point of the tool

Offset vector I2: Distance between point on **rotary axis 2** and point on **rotary axis** 1

Offset vector I3: Distance between the reference point of the tool and the point on **rotary axis** 2





Kinematic channel1

Kinematics: Inclinable head | Name: HEAD_2 | No.: 2

Retract: Z

	X	Y	Z
Retract position			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	1.000
Offset vector I3	0.000	0.000	93.800

Display opt.

Swivel mode: axial +projection angle

Direction: Rot. axis 1

Tracking TL: No

Example 3: Table 2

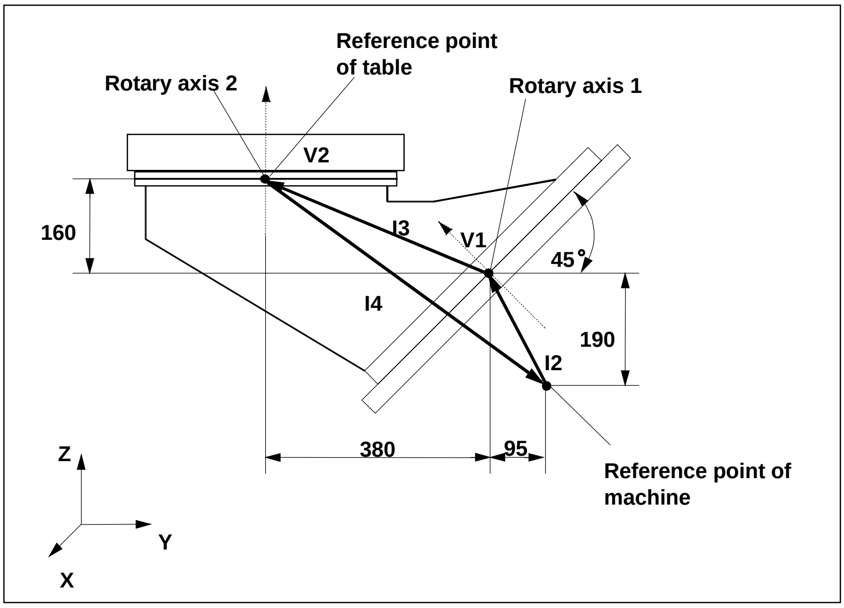
In this example, the offset vectors are not defined as described for example 3, but as a closed system as defined below:

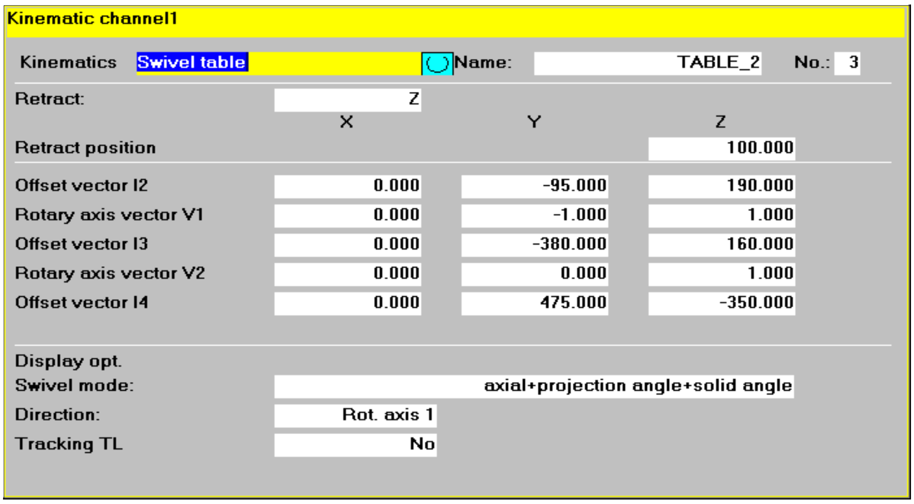
Offset vector I2: Distance between machine reference point and point on **rotary axis** 1

Offset vector I3: Distance between point on **rotary axis** 1 and point on **rotary axis** 2

Offset vector I4: Distance between point on **rotary axis** 2 and machine reference point

The advantage of this method is that the position values do not changed in the unswiveled state, regardless of whether or not the swivel table is selected.





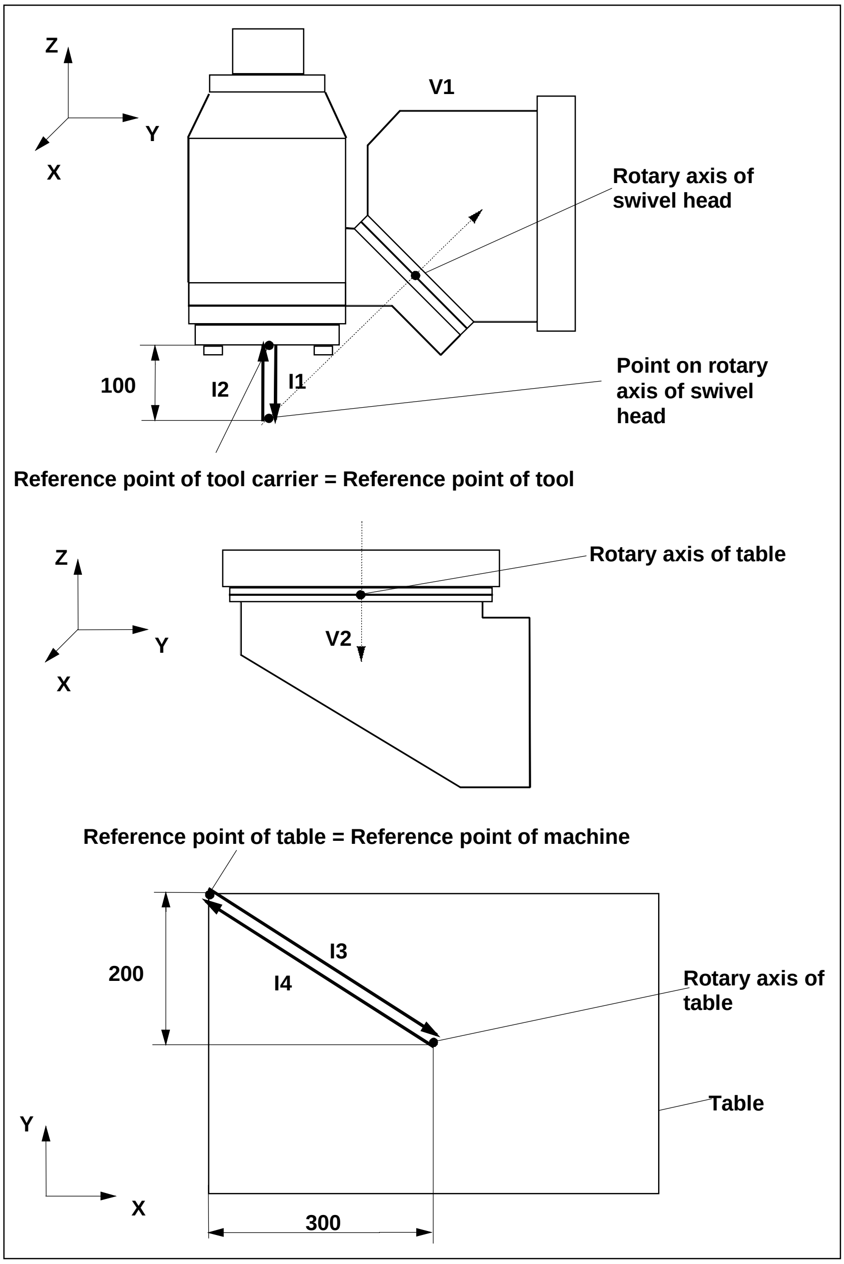
Example 4: MIXED 2

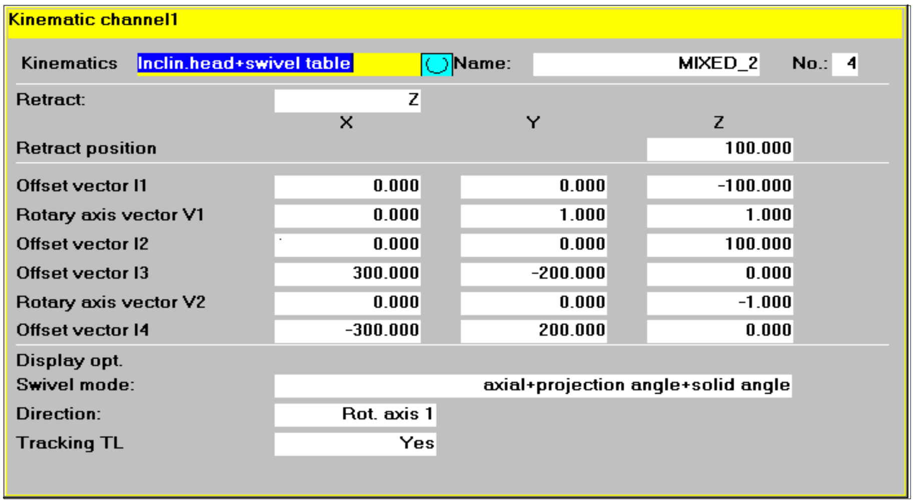
In this example, the reference points of the toolholder and the tool are identical, as well as the reference points of the table and the machine.

Thus: I1 = -I2 and I3 = -I4

resulting in each case in a closed system.

The advantage of this method is that the position values do not change in the unswiveled state, regardless of whether the swivel head or table is selected.




Kinematic channel1
Kinematics
Inclin.head+swivel table
Name:
MIXED_2
No.:
4
Retract:
Z
X
Y
Z
Retract position
100.000
Offset vector I1
0.000
0.000
-100.000
Rotary axis vector V1
0.000
1.000
1.000
Offset vector I2
0.000
0.000
100.000
Offset vector I3
300.000
-200.000
0.000
Rotary axis vector V2
0.000
0.000
-1.000
Offset vector I4
-300.000
200.000
0.000
Display opt.
Swivel mode:
axial+projection angle+solid angle
Direction:
Rot. axis 1
Tracking TL
Yes

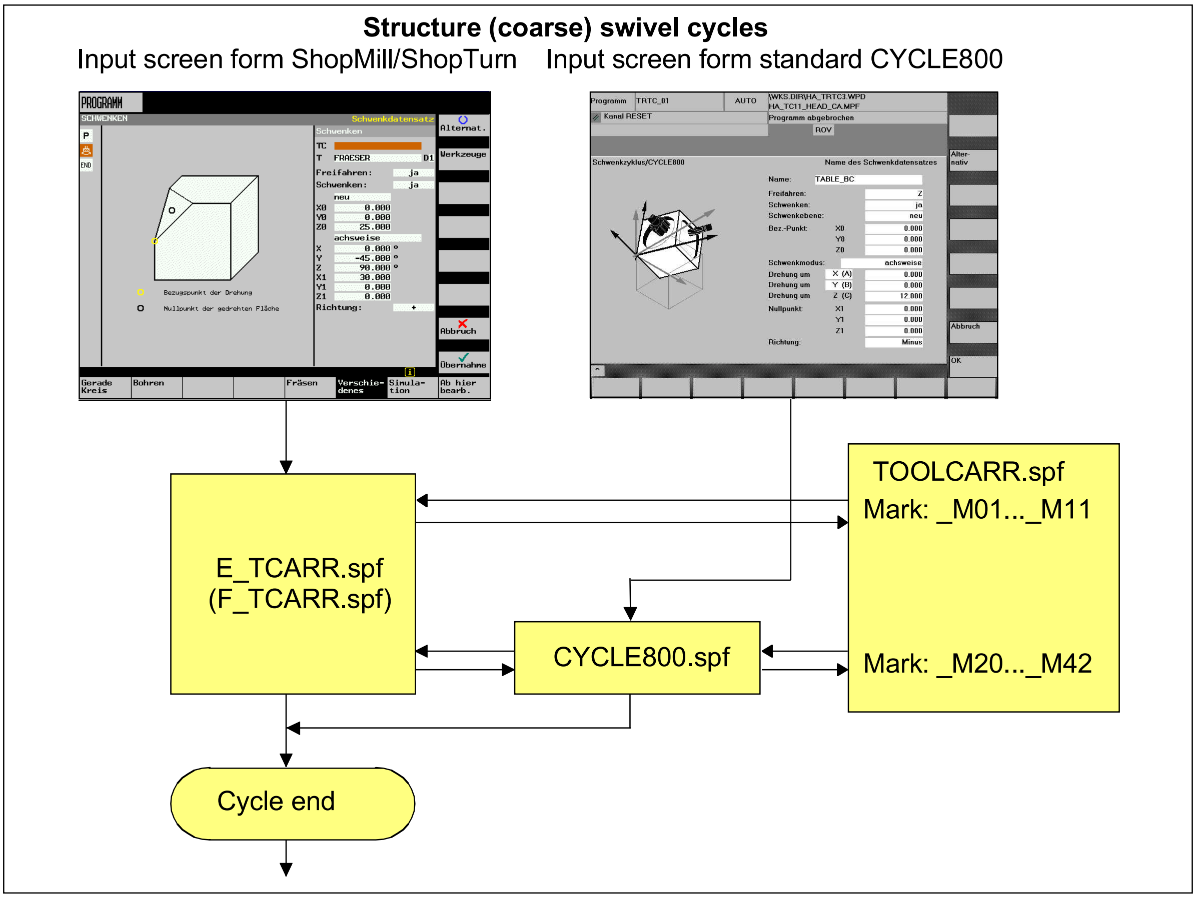
3.16.5 User cycle TOOLCARR.spf

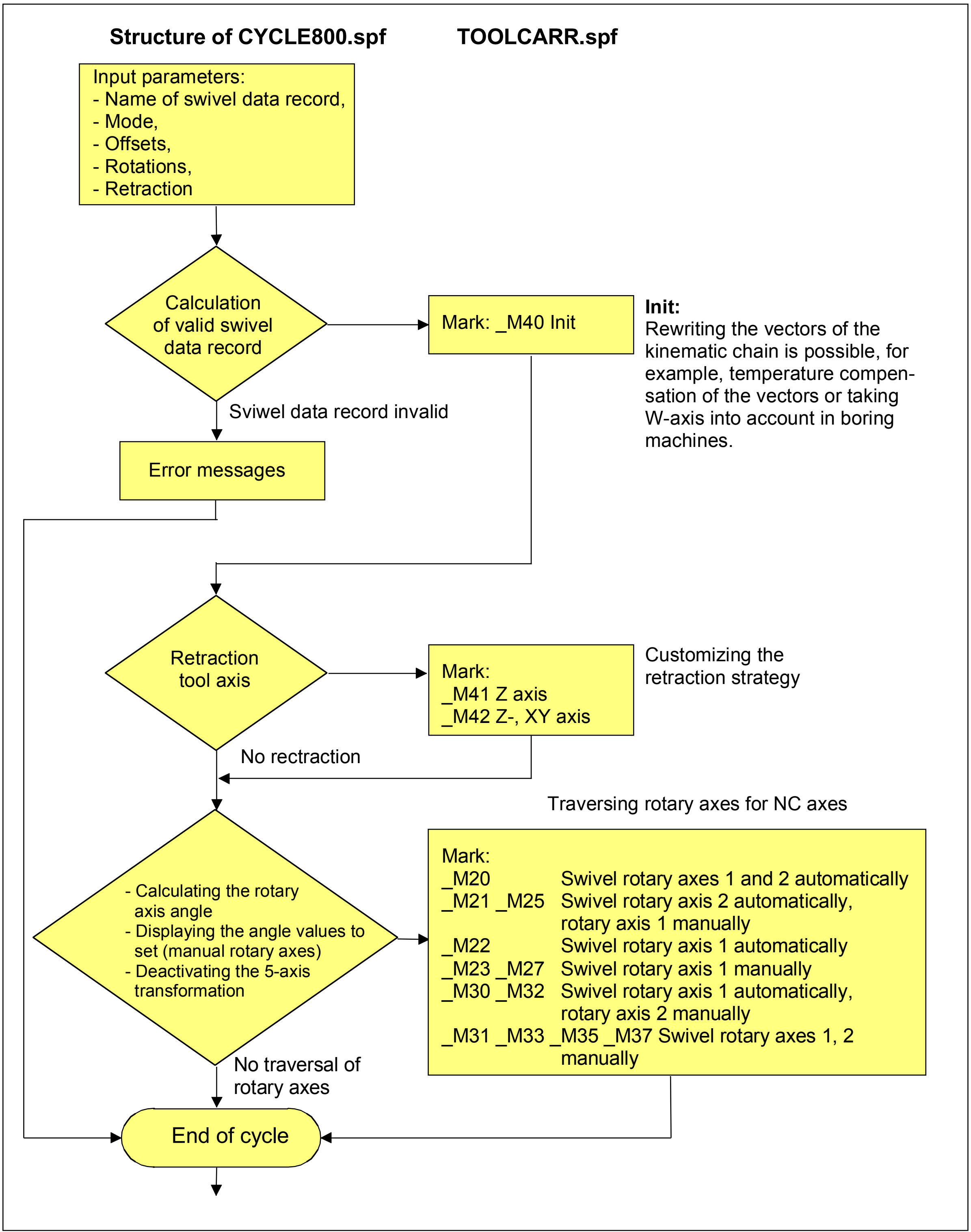
Customization by the machine manufacturer

All axis positions during swiveling are traversed by means of user cycle TOOLCARR.spf. This is called from the swivel cycle CYCLE800 or E_TCARR (ShopMill) or F_TCARR (ShopTurn). The cycle can be modified by the user (machine manufacturer during start-up) in order to cater to the properties of specific machines.

If the user cycle is not modified, during retraction before swiveling, axis Z (marker _M41) or axis Z is moved first, followed by axes X, Y (marker _M42).

The positions correspond to the start-up menu CYCLE800 "Kinematics" → retraction positions.




Structure of CYCLE800.spf
TOOLCARR.spf
Input parameters:
- Name of swivel data record,
- Mode,
- Offsets,
- Rotations,
- Retraction
Calculation of valid swivel data record
Mark: _M40 Init
Init:
Rewriting the vectors of the kinematic chain is possible, for example, temperature compensation of the vectors or taking W-axis into account in boring machines.
Sviwel data record invalid
Error messages
Retraction tool axis
Mark:
_M41 Z axis
_M42 Z-, XY axis
Customizing the retraction strategy
No rectraction
Traversing rotary axes for NC axes
- Calculating the rotary axis angle
- Displaying the angle values to set (manual rotary axes)
- Deactivating the 5-axis transformation
Mark:
_M20 Swivel rotary axes 1 and 2 automatically
_M21 _M25 Swivel rotary axis 2 automatically, rotary axis 1 manually
_M22 Swivel rotary axis 1 automatically
_M23 _M27 Swivel rotary axis 1 manually
_M30 _M32 Swivel rotary axis 1 automatically, rotary axis 2 manually
_M31 _M33 _M35 _M37 Swivel rotary axes 1, 2 manually
No traversal of rotary axes
End of cycle

Note on markers _M20 to _M37

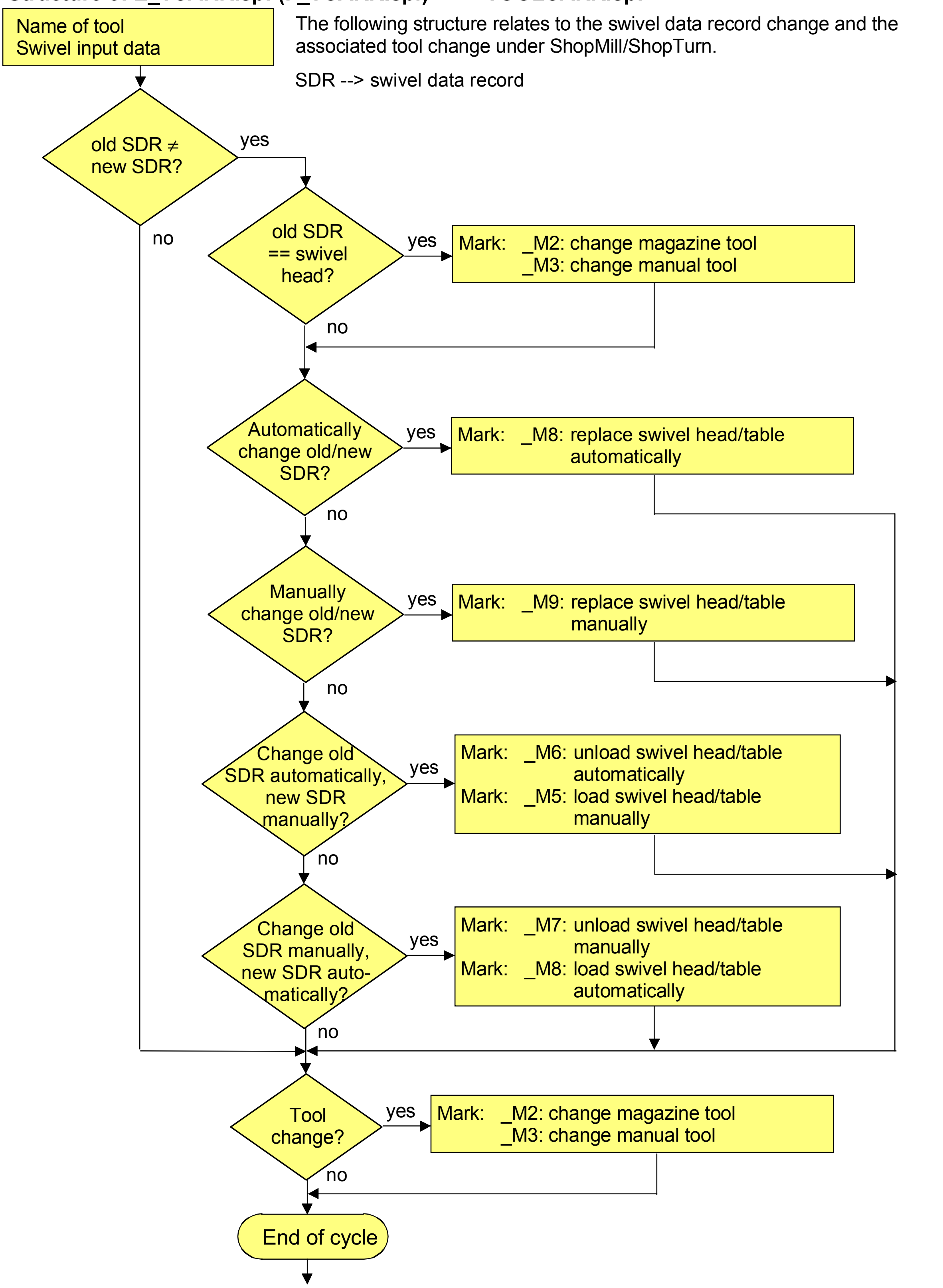
Markers _M20 to _M37 are distinguished by kinematics with two or one rotary axis. A distinction is also made between automatic rotary axes (known to the NCU) and manual rotary axes.

There is only ever one valid marker for the active swivel data record. Control via parameter/GUD7 variable _TC_ST.

Structure of E_TCARR.spf (F_TCARR.spf) TOOLCARR.spf

The following structure relates to the swivel data record change and the associated tool change under ShopMill/ShopTurn.

SDR --> swivel data record



Note on ShopMill/ShopTurn
In the user cycle Toolcarr.spf,
cycle E_SWIV_H or F_SWIV_H is called under
ShopMill/ShopTurn (see markers _M2 to _M9).

E_SWIV_H parameters (Par 1, Par 2, Par 3)

- Par 1: Number of the swivel data record (_TC1)
- Par 2: Angle of the 1st rotary axis
- Par 3: Angle of the 2nd rotary axis

Modification examples:
If the rotary axes (swivel head/table) are **not to be positioned** for swivel data change/tool change, the call of cycle E_SWIV_H can be commented out at the relevant markers.
If the rotary axes are to move to a certain position, an angle value can be transferred in parameters Par 2, Par 3.

