SIEMENS

SINUMERIK

SINUMERIK 808D
Milling Part 2: Programming (Siemens instructions)

Programming and Operating Manual

Valid for:
SINUMERIK 808D Milling (software version: V4.4.2)

Target group:
End users and service engineers

12/2012
6FC5398-4DP10-0BA0

Legal information

Warning notice system

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

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

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

⚠ CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

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

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Order number: 6FC5398-4DP10-0BA0
Ⓟ 12/2012 Technical data subject to change

Table of contents

1 Programming principles

1.1 Fundamentals of programming

1.1.1 Program names

Each program must have a program name. The program name must follow the conventions below:

- Use a maximum of 24 letters or 12 Chinese characters for a program name (the character length of the file extension excluded)
- Separate the file extension only with a decimal point
- Enter the file extension ".SPF" if the current default program type is MPF (main program) and you desire to create a subprogram
- Enter the file extension ".MPF" if the current default program type is SPF (subprogram) and you desire to create a main program
- Do not enter the file extension if you desire to take the current default program type
- Avoid using special characters for program names.

Example

WORKPIECE527

1.1.2 Program structure

Structure and content

The NC program consists of a sequence of **blocks** (see the table below). Each block represents a machining step. Instructions are written in the blocks in the form of **words**. The last block in the execution sequence contains a special word for the end of the program, for example, **M2**.

The following table shows you an example of the NC program structure.

Block	Word	Word	Word	...	; Comment
Block	N10	G0	X20	...	; First block
Block	N20	G2	Z37	...	; Second block
Block	N30	G91	...	...	; ...
Block	N40	...	...	...	
Block	N50	M2			; End of program

1.1.3 Word structure and address

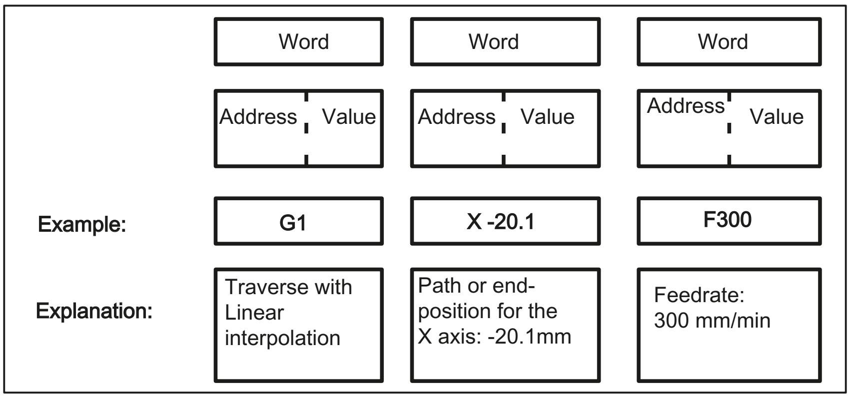
Functionality/structure

A word is a block element and mainly constitutes a control command. The word consists of the following two parts:

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

  A positive sign (+) can be omitted.

The following picture shows an example of the word structure.



Several address characters

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

Additionally, it is also possible to call G functions using a symbolic name (For more information, refer to Section "List of instructions (Page 12)".).
Example: **SCALE** ; Enable scaling factor

Extended address

With the following addresses, the address is extended by 1 to 4 digits to obtain a higher number of addresses. In this case, the value must be assigned using an equality sign "=".

R	Arithmetic parameters
H	H function
I, J, K	Interpolation parameters/intermediate point
M	Special function M, affecting the spindle with other options
S	Spindle speed

Examples: **R10=6.234 H5=12.1 I1=32.67 M2=5 S1=400**

1.1.4 Character set

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

Letters, digits

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

Printable special characters

(	Open parenthesis	"	Inverted commas
)	Close parenthesis	_	Underscore (belongs to letters)
[	Open square bracket	.	Decimal point
]	Close square bracket	,	Comma, separator
<	less than	;	Comment start
>	greater than	%	Reserved; do not use
:	Main block, end of label	&	Reserved; do not use
=	Assignment, part of equation	'	Reserved; do not use
/	skip	$	System variable identifiers
*	Multiplication	?	Reserved; do not use
+	Addition and positive sign	!	Reserved; do not use
-	Subtraction, minus sign		

Non-printable special characters

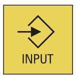
LF	End-of-block character
Blank	Separator between words; blank
Tab character	Reserved; do not use

1.1.5 Block format

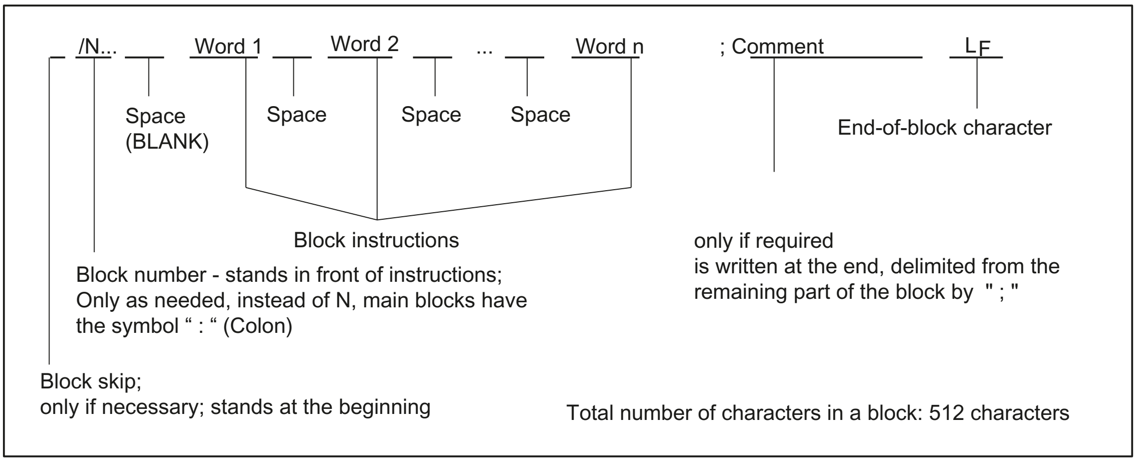
Functionality

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

Generally, a block consists of several **words** and is always completed with the **end-of-block character** " L_F " (Linefeed). When writing a block, this character is automatically generated when pressing the linefeed key on an externally connected keyboard or pressing the following key on the PPU:



See the following block structure diagram:



Word order

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

Note regarding block numbers

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

Block skip

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

Comment, remark

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

Messages

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

Programming example

```
N10                                         ; G&S company, order no. 12A71
N20                                         ; Pump part 17, drawing no.: 123 677
N30                                         ; Program created by H. Adam, Dept. TV 4
N40 MSG("DRAWING NO.: 123677")
:50 G54 F4.7 S220 D2 M3                     ;Main block
N60 G0 G90 X100 Z200
N70 G1 Z185.6
N80 X112
/N90 X118 Z180                              ; Block can be suppressed
N100 X118 Z120
N110 G0 G90 X200
N120 M2                                     ; End of program
```

1.1.6 List of instructions

The functions marked with an asterisk (*) are active at the start of the program in the CNC milling variant, unless otherwise they are programmed or the machine manufacturer has preserved the default settings for the "milling" technology.

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

Address	Significance	Value assignments	Information	Programming
G3	Circular interpolation in counter-clockwise direction (in conjunction with a third axis and TURN=... also helix interpolation -> see under TURN)			G3 ... ; otherwise as for G2
CIP	Circular interpolation through intermediate point			CIP X... Y... Z... I1=... J1=... K1=... F...
CT	Circular interpolation; tangential transition			N10 ... N20 CT X... Y... F... ;circle, tangential transition to the previous path segment
G33	Thread cutting, tapping with constant pitch			S... M... ;Spindle speed, direction G33 Z... K...; Thread drilling **with** compensating chuck, e.g. in Z axis
G331	Thread interpolation			N10 SPOS=... ; Spindle in position control N20 G331 Z... K... S... ; tapping **without** compensating chuck e.g. in Z axis; RH or LH thread is defined via the sign of the pitch (e.g. K+): + : as with M3 - : as with M4
G332	Thread interpolation - retraction			G332 Z... K... ;**Rigid tapping**, e.g. in Z axis, **retraction motion**; sign of pitch as for G331
G4	Dwell time		2: Special motions, non-modal	G4 F...;separate block, F: Time in seconds or G4 S.... ;separate block, S: in spindle revolutions
G63	Tapping with compensating chuck			G63 Z... F... S... M...
G74	Reference point approach			G74 X1=0 Y1=0 Z1=0; separate block, (machine axis identifier!)
G75	Fixed point approach			G75 X1=0 Y1 =0 Z1=0; separate block, (machine axis identifier!)
G147	SAR - Approach with a straight line			G147 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G148	SAR - Retract with a straight line			G148 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
G247	SAR - Approach with a quadrant			G247 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G248	SAR - Retract with a quadrant			G248 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
G347	SAR - Approach with a semicircle			G347 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G348	SAR - Retract with a semicircle			G348 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...

Address	Significance	Value assignments	Information	Programming
TRANS	Translation, programmable		3: Write memory, non-modal	TRANS X... Y... Z...; separate block
ROT	Rotation, programmable			ROT RPL=... ; rotation in the current plane G17 to G19, separate block
SCALE	Programmable scaling factor			SCALE X... Y... Z... ; scaling factor in the direction of the specified axis, separate block
MIRROR	Programmable mirroring			MIRROR X0; coordinate axis whose direction is changed, separate block
ATRANS	Additive translation, programming			ATRANS X... Y... Z... ; separate block
AROT	Additive programmable rotation			AROT RPL=... ; rotation in the current plane G17 to G19, separate block
ASCALE	Additive programmable scaling factor			ASCALE X... Y... Z...; scaling factor in the direction of the specified axis, separate block
AMIRROR	Additive programmable mirroring			AMIRROR X0 ; coordinate axis whose direction is changed, separate block
G110	Pole specification relative to the last programmed setpoint position			G110 X... Y... ; Pole specification, Cartesian, e.g.: for G17 G110 RP=... AP=... ; Pole specification, polar, separate block
G111	Pole specification relative to origin of current workpiece coordinate system			G111 X... Y... ; Pole specification, Cartesian, e.g.: for G17 G111 RP=... AP=... ; Pole specification, polar, separate block
G112	Pole specification, relative to the last valid POLE			G112 X... Y... ; Pole specification, Cartesian, e.g.: for G17 G112 RP=... AP=... ; Pole specification, polar, separate block
G17 *	X/Y plane		6: Plane selection, modally effective	G17 .... ; Vertical axis on this plane is tool length
G18	Z/X plane			
G19	Y/Z plane			Compensation axis
G40 *	Tool radius compensation OFF		7: Tool radius compensation, modally effective	
G41	Tool radius compensation left of contour			
G42	Tool radius compensation right of contour			
G500 *	Settable work offset OFF		8: Settable work offset,	

Address	Significance	Value assignments	Information	Programming
G54	1. Settable work offset		modally effective	
G55	2. Settable work offset			
G56	3. Settable work offset			
G57	4. Settable work offset			
G58	5. Settable work offset			
G59	6. Settable work offset			
G53	Non-modal skipping of the settable work offset		9: Suppression of settable work offset, non-modal	
G153	Non-modal skipping of the settable work offset including base frame			
G60 *	Exact stop		10: Approach behavior, modally effective	
G64	Continuous-path mode			
G62	Corner deceleration at inside corners when tool radius offset is active (G41, G42)		Only in conjunction with continuous-path mode	G62 Z... G1
G9	Non-modal exact stop		11: Exact stop, non-modal	
G601 *	Exact stop window, fine, with G60, G9		12: Exact stop window, modally effective	
G602	Exact stop window, coarse, with G60, G9			
G621	Corner deceleration at all corners		Only in conjunction with continuous-path mode.	G621 AIDS=...
G70	Inch dimension data input		13: Inch/metric dimension input, modally effective	
G71 *	Metric dimension data input			
G700	Inch dimension data input; also for feedrate F			
G710	Metric dimension data input; also for feedrate F			
G90 *	Absolute dimension data input		14: Absolute / incremental dimension, modally effective	
G91	Incremental dimension data input			
G94 *	Feed F in mm/min		15: Feedrate / spindle, modally effective	
G95	Feedrate F in mm/spindle revolutions			
CFC *	Feedrate override with circle ON		16: Feed override, modally effective	
CFTCP	Feedrate override OFF			
G450 *	Transition circle		18: Behavior at corners when working with tool radius compensation, modally effective	
G451	Point of intersection			
BRISK *	Jerking path acceleration		21: Acceleration profile, modally effective	
SOFT	Jerk-limited path acceleration			
FFWOF *	Feedforward control OFF		24: Feedforward control, modally effective	
FFWON	Feedforward control ON			
EXTCALL	Execute external subprogram			Reload program from HMI in "Execution from external source" mode.
G340 *	Approach and retraction in space (SAR)		44: Path segmentation with SAR, modally effective	
G341	Approach and retraction in the plane (SAR)			
G290 *	SIEMENS mode		47: External NC languages,	