3.16.6 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 as from NCU 6.3xx
61182	"Name of swivel data record unknown"	See Swivel cycle start-up CYCLE800 → Kinematics Name (swivel data record)
61183	"Retraction mode GUD7 _TC_FR outside value range 0..2"	See Swivel cycle start-up CYCLE800 → Retraction; 1st transfer parameter CYCLE800(x,...) is faulty >2
61184	"No solution can be found with current angle inputs"	
61185	"Rotary axis angle ranges incorrect (min>max) or not defined"	Check start-up of swivel cycle CYCLE800
61186	"Invalid rotary axis vectors"	Swivel cycle start-up CYCLE800: Rotary axis vector V1 or V2 not entered or incorrect
61187	"Block search computation end of block not valid for SWIVEL"	Select block search with calculation contour
61188	"No axis name 1st rotary axis declared"	Swivel cycle start-up CYCLE800: no entry under rotary axis 1 identifier
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

Typical display of the swivel angle to be set for a manual rotary axis in CYCLE800

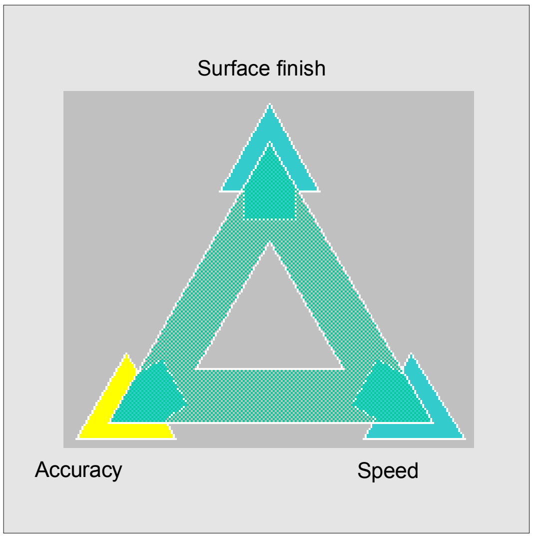
62180	"Set rotary axes B: 32.5 [deg]"

3.17 High Speed Settings – CYCLE832 (SW 6.3 and higher)

The standard cycle High Speed Settings CYCLE832 is available for the HMI in SW 6.3 and higher and NCU SW 6.3 (CCU SW 4.3).

Application of CYCLE832:

- for technological support when machining free-form contours (surfaces) in the 3- or 5-axis high-speed machining range (High Speed Cutting - HSC)
- used predominantly in the HSC milling range (possibility of use for turning or grinding)
- groups the important G codes and machine and setting data that are needed for HSC machining
- technology / geometry separation by 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 1 µm 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**, smoothing the contour puts the weighting on **speed**.
When **finishing**, enabling the NC block compressor puts the weighting on **accuracy**.
In both cases, specifying a tolerance retains the machining contour 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 the machining.

The following functions can be defined or enabled/disabled by CYCLE832:

- Tolerance band of the contour to be machined
- Smoothing (G64, G641, G642)
- NC block compressor (COMPCAD, COMPCURV, COMPOF)[1)]
- Feedforward control (FFWON, FFWOF)
- Jerk limitation (SOFT, BRISK)
- 5-axis transformation (TRAORI, TRAFOF) [1)]
- B spline

1) only if the relevant option is set.

Function

The CYCLE832 cycle groups the important G codes and the machine or setting data required for HSC machining.
In CYCLE832, a distinction is made between three technological executions:

- "Finishing",
- "Rough-finishing" and
- "Roughing".

In CAD programs, the three machining types are in the HSC range in direct relation to the accuracy and speed of the path contour (see help display). The operator / programmer uses the **tolerance value** to give relevant weighting.

Different tolerances and settings (customizing technology) can be assigned to the three machining types.

In the input screen form, the relevant G codes are preset (customizing technology) to ensure a smoothing path contour or that the CAM program is executed at optimum speed.
The cycle is located in the main menu before the CAM program (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:

- 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.

Take note of the machine manufacturer's comments!

Calling the "Machining deselection" cycle resets the changed machine/setting data to the value generated by the machine manufacturer (machine data settings).

If the machine manufacturer wants an additional CYCLE832 response that goes beyond the customization of the technology, the cycle can be copied to the directory CMA.dir (HMI manufacturer) and loaded to the NCU. In this case, CYCLE832.spf in CST.dir (HMI standard cycle directory) must be unloaded. The machine manufacturer must document the changes.

Typical CYCLE832 call

```
T1 D1
G54
M3 S12000
CYCLE832(0.2,1003)
EXTCALL "CAM_Form_Schrupp"
CYCLE832(0.01,102001)
EXTCALL "CAM_Form_Schlicht"
M02
```

CYCLE832 does not absolve the machine manufacturer from the requisite optimization tasks when commissioning the machine. This involves the optimization of the axes involved in machining and the settings of the NCU (block search, feedforward control, jerk limitation, etc.).

Reduced program call

The following call options of CYCLE832 with a reduced parameter transfer are possible:

- CYCLE832() corresponds to the selection of interactive screen form "Machining" "Deselection"
- CYCLE832(0.01) Entering the tolerance value. The active G commands are not changed in the cycle.

3.17.1 Calling CYCLE832 in the HMI menu tree

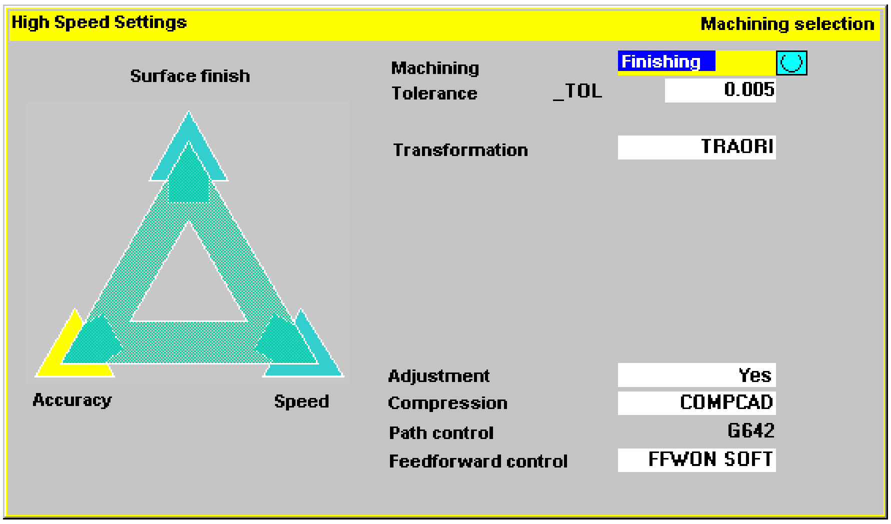
Description of parameters

Entry area Programs / **Milling**

Soft key [>>] → [High Speed Settings]

is displayed.

Interactive screen form CYCLE832 on standard interface



Machining (_TOLM)

- Finishing (default)
- Rough-finishing
- Roughing
- Deselection.

For variable _TOLM coding see Subsection 3.17.2, Parameters. At "Machining deselection", the G commands or the setting/machine data return to the value generated by the machine manufacturer.

Tolerance (_TOL)

The tolerance of the axes (corresponds to MD 33100: COMPRESS_POS_TOL[AX]) involved in machining. The tolerance value is effective for G642 and for COMPCURV or COMPCAD. If the machining axis is a rotary axis, the tolerance value is written with a factor (default factor = 8) to MD 33100: COMPRESS_POS_TOL[AX] of the rotary axis.

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**: 0.1 (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.

Transformation (_TOLM)
The transformation input field is only displayed when the NC option is set (machining package 5 axes set).

- **No**
  CAM programs with resolved rotary axis positions are supported.
- **TRAORI**
- **TRAORI(2)**

Selection of the transformation number or of the manufacturer cycle to call 5-axis transformation.
The parameter is in relation to the following GUD7 variables _TOLT2.
The name of a manufacturer cycle can be stored, resulting in calling the manufacturer cycle of transformation. If _TOLT2 is empty ("" default), selection transformation 1,2... calls the 5-axis transformation with TRAORI(1), TRAORI(2).

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".

Compression, NC block compressor (_TOLM)

- **None** (COMPOF)
- **COMPCAD** (default)
- **COMPCURV**
- **B SPLINE**

The input field is only displayed if the compressor function option is set.
The selection B spline is only made if the option spline interpolation is set.

Option → A, B and C splines/compressor function

Path control (_TOLM)

- **G642** (default)
- **G641**
- **G64**

With NC block compressor with COMPCAD, COMPCURV, G642 is always the permanent selection.

Feedforward control, velocity control (_TOLM)

- **FFWON SOFT** (default)
- **FFWOF SOFT**
- **FFWOF BRISK**

The selection of feedforward control (FFWON) and jerk limitation (SOFT) presupposes 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	int	Tolerance mode

7 6 5 4 3 2 1 0 (Tens digits)

- Digit 0:
  - 0: Deselection
  - 1: Finishing (default)[1)]
  - 2: Rough-finishing
  - 3: Roughing
- Digit 1:
  - 0:
  - 1:
- Digit 2:
  - 0: TRAFOF (default)[1)]
  - 1: TRAORI(1)
  - 2: TRAORI(2)
- Digit 3:
  - 0: G64
  - 1: G641
  - 2: G642 (default)[1)]
- Digit 4:
  - 0: FFWOF SOFT (default)[1)]
  - 1: FFWON SOFT
  - 2: FFWOF BRISK
- Digit 5:
  - 0: COMPOF
  - 1: COMPCAD (default)[1)]
  - 2: COMPCURV
  - 3: B spline
- Digit 6: reserved
- Digit 7: reserved

1) Setting may be changed by machine manufacturer, see Section "Customizing technology"

3.17.3 Customizing technology

With input field "Customizing technology" "Yes", both the machine manufacturer and the operator/programmer can customize the technology for HSC machining.
Always take note of the technology of the subsequent CAM program.

Customization by the machine manufacturer
Preconditions:

- **The manufacturer password is set**,
- Input field "Customizing technology" → "Yes"

When the interactive cycle screen form CYCLE832 is opened, the parameters are preassigned with the values of GUD7 variables _TOLV[n], _TOLT[n]. (n = machining: finishing, rough-finishing, roughing, deselection)
When the parameters are changed, the values are written directly to the GUD7 variables _TOLV[n] or _TOLT[n].
This gives the machine manufacturer the opportunity to customize the basic 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 0.3mm, 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, in CYCLE832, the velocity jumps overload factor is recalculated 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 (CYCLE832) **_OVL_on=0**.

2. The tolerance for the active NC block compressor or for smoothing is written in CYCLE832 to MD 33100: $MA_COMPRESS_POS_TOL[AX] (linear machining axes). If rotary axes are involved in the 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 local variable (CYCLE832) **_FAKTOR** can be preassigned with the relevant value.

Customization by the operator/programmer
Preconditions:

- **The manufacturer password is deleted**,
- Input variable "Customizing technology" → "Yes".

To customize the technology, the operator/programmer must have precise knowledge of the subsequent CAM machining program. The modified data is used to generate CYCLE832. The next time the tolerance cycle is called, the settings of the machine manufacturer will take effect again (GUD7 variables _TOLV[n], _TOLT[n] are not changed).

3.17.4 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),TRAOFOF
- UPATH

Note: The G commands should not be generated in the subsequent CAM program. Technology – Geometry separation.

Machine data

The following machine data is evaluated in cycle CYCLE832:

MD No.	MD identifiers	Comments
10071	$MN_IPO_CYCLE_TIME – interpolation cycle	
20480	$MC_SMOOTHING_MODE	
20482	$MC_COMPRESSOR_MODE	
20490	MC_IGNORE_OVL_FACTOR_FOR_ADIS	

The following machine data is rewritten in cycle CYCLE832:

MD No.	MD identifiers	Comments
33100	COMPRESS_POS_TOL[AX]	Geometry axes 1...3
33100	COMPRESS_POS_TOL[AX]	Rotary axes 1 and 2 [1)]
32310	MAX_ACCEL_OVL_FACTOR[AX]	Geometry axes 1...3
32310	MAX_ACCEL_OVL_FACTOR[AX]	Rotary axes 1 and 2 [1)]

1) as per 5-axis transformation machine data

Setting data

List of setting data rewritten in CYCLE832:

SD no.	SD identifier	Comments
42465	$SC_SMOOTH_CONTUR_TOL	Corresp. to the tolerance of the linear axes
42466	$SC_SMOOTH_ORI_TOL	Corresp. to the tolerance of the rotary axes
42475	$SC_COMPRESS_CONTUR_TOL	for COMPCURV only
42476	$SC_COMPRESS_ORI_TOL	for COMPCURV only
42477	$SC_COMPRESS_ORI_ROT_TOL	for COMPCURV only

1) The efficiency of setting data $SC_SMOOTH_CONTUR_TOL and $SC_SMOOTH_ORI_TOL depends on MD20480: $MC_SMOOTHING_MODE.
The efficiency of setting data $SC_COMPRESS_CONTUR_TOL and $SC_COMPRESS_ORI_TOL depends on MD20482: $MC_COMPRESSOR_MODE

GUD7 global variables

The following global variables must be activated for CYCLE832 functionality (machine manufacturer).
The definitions are a component of the GUD7 definitions of the SIEMENS standard cycle package.

Parameters	Format	Assignment	Comments
_TOLT2[2]	STRING[32]	"" (default)	Program name of the subroutine for calling 5-axis transformation
_TOLT[4]	int	**Field (4):** 0: Deselection 1: Finishing 2: Rough-finishing 3: Roughing	Field for saving the technological data settings of the machine manufacturer Coding corresponds to variable _TOLM (see parameters)
_TOLV[4]	real	**Field (4):** 0: Deselection 1: Finishing 2: Rough-finishing 3: Roughing	Field for saving the the tolerance values of the machining axes through the machine manufacturer settings (see customizing the technology). Default values: 0.1 Deselection (GUD7.def) 0.01 Finishing 0.05 Rough-finishing 0.1 Roughing

3.17.5 Error messages

Explanation

Alarm source CYCLE832

Alarm number	Alarm text	Explanation, remedy
61191	"5-axis transformation not set up"	1. Machining package five axes option or multi-axis interpolation 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"	

■

Turning Cycles

4.1 General information

The following sections describe how turning cycles are programmed. This section is intended to guide you in selecting cycles and assigning them with parameters. 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 to familiarize you with the use of cycles.

The sections are structured as follows:

- **Programming**
- **Parameters**
- **Function**
- **Sequence of operations**
- **Explanation of parameters**
- **Additional notes**
- **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 of operations", "Explanation of parameters", "Additional notes" and the "Programming example".

4.2 Preconditions

Data block for turning cycles

The turning cycles require module GUD7.DEF. It is supplied on diskette together with the cycles.

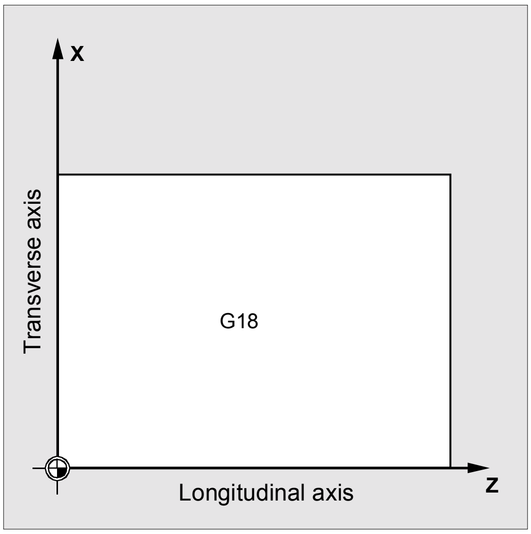
Call and return conditions

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

Plane definition

The machining plane must be defined 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 plane axis (second axis of this plane).

If diameter programming is active, the second axis of the plane is always taken as facing axis (see Programming Guide).



Spindle handling

The turning cycles are written in such a way that the spindle commands 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 (see Programming Guide).

Machining status messages

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

Cycle setting data

For the stock removal cycle CYCLE95, Software Release 4 and higher has provision for setting data that is stored in module GUD7.DEF.

Cycle setting data _ZSD[0] can be used to vary the calculation of the depth infeed MID 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

For the groove cycle CYCLE93, software release 4 and higher has provision for setting data in module GUD7.DEF. This cycle setting data _ZSD[4] can affect the retraction after the 1st groove.

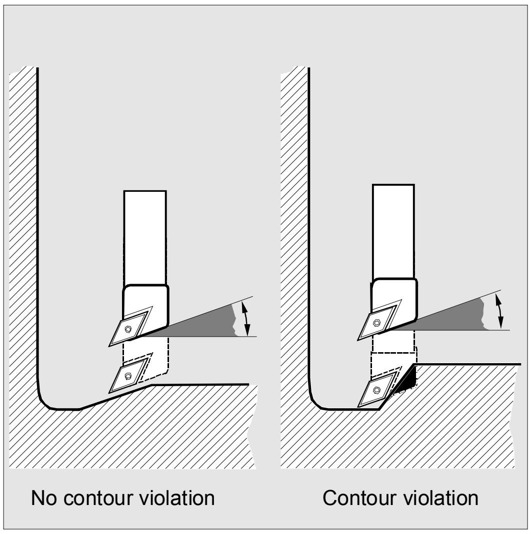
- _ZSD[4[=1 Retraction with G0
- _ZSD[4]=0 Retraction with G1 (as before)

With SW 6.2 and higher, there is an option for setting mirroring behavior in _ZSD[6] for grooving cycle CYCLE93.

- _ZSD[6]=0 The tool offsets are exchanged in the cycle when mirroring is active
  (for use without orientatable toolholder)
- _ZSD[6]=1 The tool offsets are not exchanged in the cycle with active mirroring
  (for use with orientatable toolholder)

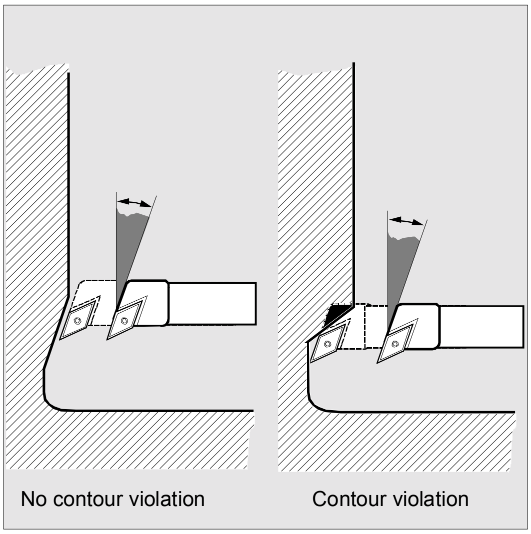
Contour monitoring with respect to tool clearance angle

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 reactions 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

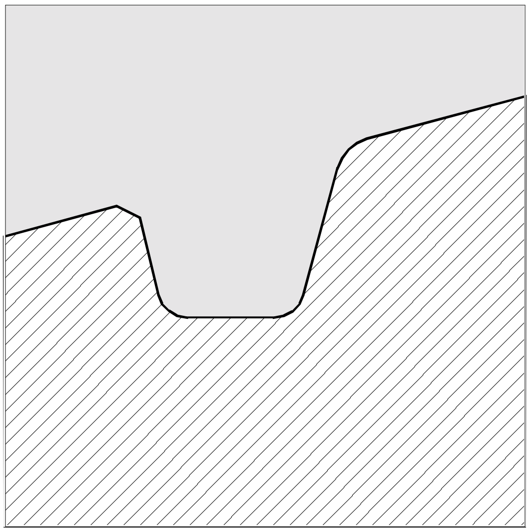
```
CYCLE93 (SPD, SPL, WIDG, DIAG, STA1, ANG1, ANG2, RCO1, RCO2, RCI1, RCI2,
FAL1, FAL2, IDEP, DTB, VARI, _VRT)
```

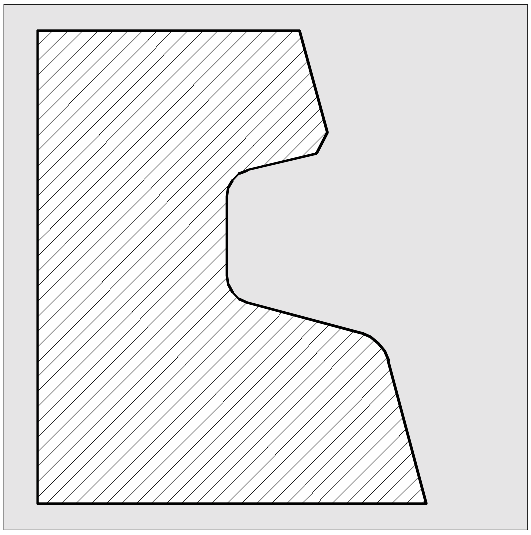
Parameters

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<=STA1<=180 degrees
ANG1	real	Flank angle 1: on the side of the groove defined by the starting point (enter without sign) Value range: 0<=ANG1<89.999 degrees
ANG2	real	Flank angle 2: on the other side (enter without sign) Value range: 0<=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 the 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	Type of machining Value range 1...8 and 11...18
_VRT from SW 6.2 and higher	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 traverse machining on straight contour elements. You can machine both external and internal grooves.



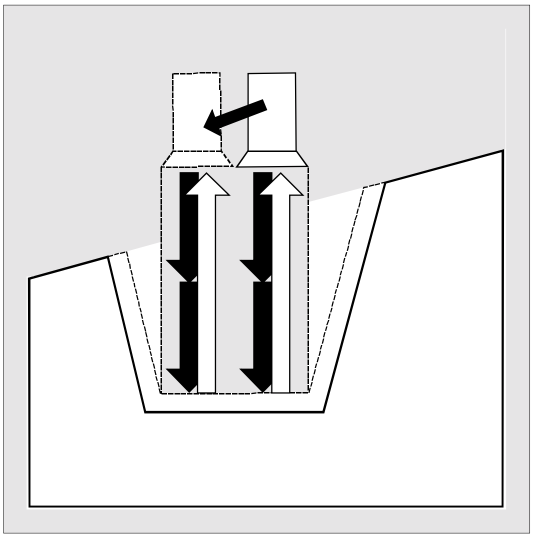


Sequence of operations

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.

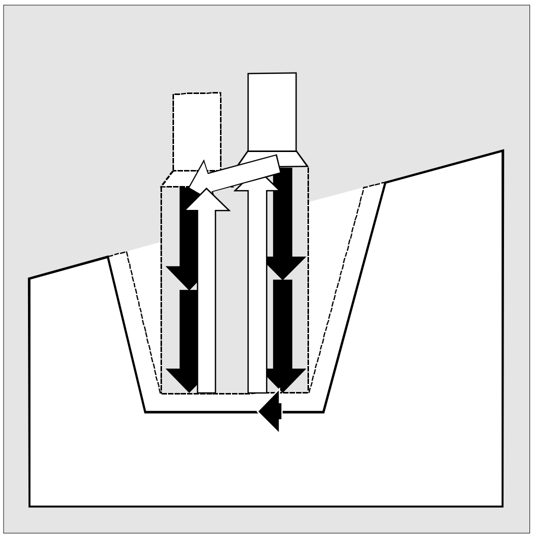
1st step

Paraxial roughing to the base of the groove in single infeed steps. After each infeed, the tool is retracted for chip breaking.



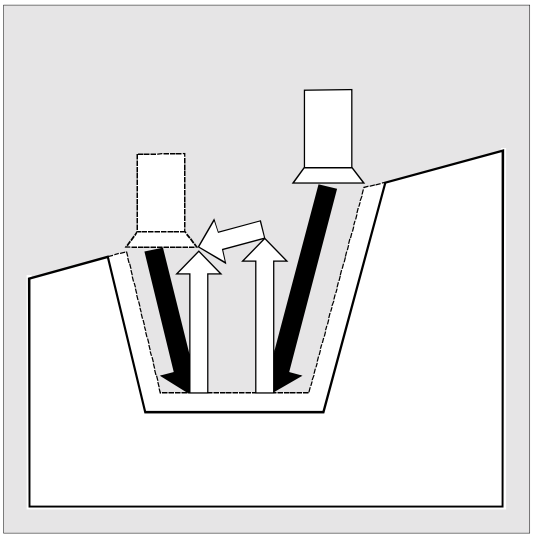
2nd step

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. From the second cut along the groove width the tool is withdrawn by 1 mm before it is fully retracted.