Address	Significance	Value assignments	Information	Programming
G291	External mode		modally effective	
H H0= to H9999=	H function	± 0.0000001 ... 9999 9999 (8 decimal places) or with specification of an exponent: ± (10^{-300} ... 10^{+300})	Value transfer to the PLC; meaning defined by the machine manufacturer	H0=... H9999=... e.g.: H7=23.456
I	Interpolation parameters	±0.001 ... 99 999.999 Thread: 0.001 ... 2000.000	Belongs to the X axis; meaning dependent on G2, G3 ->circle center or G33, G331, G332 -> thread pitch	See G2, G3, G33, G331 and G332
J	Interpolation parameters	±0.001 ... 99 999.999 Thread: 0.001 ... 2000.000	Belongs to the Y axis; otherwise, as with I	See G2, G3, G33, G331, and G332
K	Interpolation parameters	±0.001 ... 99 999.999 Thread: 0.001 ... 2000.000	Belongs to the Z axis; otherwise, as with I	See G2, G3, G33, G331, and G332
I1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the X axis; specification for circular interpolation with CIP	See CIP
J1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the Y axis; specification for circular interpolation with CIP	See CIP
K1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the Z axis; specification for circular interpolation with CIP	See CIP
L	Subroutine; name and call	7 decimals; integer only, no sign	Instead of a free name, it is also possible to select L1 ...L9999999; this also calls the subroutine (UP) in a separate block. Please note: L0001 is not always equal to L1. The name "LL6" is reserved for the tool change subroutine.	L781; separate block
M	Additional function	0 ... 99 only integer, no sign	For example, for initiating switching actions, such as "coolant ON", maximum five M functions per block.	M...
M0	Programmed stop		The machining is stopped at the end of a block containing M0; to continue, press NC START.	
M1	Optional stop		As with M0, but the stop is only performed if a special signal (Program control: "M01") is present.	
M2	End of main program with return to beginning of program		Can be found in the last block of the processing sequence	
M30	End of program (as M2)		Can be found in the last block of the processing sequence	

Address	Significance	Value assignments	Information	Programming
M17	End of subroutine		Can be found in the last block of the processing sequence	
M3	CW rotation of spindle			
M4	CCW rotation of spindle			
M5	Spindle stop			
M6	Tool change		Only if activated with M6 via the machine control panel; otherwise, change directly using the T command	
M40	Automatic gear stage changeover			
M41 to M45	Gear stage 1 to gear stage 5			
M70, M19	-		Reserved; do not use	
M...	Remaining M functions		Functionality is not defined by the control system and can therefore be used freely by the machine manufacturer	
N	Block number - subblock	0 ... 9999 9999 only integer, no sign	Can be used to identify blocks with a number; is written at the beginning of a block	N20 ...
:	Block number of a main block	0 ... 9999 9999 only integer, no sign	Special block identification, used instead of N... ; such a block should contain all instructions for a complete subsequent machining step.	:20 ...
P	Number of subroutine passes	1 ... 9999 only integer, no sign	Is used if the subroutine is run several times and is contained in the same block as the call	N10 L781 P... ; separate block N10 L871 P3 ; three cycles
R0 to R299	Arithmetic parameters	± 0.0000001 ... 9999 9999 (8 decimal places) or with specification of an exponent: ± (10^{-300} ... 10^{+300})		R1=7.9431 R2=4 with specification of an exponent: R1=-1.9876EX9; R1=-1 987 600 000
	Arithmetic functions		In addition to the 4 basic arithmetic functions using the operands + - * /, there are the following arithmetic functions:	
SIN( )	Sine	Degrees		R1=SIN(17.35)
COS()	Cosine	Degrees		R2=COS(R3)
TAN()	Tangent	Degrees		R4=TAN(R5)
ASIN()	Arc sine			R10=ASIN(0.35) ; R10: 20.487 degrees
ACOS()	Arc cosine			R20=ACOS(R2) ; R20: ... Degrees
ATAN2(,)	Arctangent2		The angle of the sum vector is calculated from 2 vectors standing vertically one on another. The 2nd vector specified is always used for angle reference. Result in the range: -180 to +180 degrees	R40=ATAN2(30.5,80.1) ; R40: 20.8455 degrees
SQRT()	Square root			R6=SQRT(R7)

Address	Significance	Value assignments	Information	Programming
POT()	Square			R12=POT(R13)
ABS()	Absolute value			R8=ABS(R9)
TRUNC()	Truncate to integer			R10=TRUNC(R11)
LN()	Natural logarithm			R12=LN(R9)
EXP()	Exponential function			R13=EXP(R1)
RET	Subroutine end		Used instead of M2 - to maintain the continuous-path control mode	RET ; separate block
S...	Spindle speed	0.001 ... 99 999.999	Unit of measurement of the spindle speed rpm	S...
S	Dwell time in block with G4	0.001 ... 99 999.999	Dwell time in spindle revolutions	G4 S... ; separate block
T	Tool number	1 ... 32 000 only integer, no sign	The tool change can be performed either directly using the T command or only with M6. This can be set in the machine data.	T...
X	Axis	±0.001 ... 99 999.999	Positional data	X...
Y	Axis	±0.001 ... 99 999.999	Positional data	Y...
Z	Axis	±0.001 ... 99 999.999	Positional data	Z...
AC	Absolute coordinate	-	The dimension can be specified for the end or center point of a certain axis, irrespective of G91.	N10 G91 X10 Z=AC(20) ;X - incremental dimension, Z - absolute dimension
ACC[*axis*]	Percentage acceleration override	1 ... 200, integer	Acceleration override for an axis or spindle; specified as a percentage	N10 ACC[X]=80 ;for the X axis 80% N20 ACC[S]=50;for the spindle: 50%
ACP	Absolute coordinate; approach position in the positive direction (for rotary axis, spindle)	-	It is also possible to specify the dimensions for the end point of a rotary axis with ACP(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=ACP(45.3) ;approach absolute position of the A axis in the positive direction N20 SPOS=ACP(33.1) ;position spindle
ACN	Absolute coordinate; approach position in the negative direction (for rotary axis, spindle)	-	It is also possible to specify the dimensions for the end point of a rotary axis with ACN(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=ACN(45.3) ;approach absolute position of the A axis in the negative direction N20 SPOS=ACN(33.1) ;position spindle
ANG	Angle for the specification of a straight line for the contour definition	±0.00001 ... 359.99999	Specified in degrees; one possibility of specifying a straight line when using G0 or G1 if only one end-point coordinate of the plane is known or if the complete end point is known with contour ranging over several blocks	N10 G1 G17 X... Y.... N11 X... ANG=... or contour over several blocks: N10 G1 G17 X... Y.... N11 ANG=... N12 X... Y... ANG=...

Address	Significance	Value assignments	Information	Programming
AP	Polar angle	0 ... ±359.99999	Specification in degrees, traversing in polar coordinates, definition of the pole; in addition: Polar radius RP	See G0, G1, G2; G3, G110, G111, G112
AR	Aperture angle for circular interpolation	0.00001 ... 359.99999	Specified in degrees; one possibility of defining the circle when using G2/G3	See G2, G3
CALL	Indirect cycle call	-	Special form of the cycle call; no parameter transfer; the name of the cycle is stored in a variable; only intended for cycle-internal use	N10 CALL VARNAME ; variable name
CHF	Chamfer; general use	0.001 ... 99 999.999	Inserts a chamfer of the specified chamfer length between two contour blocks	N10 X... Y.... CHF=... N11 X... Y...
CHR	Chamfer; in the contour definition	0.001 ... 99 999.999	Inserts a chamfer of the specified side length between two contour blocks	N10 X... Y.... CHR=... N11 X... Y...
CR	Radius for circular interpolation	0.010 ... 99 999.999 Negative sign - for selecting the circle: greater than semicircle	One possibility of defining a circle when using G2/G3	See G2, G3
CYCLE... HOLES... POCKET... SLOT...	Machining cycle	Only specified values	Call of machining cycles requires a separate block, the provided transfer parameters must be assigned values,special cycle calls are possible with additional MCALL or CALL	
CYCLE81	Drilling, centering			N5 RTP=110 RFP=100 .... ; Assign with values N10 CYCLE81(RTP, RFP, ...); separate block
CYCLE82	Drilling, counterboring			N5 RTP=110 RFP=100 .... ; Assign with values N10 CYCLE82(RTP, RFP, ...); separate block
CYCLE83	Deep-hole drilling			N10 CYCLE83(110, 100, ...) ;or transfer values directly; separate block
CYCLE84	Rigid tapping			N10 CYCLE84(...) ;separate block
CYCLE840	Tapping with compensating chuck			N10 CYCLE840(...); separate block
CYCLE85	Reaming 1			N10 CYCLE85(...) ; separate block
CYCLE86	Boring			N10 CYCLE86(...) ; separate block

Address	Significance	Value assignments	Information	Programming
CYCLE87	Drilling with stop 1			N10 CYCLE87(...); separate block
CYCLE88	Drilling with stop 2			N10 CYCLE88(...) ; separate block
CYCLE89	Reaming 2			N10 CYCLE89(...); separate block
CYCLE802	Arbitrary positions			N10 CYCLE802(...) ; separate block
HOLES1	Row of holes			N10 HOLES1(...); separate block
HOLES2	Circle of holes			N10 HOLES2(...); separate block
SLOT1	Mill slot			N10 SLOT1(...); separate block
SLOT2	Mill a circumferential slot			N10 SLOT2(...); separate block
POCKET3	Rectangular pocket			N10 POCKET3(...); separate block
POCKET4	Circular pocket			N10 POCKET4(...); separate block
CYCLE71	Face milling			N10 CYCLE71(...); separate block
CYCLE72	Contour milling			N10 CYCLE72(...); separate block
CYCLE76	Milling the rectangular spigot			N10 CYCLE76(...) ; separate block
CYCLE77	Circular spigot milling			N10 CYCLE77(...) ; separate block
CYCLE90	Thread milling			N10 CYCLE90(...); separate block
LONGHOLE	Elongated hole			N10 LONGHOLE(...); separate block
CYCLE832	High speed settings			N10 CYCLE832(...); separate block
DC	Absolute coordinate; approach position directly (for rotary axis, spindle)	-	It is also possible to specify the dimensions for the end point of a rotary axis with DC(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=DC(45.3); Approach absolute position of the A axis directly N20 SPOS=DC(33.1); Position spindle
DEF	Definition instruction		Defining a user variable of the BOOL, CHAR, INT, REAL, STRING[n] type; define directly at the beginning of the program	DEF INT VARI1=24, VARI2; two variables of the INT type; name defined by the user DEF STRING[12] VARS3="HELLO" ; max. 12 characters
DISCL	Approach / retraction distance of infeed motion for machining plane (SAR)	-	Safety clearance for speed switchover for infeed motion; note: G340, G341	See G147, G148, G247, G248, G347, G348