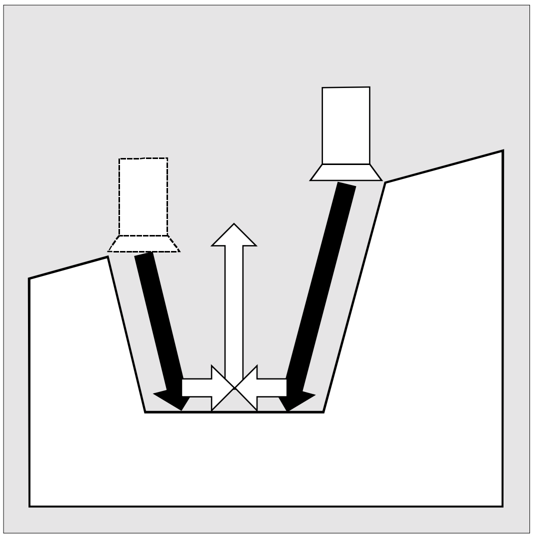
3rd step

Cutting of the flanks in one step, if angles are programmed under ANG1 or ANG2. The infeed along the groove width is performed in several steps if the flank width is larger.



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.

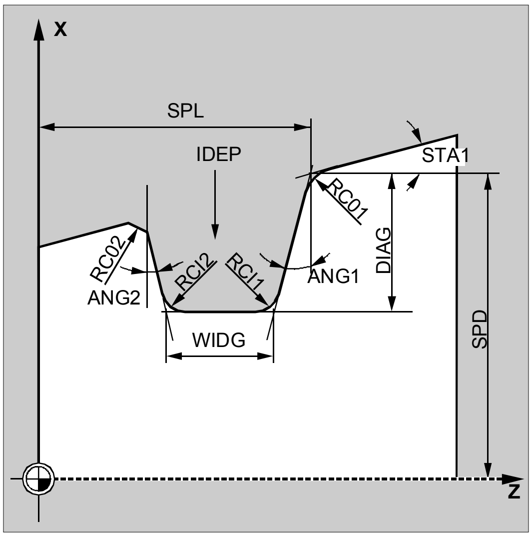


Description of parameters

SPD and SPL (starting point)

You define the starting point of the groove from where the cycle calculates the shape with these coordinates. The cycle itself determines the starting point to be approached at the beginning. In the case of an external groove, the longitudinal axis direction is first traversed and in the case of an internal groove, the facing axis direction is first traversed. Grooves on curved surfaces can be created 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.

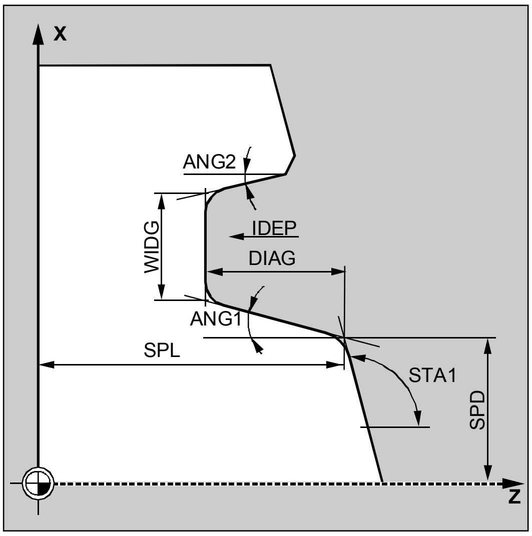
Radii and chamfers on the groove edge of a curved surface should only be programmed if the edge point in question is positioned on the straight line defined for the cycle.



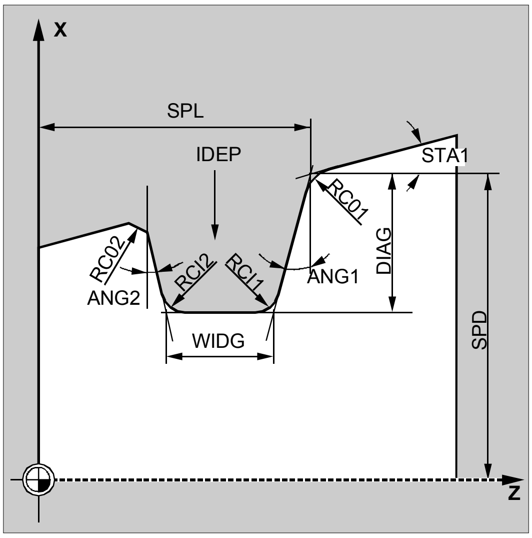
WIDG and DIAG (groove width and groove depth)

The shape of the groove is defined with the parameters groove width (WIDG) and groove depth (DIAG). The cycle always starts its calculation from the point programmed with SPD and SPL.

If the groove is wider than the active tool, the groove is machined in several steps. The total width is divided into equal sections in the cycle. The maximum infeed is 95 percent 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 error message 61602 "Tool width incorrectly defined" is output, 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.

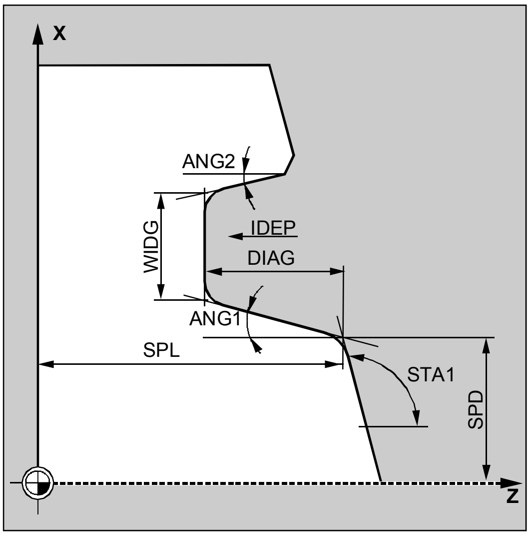


STA1 (angle)

The angle of the oblique surface on which the groove is to be machined is programmed with parameter STA1. 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.

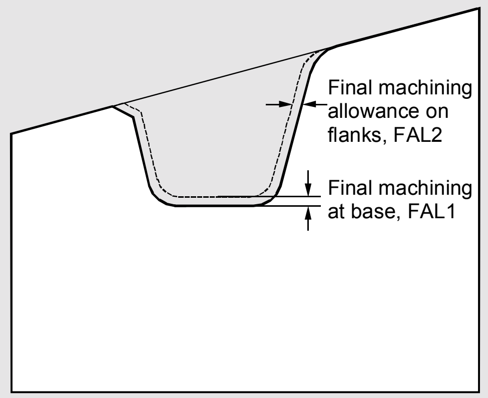


RCO1, RCO2 and RCI1, RCI2 (radius/chamfer)

The shape of the groove can be modified by entering radii/chamfers for the edge or base of the groove. The values for the radii must always be positive, the values for the chamfers must always be negative.
You can use the tens setting for the VARI parameter to determine the type of calculation for programmed milling.

- With VARI<10 (tens=0) the absolute value of this parameter is regarded as chamfer length (chamfering with CHF programming).
- With VARI>10, it is regarded as path length (chamfering with CHR programming).

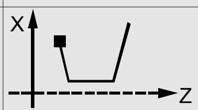
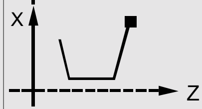
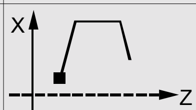
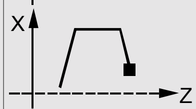
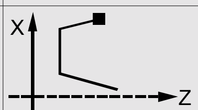
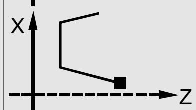
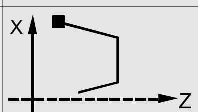
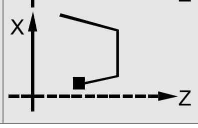
FAL1 and FAL2 (machining allowance)
You can program separate final machining allowances for the groove base and the flanks. Roughing is performed to this final machining allowance. Then, the same tool is used to machine a contour-parallel cut along the final contour.



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 for chipbreaking by 1 mm or, from SW 6.2 or higher onwards, by the distance programmed under _VRT. Parameter IDEP, anyway, is to be programmed.

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 (type of processing)
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.

VARI		Selection of cycle support
1/11		longitudinal, external, left
5/15		longitudinal, external, right
3/13		longitudinal, internal, left
7/17		longitudinal, internal, right
6/16		plane, external, top
8/18		plane, external, bottom
2/12		plane, internal, top
4/14		plane, internal, bottom

The tens value of the VARI determines the type of calculation for the chamfer.
VARI 1...8: Chamfers are calculated as CHF
VARI 11...18: Chamfers are calculated as CHF

Input of the ones position in cycle support is distributed among three selection fields:
1. Field: longitudinal/plane
2. Field: external/internal
3. Field: starting point left/right (for longitudinal) or top/bottom (for plane)

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)
With SW 6 and higher, the retraction path can be programmed on the basis of the outside or inside diameter of the groove in parameter _VRT. If _VRT=0 (parameter not programmed), liftoff is 1mm. The retraction path is always measured in 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.

Further notes

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 determines itself 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"
and the cycle is aborted.

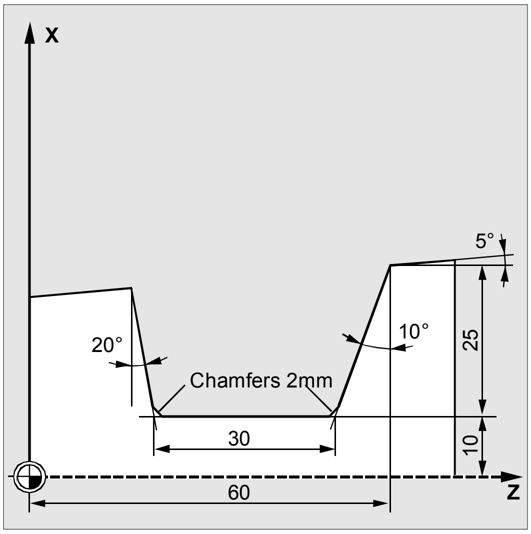
For SW 5.1 and higher, cycle setting data _ZSD[4] can be used to influence the retraction after the 1st groove.
_ZSD[4]=0 means retraction with G1 as before,
_ZSD[4]=1 means retraction with G0.

With SW 6.2 and higher, 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 correspondingly.



`DEF REAL SPD=35, SPL=60, WIDG=30, ->`
`-> DIAG=25, STA1=5, ANG1=10, ANG2=20, ->`
`-> RCO1=0, RCI1=-2, RCI2=-2, RCO2=0, ->`
`-> FAL1=1, FAL2=1, IDEP=10, DTB=1`
`DEF INT VARI=5`
	Definition of parameters with value assignments

`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, ->`
`-> STA1, ANG1, ANG2, RCO1, RCO2, ->`
`-> RCI1, RCI2, FAL1, FAL2, IDEP, ->`
`-> DTB, VARI)`
	Cycle call

`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

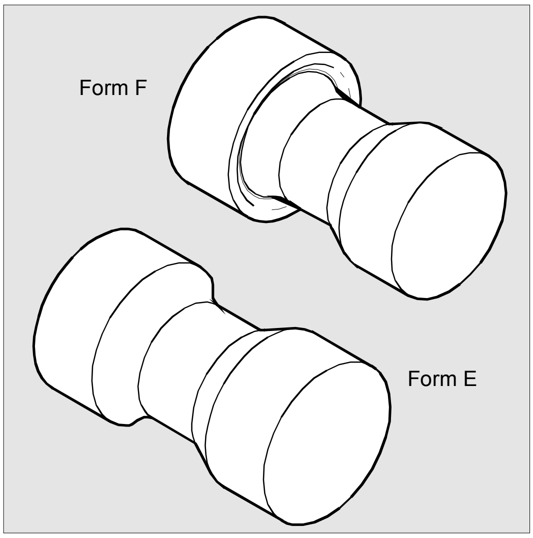
```
CYCLE94 (SPD, SPL, FORM, _VARI)
```

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 (SW 6.2 and higher)	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 >3mm.



Another cycle CYCLE96 exists for producing thread undercuts (see Section 4.6).

Sequence of operations

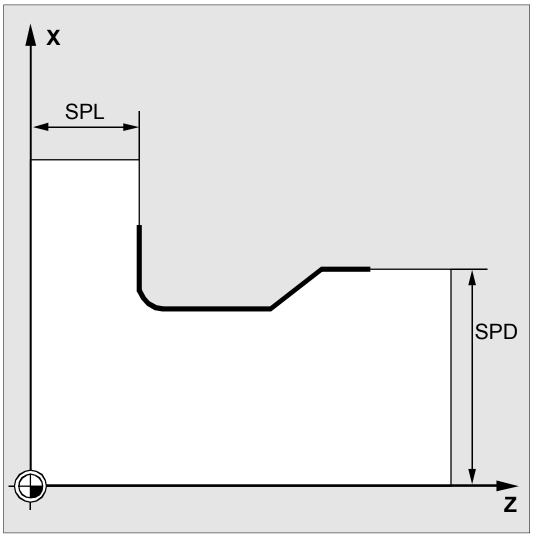
Position reached prior to cycle start:
The starting position can be any position from which the undercut can be approached without collision.

The cycle implements the following motion sequence:

- Approach to 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

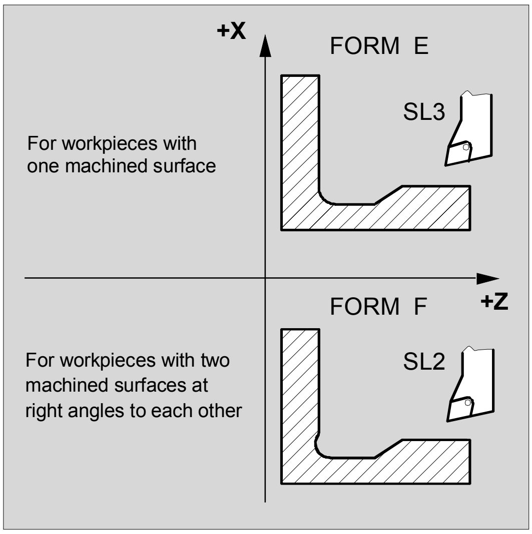
Description of parameters

SPD and SPL (starting point)
The finished part diameter for the undercut is entered in parameter SPD. With parameter SPL you define the finished part dimensions in the longitudinal axis.
If the value programmed for SPD results in a final diameter that is <3mm, the cycle is aborted with the 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 the parameter is assigned a value other than E or F, the cycle is aborted and alarm 61609 "Form incorrectly defined" is output.

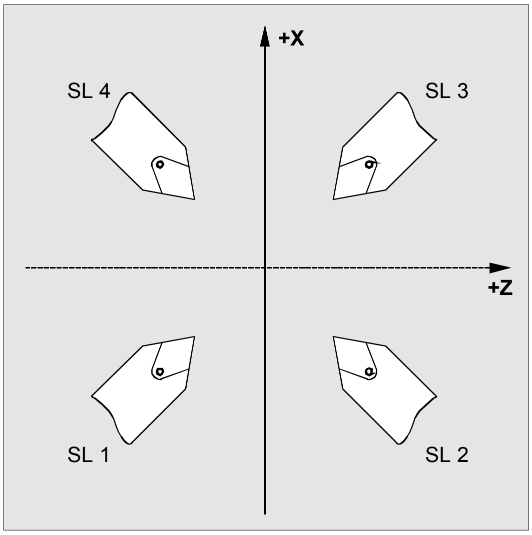


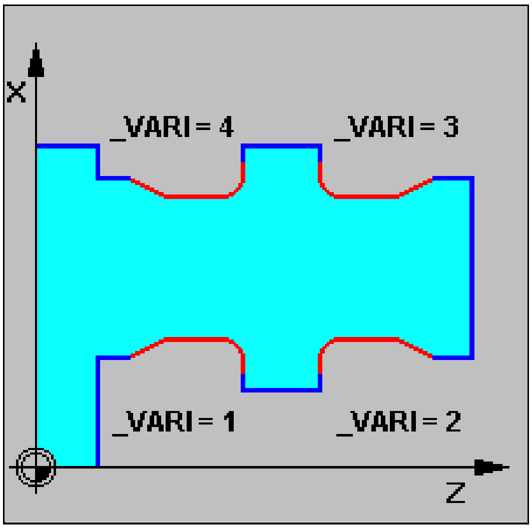
_VARI (undercut position)

The position of the undercut can be either specified directly or derived from the tool point direction with parameter _VARI.

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 recognizes a tool point direction 5 ... 9, alarm 61608 "Wrong tool point direction programmed" is output and the cycle is aborted.

_VARI=1...4: Definition of undercut position
The cycle determines the starting point automatically. This lies 2mm from the final diameter and 10mm 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" appears on the control, but machining is continued.





When _VARI<>0 the following applies:

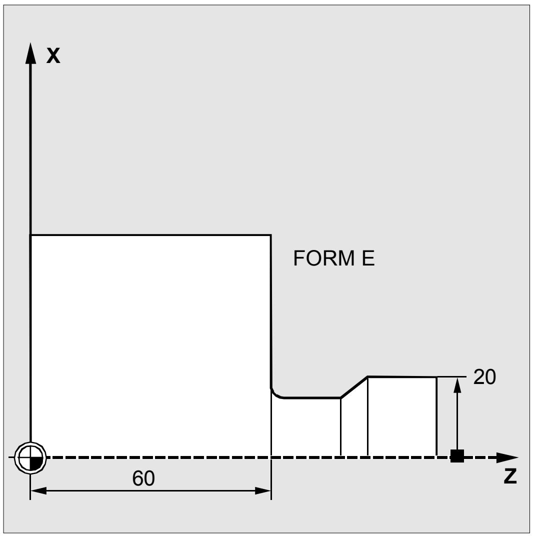
- The actual tool point direction is not checked, i.e. all directions can be used if this is meaningful technologically,
- No special consideration is given to functions such as adapter transformation, orientatable toolholder in the cycle. It is assumed that the user is familiar with the conditions on the machine and has specified the position correctly;
- Similarly, mirroring is not specially treated either, the user must be responsible for making the correct settings.

Further notes

You must activate a tool offset before you call the cycles. Otherwise alarm
61000 "No tool offset active"
is output and the cycle is aborted.

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

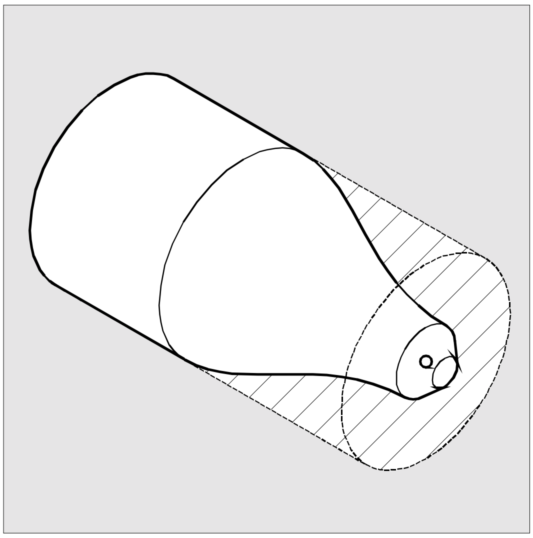
```
CYCLE95 (NPP, MID, FALZ, FALX, FAL, FF1, FF2, FF3, VARI, DT, DAM, _VRT)
```

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	Type of machining Value range: 1 ... 12 HUNDREDS DIGIT (SW 6.2 and higher): Values: 0...with rounding at the contour No residual corners will remain, the contour is rounded overlapping meaning that the contours are rounded over several points of intersection. 2...without rounding at the contour The machine always rounds to the previous roughing intersection and then retracts. Depending on the ratio of tool radius to infeed depth (MID), residual corners might 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 With SW 4.4 and higher`	real	Retraction distance from contour for roughing, incremental (enter without sign)

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 is freely selectable (roughing, finishing, complete machining). During roughing, 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 to the programmed final machining allowance.
Finishing is performed in the same direction as roughing. The tool radius compensation is automatically selected and deselected by the cycle.



Sequence of operations

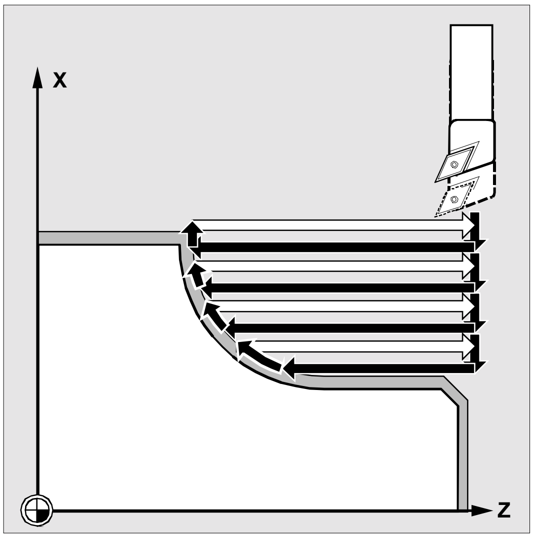
Position reached prior to cycle start:
The starting position can be any position from which the starting point of the contour can be approached without collision.

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.

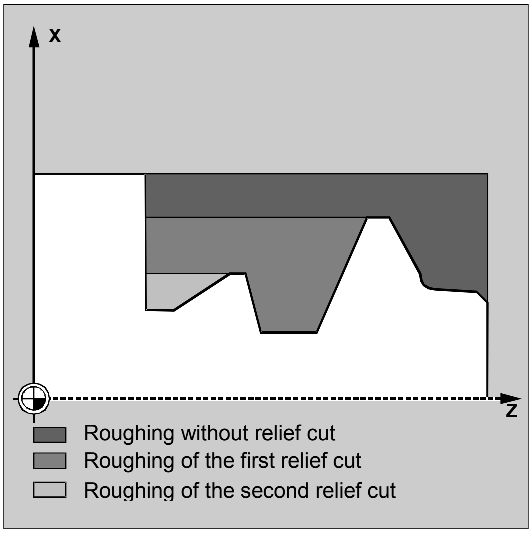
Roughing without relief cut elements:

- Paraxial infeed to the actual depth is calculated internally and then approached with G0.
- Approach roughing point paraxially with G1 and feedrate FF1.
- Machine 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 retraction with G0.
- This procedure is repeated until the total depth of the section to the machining step 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:

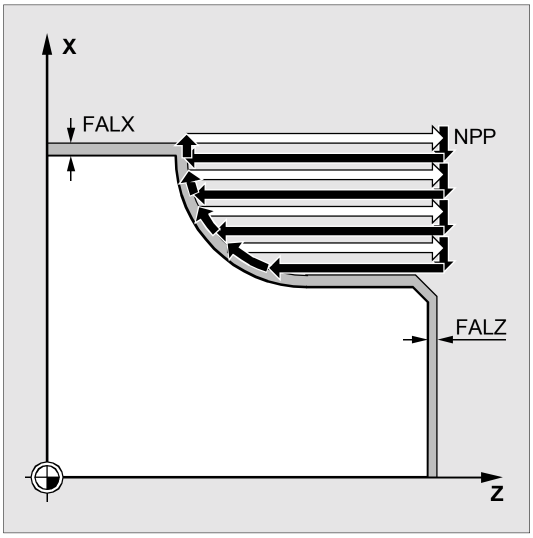
- Approach the starting point for the next relief cut axis by axis with G0. An additional safety distance is calculated internally.
- Infeed parallel to the contour + final machining allowance with G1/G2/G3 and FF2.
- Approach roughing point paraxially with G1 and feedrate FF1.
- Machine to the last roughing point. 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 an amount 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.

Description 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.

In SW 5.2 and higher, 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:

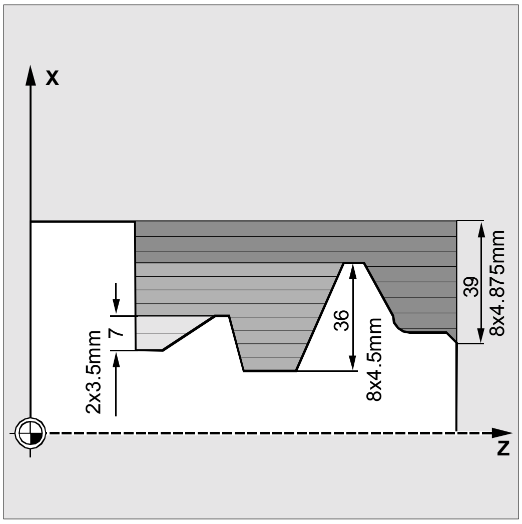
NPP="CONTOUR_1"	The machining contour is the complete program "Contour_1".
NPP="START:END"	The machining contour is defined as the section starting from the block labeled START to the block labeled END in the calling routine.
NPP="/_N_SPF_DIR/_N_CONTOUR_1_SPF:N130:N210"	The machining contour is defined in blocks N130 to N210 in program CONTOUR_1. Write the full program name together with path and extension, refer to CALL description in the References: /PGA/ Programming Guide Advanced.

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

MID (infeed depth)

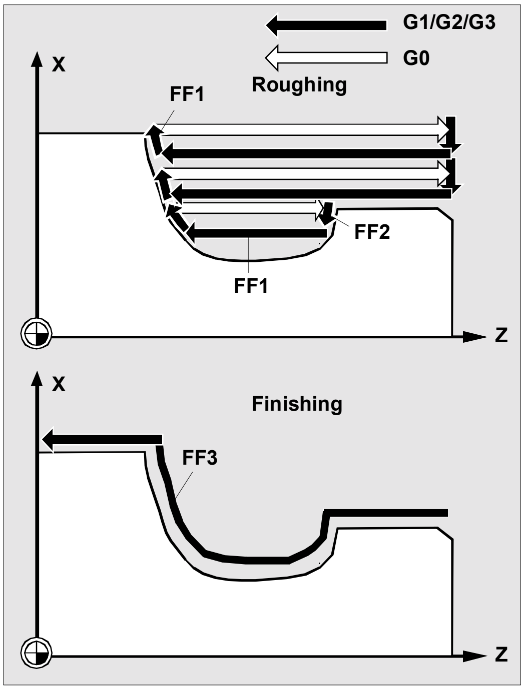
Under parameter MID you define the maximum possible infeed depth for the roughing operation.
The interpretation of this parameter depends on the cycle setting data for software release 4 and higher _ZSD[0] (see Section 4.2).
The cycle automatically calculates the actual infeed depth for roughing.
Where contours with relief cut elements are to be machined, the cycle divides up the roughing operation into single roughing steps. The cycle recalculates the actual infeed depth for every roughing step. This actual 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 for the calculation of the actual infeed depths:
Machining section 1 has a total depth of 39mm. If the maximum infeed depth is 5mm, eight roughing cuts are required. These are performed with an infeed of 4.875mm.
In machining section 2, eight roughing cuts, each with an infeed of 4.5mm are also executed (total difference 36mm).
Machining section 3 is roughed twice with an actual infeed of 3.5 (total difference 7mm).



FAL, FALZ and FALX (final machining allowance)
The final machining allowance for the roughing operation is either defined in parameters FALZ and FALX if you wish to enter different final machining allowances for each axis or in parameter FAL 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 **finish cut** machining mode does not consider programmed values. Turning always takes place down to finishing dimension.

FF1, FF2 and FF3 (feedrate)
You can define different feedrates for the different machining steps as is shown in the figure on the right.



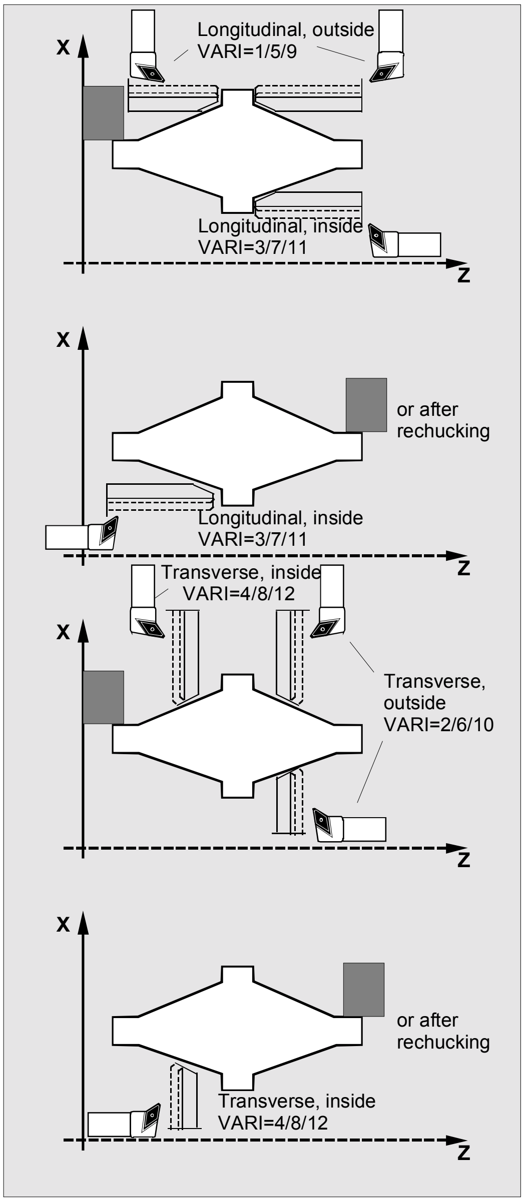
VARI (machining types)

You can call the machining types as follows:

Machining Roughing
(Roughing/Finishing/Complete)

Selection Longitudinal
(Longitudinal/Transverse)

Selection Outside
(Outside/Inside)



The machining types are shown in the table below.

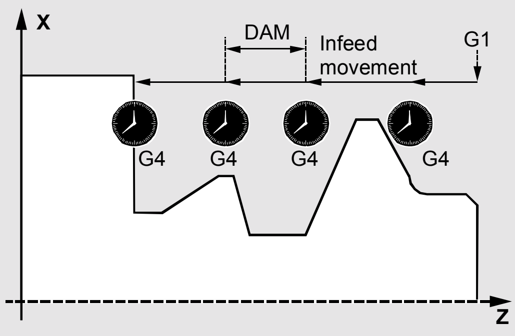
Value	Machining	Selection	Selection
1	Roughing	long.	outside
2	Roughing	transv.	outside
3	Roughing	long.	inside
4	Roughing	transv.	inside
5	Finishing	long.	outside
6	Finishing	transv.	outside
7	Finishing	long.	inside
8	Finishing	transv.	inside
9	Complete machining	long.	outside
10	Complete machining	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.
Outside machining means that infeed is performed in the direction of the negative axis. In inside 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 SW 6.2 and higher. For doing so, the HUNDREDS DIGIT is implemented in parameter VARI.

A plausibility check is performed on parameter VARI. If you select an invalid value, the cycle is aborted and alarm 61002 "Wrong machining type defined" is put out.

DT and DAM (dwell time and path length)
With these two parameters you can program an interruption in the individual roughing cuts after a defined path for the purposes of chip breaking. These parameters only apply to roughing. In parameter DAM you define the maximum path after which chip breaking is to be performed. In DT you can also program a dwell time to be included at each of the interruption points. If no path has been specified for cut interruption (DAM = 0), then uninterrupted roughing cuts without dwell times are generated.

Interrupted paraxial cut



_VRT (lift)
With SW 4.4. and higher, the amount by which the tool is lifted off the contour in both axes during roughing operations can be programmed in parameter _VRT.
If _VRT=0 (parameter not programmed), liftoff is 1mm.
The retraction distance is always measured in inches or metric depending on the programmed system, i.e. _VRT=1 for inch →programming produces a tool retract of 1 inch.

Further notes

Contour definition
The contour is programmed in a subroutine whose name is defined as a parameter.
The contour subroutine must contain at least three 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" 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 preprocessed in the cycle as only these axes are involved in the machining operation. 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, it is aborted with alarm 10930 "Illegal interpolation type in the machining contour".

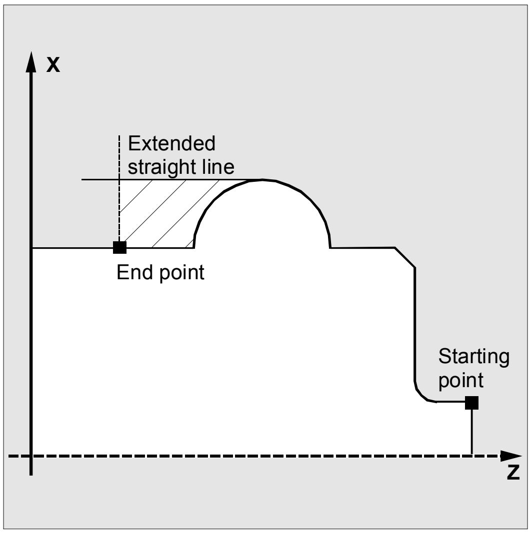
The first block containing a traversing movement in the current machining plane must contain a travel command G0, G1, G2 or G3, otherwise the cycle is aborted with the alarm
15800 "Wrong starting conditions for CONTPRON".
This alarm is also activated when G41/G42 is active.
The starting point of the contour is the first position on the machining plane programmed in the contour subroutine.

The maximum possible number of blocks for the contour containing travel commands for the current plane 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 called separately.

If the maximum diameter in one contour subroutine is not 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.



If any of the following are programmed in the contour subroutine:

- Radius compensation plane with G17/G18/G19
- Frames
- An axis of the plane in which machining is performed is traversed as a positioning axis
- Selection of tool radius compensation with G41/G42

alarm 10931 "Incorrect machining contour" is activated and aborts the cycle.

Contour direction

With SW 4.4 and higher, the direction in which the stock removal contour can be programmed is freely selectable. The machining direction is automatically defined in the cycle. With complete machining operations, the contour is finished in the same direction in which rough cutting took place.

If only finishing is selected, the contour is always traversed in the programmed direction.

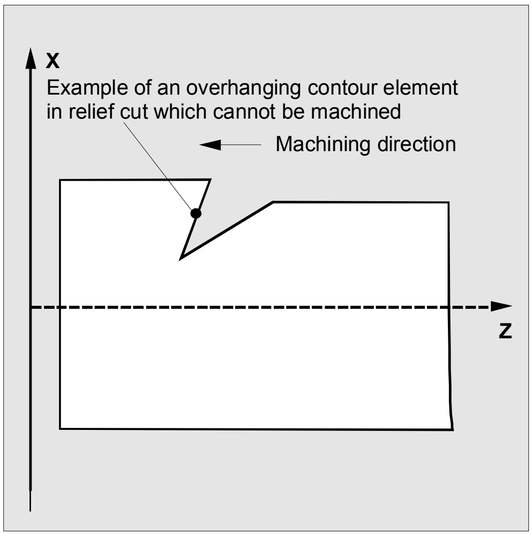
The first and last programmed contour points are the criteria for selecting the machining direction. For this reason, both coordinates must always be programmed in the first block of the contour subroutine.

Contour monitoring

The cycle performs contour monitoring of the following:

- Clearance angles of the active tool
- Programming of arcs with an aperture angle of > 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 is leading 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, alarm 10931 "Incorrect machining contour" is output.

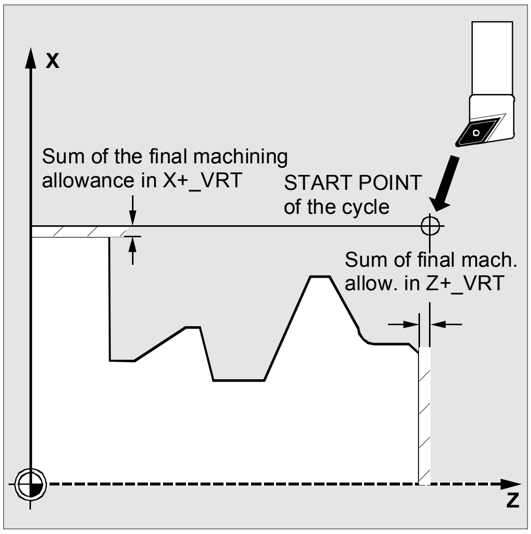
Overhanging contours cannot be machined by CYCLE95. Contours of this type are not monitored by the cycle and consequently there is no alarm.

Starting point

The cycle determines the starting point of the machining operation automatically. The starting point is positioned on the axis in which infeed is performed at a distance from the contour corresponding to final machining allowance + liftoff distance (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 noise radius compensation is selected internally in the cycle when the starting point is approached.

The last point before the cycle is called must therefore be selected such that it can be approached without risk of collision and adequate space is available for the compensating movement.

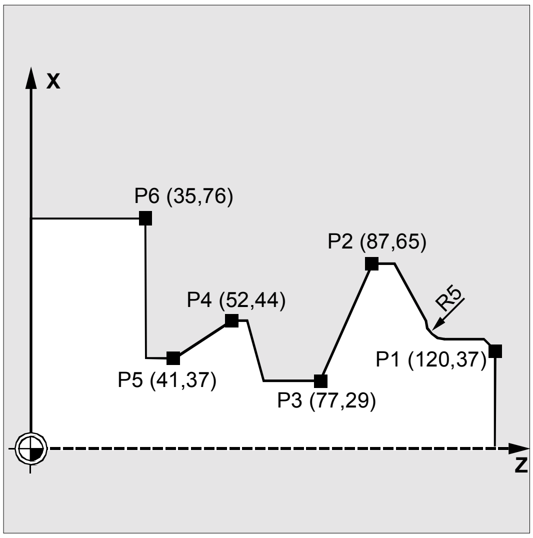


Approach strategy of the cycle

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.

Programming example 1

Stock removal cycle
The contour illustrated in the figure explaining the assignment parameters must be machined completely (longitudinal, outside). Axis-specific final machining allowances have been defined. No interruption between cuts has been programmed. The maximum infeed is 5mm.
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 to starting position
`N40 Z125`	Traverse in each axis separately
`N50 M30`	End of program
`PROC CONTOUR_1`	Beginning of contour subroutine
`N100 G1 Z120 X37` `N110 Z117 X40`	Traverse in each axis separately
`N120 Z112`	Rounding with radius 5

`N130 G1 Z95 X65 RND=5`
`N140 Z87`
`N150 Z77 X29`
`N160 Z62`
`N170 Z58 X44`
`N180 Z52`
`N190 Z41 X37`
`N200 Z35`
`N210 G1 X76`
	Traverse in each axis separately

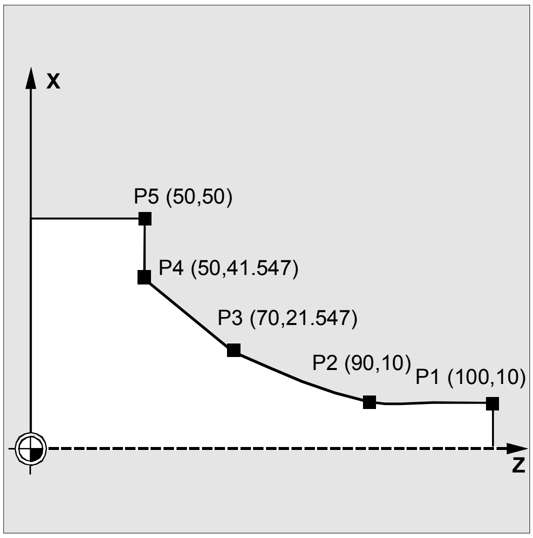
`N220 M17`	End of subroutine

-> Must be programmed in a single block

Programming example 2

Stock removal cycle

The stock removal contour is defined in the calling program. After the stock removal cycle, the program is ended.



`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, -> 0.8,0,0.8,0.75,0.6,1)`	Cycle call
`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

Programming

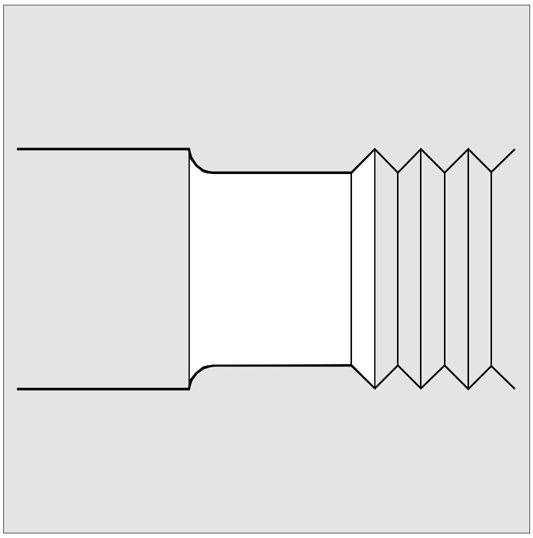
```
CYCLE96 (DIATH, SPL, FORM, _VARI)
```

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 (SW 6.2 and higher)	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 of operations

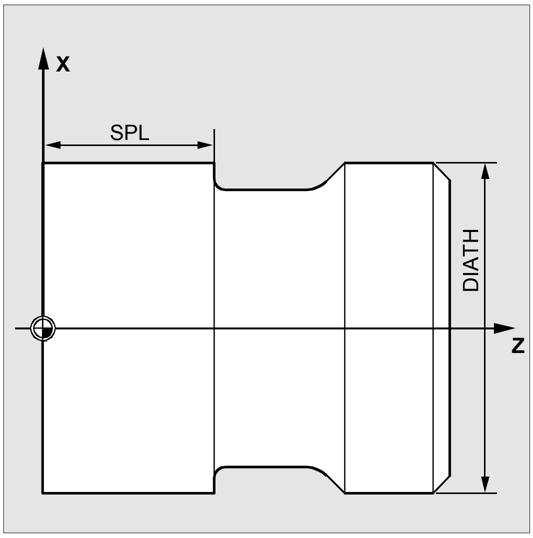
Position reached prior to cycle start:
The starting position can be any position from which any thread undercut can be approached without collision.

The cycle implements the following motion sequence:

- Approach to 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.

Description of parameters

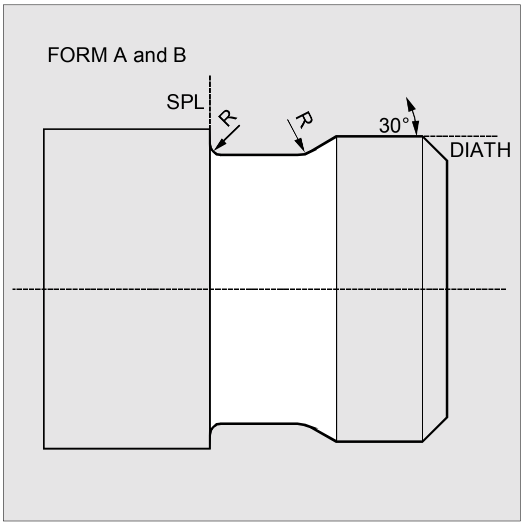
DIATH (nominal diameter)
With this cycle you can machine thread undercuts for metrical ISO threads from M3 to M68.
If the value programmed in DIATH results in a final diameter of <3mm, the cycle is aborted and alarm 61601 "Finished part diameter too small" is output.
If the parameter is assigned a value other than that defined by DIN76 Part 1, again the cycle is aborted and the alarm
61001 "Thread pitch incorrectly defined" is output.



SPL (starting point)
With parameter SPL you define the final dimension in the longitudinal axis.

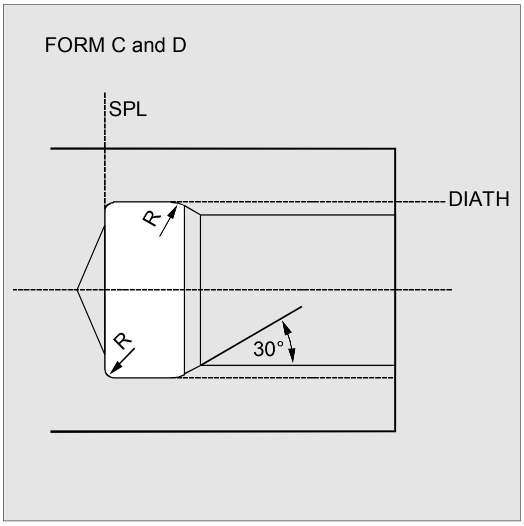
FORM (definition)

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 for used for internal threads, form C for normal thread runouts, form D for short thread run-outs.
If the parameter is assigned a value other than A ... D, the cycle is aborted and alarm 61609 "Form incorrectly defined" is output.
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 with parameter _VARI.
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.

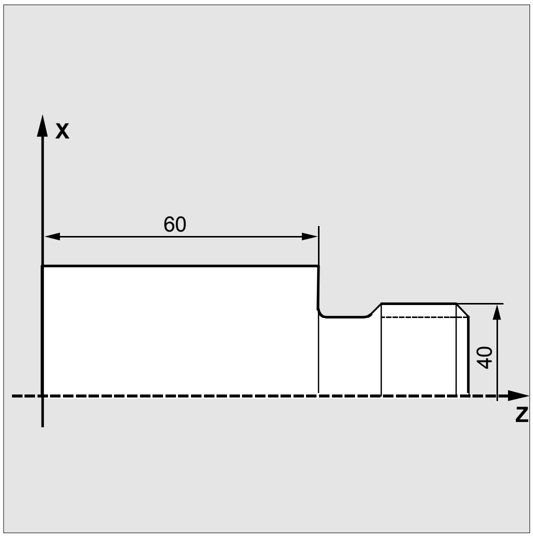
Further notes

You must activate a tool offset before the cycle is called. Otherwise error message
61000 "No tool offset active"
is output 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

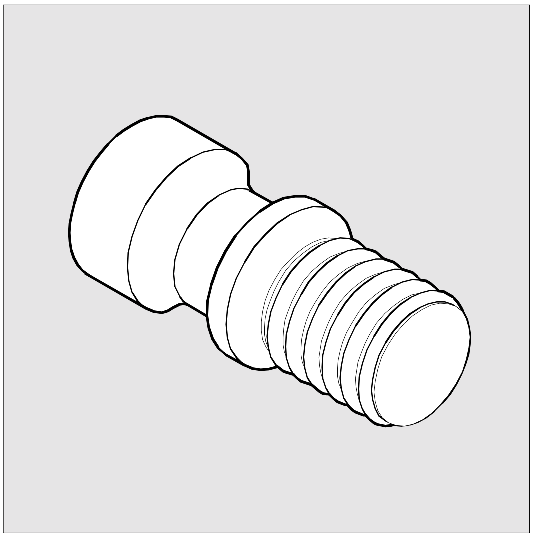
4.7 Thread cutting – CYCLE97

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 (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 threads (enter without sign)
_VRT from SW 6.2 or higher	real	Variable retraction distance based on initial diameter, incremental (enter without sign)



Function

With this cycle you can machine cylindrical and tapered outside and inside threads with constant lead in longitudinal or face machining. Both single threads and multiple threads can be cut. In multiple thread cutting, the threads 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 programmed before the cycle call.
Feedrate and spindle override both have no effect in thread travel blocks.

A speed-controlled spindle with a position measuring system is required to operate this cycle.

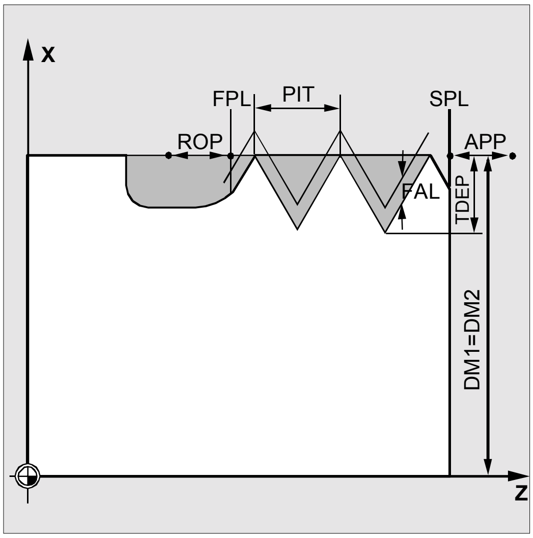
Sequence of operations

Position reached prior to cycle start:
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 to the starting point determined by the cycle at the beginning of the arc-in section for the first thread 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.
- In the next cut with G33 the final machining allowance is removed.
- This cut is repeated according to the number of programmed noncuts.
- The total motion sequence is repeated for each additional thread.