Address	Significance	Value assignments	Information	Programming
DISR	Approach / retraction distance or radius (SAR)	-	G147/G148: Distance of the cutter edge from the starting or end point of the contour G247, G347/G248, G348: Radius of the tool center pointpath	See G147, G148, G247, G248, G347, G348
FAD	Velocity for the infeed (SAR)	-	Speed takes effect after the safety clearance is reached for the infeed; note: G340, G341	See G147, G148, G247, G248, G347, G348
FRC	Non-modal feedrate for chamfer/rounding	0, >0	When FRC=0, feedrate F will act	For the unit, see F and G94, G95; for chamfer/rounding, see CHF, CHR, RND
FRCM	Modal feedrate for chamfer/rounding	0, >0	When FRCM=0, feedrate F will act	For the unit, see F and G94, G95; for rounding/modal rounding, see RND, RNDM
GOTOB	GoBack instruction	-	A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the program start.	N10 LABEL1: ... ... N100 GOTOB LABEL1
GOTOF	GoForward instruction	-	A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the end of the program.	N10 GOTOF LABEL2 ... N130 LABEL2: ...
IC	Coordinate specified using incremental dimensions		The dimension can be specified for the end or center point of a certain axis irrespective of G90.	N10 G90 X10 Z=IC(20); Z - incremental dimension, X - absolute dimension
IF	Jump condition	-	If the jump condition is fulfilled, the jump to the block with *Label: is performed*; otherwise, next instruction/block; several IF instructions per block are possible **Relational operators:** = = equal, <> not equal, > greater than, < less than, >= greater than or equal to, <= less than or equal to	N10 IF R1>5 GOTOF LABEL3 ... N80 LABEL3: ...
MEAS	Measuring **with** deletion of distance-to-go	+1 -1	=+1: Measuring input 1, rising edge =-1: Measuring input1, falling edge	N10 **MEAS=-1** G1 X... Y... Z... F...
MEAW	Measuring **without** deletion of distance-to-go	+1 -1	=+1: Measuring input 1, rising edge =-1: Measuring input1, falling edge	N10 MEAW=-1 G1 X... Y... Z... F...

Address	Significance	Value assignments	Information	Programming
$A_DBB[n] $A_DBW[n] $A_DBD[n] $A_DBR[n]	Data byte Data word Data double-word Real data		Reading and writing PLC variables	N10 $A_DBR[5]=16.3 ; Write Real variables; with offset position 5; (position, type and meaning are agreed between NC and PLC)
$AA_MM[*axis**	Measurement result for an axis in the machine coordinate system	-	*Axis*: Identifier of an axis (X, Y, Z ...) traversing when measuring	N10 R1=$AA_MM[X]
$AA_MW[*axis*]	Measurement result for an axis in the workpiece coordinate system	-	*Axis*: Identifier of an axis (X, Y, Z ...) traversing when measuring	N10 R2=$AA_MW[X]
$A..._..._TIME	Timer for runtime: $AN_SETUP_TIME $AN_POWERON_TIME $AC_OPERATING_TIME $AC_CYCLE_TIME $AC_CUTTING_TIME	0.0 ... 10^{+300} min (value read-only) min (value read-only) s s s	System variable: Time since the control system has last booted Time since the control system has last booted normally Total runtime of all NC programs Runtime of the NC program (only of the selected program) Tool action time	N10 IF $AC_CYCLE_TIME==50.5 ....
$AC_..._PARTS	Workpiece counter: $AC_TOTAL_PARTS $AC_REQUIRED_PARTS $AC_ACTUAL_PARTS $AC_SPECIAL_PARTS	0 ... 999 999 999, integer	System variable: Total actual count Set number of workpiece Current actual count Count of workpieces - specified by the user	N10 IF $AC_ACTUAL_PARTS==15 ....
$AC_MEA[1]	Measuring task status	-	Default condition: 0: Default condition, probe did not switch 1: Probe switched	N10 IF $AC_MEAS[1]==1 GOTOF .... ; Continue program when probe has switched ...
$P_TOOLNO	Number of the active tool T	-	read-only	N10 IF $P_TOOLNO==12 GOTOF ....
$P_TOOL	Active D number of the active tool	-	read-only	N10 IF $P_TOOL==1 GOTOF ....
MCALL	Modal subprogram call	-	The subroutine in the block containing MCALL is called automatically after each successive block containing a path motion. The call acts until the next MCALL is called. Application example: Drilling a hole pattern	N10 MCALL CYCLE82(...); Separate block, drilling cycle N20 HOLES1(...); Row of holes N30 MCALL; Separate block, modal call of CYCLE82(...) completed

Address	Significance	Value assignments	Information	Programming
MSG ()	Signal	Max. 65 characters	Message text in inverted commas	N10 MSG("MESSAGE TEXT"); separate block ... N150 MSG() ; Clear previous message
OFFN	Dimension specification	-	Only effective with the tool radius compensation G41, G42 active	N10 OFFN=12.4
RND	Rounding	0.010 ... 99 999.999	Inserts a rounding with the specified radius value tangentially between two contour blocks, special FRC= ... feed possible	N10 X... Y.... RND=4.5 N11 X... Y...
RNDM	Modal rounding	0.010 ... 99 999.999 0	• Inserts roundings with the specified radius value tangentially at the following contour corners; special feedrate possible: FRCM= ... • Modal rounding OFF	N10 X... Y.... RNDM=.7.3; modal rounding ON N11 X... Y... .... N100 RNDM=.0; modal rounding OFF
RP	Polar radius	0.001 ... 99 999.999	Traversing in polar coordinates, pole specification; in addition: Polar angle AP	See G0, G1, G2; G3, G110, G111, G112
RPL	Angle of rotation with ROT, AROT	±0.00001 ... 359.9999	Specification in degrees; angle for a programmable rotation in the current plane G17 to G19	See ROT, AROT
SET(, , ,) REP()	Set values for the variable fields		SET: Various values, from the specified element up to: according to the number of values REP: the same value, from the specified element up to the end of the field	DEF REAL VAR2[12]=REP(4.5) ; all elements value 4.5 N10 R10=SET(1.1,2.3,4.4) ; R10=1.1, R11=2.3, R4=4.4
SF	Thread starting point when using G33	0.001 ... 359.999	Specified in degrees; the thread starting point with G33 will be offset by the specified value (not applicable for tapping)	See G33
SPI(n)	Converts the spindle number n into the axis identifier		n =1, axis identifier: e.g. "SP1" or "C"	
SPOS	Spindle position	0.0000 ... 359.9999 with incremental specification (IC): ±0.001 ... 99 999.999	Specified in degrees; the spindle stops at the specified position (to achieve this, the spindle must provide the appropriate technical prerequisites: position control	N10 SPOS=.... N10 SPOS=ACP(...) N10 SPOS=ACN(...) N10 SPOS=IC(...) N10 SPOS=DC(...)

Address	Significance	Value assignments	Information	Programming
STOPFIFO	Stops the fast machining step	-	Special function; filling of the buffer memory until STARTFIFO, "Buffer memory full" or "End of program" is detected.	STOPFIFO; separate block, start of filling N10 X... N20 X...
STARTFIFO	Start of fast machining step	-	Special function; the buffer memory is filled at the same time.	N30 X... STARTFIFO ;separate block, end of filling
STOPRE	Preprocessing stop	-	Special function; the next block is only decoded if the block before STOPRE is completed.	STOPRE ; separate block
TURN	Number of additional circle passes with helix interpolation	0 ... 999	In conjunction with circular interpolation G2/G3 in a plane G17 to G19 and infeed motion of the axis vertical to this plane	N10 G0 G17 X20 Y5 Z3 N20 G1 Z-5 F50 N30 G3 X20 Y5 Z-20 I0 J7.5 **TURN=2**; total of three full circles

1.2 Positional data

1.2.1 Programming dimensions

In this section you will find descriptions of the commands, with which you can directly program dimensions taken from a drawing. This has the advantage that no extensive calculations have to be made for NC programming.

Note

The commands described in this section stand in most cases at the start of a NC program. The way, in which these functions are combined, is not intended to be a patent remedy. For example, the choice of working plane may be made at another point in the NC program. The real purpose of this and the following sections is to illustrate the conventional structure of an NC program.

Overview of typical dimensions

The basis of most NC programs is a drawing with concrete dimensions.

When implementing in a NC program, it is helpful to take over exactly the dimensions of a workpiece drawing into the machining program. These can be:

- Absolute dimension, G90 modally effective applies for all axes in the block, up to revocation by G91 in a following block.
- Absolute dimension, X=AC(value) only this value applies only for the stated axis and is not influenced by G90/G91. This is possible for all axes and also for SPOS, SPOSA spindle positionings, and interpolation parameters I, J, K.

- Absolute dimension, X=CC(value) directly approaching the position by the shortest route, only this value applies only for the stated rotary axis and is not influenced by G90/G91. This is also possible for SPOS, SPOSA spindle positionings.
- Absolute dimension, X=ACP(value) approaching the position in positive direction, only this value is set only for the rotary axis, the range of which is set to 0... < 360 degrees in the machine data.
- Absolute dimension, X=ACN(value) approaching the position in negative direction, only this value is set only for the rotary axis, the range of which is set to 0... < 360 degrees in the machine data.
- Incremental dimension, G91 modally effective applies for all axes in the block, until it is revoked by G90 in a following block.
- Incremental dimension, X=IC(value) only this value applies exclusively for the stated axis and is not influenced by G90/G91. This is possible for all axes and also for SPOS, SPOSA spindle positionings, and interpolation parameters I, J, K.
- Inch dimension, G70 applies for all linear axes in the block, until revoked by G71 in a following block.
- Metric dimension, G71 applies for all linear axes in the block, until revoked by G70 in a following block.
- Inch dimension as for G70, but applies also for feedrate and length-related setting data.
- Metric dimension as for G71, but applies also for feedrate and length-related setting data.
- Diameter programming, DIAMON on
- Diameter programming, DIAMOF off

Diameter programming, DIAM90 for traversing blocks with G90. Radius programming for traversing blocks with G91.

1.2.2 Plane selection: G17 to G19

Functionality

To assign, for example, **tool radius and tool length compensations**, a plane with two axes is selected from the three axes X, Y and Z. In this plane, you can activate a tool radius compensation.

For drill and cutter, the length compensation (length1) is assigned to the axis standing vertically on the selected plane. It is also possible to use a 3-dimensional length compensation for special cases.

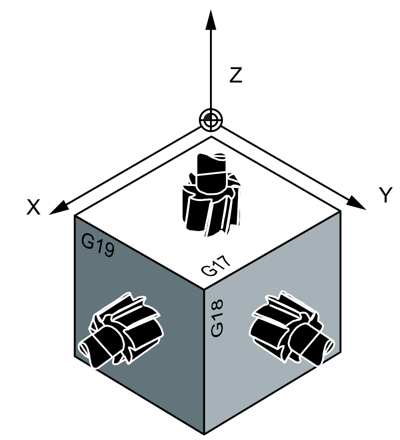
Another influence of plane selection is described with the appropriate functions (e.g. Section "Support for the contour definition programming").

The individual planes are also used to define the **direction of rotation of the circle for the circular interpolation** CW or CCW. In the plane in which the circle is traversed, the abscissa and the ordinate are designed and thus also the direction of rotation of the circle. Circles can also be traversed in a plane other than that of the currently active G17 to G19 plane (For more information , refer to Section "Axis movements (Page 39)".).

The following plane and axis assignments are possible:

G function	Plane (abscissa/ordinate)	Vertical axis on plane (length compensation axis when drilling/milling)
G17	X/Y	Z
G18	Z/X	Y
G19	Y/Z	X

See the following illustration for planes and axes when drilling/milling:



Programming example

```
N10 G17 T... D... M...    ; X/Y plane selected
N20 ... X... Y... Z...    ; tool length compensation (length1) in Z axis
```

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

Functionality

With the instructions G90/G91, the written positional data X, Y, Z, ... are evaluated as a coordinate point (G90) or as an axis position to traverse to (G91). G90/G91 applies to all axes.

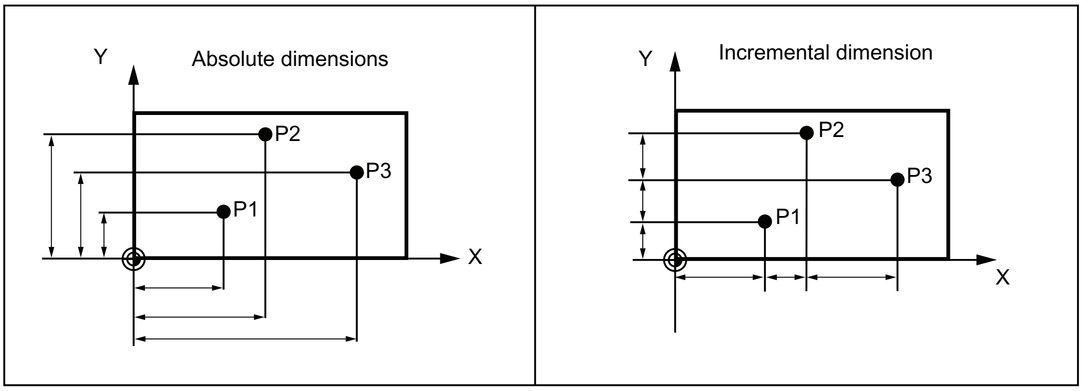
Irrespective of G90/G91, certain positional data can be specified for certain blocks in absolute/incremental dimensions using AC/IC.

These instructions do **not determine the path** by which the end points are reached; this is provided by a G group (G0, G1, G2 and G3.... For more information, refer to Section "Axis movements (Page 39)".).

Programming

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

See the following illustration for different dimensioning types in the drawing:



Absolute dimensioning G90

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

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

Incremental dimensioning G91

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

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

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

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

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

Programming example

```
N10 G90 X20 Z90          ; Absolute dimensions
N20 X75 Z=IC(-32)        ; X-dimensions remain absolute, incremental Z dimension
N180 G91 X40 Z20         ; Switch-over to incremental dimensioning
N190 X-12 Z=AC(17)       ; X-remains incremental dimensioning, Z-absolute
```

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

Functionality

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

Programming

G70	; Inch dimensions
G71	' Metric dimensions
G700	; Inch dimensions, also for feedrate F
G710	; Metric dimensions, also for feedrate F

Programming example

```
N10 G70 X10 Z30          ; Inch dimensions
N20 X40 Z50              ;G70 continues to act
N80 G71 X19 Z17.3        ; metric dimensioning from this point on
```

Information

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

All examples listed in this manual are based on a **metric default setting**.

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

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

All remaining geometric parameters that are not direct workpiece parameters, such as feedrates, tool offsets, and **settable** work offsets, are not affected by **G70/G71**.

G700/G710 however, also affects the feedrate F (inch/min, inch/rev. or mm/min, mm/rev.).

1.2.5 Polar coordinates, pole definition: G110, G111, G112

Functionality

In addition to the common specification in Cartesian coordinates (X, Y, Z), the points of a workpiece can be specified using the polar coordinates.

Polar coordinates are also helpful if a workpiece or a part of it is dimensioned from a central point (pole) with specification of the radius and the angle.

Plane

The polar coordinates refer to the plane activated with G17 to G19. In addition, the third axis standing vertically on this plane can be specified. When doing so, spatial specifications can be programmed as cylinder coordinates.

Polar radius RP=...

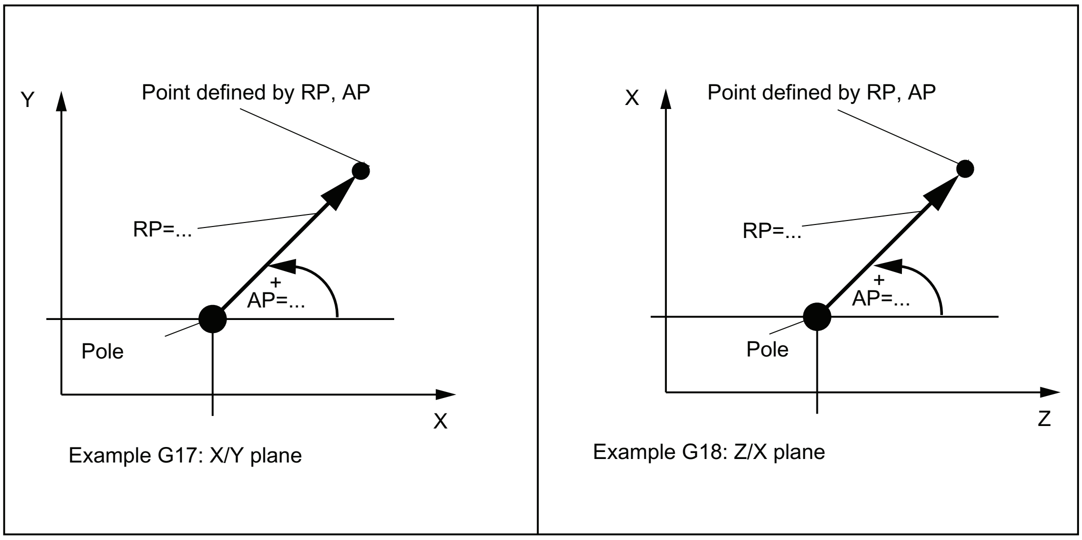
The polar radius specifies the distance of the point to the pole. It is stored and must only be written in blocks in which it changes, after changing the pole or when switching the plane.

Polar angle AP=...

The angle is always referred to the horizontal axis (abscissa) of the plane (for example, with G17: X axis). Positive or negative angle specifications are possible.

The polar angle remains stored and must only be written in blocks in which it changes, after changing the pole or when switching the plane.

See the following illustration for polar radius and polar angle with definition of the positive direction in different planes:



Pole definition, programming

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

Note

Pole specifications

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

Programming example

```
N10 G17                     ; X/Y plane
N20 G0 X0 Y0
N30 G111 X20 Y10            ; Pole coordinates in the current workpiece
                            coordinate system
N40 G1 RP=50 AP=30 F1000
N50 G110 X-10 Y20
N60 G1 RP=30 AP=45 F1000
N70 G112 X40 Y20            ; New pole, relative to the last pole as a polar
                            coordinate
N80 G1 RP=30 AP=135         ; Polar coordinate
M30
```

Traversing with polar coordinates

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

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

(See also Section "Axis movements (Page 39)".)

1.2.6 Programmable work offset: TRANS, ATRANS

Functionality

The programmable work offset can be used:

- for recurring shapes/arrangements in various positions on the workpiece
- when selecting a new reference point for the dimensioning
- as a stock allowance when roughing

This results in the current workpiece coordinate system. The rewritten dimensions use this as a reference.

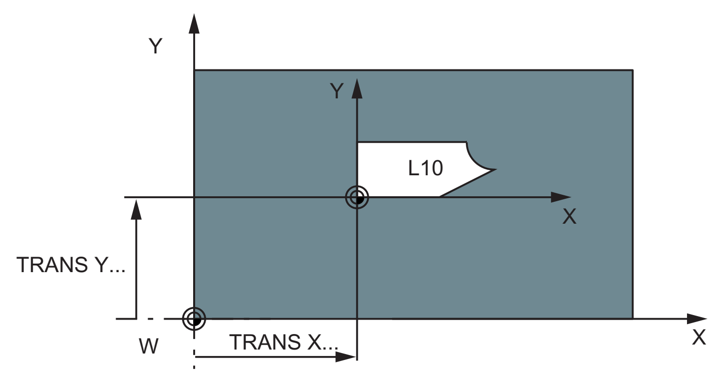
The offset is possible in all axes.

Programming

TRANS X... Y... Z...	; programmable offset, deletes old instructions for offsetting, rotation, scaling factor, mirroring
ATRANS X... Y... Z...	; programmable offset, additive to existing instructions
TRANS	; without values: clears old instructions for offset, rotation, scaling factor, mirroring

The instructions which contain TRANS or ATRANS each require a separate block.

See the following illustration for the example for programmable offset:



Programming example

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

Subroutine call - see Section "Subroutine technique (Page 97)".

1.2.7 Programmable rotation: ROT, AROT

Functionality

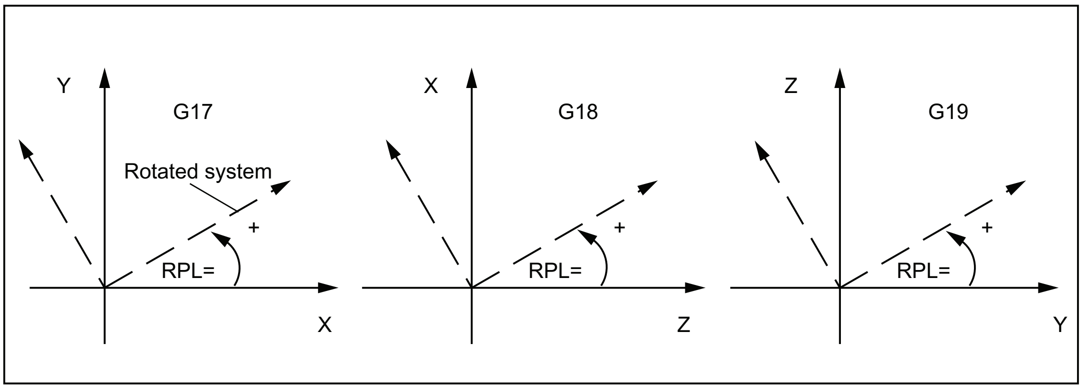
The rotation is performed in the current plane G17 or G18 or G19 using the value of RPL=... specified in degrees.

Programming

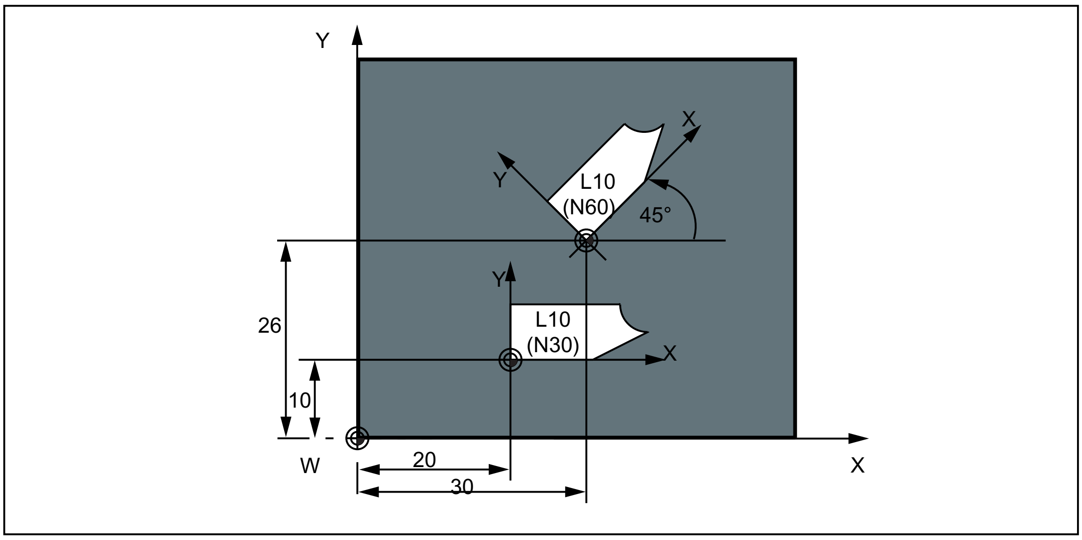
ROT RPL=...	; Programmable rotation, deletes old instructions for offsetting, rotation, scaling factor, mirroring
AROT RPL=...	; Programmable rotation, additive to existing instructions
ROT	; Without values: clears old instructions for offset, rotation, scaling factor, mirroring

The instructions which contain ROT or AROT each require a separate block.

See the following illustration for definition of the positive direction of the angle of rotation in the individual planes:



See the following illustration for programming example for programmable offset and rotation:



Programming example

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

Subroutine call - see Section "Subroutine technique (Page 97)".