Description of parameters



PIT and MPIT (value and thread size)

The thread pitch is a paraxial value and entered without a sign. If metric cylindrical threads are being machined it is also possible to define the thread pitch in parameter MPIT 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 wrong"
and is aborted.

DM1 and DM2 (diameter)

This parameter is set to program 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) and end point (FPL) are the basis 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 has an effect parallel to the axis and is subtracted from the preset thread depth TDEP and the remainder is divided into roughing cuts.

The cycle automatically calculates the individual actual infeed depths depending on the parameter VARI.

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.

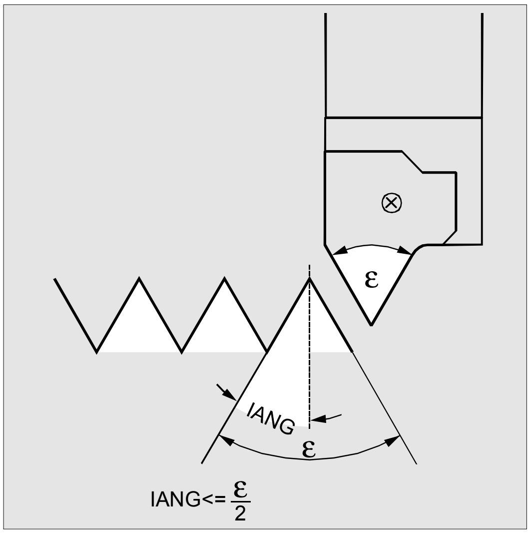
In a second method, the total thread depth is divided into constant infeed depths. The cross-section of cut gets larger from cut to cut. However, if the values for the thread depth are small, this method can create better cutting conditions.

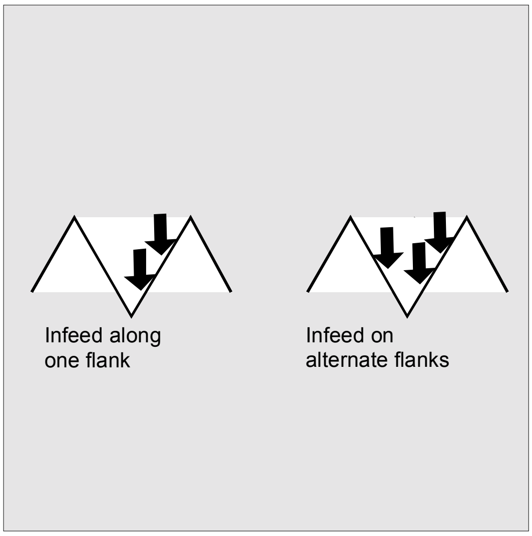
The final machining allowance FAL is removed in one cut after roughing. After this, the noncuts programmed under parameter NID are executed.

IANG (infeed angle)

With parameter IANG you define the infeed angle. 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 infeed type on both flanks alternately 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.



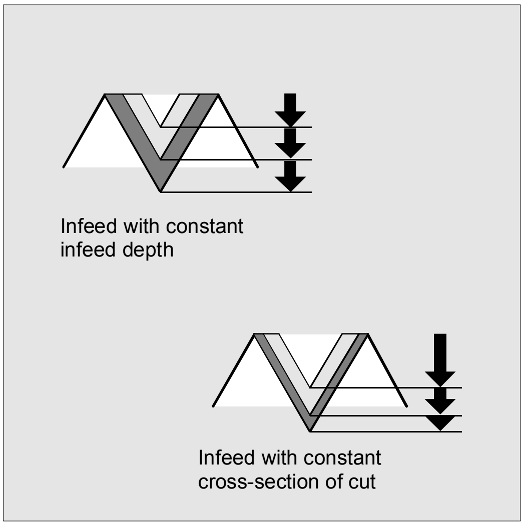


NSP (starting point offset)

Under this parameter you can program the angular value that defines the point of the first cut for the first thread turn on the circumference on 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 parameter VARI, you define if machining is to be internal or external and with which technology the infeed will be machined during roughing. The parameter VARI can be one of the values between 1 and 4 with the following meaning:

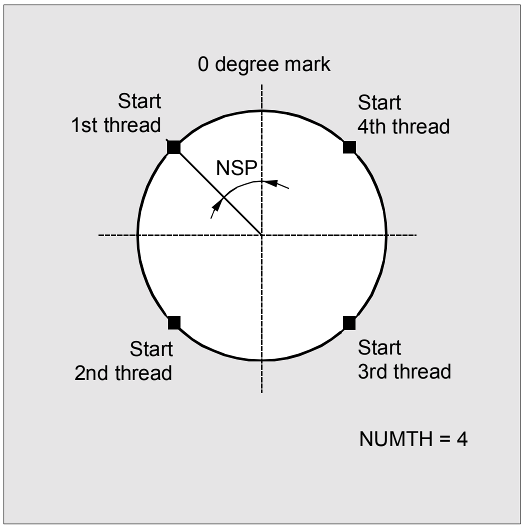


Value	Outside/inside	Const. infeed/const. cross-section of cut
1	Outside	Constant infeed
2	Inside	Constant infeed
3	Outside	Constant cross-section of cut
4	Inside	Constant cross-section of cut

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

NUMT (number of thread starts)

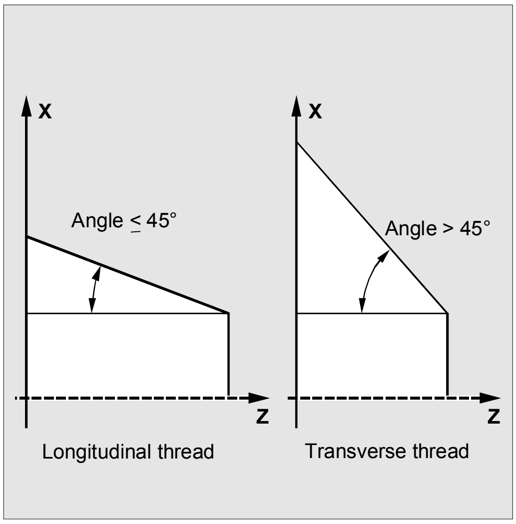
With parameter NUMT you specify the number of thread starts for a multiple thread. If you require a single thread, either assign the value zero to the parameter or omit it from the parameter list.
The thread starts are distributed uniformly around the circumference of the turned part, the first thread is defined in parameter NSP.
If a multiple-start thread with a non-uniform distribution of threads 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)
With SW 6 and higher, the retraction path can be programmed on the basis of the initial thread diameter in parameter _VRT.
When _VRT = 0 (parameter not programmed), the retraction path is 1mm. The retraction path is always measured in the programmed system of units inch or metric.

Further notes

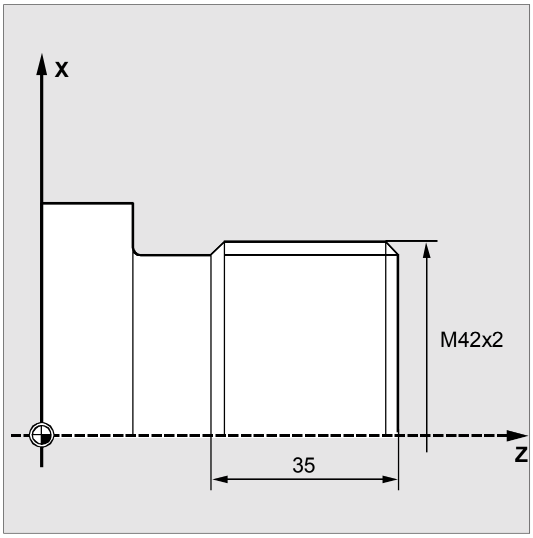
Difference between a longitudinal thread and a face thread
The cycle automatically calculates whether a longitudinal or face thread is to be machined. This depends on the angle of the taper on which the thread is to be machined. If the angle at the taper ≤45 degrees, the longitudinal axis thread is machined, otherwise it will be the face thread.



Programming example

Thread cutting

With this program you can cut a metric outside thread M42x2 with flank infeed. The infeed is performed with a constant cross-section of cut. Five roughing cuts are made to a thread depth of 1.23mm 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

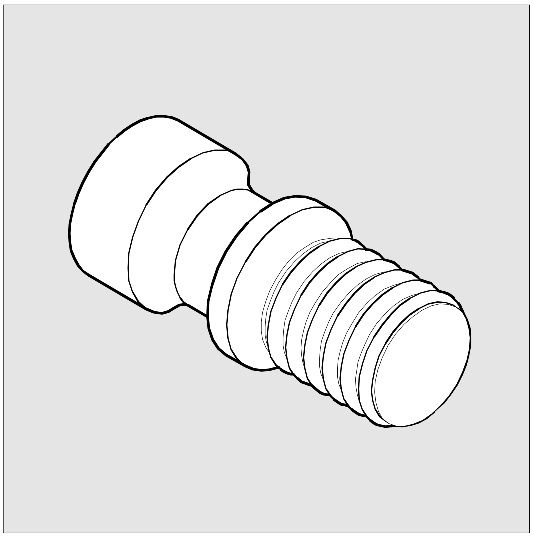
```
CYCLE98 (PO1, DM1, PO2, DM2, PO3, DM3, PO4, DM4, APP, ROP, TDEP, FAL,
IANG, NSP, NRC, NID, PP1, PP2, PP3, VARI, NUMT, _VRT)
```

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 threads (enter without sign)
_VRT SW 6.2 and higher	real	Variable retraction distance based on initial diameter, incremental (enter without sign)

Function

With this cycle you can produce several concatenated cylindrical or tapered threads with a constant lead in longitudinal or face machining, all of which can have different thread leads.



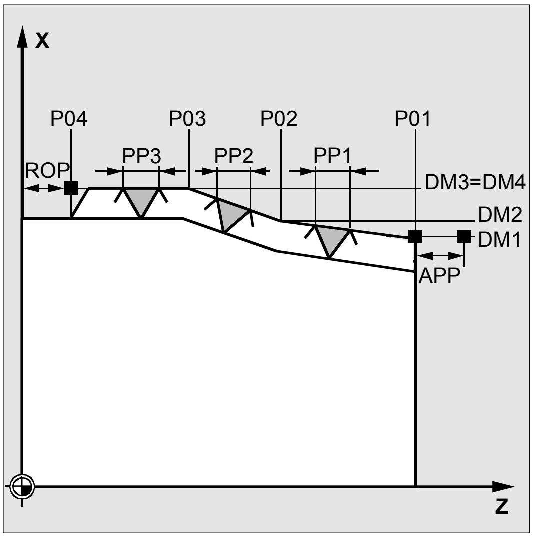
Sequence of operations

Position reached prior to cycle start:
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 to the starting point determined by the cycle at the beginning of the arc-in section for the first thread 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.
- In the next cut with G33 the final machining allowance is cut.
- This cut is repeated according to the number of programmed noncuts.
- The total motion sequence is repeated for each additional thread.

Description of parameters



PO1 and DM1 (starting point and diameter)
With these parameters you define the original starting point of the thread chain. The starting point calculated by the cycle that is approached at the beginning with G0 is the length of the arc-in section in front of the programmed starting point (starting point A).

PO2, DM2 and PO3, DM3 (intermediate point and diameter)
With these parameters you define two intermediate points in the thread.

PO4 and DM4 (endpoint and diameter)
The original end point of the thread is programmed under parameters PO4 and DM4.

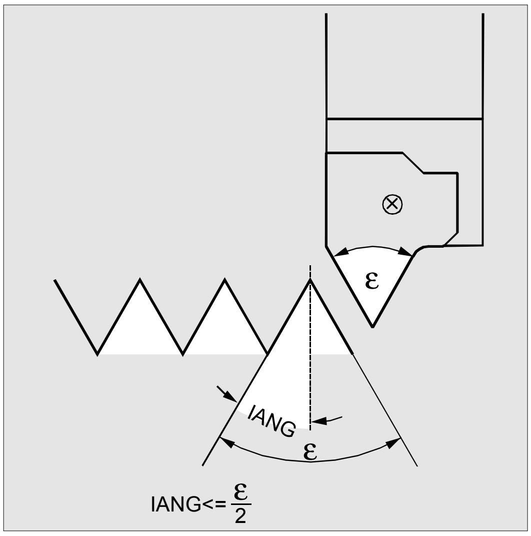
With an inside thread, DM1...DM4 corresponds to the tap hole diameter.

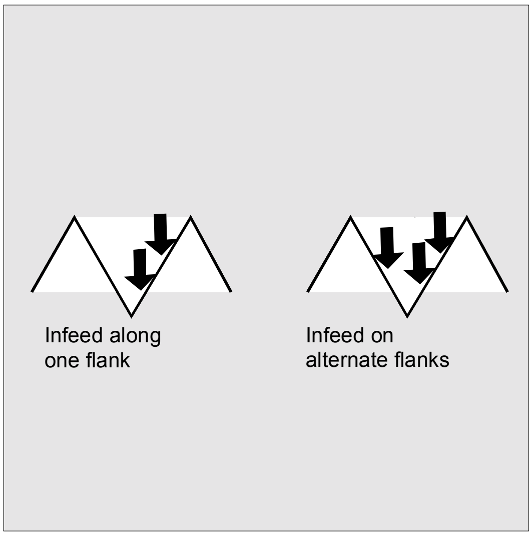
Connection between APP and ROP (arc-in, arc-out sections)
The starting point used in the cycle is the starting point brought forward by the arc-in section APP and, in the same way, the end point is the programmed end point brought back by the arc-out section ROP.
The starting point defined by the cycle always lies 1mm 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 divided into rough cuts. The cycle automatically calculates the individual actual infeed depths depending on the parameter VARI. 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 gets larger 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 parameter NID are executed.

IANG (infeed angle)
With parameter IANG you define the infeed angle. 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 infeed type on both flanks alternately 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.

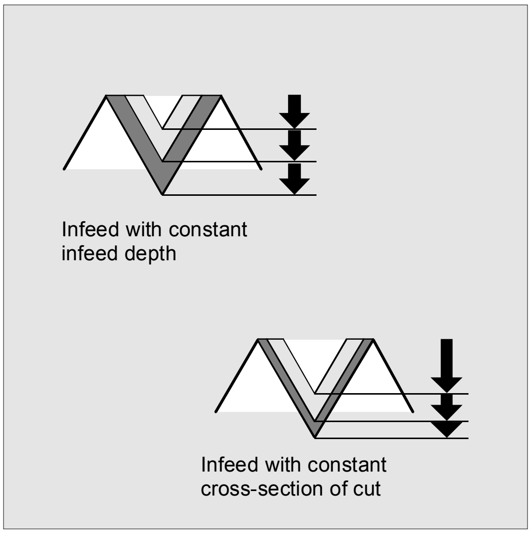




NSP (starting point offset)
Under this parameter you can program the angular value that defines the point of the first cut for the first thread turn on the circumference on the turned part. This value is a starting point offset. The parameter can be assigned any value between 0.0001 and +359.9999 degrees. 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.

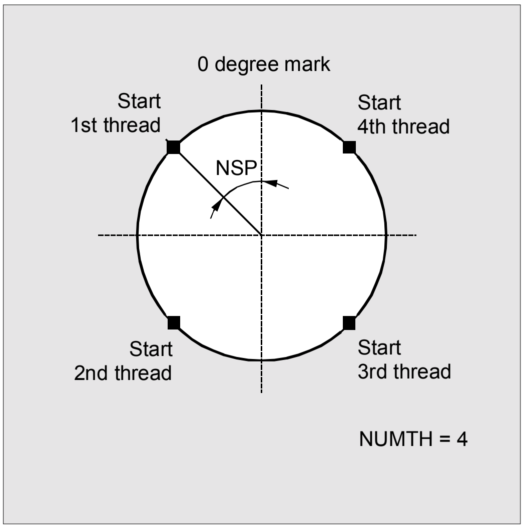
VARI (machining type)
With parameter VARI, you define if machining is to be internal or external and with which technology the infeed will be machined during roughing. The parameter VARI can be one of the values between 1 and 4 with the following meaning:



Value	Outside/inside	Const. infeed/const. cross-section of cut
1	Outside	Constant infeed
2	Inside	Constant infeed
3	Outside	Constant cross-section of cut
4	Inside	Constant cross-section of cut

If another value is assigned to parameter VARI, the cycle is aborted and alarm 61002 "Machining type incorrectly programmed" is output.

NUMT (number of thread starts)
With parameter NUMT you specify the number of thread starts for a multiple thread. 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 is defined in parameter NSP.
If a multiple-start thread with a non-uniform distribution of threads 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.

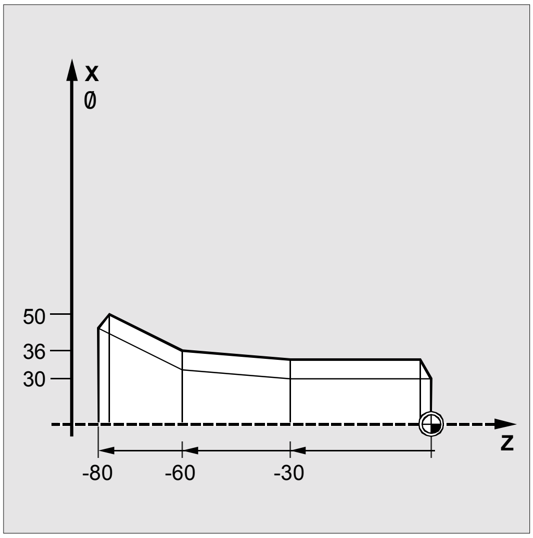


_VRT (variable retraction path)
With SW 6.2 and higher, the retraction path can be programmed on the basis of the initial thread diameter in parameter _VRT.
When _VRT = 0 (parameter not programmed), the retraction path is 1mm. The retraction path is always measured in the programmed system of units inch 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 defined is longitudinal, outside, 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`
`N50 Z10`
`N60 X40`
	Traverse in each axis separately

`N70 M30`	End of program

-> Must be programmed in a single block

4.9 Thread recutting (SW 5.3 and higher)

SW 5.3 contains thread cutting cycles CYCLE97 and CYCLE98 which allow threads to be recut.

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.

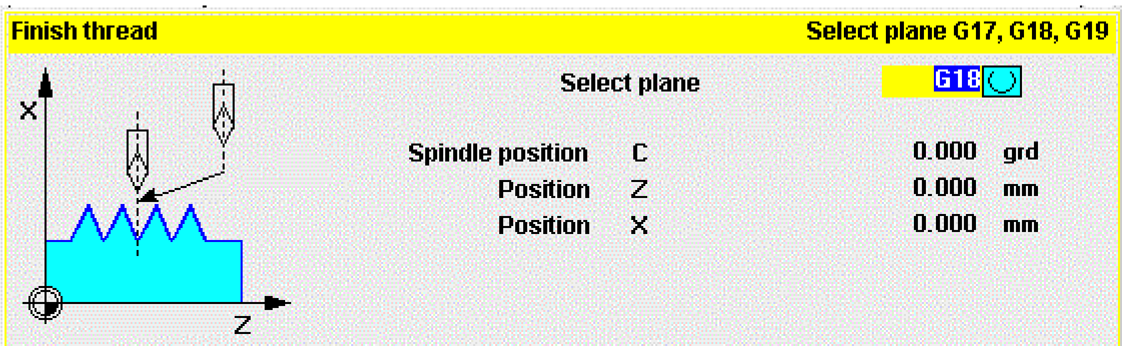
The cycles calculate an additional offset angle for each thread, which is applied in addition to the programmed starting point offset, from the data stored in the thread start during synchronization.

Preconditions

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 of operations

- Select JOG in "Machine" operating area.
- Press soft key "Finish thread"
  ➔ Open screen form for this function.



- Thread into thread start using the threading tool.
- Select soft key "Sync Point" when the cutting tool is positioned exactly in the thread start.

- Press soft key "Cancel" to return to the next-higher soft key menu without activating the function (no data are then stored in the NC).
- Select soft key "OK" to transfer all values to the GUD in the NC.
- Then retract the tool and move it to its starting position.
- Select "Automatic" and position the program pointer using block search in front of the thread cycle call.
- Start the program with NC start.

Special functions

You can delete values stored earlier by selecting another soft key labeled "Delete".

If several spindles are operating in the channel, another box is displayed in the screenform in which you can select a spindle to machine the thread.

4.10 Extended stock removal cycle – CYCLE950 (SW 5.3 and higher)

The extended stock removal cycle is an option.
It requires SW 5.3 in the NCK and MMC.

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

`_NP1`	string	Name of the contour subroutine for the finished part contour
`_NP2`	string	Label / block number start of finished part contour, optional (this can be used to define contour sections)
`_NP3`	string	Label / block number end of finished part contour, optional (this can be used to define contour sections)
`_NP4`	string	Name of the stock removal program to be generated
`_VARI`	int	Type of machining: (enter without sign) ONES 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+
`_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)
`_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 parameter _APZ 90=absolute, 91=incremental
_APX	real	Axial value for defining blank for facing axis
_APXA	int	Absolute or incremental evaluation of parameter _APX 90=absolute, 91=incremental
_TOL1	real	Blank tolerance

Function

With the extended stock removal cycle CYCLE950 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 in the longitudinal and facing directions with this cycle. You can freely select a technology (roughing, finishing, complete machining, machining and infeed directions). It is possible to update a blank.

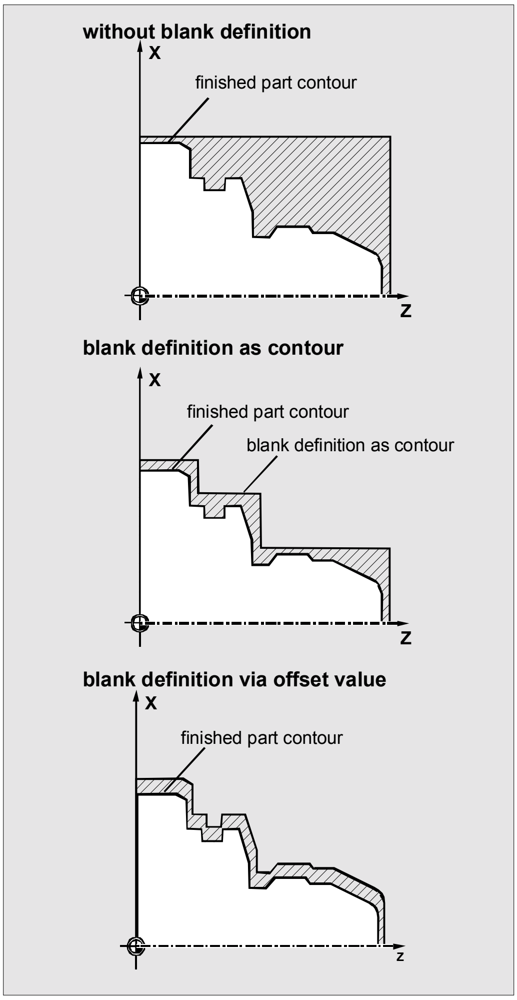
For roughing, the programmed infeed depth is observed precisely; the last two roughing steps are divided equally. Roughing is performed to the programmed final machining allowance.

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 current tool. The cycle can generate an updated blank, 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 section of any given program.
- During 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 is programmable.
- Optionally, relief cuts can be machined or skipped during roughing.



Sequence of operations

Position reached prior to cycle start:

The initial position can be any position from which the blank contour can be approached collision-free. The cycle calculates collision-free approach movements to the starting point for machining but does not consider the tool holder 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 parameter _MID, is carried out with G1, and paraxial roughing then performed with G1. The feedrate during roughing is calculated internally in the cycle according to the path as 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 machining "Without rounding along contour", 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 liftoff path _VRT is also followed precisely; it is not exceeded for other angles.
- This procedure is repeated until the full 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 is carried out in the same way as for paraxial roughing.

Description 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 are 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 need 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. This program is stored in the same directory as the calling program in the parts program memory if no other path is specified when it is generated. If a path is entered, it is stored accordingly in the file system.
The program is a main program (type MPF) if no other type is specified.
Parameter _NP4 defines the name of this program.

_VARI (machining type)

Parameter _VARI 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)
The selection with or without rounding along the contour determines whether or stock removal starts at the roughing intersection immediately or whether machining is performed along the contour up to the previous intersection so that there are no residual corners.

Ten thousands digit:

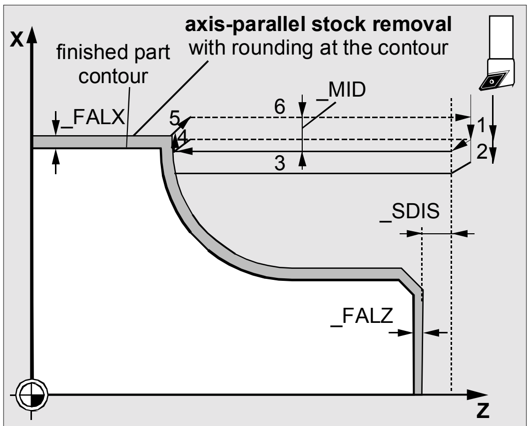
1=Machine relief cuts
2=Do not machine relief cuts

Hundred thousands digit:

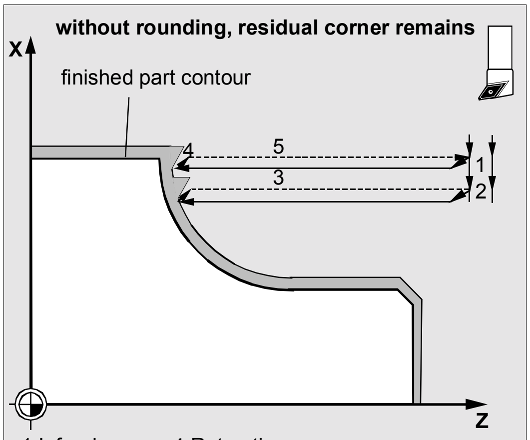
1=Programmed machining direction X-
2=Programmed machining direction X+
3=Programmed machining direction Z-
4=Programmed machining direction Z+

Example:

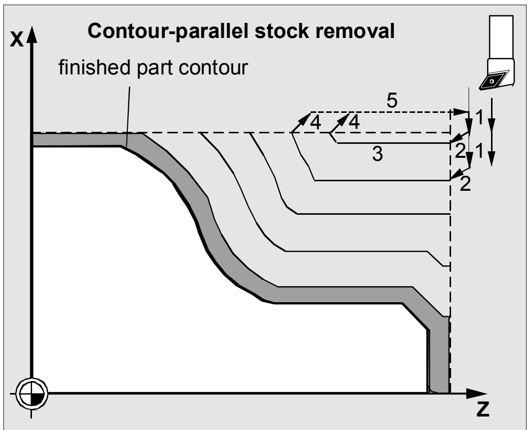
_VARI=312311 means machining:
longitudinal,
infeed direction X- (i.e. external),
complete;
the workpiece is not rounded along the contour, relief cuts are machined,
machining direction Z-.



1 Infeed
2 Approach
3 Roughing
4 Stock removal of residual corners (rounding)
5 Retraction
6 Returning



1 Infeed
2 Approach
3 Roughing
4 Retraction
5 Returning



1 Infeed
2 Approach
3 Roughing
4 Retraction
5 Returning

_MID (infeed depth for roughing)
The infeed depth for roughing is programmed with the parameter _MID. Roughing steps are generated with this infeed until the remaining depth is less than twice the infeed depth. Then two steps are performed each at half of the remaining depth. _MID is interpreted as a radius or diameter depending on the value of cycle setting data _ZSD[0] if the facing axis is involved in the infeed for roughing.

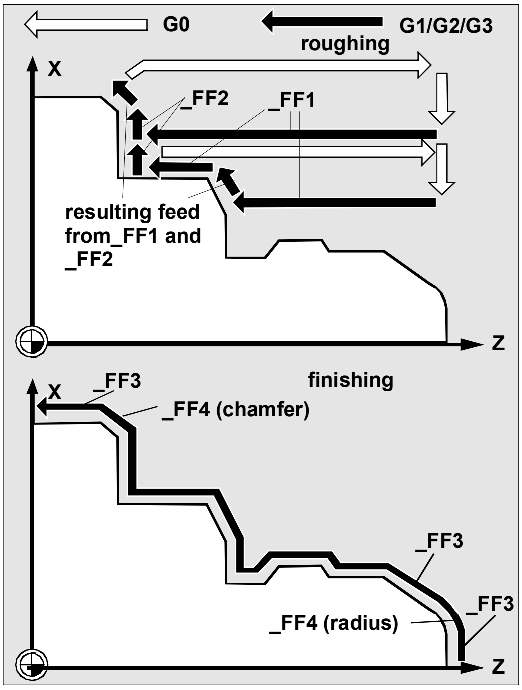
_ZSD[0]=0:	_MID is interpreted according to the G group for radius/ diameter programming, as a radius with DIAMOF, otherwise as a diameter.
_ZSD[0]=1:	_MID is a radius value
_ZSD[0]=2:	_MID is a diameter value

When rough cutting 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 rough-cutting with the same value for infeed depth.

_FALZ, _FALX (machining allowance)
The default for a final machining allowance for roughing is provided by the parameters FALZ (for Z axis) and FALX (for X axis). Roughing always takes place down to theses final machining allowances. If no machining allowances are programmed, stock removal is performed up to the end contour during roughing.

If final machining allowances are programmed, these are applied correspondingly.

_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 sample program 1 for programming of the parts in the figure below)



_VRT (liftoff) and _ANGB (lift angle)
The parameter _VRT 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 parameter _ANGB. If nothing is programmed, the angle is 45°.

_SDIS (safety distance)
Parameter _SDIS 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 parameter _NP5 or as a program section with parameters _NP6 and _NP7.
Otherwise, programming is carried out as for finished parts (see _NP1, _NP2, _NP3).

_NP8 (name of contour program for updated blank contour)
Cycle CYCLE950 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 parameter _NP8 with or without path details (see sample program 3).
An updated blank contour is always generated when a travel program is generated.

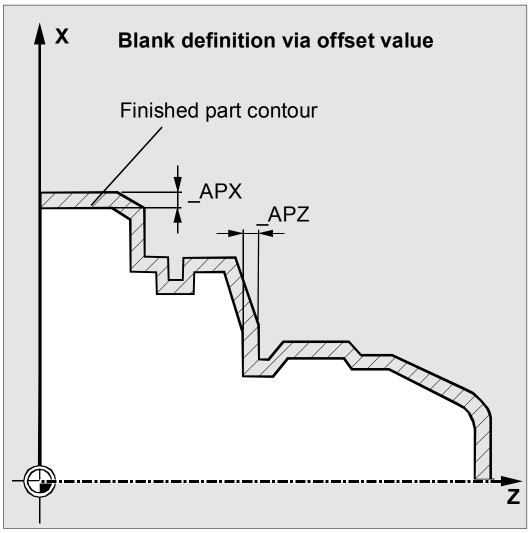
_APZ, _APZA, _APX, _APXA (blank definition)

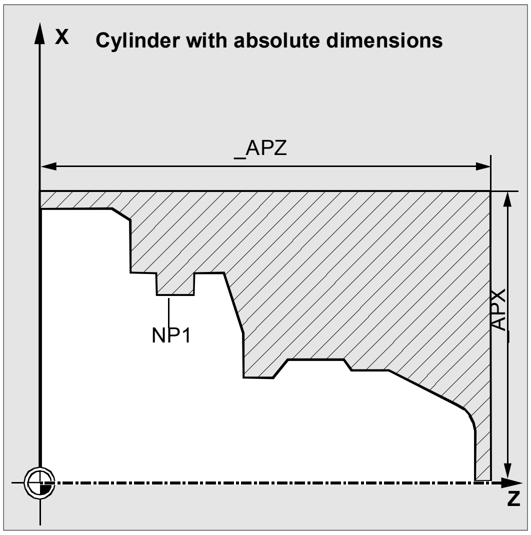
You can also define a blank by entering the dimensions of a blank cylinder (or hollow cylinder) or as an allowance on the finished-part contour in parameters _APZ and _APX.

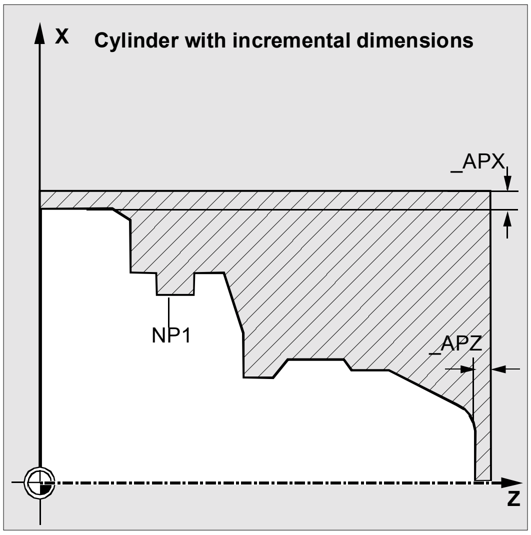
You can enter the cylinder dimensions as either absolute or incremental values, although an allowance on the finished-part contour is always interpreted incrementally.

Absolute or incremental values are selected via parameters _APZA and _APXA

(_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 to compensate for any tolerances.
Parameter _TOL1 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 1mm.

Further notes

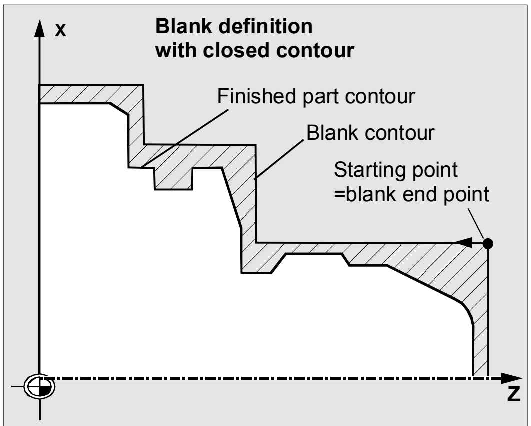
Contour definition

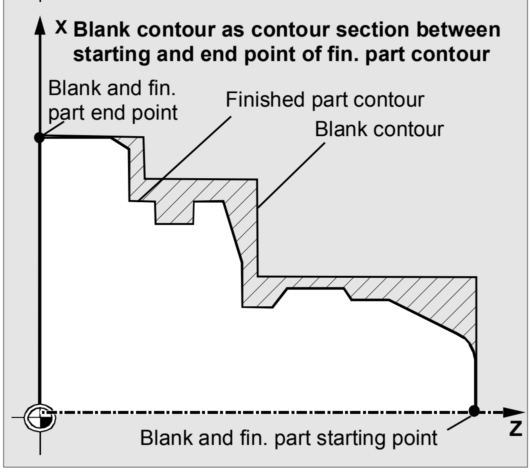
Unlike CYCLE95, one block that contains a link to the current plane is sufficient for contour programming.

For further details of contour definition, see CYCLE95.

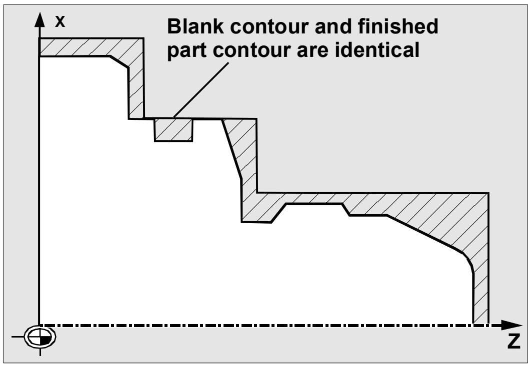
Blank contour definition

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 always must be described in a way that they are not partly identical with 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 MMC.

The calculation is started in the cycle and a program with traversing blocks for stock removal generated in its result and stored in the file system of the control, 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 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 for the search in the file system of the control:

- If the calling program is stored in a workpiece directory, then the programs which define the finished-part 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), a search is also made there for the programs is no path is specified.

The cycle creates a program that contains the traversing blocks for stock removal and, optionally, an updated blank contour.

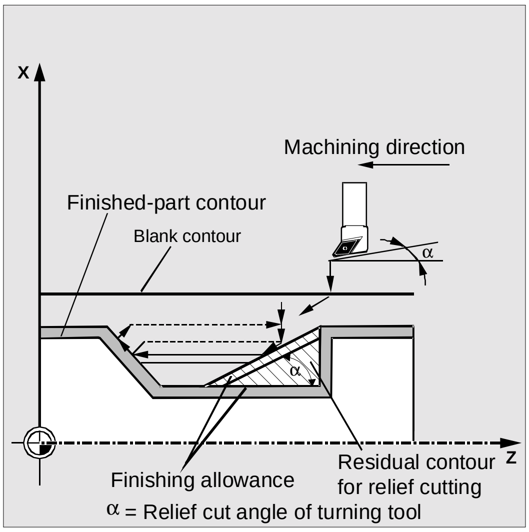
These are either stored in the same directory as the cycle-calling program or in accordance with the specified path.

Note on simulation

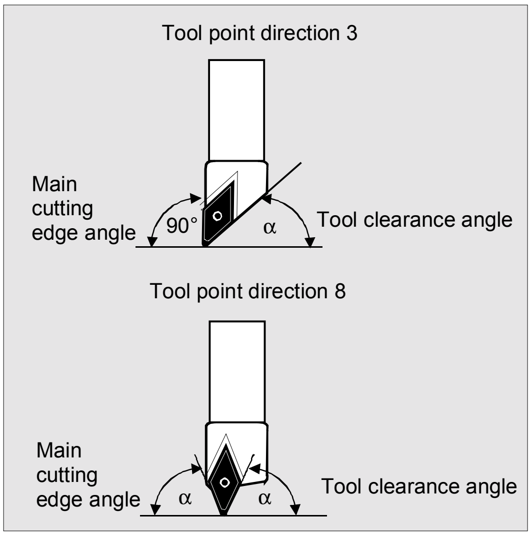
In the simulation of the extended stock removal cycle CYCLE950, the generated programs are saved to the NCU file system. Therefore, only the "NC Active Data" setting is practical since tool offset data are included in program calculation.

Blank updating

The extended stock removal cycle CYCLE950 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.

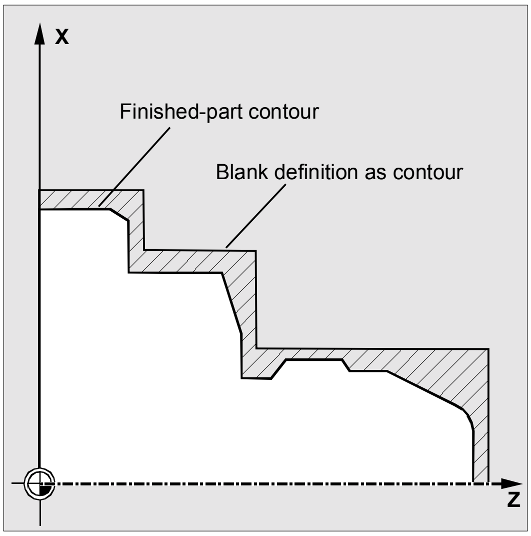


To do this, the cycle internally considers the angle at the tool point.
The relief cut angle of the tool must be entered in the tool offset data (parameter 24).
The cycle defines the main cutting edge angle automatically according to the tool point position.
For tool point positions 1 to 4, the blank update is calculated with a main cutting edge angle of 90°. For tool point positions 5 to 9, the main cutting angle is assumed to be identical to the relief cut angle.
If CYCLE950 is called more than once, each time with blank update, in the same program, different names for the generated blank contours must be assigned; it is not permissible to use the program name (parameter _NP8) more than once.



Extended stock removal cannot be performed in m:n configurations.

Programming example 1



From a preshaped blank, the contour saved in program PART1.MPF is to be machined.
The type of machining for the stock removal process is

- only roughing,
- longitudinal,
- outside,
- with rounding (so that no corners are left over),
- relief cuts are to be machined.

The blank contour is specified in program BLANK1.MPF.
A turning steel with tool point position 3 and a radius of 0.8mm is used.

Machining program:

```
%_N_EXAMPLE_1_MPF
;$PATH=/_N_WKS_DIR/_N_STOCK_REMOVAL_NEW_WPD
; Example 1: Stock removal with blank
; Sca, 01.04.99
;
; Tool offset data
N10 $TC_DP1[3,1]=500 $TC_DP2[3,1]=3
$TC_DP6[3,1]=0.8 $TC_DP24[3,1]=60
N15 G18 G0 G90 DIAMON
N20 T3 D1
N25 X300
N30 Z150
N35 G96 S500 M3 F2
N45 CYCLE950("Part1",,,"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
```

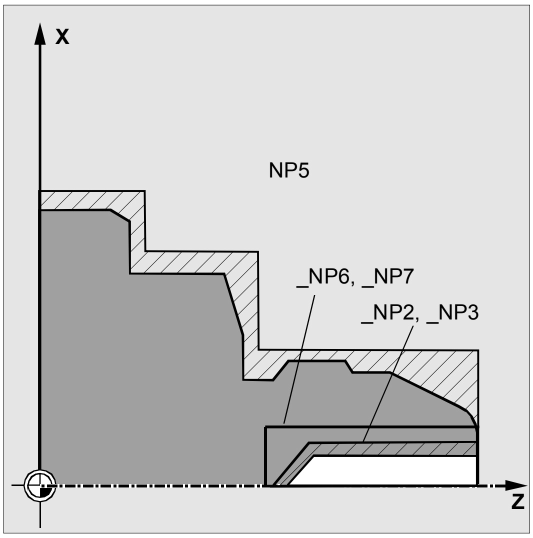
Blank contour:

`%_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, a new program called MACHINING_PART1.MPF is present in the workpiece STOCK_REMOVAL_NEW.WPD. 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 sample program 1.
A center bore is made first using a diameter-10 drill.
Then, the inside contour is roughed parallel to the contour, since the hole roughly corresponds to the end contour.
This is done by defining a blank contour again 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 1: inside 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 as in sample program 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 with the name blank3.MPF.
The type of machining for this step is:

- only roughing,
- longitudinal,
- outside,
- with rounding,
- relief cuts are not be machined.

In the second machining step (N70), the residual material on this blank is machined with a different tool and then finished.
The type of machining for this step is:

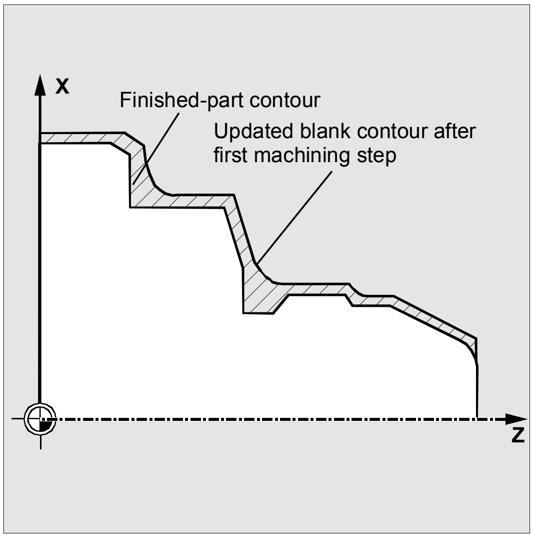
- complete machining (roughing and finishing)
- longitudinal,
- outside,
- with rounding (so that there are no residual corners),
- relief cuts are to be machined.