1.2.8 Programmable scaling factor: SCALE, ASCALE

Functionality

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

Programming

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

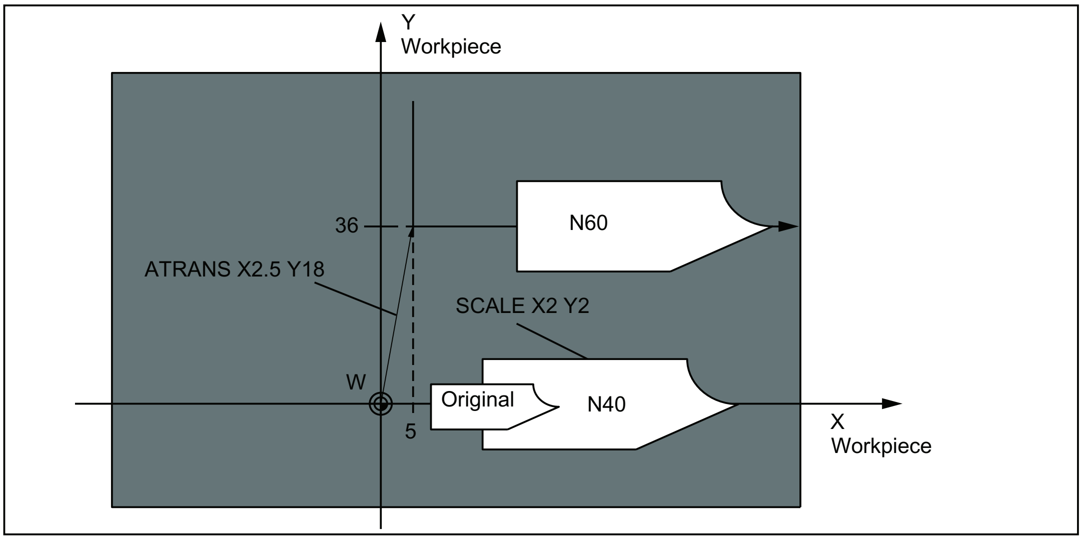
The instructions which contain SCALE or ASCALE each require a separate block.

Note

For circles, the same factor should be used in both axes.

If ATRANS is programmed with SCALE/ASCALE active, these offset values are also scaled.

See the following illustration for example for scaling and offset:



Programming example

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

Subroutine call - see Section "Subroutine technique (Page 97)".

1.2.9 Programmable mirroring: MIRROR, AMIRROR

Functionality

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

Programming

MIRROR X0 Y0 Z0	; Programmable mirroring, clears old instructions for offset, rotation, scaling factor, mirroring
AMIRROR X0 Y0 Z0	; Programmable mirroring, additive to existing instructions
MIRROR	; Without values: clears old instructions for offset, rotation, scaling factor, mirroring

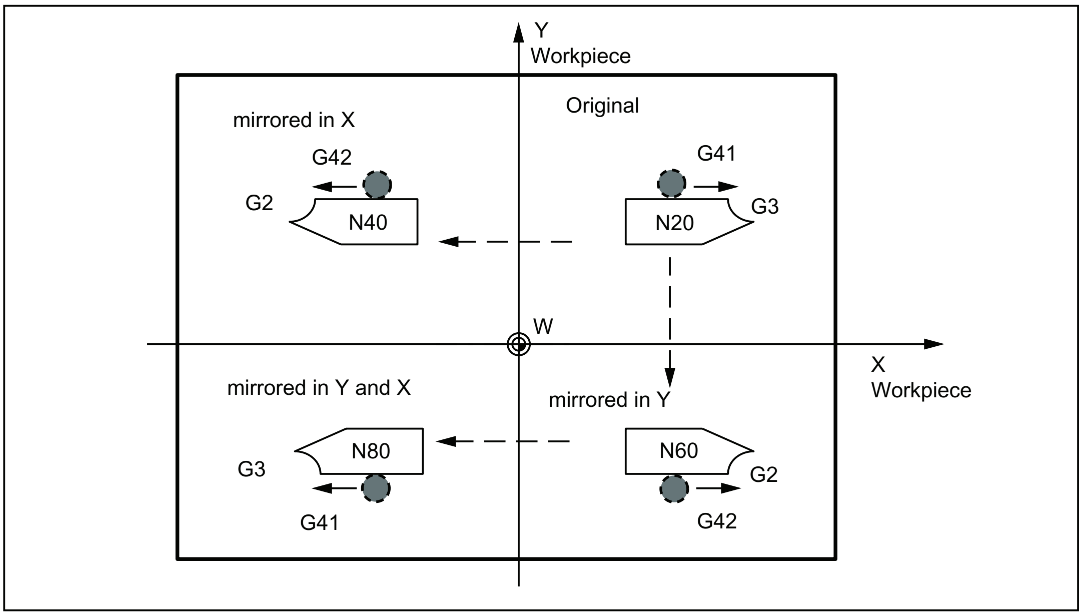
The instructions that contain MIRROR or AMIRROR each require a separate block. The axis value has no influence. A value, however, must be specified.

Note

Any active tool radius compensation (G41/G42) is reversed automatically when mirroring.

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

See the following illustration for example for mirroring with the tool position shown:



Programming example

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

```
...
N10 G17                      ; X/Y plane, Z standing vertically on it
N20 L10                      ; Programmed contour with G41
N30 MIRROR X0                ; Direction changed in X
N40 L10                      ; Mirrored contour
N50 MIRROR Y0                ; Direction changed in Y
N60 L10
N70 AMIRROR X0               ; Mirroring once more, but now in X
N80 L10                      ; Twice-mirrored contour
N90 MIRROR                   ; Mirroring off
```

Subroutine call - see Section "Subroutine technique (Page 97)".

1.2.10 Workpiece clamping - settable work offset: G54 to G59, G500, G53, G153

Functionality

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

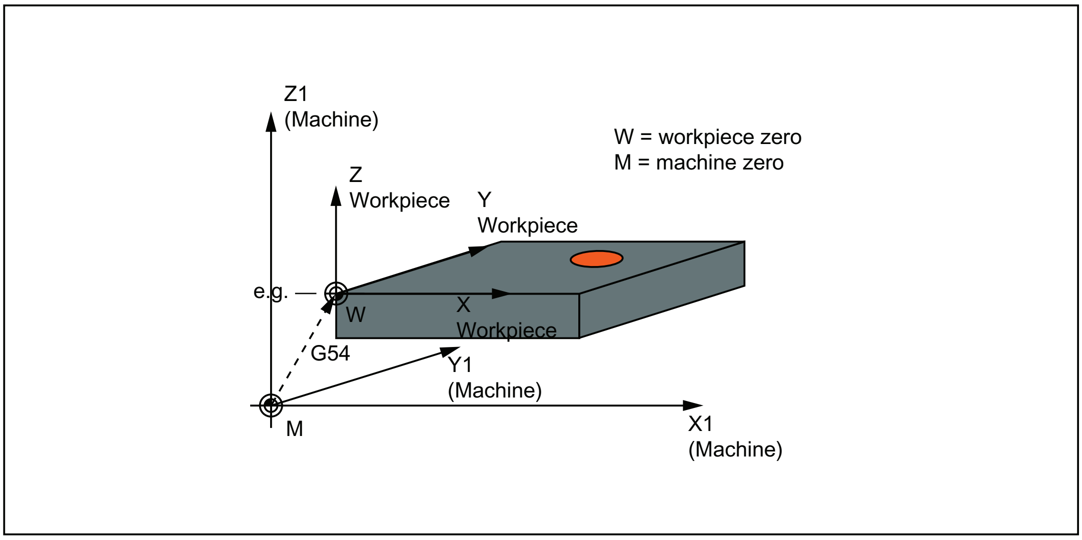
Note

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

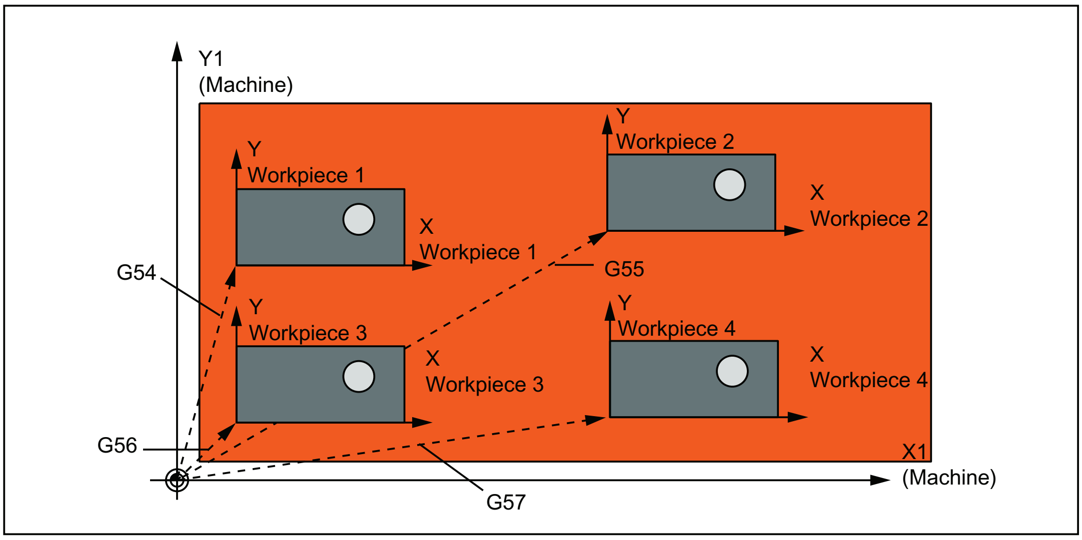
Programming

G54 to G59	; 1. to 6th settable work offset
G500	; Settable work offset OFF - modal
G53	; settable work offset OFF, non-modal, also suppresses programmable offset
G153	;settable work offset OFF, non-modal; additionally suppresses base frame

See the following illustration for settable work offset:



See the following illustration for various workpiece clamping positions when drilling/milling:



Programming example

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

Subroutine call - see Section "Subroutine technique (Page 97)".

1.3 Axis movements

1.3.1 Linear interpolation with rapid traverse: G0

Functionality

The rapid traverse movement G0 is used for rapid positioning of the tool, but not for direct workpiece machining.

All the axes can be traversed simultaneously - on a straight path.

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

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

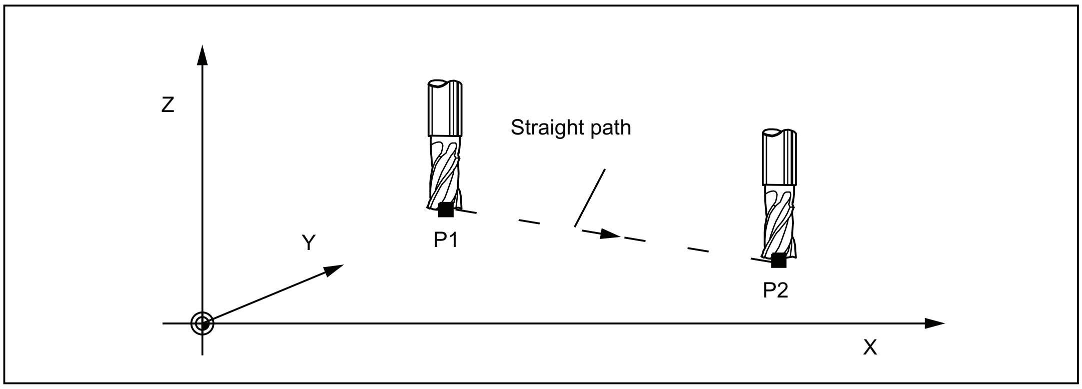
Programming

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

Note

Another option for linear programming is available with the angle specification ANG=... (For more information, see Section "Contour definition programming (Page 66)".).

See the illustration for linear interpolation with rapid traverse from point P1 to P2:



Programming example

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

Information

Another group of G functions exists for movement to the position (see Section "Exact stop / continuous-path control mode: G9, G60, G64 (Page 59)").

For G60 exact stop, a window with various precision values can be selected with another G group. For exact stop, an alternative instruction with non-modal effectiveness exists: G9.

You should consider these options for adaptation to your positioning tasks.

1.3.2 Feedrate F

Functionality

The feed F is the path velocity and represents the value of the geometric sum of the velocity components of all axes involved. The individual axis velocities therefore result from the portion of the axis path in the overall distance to be traversed.