Machining program:

```
%_N_EXAMPLE_3_MPF
;$PATH=/_N_WKS_DIR/_N_STOCK_REMOVAL_NEW_W
PD
; Example 3: stock removal in two steps
with blank update
; 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 steel 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 sample program 1



Explanation

Alarm source CYCLE950

Alarm number	Alarm text	Explanation, remedy
61701	"Error in contour description of finished part"	Either none of parameters _NP1, _NP2 or _NP3 is assigned or error in programming of finished-part contour
61702	"Error in contour description of blank"	Either none of parameters _NP5, _NP6 or _NP7 is assigned or error in programming of 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"	

Alarm number	Alarm text	Source	Explanation, remedy
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"		
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 position of tool 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 information

If error conditions are detected in the cycles, an alarm is output and execution of the cycle is aborted. The cycles also output messages in the dialog line of the control. These messages do not interrupt processing.

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 Troubleshooting 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 acknowledgment criteria.
The text displayed with the number provides an explanation of the cause of the error.

Alarm number	Cancel criterion	Alarm reaction
61000 ... 61999	NC_RESET	Block preprocessing in the NC is aborted
62000 ... 62999	Cancel key	Block preprocessing is interrupted, the cycle can be continued with NC Start once the alarm has been canceled

5.3 Overview of cycle alarms

The alarm numbers are classified as follows:

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 is called
61001	"Thread lead incorrect"	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 parameter VARI for the machining type is incorrect and must be altered

Alarm number	Alarm text	Source	Explanation, remedy
61003	"No feedrate programmed in the cycle"	CYCLE71 CYCLE72	The parameter for feedrate has been incorrectly set and must be altered.
61005	"3rd geometry axis not available"	CYCLE86	When used on turning machines without Y axis in G18 plane.
61009	"Active tool number = 0"	CYCLE71 CYCLE72	No tool (T) is programmed prior to the cycle call.
61010	"Final machining allowance too great"	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	"Scaling in the plane different"	CYCLE76 CYCLE77	
61101	"Reference plane incorrectly defined"	CYCLE71 CYCLE72 CYCLE81 to CYCLE90 CYCLE840 SLOT1 SLOT2 POCKET1 to POCKET4 LONGHOLE	Either different values must be entered for the reference plane and the retraction plane if they are relative values or an absolute value must be entered for the depth
61102	"No spindle direction programmed"	CYCLE86 CYCLE87 CYCLE88 CYCLE840 POCKET3 POCKET4	Parameter SDIR (or SDR in CYCLE840) must be programmed
61103	"Number of holes equals zero"	HOLES1 HOLES2	No value has been programmed for the number of holes
61104	"Contour violation of the slots/elongated holes"	SLOT1 SLOT2 LONGHOLE	Incorrect parameterization of the milling pattern in the parameters that define the position of the slots/elongated holes in the cycle and their shape

Alarm number	Alarm text	Source	Explanation, remedy
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
61106	"Number of or distance between circular 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 incompatible with final drilling depth
61108	"No admissible values for parameters _RAD1 and _DP1"	POCKET3 POCKET4	Parameters _RAD1 and _DP which define the path for depth infeed have been incorrectly set.
61109	"Parameter _CDIR incorrectly defined"	POCKET3 POCKET4	The value of the parameter for milling direction _CDIR has been incorrectly set and must be altered.
61110	"Final machining allowance on the base > depth infeed"	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	"Negative tool radius"	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 high"	POCKET3	The parameter for corner radius _CRAD has been set too high and must be reduced.
61114	"Machining direction G41/G42 incorrectly defined"	CYCLE72	The machining direction of the cutter radius compensation G41/G42 has been incorrectly set.

Alarm number	Alarm text	Source	Explanation, remedy
61115	"Contour approach or return mode (straight line/circle/plane/ space) incorrectly defined"	CYCLE72	The contour approach or return mode has been incorrectly programmed; check parameter _AS1 or AS2.
61116	"Approach or return travel=0"	CYCLE72	The approach or return travel is set to zero and must be increased; check parameter _LP1 or _LP2.
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 parameters _LENG and _WID.
61124	"Infeed width has not been programmed"	CYCLE71	A value for the infeed width _MIDA must always be programmed for active simulation without a tool.
61125	"Technology selection in parameter _TECHNO incorrectly defined"	CYCLE84 CYCLE840	Check parameter _TECHNO.
61126	"Thread length too short"	CYCLE840	Program lower spindle speed/raise reference plane
61127	"Gear 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 parameter _STA2
61180	"No name assigned to swivel data record even though machine data $MN_MM_NUM_TOOL_CARRIER > 1"	CYCLE800	No name assigned to swivel data record even though several swivel data records exist ($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)"	CYCLE800	TOOLCARRIER functionality as from NCU 6.3xx
61182	"Name of swivel data record unknown"	CYCLE800	See Swivel cycle start-up CYCLE800 → Kinematics Name (swivel data record)

Alarm number	Alarm text	Source	Explanation, remedy
61183	"Retraction mode GUD7 _TC_FR outside value range 0..2"	CYCLE800	See Swivel cycle start-up CYCLE800 → Retraction; 1st transfer parameter CYCLE800(x,...) is faulty >2
61184	"No solution can be found with current angle inputs"	CYCLE800	
61185	"Rotary axis angle ranges incorrect (min>max) or not defined"	CYCLE800	Check start-up of swivel cycle CYCLE800
61186	"Invalid rotary axis vectors"	CYCLE800	Swivel cycle start-up CYCLE800: Rotary axis vector V1 or V2 not entered or incorrect
61187	"Block search computation end of block not valid for SWIVEL"	CYCLE800	Select block search with calculation contour
61188	"No axis name 1st rotary axis declared"		Swivel cycle start-up CYCLE800: no 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, if necessary deleting elements
61213	"Circle radius too small"	CYCLE77	Enter higher value for circle radius
61215	"Blank dimension incorrectly programmed"	CYCLE76 CYCLE77	Check unmachined part spigot dimensions. The unmachined part spigot must be larger than the machined part spigot.
61601	"Finished part diameter too small"	CYCLE94 CYCLE96	A finished part diameter has been programmed
61602	"Tool width incorrectly defined"	CYCLE93	Parting tool is larger than programmed groove width
61603	"Groove shape incorrectly defined"	CYCLE93	• Radii/chamfers on the groove base are not suitable for the groove width • Face groove of a contour element parallel to the longitudinal axis is not possible

Alarm number	Alarm text	Source	Explanation, remedy
61604	"Active tool violates 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 on 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 direction programmed"	CYCLE94 CYCLE96	A tool point direction between 1 ... 4 that matches the undercut form must be programmed
61609	"Form incorrectly programmed"	CYCLE94 CYCLE96	Check parameters for the undercut form
61610	"No infeed depth programmed"	CYCLE76 CYCLE77 CYCLE96	
61611	"No intersection found"	CYCLE95	The system cannot calculate an intersection 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 parameter _VARI
61803	"Programmed axis does not exist"	CYCLE83 CYCLE84 CYCLE840	Check parameter _AXN
61807	"Incorrect spindle direction programmed (active)"	CYCLE840	Check parameters SDR and SDAC
62100	"No drilling cycle active"	HOLES1 HOLES2	No drilling cycle was called modally before the drilling pattern cycle was called
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 Messages in the cycles

The cycles output messages in the dialog line of the control. These messages do not interrupt processing.
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
"Wrong 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 ≥ tool diameter!"	POCKET1, POCKET2
"Thread: – longitudinal thread machining"	CYCLE97, CYCLE98
"Thread: – transversal thread machining"	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

AS	Automation System
ASCII	American Standard Code for Information Interchange
ASIC	Application Specific Integrated Circuit: User switching circuit
ASUB	Asynchronous Subroutine
AuxF	Auxiliary Function
AV	Production planning
BA	Operating mode
BAG	Mode group
BB	Ready
BCD	Binary Coded Decimals: Decimals number coded in binary format
BCS	Basic Coordinate System
BIN	Binary Files
BIOS	Basic Input Output System
BOT	Boot Files: Boot files for SIMODRIVE 611D
BP	Basic Program
C Bus	Communications Bus
C1 .. C4	Channel 1 to channel 4
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing

CNC	Computerized Numerical Control
COM	Communication
COR	COordinate Rotation
CP	Communications processor
CPU	Central Processing Unit
CR	Carriage Return
CRC	Cutter Radius Compensation
CRT	Cathode Ray Tube: Teletube
CSB	Central Service Board: PLC module
CSF	Control System Flowchart (programming method for PLC)
CTS	Clear To Send: Clear to send for serial interfaces
CUTOM	CUTter radius cOMpensation: Tool radius compensation
DAC	Digital Analog Converter
DB	Data Block on the PLC
DBB	Data Block Byte on the PLC
DBW	Data Block Word on the PLC
DBX	Data Block bit on the PLC
DC	Direct Control: Movement of the rotary axis across the shortest path to the absolute position within one revolution
DCD	Data Carrier Detect
DCE	Data Communications Equipment
DDE	Dynamic Data Exchange
DIN	German Industrial Standard

DIO	Data Input/Output: Data transfer display
DIR	Directory
DLL	Dynamic Link Library
DOS	Disk Operating System
DPM	Dual Port Memory
DPR	Dual port RAM
DRAM	Dynamic Random Access Memory
DRF	Differential Resolver Function (handwheel)
DRY	DRY Run: Dry run feedrate
DSB	Decoding Single Block
DTE	Data terminal equipment
DW	Data word
EIA Code	Special punch tape code: Number of punched holes per character is always odd
ENC	ENCoder
EPROM	Erasable Programmable Read Only Memory
ERROR	ERROR from printer
FB	Function Block
FBS	Slimline screen
FC	Function Call: Function block on the PLC
FDB	Product database
FDD	Floppy Disk Drive

FDD	Feed drive
FEPROM	Flash EPROM: Readable and writable memory
FIFO	First in first out: Memory that operates without addresses where the data are always read out in the same order in which they were stored.
FIPO	Fine InterPolator
FM	Function Module
FM-NC	Function Module Numerical Control
FPU	Floating Point Unit
FRA	FRAme block
FRAME	Data block (frame)
FST	Feed STop
GUD	Global User Data
GWRC	Grinding Wheel Radius Compensation
HD	Hard Disk
HEX	Hexadecimal number
HHU	Hand-held unit
HMI	Operator control and monitoring
HMS	High resolution measuring system
HW	Hardware
I	Input
I/O	Input/output

I/RF	Power feed/return converter unit on the SIMODRIVE 611(D)
IBN	Installation and start-up
IC (GD)	Implicit Communication (Global Data)
ICA	Interpolatory Compensation with Absolute values
IF	Pulse enable for drive module
IM	Interface Module
IMR	Interface Module Receive: Interface module for receiving data
IMS	Interface Module Send: Interface module for transmitting data
INC	Increment: Incremental dimension
INI	INItializing data
IPO	InterPOlator
IS	Interface
IS	Interface Signal
ISA	International Standard Architecture
ISO	International Standard Organization
ISO Code	Special punch tape code: number of punched holes per character is always even
JOG	JOGging: Jog mode
$K_Ü$	Transmission ratio
K_v	Servo gain factor
LAD	LADder logic (programming method for PLC)
LCD	Liquid Crystal Display

LEC	Leadscrew Error Compensation
LED	Light Emitting Diode
LF	Line Feed
LR	Position controller
LUD	Local User Data
MB	Megabyte(s)
MC	Measuring circuit
MCP	Machine Control Panel
MCS	Machine Coordinate System
MD	Machine Data
MDA	Manual Data Automatic: Manual input (MDI)
MLFB	Machine-readable product designation
MMC	Man Machine Communication: User interface on numerical control systems for operator control, programming and simulation
MPF	Main Program File: NC parts program (main program)
MPI	MultiPort Interface
MS	Microsoft (software manufacturer)
MSD	Main Spindle Drive
NC	Numerical Control
NCK	Numerical Control Kernel: Numerical kernel with block preparation, positioning range etc.
NCU	Numerical Control Unit: NCK hardware unit
NRK	Numeric Robotic Kernel: Name of NCK operating system

NURBS	Non-Uniform Rational Basis (B) Spline
O	Output
OB	Organization Block on PLC
OEM	Original Equipment Manufacturer
OI	Operator Interface
OP	Operator Panel
OPI	Operator Panel Interface: operator panel interface module
OPT	Options
OSI	Open Systems Interconnection: Standardization for computer communication
P Bus	Peripheral Bus
PC	Personal Computer
PCIN	Name of SW for data exchange with the control
PCMCIA	Personal Computer Memory Card International Association: Memory plug-in board normalization
PG	Programming device
PLC	Programmable Logic Control
PMS	Position Measuring System
POS	POSitioning
RAM	Random Access Memory: in which data can be read and written
REF	REFerence point approach function
REPOS	REPOSitioning function

RISC	Reduced Instruction Set Computer: type of processor with small instruction set and ability to process instructions at high speed
ROV	Rapid OVerride: Input adjustment
RPA	R Parameter Active: memory area on the NCK for R parameter numbers (RVA)
RPY	Roll Pitch Yaw: Type of rotation of a coordinate system
RS-232-C	Serial interface (definition of interchange circuit between DTE and DCE) (V.24)
RTS	Request To Send: control signal on serial data interfaces
RVA	R Variable Active (RPA)
SBL	Single BLock
SD	Setting Data
SDB	System Data Block
SEA	SEtting data Active: Identifier (file type) for setting data
SFB	System Function Block
SFC	System Function Call
SK	Soft Key
SKP	SKiP: Skip block
SM	Stepper Motor
SPF	Subroutine file: subroutine
SR	Subroutine
SRAM	Static RAM
SSI	Serial Synchronous Interface

STL	Statement list
SW	Software
SYF	System Files
T	Tool
TC	Tool Change
TEA	TEsting data Active: Refers to machine data
TLC	Tool Length Compensation
TO	Tool Offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
TRANSMIT	TRANSform Milling Into Turning: Coordinate conversion on turning machine for milling operations
TRC	Tool Radius Compensation
UFR	User Frame: Zero offset/work offset
UI	User Interface
WCS	Workpiece Coordinate System (Work)
WO	Work Offset (ZO)
WOP	Workshop-Oriented Programming
WPD	WorkPiece Directory
ZO	Zero Offset (WO)
ZOA	Zero Offset Active: Identifier (file type) for zero offset data
µC	Microcontroller

B Terms

Important terms are listed in alphabetical order. The symbol "->" precedes terms which are explained under a separate entry in this list

A

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 reset criterion. Alarms and messages are displayed separately.
Alarms and messages in the parts program
Alarms and messages can be displayed directly from the parts program in plain text.
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.
Cycle alarms lie in the number range 60000 to 69999.

B

Blank

The part used to start machining a workpiece.

Block

A section of a -> parts program terminated with a line feed. A distinction is made between -> main blocks and -> subblocks.

Block search

The block search function allows selection of any point in the parts program 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.

Booting

Loading the system program after power ON.

C

CNC

-> NC

CNC high-level language

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.

COM — Component of the NC control for the implementation and coordination of communication.

Contour — Outline of a -> workpiece.

Coordinate system — See -> machine coordinate system, -> workpiece coordinate system

CPU — Central Processor Unit, -> PLC

Cycle — Protected subroutine for executing a machining process that is repeated on the -> workpiece

Cycle setting data — Using these special setting data the cycle parameter calculation can be varied.

Cycle support — The available cycles are listed in menu "Cycle support" 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 module — A data unit on the -> PLC which can be accessed by -> HIGHSTEP programs.
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.

Diagnosis — Operating area of the control.
The control has both a self-diagnosis program as well as test functions for servicing purposes: status, alarm and service displays.

Dimensions specification, metric and inches — Position and lead 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).
The cycle are programmed independent of the measuring system.

E

Editor — The editor makes it possible to create, modify, extend, join and import programs/texts/program blocks.

F

Finished-part contour Contour of the finished workpiece. See also -> Blank.

Frame A frame is a calculation rule that translates one Cartesian coordinate system into another Cartesian coordinate system. A frame contains the components -> zero offset, -> rotation, -> scaling, -> mirroring. Inside the cycle, additive frames programed, which have an effect on the actual-value display during the cycle.
After the cycle, the same workpiece coordinate system is active as before the call.

G

Geometry axis Geometry axes are used to describe a 2 or 3-dimensional area in the workpiece coordinate system.

Global main program/subroutine 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.

I

Identifier Words in compliance with DIN 66025 are supplemented by identifiers (names) for variables (arithmetic variables, system variables, user variables), for subroutines, for keywords and for words with several address letters. These supplements have the same meaning as the words with respect to block format. Identifiers must be unambiguous. It is not permissible to use the same identifier for different objects.

Imperial measurement system Measurement system which defines distances in "inches" and fractions of inches.

J

JOG Control operating mode (setup operation): The machine can be set up in the Jog mode. Individual axes and spindle can be traversed in jog mode by means of the direction keys. Other functions which are executed in jog mode are -> reference point approach, -> REPOS and -> preset (set actual value).

L

Languages
The operator-prompt display texts, system messages and system alarms are available (on diskette) in five system languages: **English, French, German, Italian** and **Spanish**.
The user can select **two** of the listed languages at a time in the control.

M

Machine
Operating area of the control.

Machine coordinate system
A coordinate system which is related to the axes of the machine tool.

Machine origin
A fixed point on the machine tool which can be referenced by all (derived) measurements systems.

Macros
A collections of instructions under a common identifier. The identifier in the program refers to the collected sequence of instructions.

Main program
-> Parts program identified by a number or name in which further main programs, subroutines or -> cycles may be called.

MDA
A mode in the control: 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.

Messages
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 reset criterion. Alarms and messages are displayed separately.

Metric measurement system
Standardized system of units: for lengths in millimeters (mm), meters (m), etc.

Mirroring
Mirroring exchanges the leading signs of the coordinate values of a contour in relation to an axis. Mirroring can be performed simultaneously in relation to several axes.

Module
"Module" is the term given to any files required for creating and processing programs.

N

NC Numerical control It incorporates all the components of the of the machine tool control system: -> NCK, -> PLC, -> MMC, -> COM. Note CNC (computerized numerical control) would be a more appropriate description for the SINUMERIK 810D or 840D.

NCK Numeric Control Kernel: Components of the NC control which executes -> parts programs and essentially coordinates the movements on the machine tool.

O

Oriented spindle stop Stops the workpiece spindle with a specified orientation angle, e.g. to perform an additional machining operation at a specific position. This function is used in several drilling cycles.

P

Parameter **840D**:
- Operating area of the control unit
- Computation parameter, can be set or scanned in the program at the discretion of the programmer for any purposes he may deem meaningful.

Parts program A sequence of instructions to the NC control which combine to produce a specific -> workpiece by performing certain machining operation on a given -> blank.

Parts program management 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.

PG Programming device

PLC Programmable logic control: -> Programmable logic control. Component of the -> NC control: A control which can be programmed to control the logic on a machine tool.

Polar coordinates A coordinate system which defines the position of a point on a plane in terms of its distance from the origin and the angle formed by the radius vector with a defined axis.

Power ON Control is switched off and then switched on again.
After loading the cycles it is always necessary to carry out a Power ON.

Program Operating area of the control.
Sequence of instructions to the control.

R

Rapid traverse The highest traversing speed of an axis. It is used to move the tool from rest to the -> workpiece contour or retract the tool from the contour.

Rigid tapping 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 end of the drilling depth, e.g. tapped blind hole (precondition: spindle is operating in axis mode). -> CYCLE84

Rotation Component of a -> frame which defines a rotation of the coordinate system through a specific angle.

R-Parameter Calculation parameter. The programmer of the -> parts program can assign or request the values of the R parameter as required.

S

Scaling Component of a -> frame which causes axis-specific alterations in the scale.

Serial RS-232 interface For the purpose of data input and output, one serial RS-232 interface (similar to European Standard V.24) (RS232) is provided on the MMC module MMC 100 and two RS-232 interfaces on the MMC modules MMC 101 and MMC 102. It is possible to load and save machining programs, cycles as well as manufacturer and user data via these interfaces.

Services Operating area of the control.

Setting data Data which provide the NC control with information on properties of the machine tool in a way defined by the system software.

Standard cycles

Standard cycles are provided for machining operations which are frequently repeated:
Cycles for drilling/milling applications
Cycles for turning applications (SINUMERIK FM-NC)
The available cycles are listed in menu "Cycle support" in the "Program" operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plain text.

Subroutine

A sequence of instructions of a -> parts program which can be called repetitively 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.

T

Tapping with compensating chuck

Tapping is carried out without spindle encoder (G33 or G63) -> CYCLE840

Text editor

-> Editor

Tool

A part used on the machine tool for machining. Examples of tools include cutting tools, mills, drills, laser beams, etc.

Tool edge radius compensation

When a contour is programmed, it is assumed that a pointed tool is used. Since this is not always possible, the control makes allowance for the curvature radius of the tool being used. The curvature centre point displaced by the curvature radius is guided equidistantly to the contour.
Turning and milling cycles select and deselect the cutting radius offset internally.

Tool offset

A tool is selected through the programming of a **T function** (5 decades, integer) in the block. Up to nine cutting edges (D addresses) can be assigned to each T number. The number of tools to be managed in the control is set at the configuration stage.

Tool radius compensation

In order to program a desired -> workpiece contour directly, the control must traverse a path equidistant to the programmed contour with allowance for the radius (G41/G42).

U

User-defined variables

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. Cycles work internally with user-defined variables.

V

Variable definition

A variable definition includes the specification of a data type and a variable name. The variable name can be used to address the value of the variable.

W

Workpiece

Part to be created/machined by the machine tool.

Workpiece contour

Setpoint contour of the -> workpiece to be created/machined.

Workpiece coordinate system

The starting position of the workpiece coordinate system is the -> workpiece origin. When programming in the workpiece coordinate system, the dimensions and directions refer to this system.

Workpiece origin

The workpiece origin is the starting point for the -> workpiece coordinate system. It is defined by the distance to the machine origin.

Z

Zero offset

Specification of a new reference point for a coordinate system through reference to an existing origin and a -> frame 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.
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
- from the PLC.

Programmable
It is possible to program zero offsets for all path and positioning axes by means of the TRANS statement.

C References

General Documentation

/BU/ SINUMERIK 840D/840Di/810D/802S, C, D
Ordering Information
Catalog NC 60
Order number: E86060-K4460-A101-A9-7600

/ST7/ **SIMATIC**
SIMATIC S7 Programmable Logic Controllers
Catalog ST 70
Order number: E86060-K4670-A111-A3-7600

/Z/ SINUMERIK, SIROTEC, SIMODRIVE
Connections & System Components
Catalog NC Z
Order number: E86060-K4490-A001-A8-7600

Electronic Documentation

/CD1/ The SINUMERIK System (11.02 Edition)
DOC ON CD
(includes all SINUMERIK 840D/840Di/810D/802 and SIMODRIVE publications)
Order number: 6FC5 298-6CA00-0BG3

User Documentation

/AUK/ SINUMERIK 840D/810D
AutoTurn Short Operating Guide (09.01 Edition)
Order number: 6FC5 298-4AA30-0BP3

/AUP/ SINUMERIK 840D/810D
AutoTurn Graphic Programming System (02.02 Edition)
Operator's Guide
Programming/Setup
Order number: 6FC5 298-4AA40-0BP3

/BA/ SINUMERIK 840D/810D
Operator's Guide MMC (10.00 Edition)
Order number: 6FC5 298-6AA00-0BP0

/BAD/ SINUMERIK 840D/840Di/810D
Operator's Guide: HMI Advanced (11.02 Edition)
Order number: 6FC5 298-6AF00-0BP2

/BEM/ SINUMERIK 840D/810D
Operator's Guide: HMI Embedded (11.02 Edition)
Order number: 6FC5 298-6AC00-0BP2

/BAH/ SINUMERIK 840D/840Di/810D
Operator's Guide HT 6 (HPU New) (06.02 Edition)
Order number: 6FC5 298-0AD60-0BP2

/BAK/ SINUMERIK 840D/840Di/810D
Short Operating Guide (02.01 Edition)
Order number: 6FC5 298-6AA10-0BP0

/BAM/ SINUMERIK 810D/840D
Operator's Guide ManualTurn (08.02 Edition)
Order number: 6FC5 298-6AD00-0BP0

/BAS/ SINUMERIK 840D/810D
Operator's Guide ShopMill (09.02 Edition)
Order number: 6FC5 298-6AD10-0BP1

/BAT/ SINUMERIK 810D/840D
Operator's Guide ShopTurn (10.02 Edition)
Order number: 6FC5 298-6AD50-0BP2

/BAP/ SINUMERIK 840D/840Di/810D
Handheld Programming Unit (04.00 Edition)
Order number: 6FC5 298-5AD20-0BP1

/BNM/ SINUMERIK 840D/840Di/810D
User's Guide Measuring Cycles (11.02 Edition)
Order number: 6FC5 298-6AA70-0BP2

/DA/ SINUMERIK 840D/840Di/810D
Diagnostics Guide (11.02 Edition)
Order number: 6FC5 298-6AA20-0BP3

/KAM/ SINUMERIK 840D/810D
Short Guide ManualTurn (04.01 Edition)
Order number: 6FC5 298-5AD40-0BP0

/KAS/ SINUMERIK 840D/810D
Short Guide ShopMill (04.01 Edition)
Order number: 6FC5 298-5AD30-0BP0

/PG/ SINUMERIK 840D/840Di/810D
Programming Guide Fundamentals (11.02 Edition)
Order number: 6FC5 298-6AB00-0BP2

/PGA/ SINUMERIK 840D/840Di/810D
Programming Guide Advanced (11.02 Edition)
Order number: 6FC5 298-6AB10-0BP2

/PGK/ SINUMERIK 840D/840Di/810D
Short Guide Programming (02.01 Edition)
Order number: 6FC5 298-6AB30-0BP1

/PGM/ SINUMERIK 840D/840Di/810D
Programming Guide ISO Milling (11.02 Edition)
Order number: 6FC5 298-6AC20-0BP2

/PGT/ SINUMERIK 840D/840Di/810D
Programming Guide ISO Turning (11.02 Edition)
Order number: 6FC5 298-6AC10-0BP2

/PGZ/ SINUMERIK 840D/840Di/810D
Programming Guide Cycles (11.02 Edition)
Order number: 6FC5 298-6AB40-0BP2

/PI/ **PCIN 4.4**
Software for Data Transfer to/from MMC Module
Order number: 6FX2 060-4AA00-4XB0 (English, French, German)
Order from: WK Fürth

/SYI/ SINUMERIK 840Di
System Overview (02.01 Edition)
Order number: 6FC5 298-6AE40-0BP0

Manufacturer/Service Documentation

a) Lists

/LIS/ SINUMERIK 840D/840Di/810D/
SIMODRIVE 611D
Lists (11.02 Edition)
Order number: 6FC5 297-6AB70-0BP3

b) Hardware

/BH/	SINUMERIK 840D/840Di/810D **Operator Components Manual** (HW) Order number: 6FC5 297-6AA50-0BP2	(11.02 Edition)
/BHA/	SIMODRIVE **Sensor** **Absolute Encoder with PROFIBUS DP** User's Guide (HW) Order number: 6SN1 197-0AB10-0YP1	(02.99 Edition)
/EMV/	SINUMERIK, SIROTEC, SIMODRIVE **EMC Installation Guideline** Planning Guide (HW) Order number: 6FC5 297-0AD30-0BP1	(06.99 Edition)
/PHC/	SINUMERIK 810D **Configuring Manual** (HW) Order number: 6FC5 297-6AD10-0BP0	(03.02 Edition)
/PHD/	SINUMERIK 840D **Configuring Manual NCU 561.2-573.2** (HW) Order number: 6FC5 297-6AC10-0BP2	(10.02 Edition)
/PHF/	SINUMERIK FM-NC **Configuring Manual NCU 570** (HW) Order number: 6FC5 297-3AC00-0BP0	(04.96 Edition)
/PMH/	SIMODRIVE **Sensor** **Hollow-Shaft Measuring System** Configuring/Installation Guide, SIMAG H (HW) Order number: 6SN1197-0AB30-0BP0	(05.99 Edition)

c) Software

/FB1/ SINUMERIK 840D/840Di/810D
Description of Functions, Basic Machine (Part 1) (11.02 Edition)
(the various manuals are listed below)
Order number: 6FC5 297-6AC20-0BP2

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 Coordinate Systems, Axis Types, Axis Configurations,
Actual-Value System for Workpiece, External Zero Offset
K4 Communication
N2 EMERGENCY STOP
P1 Transverse Axes
P3 Basic PLC Program
R1 Reference Point Approach
S1 Spindles
V1 Feeds
W1 Tool Compensation

/FB2/ SINUMERIK 840D/840Di/810D(CCU2)
Description of Functions, Extended Functions (Part 2) (11.02 Edition)
including FM-NC: Turning, Stepping Motor
(the various manuals are listed below)
Order number: 6FC5 297-6AC30-0BP2

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
Description of Functions Special Functions (Part 3) (11.02 Edition)
(the various manuals are listed below)
Order number: 6FC5 297-6AC80-0BP2

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
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 Unclocked Path-Synchronous Switching Signal Output
V2 Preprocessing
W3 3D Tool Radius Compensation

/FBA/ SIMODRIVE 611D/SINUMERIK 840D/810D
Description of Functions, Drive Functions (11.02 Edition)
(the various sections are listed below)
Order number: 6SN1 197-0AA80-0BP9

DB1 Operational Messages/Alarm Reactions
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
Description of Functions **ANA-MODULE** (02.00 Edition)
Order number: 6SN1 197-0AB80-0BP0

/FBD/ SINUMERIK 840D
Description of Functions **Digitizing** (07.99 Edition)
Order number: 6FC5 297-4AC50-0BP0
DI1 Start-Up
DI2 Scanning with Tactile Sensors (scancad scan)
DI3 Scanning with Lasers (scancad laser)
DI4 Milling Program Generation (scancad mill)

/FBDN/ CAM Integration DNC NT-2000
Description of Functions
System for NC Data Management and
Data Distribution (01.02 Edition)
Order number: 6FC5 297-5AE50-0BP2

/FBDT/ SINUMERIK 840D/840Di/810D
IT Solutions
Data Transfer via Network (SinDNC) (09.01 Edition)
Description of Functions
Order number: 6FC5 297-1AE70-0BP1

/FBFA/ SINUMERIK 840D/840Di/810D
Description of Functions
ISO Dialects for SINUMERIK (11.02 Edition)
Order number: 6FC5 297-6AE10-0BP2

/FBFE/ SINUMERIK 840D/810D
Description of Functions **Remote Diagnosis** (11.02 Edition)
Order number: 6FC5 297-0AF00-0BP2

/FBH/ SINUMERIK 840D/810D
HMI Programming Package (11.02 Edition)
Order number: (is part of the SW delivery)
Part 1 User's Guide
Part 2 Description of Functions

/FBHLA/ SINUMERIK 840D/SIMODRIVE 611 digital
Description of Functions
HLA Module (04.00 Edition)
Order number: 6SN1 197-0AB60-0BP2

/FBMA/ SINUMERIK 840D/810D
Description of Functions **ManualTurn** (08.02 Edition)
Order number: 6FC5 297-6AD50-0BP0

/FBO/ SINUMERIK 840D/810D
Description of Functions
Configuring of OP 030 Operator Interface (09.01 Edition)
(the various sections are listed below)
Order number: 6FC5 297-6AC40-0BP0

BA Operator's Guide
EU Development Environment (Configuring Package)
PSE Introduction to Configuring of Operator Interface
IK Screen Kit: Software Update and Configuration
PS Online only: Configuring Syntax (Configuring Package)

/FBP/ SINUMERIK 840D
Description of Functions **C-PLC Programming** (03.96 Edition)
Order number: 6FC5 297-3AB60-0BP0

/FBR/ SINUMERIK 840D/810D
IT Solutions
Description of Functions **Computer Link** (SINCOM) (09.01 Edition)
Order number: 6FC5 297-6AD60-0BP0

NFL Host Computer Interface
NPL PLC/NCK Interface

/FBSI/ SINUMERIK 840D/SIMODRIVE
Description of Functions **SINUMERIK Safety Integrated** (09.02 Edition)
Order number: 6FC5 297-6AB80-0BP1

/FBSP/ SINUMERIK 840D/810D
Description of Functions **ShopMill** (09.02 Edition)
Order number: 6FC5 297-6AD80-0BP1

/FBST/ **SIMATIC**
FM STEPDRIVE/SIMOSTEP (01.01 Edition)
Description of Functions
Order number: 6SN1 197-0AA70-0YP4

/FBSY/ SINUMERIK 840D/810D
Description of Functions **Synchronized Actions** (10.02 Edition)
for Wood, Glass, Ceramics and Presses
Order number: 6FC5 297-6AD40-0BP2

/FBT/ SINUMERIK 840D/810D
Description of Functions **ShopTurn** (10.02 Edition)
Order number: 6FC5 297-6AD70-0BP2

/FBTC/ SINUMERIK 840D/810D
IT Solutions
SINUMERIK Tool Data Communication SinTDC (01.02 Edition)
Description of Function
Order number: 6FC5 297-5AF30-0BP0

/FBTD/ SINUMERIK 840D/810D
Description of Functions
Tool Information System (SinTDI) with Online Help (02.01 Edition)
Order number: 6FC5 297-6AE00-0BP0

/FBU/ **SIMODRIVE 611 universal**
Description of Functions
Closed-Loop Control Component for Speed Control and Positioning
Description of Functions (08.02 Edition)
Order number: 6SN1 197-0AB20-0BP6

/FBW/ SINUMERIK 840D/810D
Description of Functions **Tool Management** (10.02 Edition)
Order number: 6FC5 297-6AC60-0BP1

/FBWI/ SINUMERIK 840D/840Di/810D
Description of Functions **WinTPM** (02.02 Edition)
Order number: This document is part of the software

/HBA/ SINUMERIK 840D/840Di/810D
Manual @Event (01.02 Edition)
Order number: 6AU1900-0CL20-0BA0

/HBI/ SINUMERIK 840Di
Manual (09.02 Edition)
Order number: 6FC5 297-6AE60-0BP0

/INC/ SINUMERIK 840D/840Di/810D
Commissioning Tool **SINUMERIK SinuCOM NC** (02.02 Edition)
Order number: (an integral part of the online Help for the start-up tool)

/PFK/ **SIMODRIVE**
Planning Guide **1FT5/1FT6/1FK6 Motors** (12.01 Edition)
AC Servo Motors for Feedrate and Main Spindle Drives
Order number: 6SN1 197-0AC20-0BP0

/PJE/ SINUMERIK 840D/810D
HMI Embedded Configuring Package (08.01 Edition)
Description of Functions: Software Update, Configuration, Installation
Order number: 6FC5 297-6EA10-0BP0
(the document PS Configuring Syntax is supplied with the software and available as a pdf file)

/PJFE/ **SIMODRIVE**
Planning Guide (09.01 Edition)
Built-In Synchronous Motors 1FE1
Three-Phase AC Motors for Main Spindle Drives
Order number: 6SN1 197-0AC00-0BP1

/PJLM/ **SIMODRIVE**
Planning Guide **Linear Motors 1FN1, 1FN3** (11.01 Edition)
ALL General Information about Linear Motors
1FN1 1FN1 Three-Phase Linear Motor
1FN3 1FN3 Three-Phase Linear Motor
CON Connections
Order number: 6SN1 197-0AB70-0BP2

/PJM/ **SIMODRIVE**
Planning Guide **Motors** (11.00 Edition)
Three-Phase AC Motors for Feed and Main Spindle Drives
Order number: 6SN1 197-0AA20-0BP5

/PJU/ **SIMODRIVE 611**
Planning Guide **Inverters** (08.02 Edition)
Order number: 6SN1 197-0AA00-0BP6

/PMS/ **SIMODRIVE** (04.02 Edition)
Planning Guide **ECO Motor Spindle** for Main Spindle Drives
Order number: 6SN1 197-0AD04-0BP0

/POS1/ **SIMODRIVE POSMO A**
User's Guide (08.02 Edition)
Distributed Positioning Motor on PROFIBUS DP
Order number: 6SN2 197-0AA00-0BP3

/POS2/ **SIMODRIVE POSMO A**
Installation Guide (enclosed with every POSMO A) (12.98 Edition)
Order number: 462 008 0815 00

/POS3/	**SIMODRIVE POSMO SI/CD/CA** Distributed Servo Drive Systems, Operator's Guide Order number: 6SN2 197-0AA20-0BP3	(08.02 Edition)
/PPH/	**SIMODRIVE** Planning Guide **1PH2/1PH4/1PH7 Motors** AC Induction Motors for Main Spindle Drives Order number: 6SN1 197-0AC60-0BP0	(12.01 Edition)
/PPM/	**SIMODRIVE** Planning Guide **Hollow-Shaft Motors** Hollow-Shaft Motors for Main Spindle Drives 1PM4 and 1PM6 Order number: 6SN1 197-0AD03-0BP0	(10.01 Edition)
/S7H/	**SIMATIC S7-300** Reference Manual: CPU Data (HW Description) Reference Manual: Module Data Order number: 6ES7 398-8AA03-8AA0	(10.98 Edition)
/S7HT/	**SIMATIC S7-300** Manual STEP 7, Fundamentals, V. 3.1 Order number: 6ES7 810-4CA02-8AA0	(03.97 Edition)
/S7HR/	**SIMATIC S7-300** Manual STEP7, Reference Manuals, V3.1 Order number: 6ES7 810-4CA02-8AR0	(03.97 Edition)
/S7S/	**SIMATIC S7-300** **FM 353** Stepper Drive Positioning Module Order in conjunction with configuring package	(04.97 Edition)
/S7L/	**SIMATIC S7-300** **FM 354** Positioning Module for Servo Drive Order together with configuring package	(04.97 Edition)
/S7M/	**SIMATIC S7-300** **FM 357** Multimodule for Servo and Stepper Drives Order together with configuring package	(10.99 Edition)

/SP/ **SIMODRIVE 611-A/611-D,**
SimoPro 3.1
Program for Configuring Machine-Tool Drives
Order number: 6SC6 111-6PC00-0AA❑,
Order from: WK Fürth

d) Installation and Start-Up

/IAA/ **SIMODRIVE 611A**
Installation and Start-Up Guide (10.00 Edition)
(including description of SIMODRIVE 611D start-up software)
Order number: 6SN 1197-0AA60-0BP6

/IAC/ SINUMERIK 810D
Installation and Start-Up Guide (11.02 Edition)
(including description of SIMODRIVE 611D start-up software)
Order number: 6FC5 297-6AD20-0BP0

/IAD/ SINUMERIK 840D/SIMODRIVE 611D
Installation and Start-Up Guide (11.02 Edition)
(including description of SIMODRIVE 611D start-up software)
Order number: 6FC5 297-6AB10-0BP2

/IAM/ SINUMERIK 840D/840Di/810D
HMI/MMC Installation and Start-Up Guide (11.02 Edition)
Order number: 6FC5 297-6AE20-0BP2

AE1 Updates/Options
BE1 Expand the operator interface
HE1 Online Help
IM2 Start-Up HMI Embedded
IM4 Start-Up HMI Advanced
TX1 Setting Foreign Language Texts

D Index

Notes

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810 D | 840Di

E Identifiers

List of input/output variables for measuring cycles

Name	*Meaning in English*	*Meaning in German*
AD	Allowance depth	Rohmaß Taschentiefe von Referenz-ebene
AFSL	Angle for slot length	Winkel für die Nutlänge
ANG1, ANG2	Flank angle	Flankenwinkel
ANGB	Liftoff angle for roughing	Abhebwinkel beim Schruppen
AP1	Unfinished dimension in plane	Rohmaß Taschenlänge/Taschenradius
AP2	Unfinished dimension in plane	Rohmaß Taschenbreite
APP	Approach path	Einlaufweg
APX	Axial value for defining blank for facing axis	achsweiser Wert zur Rohteildefinition für Planachse
APXA	Absolute or incremental evaluation of parameter APX	Bewertung des Parameters _APX absolut oder inkrementell
APZ	Axial value for defining blank for longitudinal axis	achsweiser Wert zur Rohteildefinition für Längsachse
APZA	Absolute or incremental evaluation of parameter APZ	Bewertung des Parameters _APZ absolut oder inkrementell
AS1, AS2	Direction of approach/approach travel	Spezifikation der Anfahrrichtung/-bahn
AXN	Tool axis	Werkzeugachse
BNAME	Name for program of drill positions	Name für Programm der Bohrpositionen
CDIR	Circle direction	Drehrichtung, Fräsrichtung
CPA	Center point, abscissa	Mittelpunkt des Lochkreises, Abszisse (absolut)
CPO	Center point,ordinate	Mittelpunkt des Lochkreises, Ordinate (absolut)
CRAD	Corner radius	Eckenradius
DAM	Degression value, Path for roughing interrupt	Degressionsfaktor / Weglänge
DBH	Distance between holes	Abstand zwischen den Bohrungen
DIAG	Groove depth	Einstichtiefe
DIATH	Diameter of thread	Nenndurchmesser, Außendurchmesser des Gewindes
DIS1	Distance	programmierbarer Vorhalteabstand
DIS1	Distance between columns	Abstand der Spalten
DIS2	Number of lines, Distance between rows	Abstand der Zeilen

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810 D | 840Di

DM1 ... DM4	Diameter	Durchmesser des Gewindes am Anfangspunkt
DP	Depth	Tiefe (absolut)
DP1	First depth	Eintauchtiefe
DPR	Depth, relative	Tiefe relativ zur Referenzebene
DT	Dwell time	Verweilzeit zum Spänebrechen beim Schruppen
DTB	Dwell time at bottom	Verweilzeit auf Endbohrtiefe/am Einstichgrund
DTD	Dwell time at depth	Verweilzeit auf Endbohrtiefe
DTS	Dwell time at starting point	Verweilzeit am Anfangspunkt
ENC	Tapping with/without encoder	Gewindebohren mit/ohne Geber
FAL	Finish allowance	konturgerechtes Schlichtaufmaß am Nutrand/Taschenrand
FAL1	Finish allowance on groove base	Schlichtaufmaß am Einstichgrund
FAL2	Finish allowance on flanks	Schlichtaufmaß an den Flanken
FALD	Finish allowance depth	Schlichtaufmaß am Grund
FALZ	Finish allowance, z axis	Schlichtaufmaß in der Längsachse
FALX	Finish allowance, x axis	Schlichtaufmaß in der Planachse
FDEP	First depth	erste Bohrtiefe (absolut)
FDIS	First distance	Abstand der ersten Bohrung vom Bezugspunkt
FDP1	Overrun path in direction to plane	Überlaufweg in Richtung der Ebenenzustellung
FDPR	First depth, relative	erste Bohrtiefe relativ zur Referenzebene
FF1	Feedrate for roughing	Vorschub für Schruppen
FF2	Feedrate for insertion	Vorschub zum Eintauchen
FF3	Feedrate for finishing	Vorschub für Schlichten
FF4	Feedrate for contour transition elements	Vorschub an Konturübergangselementen
FFCP (ab SW 6.3)	Feedrate for circular positioning	Vorschub für Zwischenpositionierung auf Kreisbahn
FFD	Feedrate for depth	Vorschub für Tiefenzustellung
FFP1	Feedrate surface	Vorschub für Flächenbearbeitung
FFP2	Feedrate for finishing	Vorschub für Schlichtbearbeitung
FFR	Feedrate	Vorschub
FORM	Definition of form	Definition der Form
FPL	Final point along longitudinal axis	Endpunkt in der Längsachse
FRF	Feedrate factor	Vorschubfaktor
IANG	Infeed angle	Zustellwinkel
INDA	Incremental angle	Fortschaltwinkel

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810 D | 840Di

IDEP	Infeed depth	Zustelltiefe
KNAME	Name of the contour subroutine	Name des Konturunterprogramms
LSANF	Block number/label identifying start of contour definition	Satznummer/Label des Beginns der Konturbeschreibung
LSEND	Block number/label identifying end of contour definition	Satznummer/Label des Endes der Konturbeschreibung
LP1	Length of approach travel, radius	Länge des Anfahrwegs, Radius
LP2	Length of return travel, radius	Länge des Abfahrwegs, Radius
KDIAM	Internal diameter of thread	Kerndurchmesser, Innendurchmesser des Gewindes
LENG	Elongated hole length, pocket length	Langlochlänge, Taschenlänge
MDEP	Minimum depth	Mindestbohrtiefe
MID	Maximum infeed depth	maximale Zustelltiefe für eine Zustellung
MIDA	Maximum infeed width	maximale Zustellbreite
MIDF	Maximum infeed depth for finishing	maximale Zustelltiefe für Schlichtbearbeitung
MPIT	Thread lead as thread size	Gewindesteigung als Gewindegröße
NID	Number of noncuts	Anzahl der Leerschnitte
NP1 ... NP8	Name/Label ...	Name des Konturunterprogramms der Fertigteilkontur/Label
NPP	Name of parts program	Name des Konturunterprogramms
NRC	Number of roughing cuts	Anzahl der Schruppschnitte
NSP	Start point offset first thread	Startpunktversatz für den ersten Gewindegang
NUM	Number of holes	Anzahl der Bohrungen
NUM1	Number of columns	Anzahl der Spalten
NUM2	Number of lines	Anzahl der Zeilen
NUMT	Number of threads	Anzahl der Gewindegänge
PA	Reference point, abscissa	Bezugspunkt Tasche
PO	Reference point, ordinate	Bezugspunkt Tasche
PO1 ... PO4	Point in longitudinal axis	Anfangspunkt/Zwischenpunkt/Endpunkt des Gewindes in der Längsachse
PIT	Thread lead	Gewindesteigung als Wert
PNAME	Name for pocket milling machining program	Name für Taschenfräsen Bearbeitungsprogramm
POSS	Position for oriented spindle stop	Spindelposition
PP1 ... PP3	Thread pitch 1...3 as value	Gewindesteigung 1...3 als Wert
PRAD	Pocket radius	Taschenradius
RAD	Radius	Radius des Kreises
RAD1	Radius	Radius der Helixbahn beim Eintauchen
RCO1, RCO2	Radius/chamfer outside	Radius/Fase, außen

840 D NCU 571 | 840 D NCU 572 NCU 573 | 810 D | 840Di

RCI1, RCI2	Radius/chamfer inside	Radius/Fase, innen
RFF	Retract feed	Rückzugsvorschub
RFP	Reference plane	Referenzebene (absolut)
ROP	Run out path	Auslaufweg
RPA	Retract path, abscissa	Rückzugsweg in der Abszisse
RPAP	Retract path, applicate	Rückzugsweg in der Applikate
RPO	Retract path, ordinate	Rückzugsweg in der Ordinate
RL	Bypass contour	Umfahren der Kontur mittig
RTP	Retract plane	Rückzugsebene (absolut)
SDAC	Spindle direction after cyle	Drehrichtung nach Zyklusende
SDIR	Spindle direction	Spindelrichtung
SDIS	Safety distance	Sicherheitsabstand
SDR	Spindle direction for retraction	Drehrichtung für Rückzug
SPCA	Reference point, abscissa	Abszisse eines Bezugspunktes auf der Geraden (absolut)
SPCO	Reference point, ordinate	Ordinate dieses Bezugspunktes (absolut)
SPD	Starting point in the facing axis	Anfangspunkt in der Planachse
SPL	Starting point along longitudinal axis	Anfangspunkt in der Längsachse
SSF	Speed for finishing	Drehzahl bei Schlichtbearbeitung
SST	Speed for tapping	Drehzahl für Gewindebohren
SST1	Speed for retraction	Drehzahl für Rückzug
STA, STA1	Angle	Winkel
STA2	Insertion angle	Maximaler Eintauchwinkel für Pendelbewegung
TDEP	Thread depth	Gewindetiefe
TN	Name of stock removal tool	Name des Ausräumwerkzeuges
TOL1	Blank tolerance	Rohteiltoleranz
TYPTH	Typ of thread	Gewindetyp
VARI	Working	Bearbeitungsart
VRT	Variable return path	variabler Rückzugsbetrag / Rückzugsweg
WID	(Pocket) width	Taschenbreite
WIDG	Groove width	Einstichbreite

To SIEMENS AG A&D MC BMS P.O. Box 3180 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
From Name Company/Department Address: Zip Code: Phone: / Fax: /	Programming Guide Order no.: 6FC5 298-6AB40-0BP2 Edition: 11.02 If you come across any printing errors in this document, please let us know using form provided. We would also appreciate and new ideas and/or suggestion for improvements.

Suggestions and/or corrections

Overview of SINUMERIK 840D/840Di/810D Documentation (11.2002)

General Documentation

- *SINUMERIK* 840D/810D — Brochure
- *SINUMERIK* 840D/840Di/810D/ — Catalog Ordering Info. NC 60 ***)**
- *SINUMERIK SIROTEC SIMODRIVE* Accessories — Catalog Accessories NC-Z

User Documentation

- *SINUMERIK* 840D/810D/FM-NC — **AutoTurn**
  – Short Guide
  – Programming/ Setup
- *SINUMERIK* 840D/840Di/810D — Operator's Guide **– HT 6**
- *SINUMERIK* 840D/840Di/810D — **Diagnostics Guide *)**
- *SINUMERIK* 840D/840Di/810D — **Operator's Guide *)**
  – Short Guide
  – HMI Embedded
  – HMI Advanced

User Documentation

- *SINUMERIK* 840D/840Di/810D — **Program. Guide**
  – Short Guide
  – Fundamentals ***)**
  – Advanced ***)**
  – Cycles
  – Measuring Cycles
  – ISO Turning/Milling
- *SINUMERIK* 840D/810D — Operator's Guide
  – **ManualTurn**
  – Short Guide ManualTurn
  – **ShopMill**
  – Short Guide ShopMill
  – **ShopTurn**
  – Short Guide ShopTurn
- *SINUMERIK* 840Di — **System Overview**

Manufacturer/Service Documentation

- *SINUMERIK* — **Configuring** (HW) ***)**
  – 810D
  – 840D
- *SINUMERIK* 840D/840Di/810D — **Operator Components** (HW) ***)**
- *SINUMERIK* 840D/810D — Description of Functions
  – ManualTurn
  – ShopMill
  – ShopTurn
- *SINUMERIK* 840D/840Di/810D — Description of Functions **Synchronized Actions**

Manufacturer/Service Documentation

- *SINUMERIK SIMODRIVE* 611D 840D/810D — Description of Functions **Drive Functions *)**
- *SINUMERIK* 840D/840Di/810D — **Description of Functions**
  – Basic Machine ***)**
  – Extended Functions
  – Special Functions
- *SINUMERIK* 840D/840Di/810D — Description of Functions **Tool Manage- ment**
- *SINUMERIK* 840D/810D — **Configuring Kit** HMI Embedded
- *SINUMERIK* 840D/810D — Description of Functions Operator Interface **OP 030**
- *SINUMERIK* 840D/810D — **IT Solutions**
  – Computer Link
  – Tool Data Information System
  – NC Data Management
  – NC Data Transfer
  – Tool Data Communication

Manufacturer/Service Documentation

- *SINUMERIK SIMODRIVE* — Description of Functions SINUMERIK **Safety Integrated**
- *SINUMERIK* 840D — Description of Functions **Digitizing**
- *SINUMERIK SIMODRIVE* — **Installation & Start-Up Guide *)**
  – 810D
  – 840D/611D
  – HMI
- *SINUMERIK SIMODRIVE* 840D/840Di 810D 611D — **Lists *)**
- *SINUMERIK SIMODRIVE* 840D 611D — Description of Functions **Linear Motor**
- *SINUMERIK SIMODRIVE* 840D 611D — Description of Functions
  – Hydraulics Module
  – Analog Module
- *SINUMERIK SIMODRIVE SIROTEC* — **EMC** Guidelines

Electronic Documentation

- *SINUMERIK SIMODRIVE* 840D/840Di/810D 611, Motors — **DOC ON CD *)** The SINUMERIK System

Manufacturer/Service Documentation

- *SINUMERIK* 840D/840Di/810D — Description of Functions **ISO Dialects** for SINUMERIK
- *SINUMERIK* 840Di — **Manual** (HW + Installation and Start-Up)
- *SINUMERIK* 840D/810D — Description of Functions **Remote Diagnosis**
- *SINUMERIK* 840D/840Di/810D — **Manual @Event**

***)** These documents are a minimum requirement

Siemens AG
Automation & Drives
Motion Control Systems
P.O. Box 3180, D-91050 Erlangen
Germany

www.ad.siemens.de

Order No: 6FC5 298-6AB40-0BP2

Printed in Germany