The feedrate F is effective for the interpolation types G1, G2, G3, CIP, and CT and is retained until a new F word is written.

Programming

F...

Note

For integer values, the decimal point is not required, e.g. F300.

Unit of measure for F with G94, G95

The dimension unit for the F word is determined by G functions:

- G94: F as the feedrate in mm/min
- G95: Feedrate F in mm/spindle revolutions

  (only meaningful when the spindle is running)

  Note

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

Programming example

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

Note

Write a new F word if you change G94 - G95.

1.3.3 Linear interpolation with feedrate: G1

Functionality

The tool moves from the starting point to the end point along a straight path. The **path velocity** is determined by the programmed **F word**.

All axes can be traversed simultaneously.

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

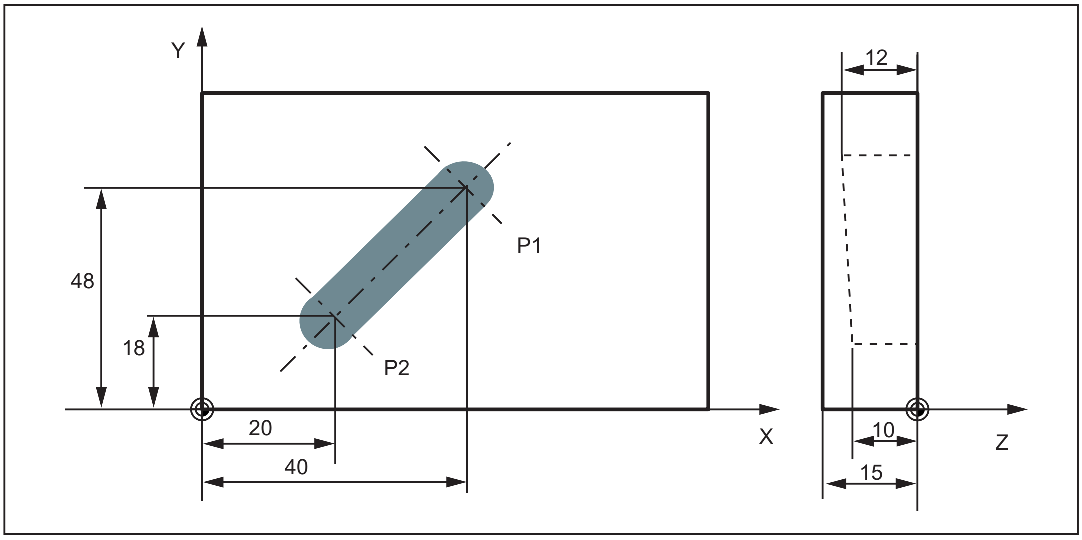
Programming

```
G1 X... Y... Z... F...                ; Cartesian coordinates
G1 AP=... RP=... F...                 ; Polar coordinates
G1 AP=... RP=... Z... F...            ; cylindrical coordinates (3dimensional)
```

Note

Another option for linear programming is available with the angle specification ANG=... (see Section "Contour definition programming (Page 66)").

See the illustration for linear interpolation in three axes using the example of a slot:



Programming example

`N05 G0 G90 X40 Y48 Z2 S500 M3`	; The tool traverses in rapid traverse on P1, three axes concurrently, spindle speed = 500 rpm, clockwise
`N10 G1 Z-12 F100`	; Infeed on Z-12, feed 100 mm/min
`N15 X20 Y18 Z-10`	; Tool travels on a straight line in space on P2
`N20 G0 Z100`	; Retraction in rapid traverse
`N25 X-20 Y80`	
`N30 M2`	; End of program

To machine a workpiece, spindle speed S ... and direction M3/M4 are required (see Section "Spindle movements (Page 63)").

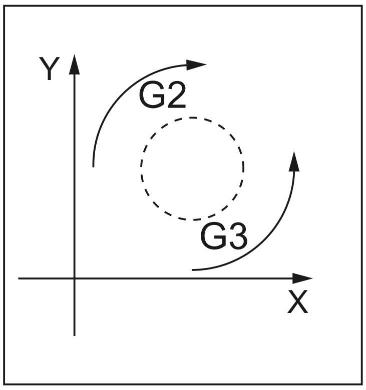
1.3.4 Circular interpolation: G2, G3

Functionality

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

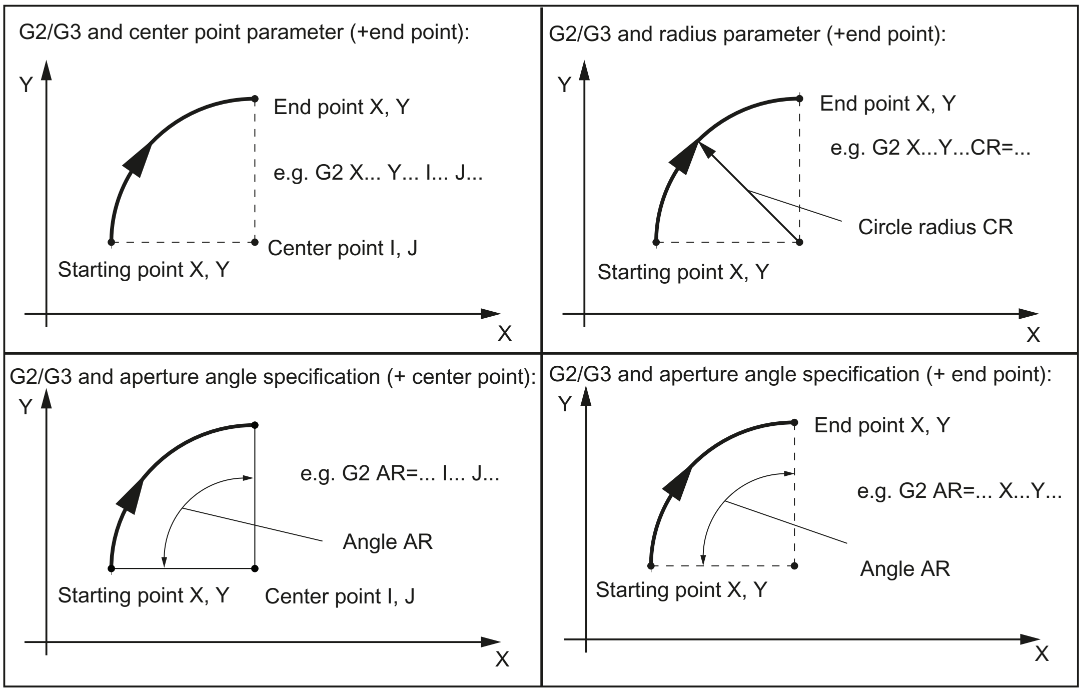
G2: clockwise

G3: counter-clockwise



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

See the following illustration for possibilities of circle programming with G2/G3 using the example of the axes X/Y and G2:



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

The **path velocity** is determined by the programmed **F word.**

Programming

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

Note

Further possibilities for circle programming result from:

CT - circle with tangential connection and

CIP - circle via intermediate point (see next sections).

Input tolerances for the circle

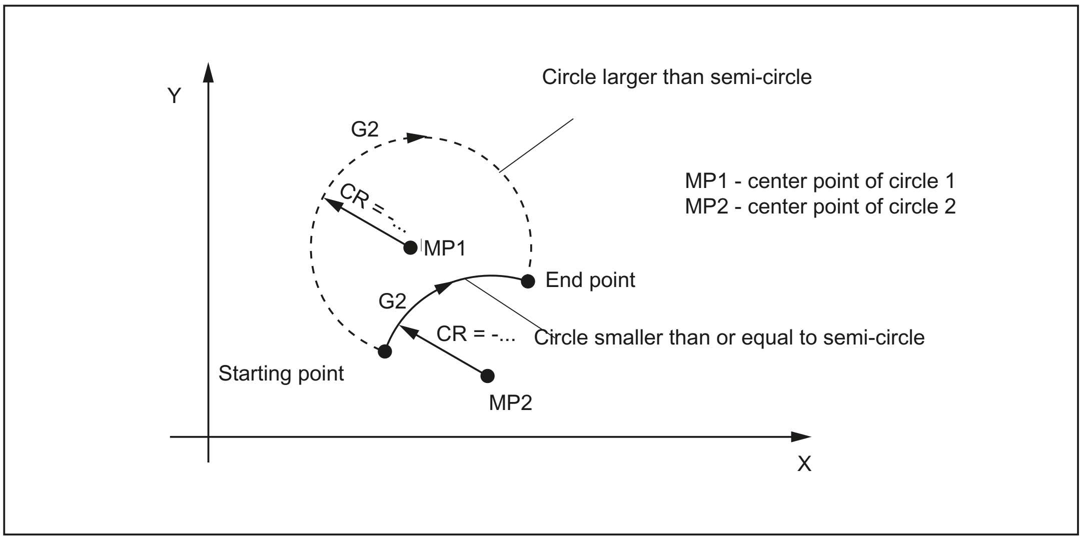
Circles are only accepted by the control system with a certain dimensional tolerance. The circle radius at the starting and end points are compared here. If the difference is within the tolerance, the center point is exactly set internally. Otherwise, an alarm message is issued.

Information

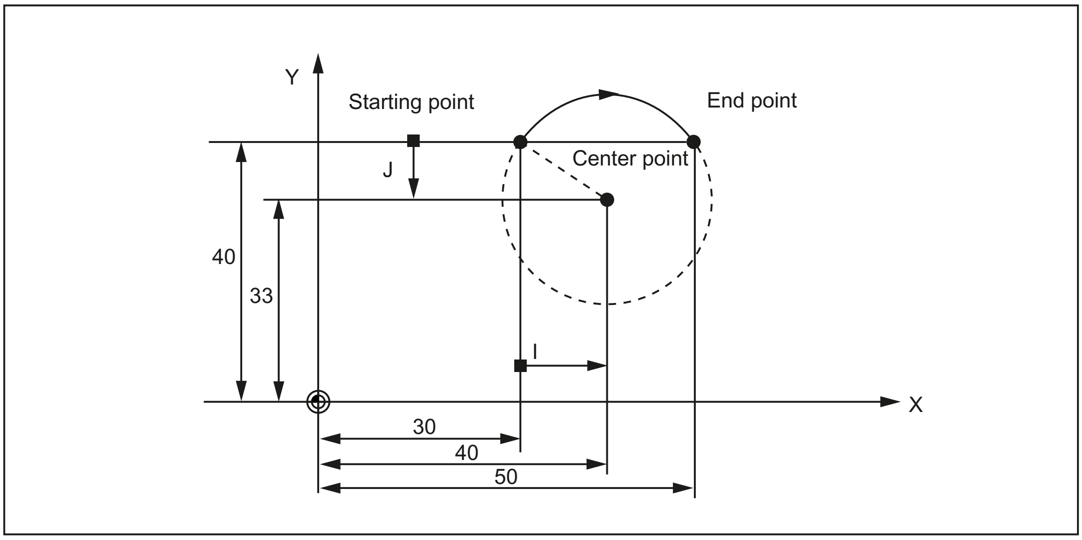
Full circles in a block are only possible if the center point and the end point are specified.

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

See the following illustration for selection of the circle from two possible circles with radius specification:



Programming example: Definition of center point and end point

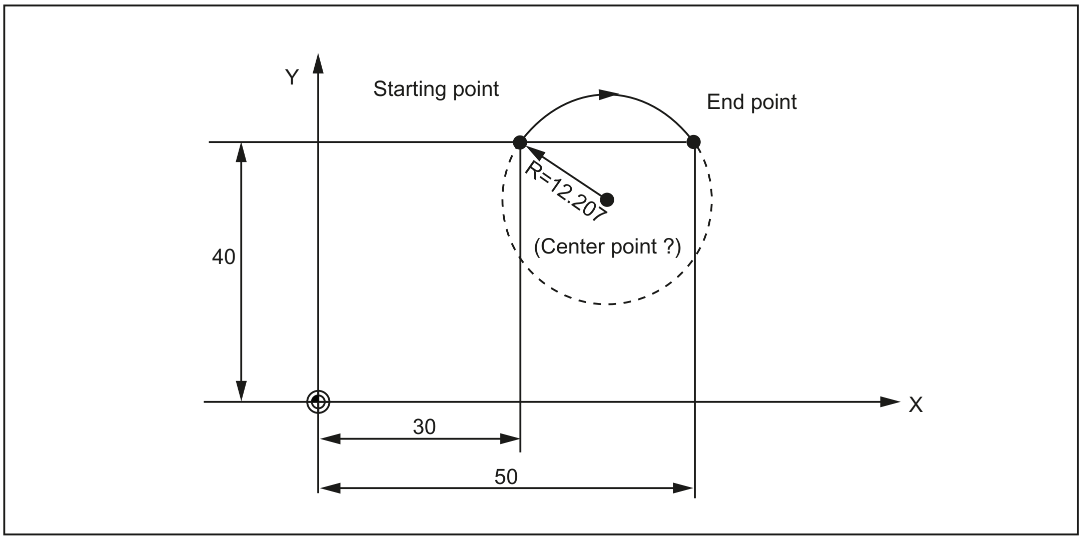



```
N5 G90 X30 Y40                          ; Starting point circle for N10
N10 G2 X50 Y40 I10 J-7                  ; End point and center point
```

Note

Center point values refer to the circle starting point!

Programming example: End point and radius specification

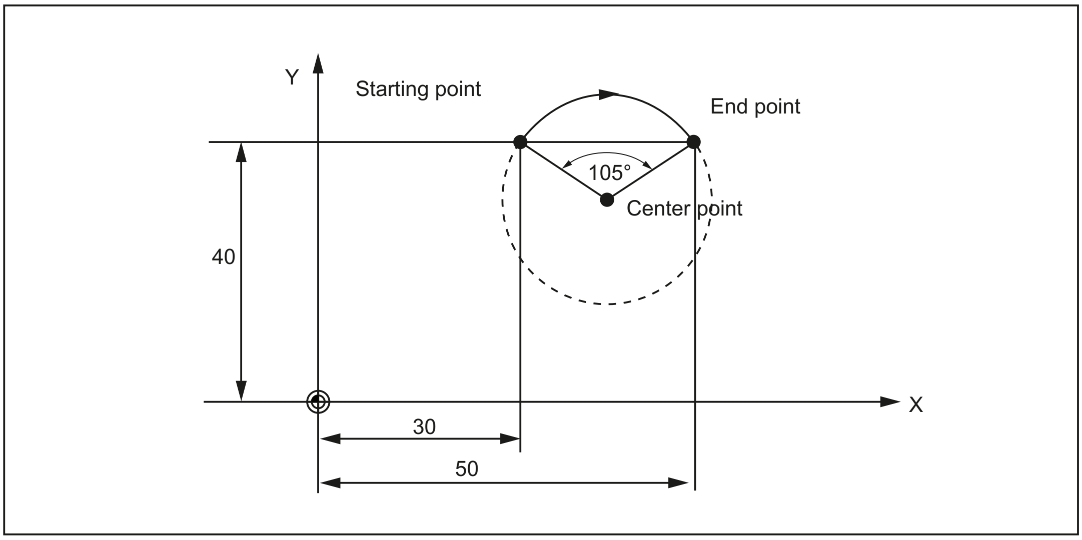



```
N5 G90 X30 Y40                          ; Starting point circle for N10
N10 G2 X50 Y40 CR=12.207                ; End point and radius
```

Note

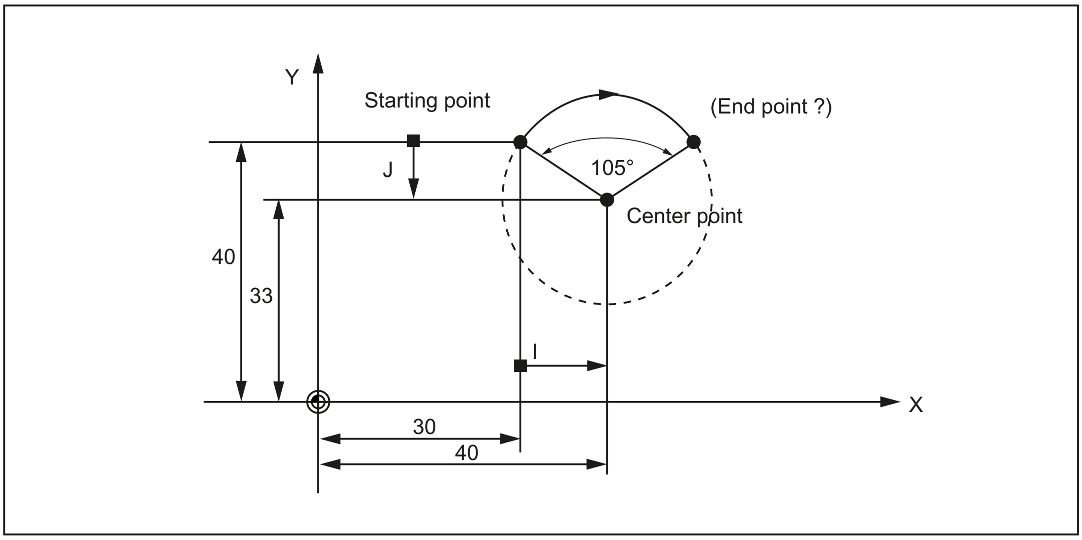
With a negative leading sign for the value with CR=-..., a circular segment larger than a semi-circle is selected.

Programming example: Definition of end point and aperture angle




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

Programming example: Definition of center point and aperture angle

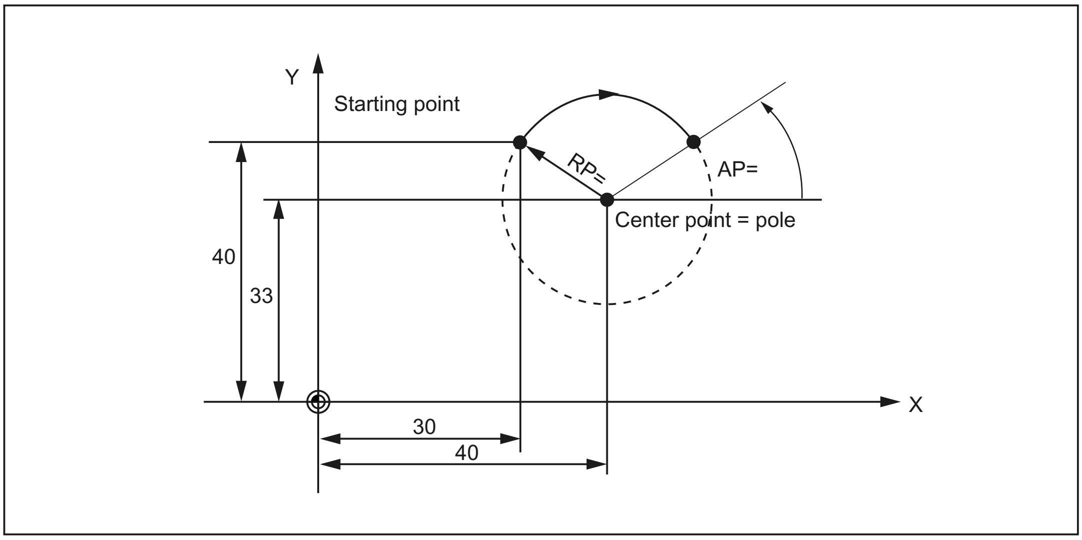



```
N5 G90 X30 Y40                      ; Starting point circle for N10
N10 G2 I10 J-7 AR=105               ; Center point and aperture angle
```

Note

Center point values refer to the circle starting point!

Programming example: Polar coordinates




```
N1 G17                          ; X/Y plane
N5 G90 G0 X30 Y40               ; Starting point circle for N10
N10 G111 X40 Y33                ; Pole = circle center
N20 G2 RP=12.207 AP=21          ; Polar specifications
```

1.3.5 Circular interpolation via intermediate point: CIP

Functionality

If you know **three contour points** of the circle, instead of center point or radius or aperture angle, then it is advantageous to use the CIP function.

The direction of the circle results here from the position of the intermediate point (between starting and end points). The intermediate point is written according to the following axis assignment:

I1=... for the X axis,

J1=... for the Y axis,

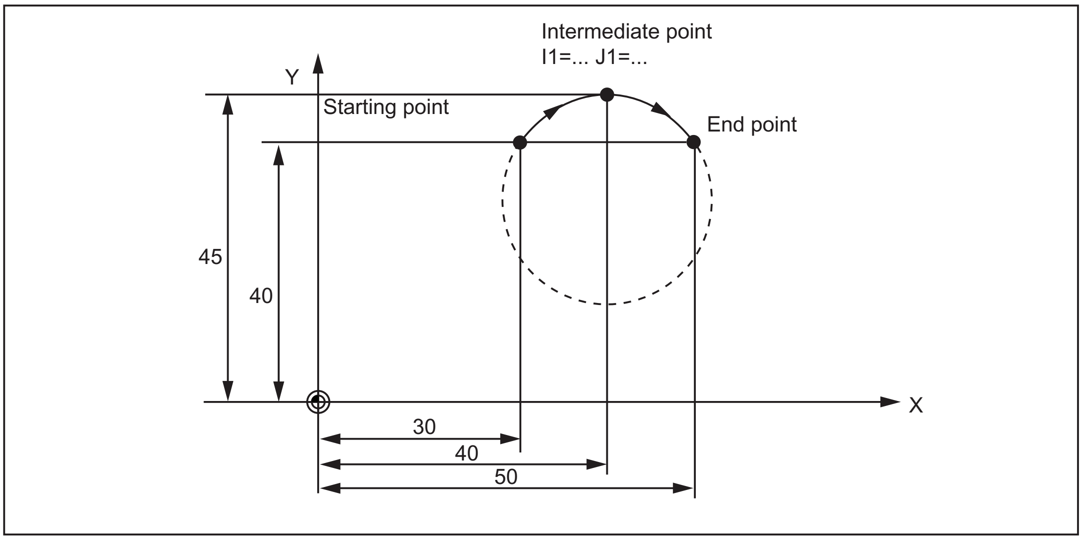
K1=... for the Z axis.

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

Note

The configured dimensional data G90 or G91 applies to the end point **and** the intermediate point.

See the following illustration for circle with end point and intermediate point specification using the example of G90:



Programming example

```
N5 G90 X30 Y40                          ;Starting point circle for N10
N10 CIP X50 Y40 I1=40 J1=45             ; End point and intermediate point
```

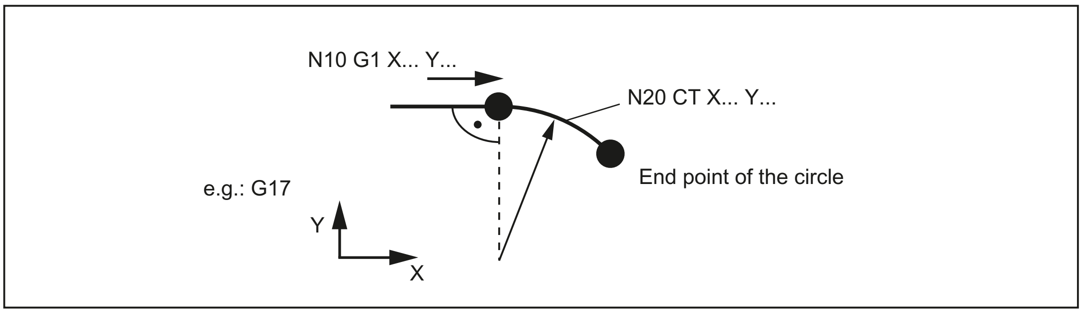
1.3.6 Circle with tangential transition: CT

Functionality

With CT and the programmed end point in the current plane G17 through G19, a circle is generated which is connected tangentially to the previous path segment (circle or straight line) in this plane.

This defines the radius and center point of the circle from the geometric relationships of the previous path section and the programmed circle end point.

See the following illustration for circle with tangential transition to the previous path section:



Programming example

```
N10 G1 X20 F300                              ; Straight line
N20 CT X... Y...                             ; Circle with tangential connection
```

1.3.7 Helix interpolation: G2/G3, TURN

Functionality

With helix interpolation, two movements are overlaid:

- Circular movement in the G17, G18 or G19 plane
- Linear movement of the axis standing vertically on this plane.

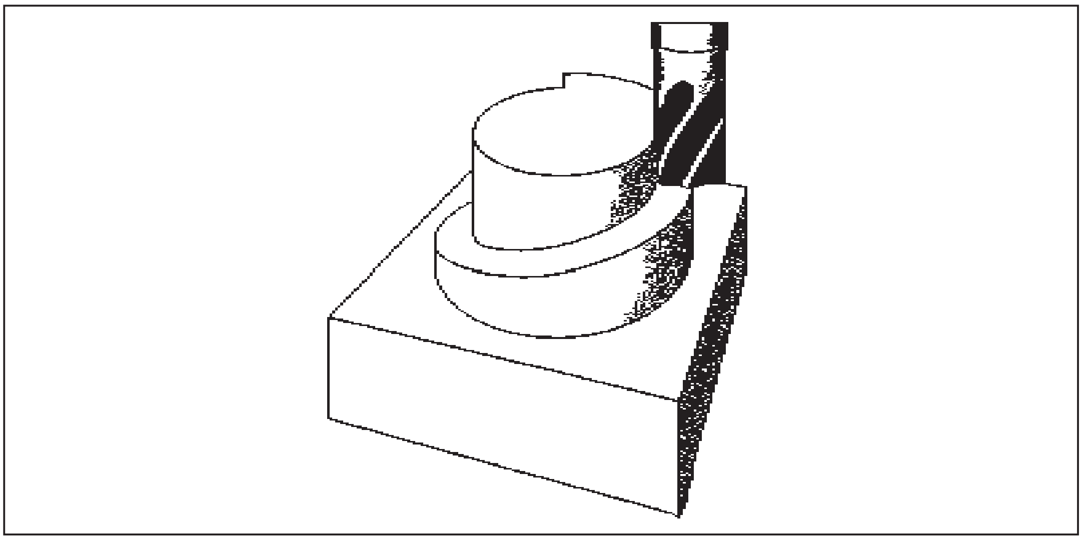
  The number of additional full-circle passes is programmed with TURN=. These are added to the actual circle programming.

  The helix interpolation can preferably be used for the milling of threads or of lubricating grooves in cylinders.

Programming

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

See the following illustration for helical interpolation:



Programming example

```
N10 G17                              ; X/Y plane, Z standing vertically on it
N20 G0 Z50
N30 G1 X0 Y50 F300                   ; Approach starting point
N40 G3 X0 Y0 Z33 I0 J-25 TURN= 3     ; Helix
M30
```

1.3.8 Thread cutting with constant lead: G33

Functionality

This requires a spindle with position measuring system.

The function G33 can be used to machine threads with constant lead of the following type: If an appropriate tool is used, tapping with compensating chuck is possible.

The compensating chuck compensates the resulting path differences to a certain limited degree.

The drilling depth is specified by specifying one of the axes X, Y or Z; the thread pitch is specified via the relevant I, J or K.

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

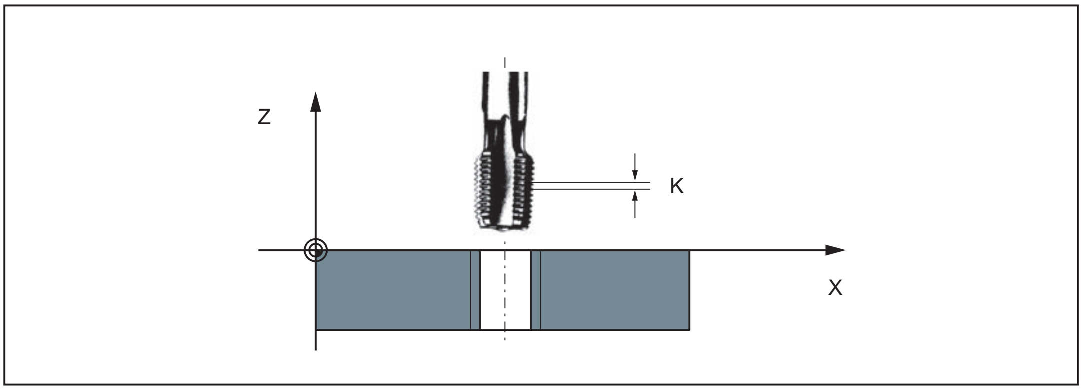
Right-hand or left-hand thread

Right-hand or left-hand thread is set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) - see Section "Spindle movements (Page 63)"). To do this, the rotation value must be programmed under address S or a rotation speed must be set.

Note

A complete cycle of tapping with compensating chuck is provided by the standard cycle CYCLE840.

See the following illustration for tapping using G33:



Programming example

```
                                      ; metric thread 5,
                                      ; pitch as per table: 0.8 mm/rev., hole
                                      already premachined
N10 G54 G0 G90 X10 Y10 Z5 S600 M3     ; Approach starting point, clockwise
                                      spindle rotation
N20 G33 Z-25 K0.8                     ; Tapping, end point -25 mm
N40 Z5 K0.8 M4                        ; Retraction, counter-clockwise spindle
                                      rotation
N50 G0 X30 Y30 Z20
N60 M30
```

Axis velocity

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

Note

Override switch

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

1.3.9 Tapping with compensating chuck: G63

Functionality

G63 can be used for tapping with compensating chuck. The programmed feedrate F must match with the spindle speed S (programmed under the address "S" or specified speed) and with the thread pitch of the drill:

F [mm/min] = S [rpm] x thread pitch [mm/rev.]

The compensating chuck compensates the resulting path differences to a certain limited degree.

The drill is retracted using G63, too, but with the spindle rotating in the opposite direction M3 <-> M4.

G63 is non-modal. In the block after G63, the previous G command of the "Interpolation type" group (G0, G1,G2, ...) is active again.

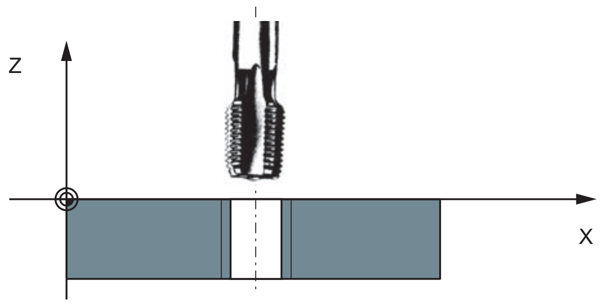
Right-hand or left-hand thread

Right-hand or left-hand thread is set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) - see Section "Spindle movements (Page 63)").

Note

The standard cycle CYCLE840 provides a complete tapping cycle with compensating chuck (but with G33 and the relevant prerequisites).

See the following illustration for tapping using G63:



Programming example

```
                                        ; metric thread 5,
                                        ; lead as per table: 0.8 mm/rev., hole
                                        already premachined
N10 G54 G0 G90 X10 Y10 Z5 S600 M3       ; Approach starting point, clockwise
                                        spindle rotation
N20 G63 Z-25 F480                       ; Tapping, end point -25 mm
N40 G63 Z5 M4                           ; Retraction, counter-clockwise spindle
                                        rotation
N50 X30 Y30 Z20
M30
```

1.3.10 Thread interpolation: G331, G332

Functionality

This requires a position-controlled spindle with a position measuring system.

By using G331/G332, the threads can be tapped **without** compensating chuck if the dynamic properties of the spindle and axis allow it.

If, however, a compensating chuck is used, the path differences to be compensated by the compensating chuck are reduced. This allows tapping at higher spindle speeds.

Drilling is done using G331, retraction is done using G332.

The drilling depth is specified by specifying one of the axes X, Y or Z; the thread pitch is specified via the relevant I, J or K.

For G332, the same lead is programmed as for G331. Reversal of the spindle direction of rotation occurs automatically.

The spindle speed is programmed with S and without M3/M4.

Before tapping the thread using G331/G332, the spindle must be switched to the position-controlled mode with SPOS=....

Right-hand or left-hand thread

The **sign of the** thread lead determines the direction of spindle rotation:

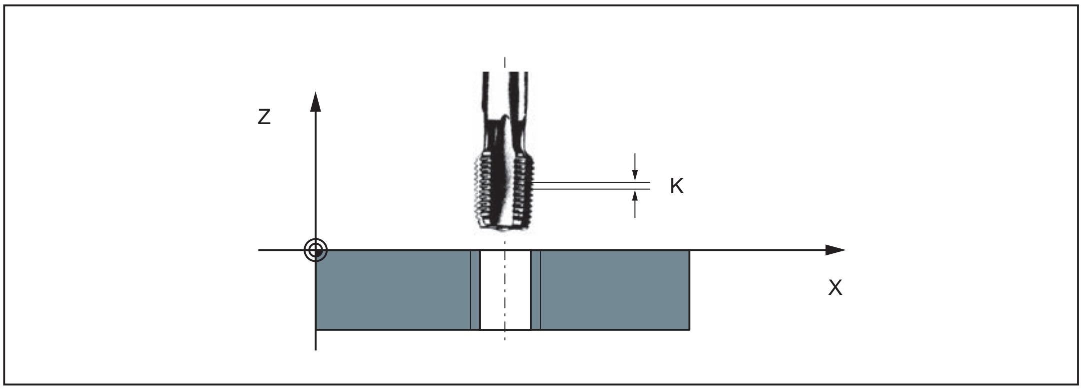
Positive: right-hand (as with M3)

Negative: left-hand (as with M4)

Note

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

See the following illustration for tapping using G331/G332:



Axis velocity

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

Programming example

```
metric thread 5,
lead as per table: 0.8 mm/rev., hole
already premachined:
N5 G54 G0 G90 X10 Y10 Z5               ; Approach starting point
N10 SPOS=0                             ; Spindle in position control
N20 G331 Z-25 K0.8 S600                ; Tapping, K positive = clockwise
                                        of the spindle, end point Z=-25 mm
N40 G332 Z5 K0.8                       ; Retraction
N50 G0 X30 Y30 Z20
N60 M30
```

1.3.11 Fixed point approach: G75

Functionality

By using G75, a fixed point on the machine, e.g. tool change point, can be approached. The position is stored permanently in the machine data for all axes. A maximum of four fixed points can be defined for each axis.

No offset is effective. The speed of each axis is its rapid traverse.

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

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

Programming

G75 FP=<n> X1=0 Y1=0 Z1=0

Note

FPn references with axis machine date MD30600 $MA_FIX_POINT_POS[n-1]. If no FP has been programmed, then the first fixed point will be selected.

Table 1- 1 Explanation

Command	Significance
G75	Fixed point approach
FP=<n>	Fixed point that is to be approached. The fixed point number is specified: <n> Value range of <n>: 1, 2, 3, 4 If no fixed point is specified, fixed point 1 is approached automatically.
X1=0 Y1=0 Z1=0	Machine axes to be traversed to the fixed point. Here, specify the axes with value "0" with which the fixed point is to be approached simultaneously. Each axis is traversed with the maximum axial velocity.

Programming example

```
N05 G75 FP=1 Z1=0                          ; Approach fixed point 1 in Z
N10 G75 FP=2 X1=0 Y1=0                     ; Approach fixed point 2 in X and Y,
                                           e. g. to change a tool
N30 M30                                    ; End of program
```

Note

The programmed position values for X1, Y1, Z1 (any value, here = 0) are ignored, but must still be written.

1.3.12 Reference point approach: G74

Functionality

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

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

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

Programming example

```
N10 G74 X1=0 Y1=0 Z1=0
```

Note

The programmed position values for X1, Y1, Z1 (any value, here = 0) are ignored, but must still be written.

1.3.13 Feedrate override for circles: CFTCP, CFC

Functionality

For activated tool radius compensation (G41/G42) and circle programming, it is imperative to correct the feedrate at the cutter center point if the programmed F value is to act at the circle contour.

Internal and external machining of a circle and the current tool radius are taken into account automatically if the tool radius compensation is enabled.

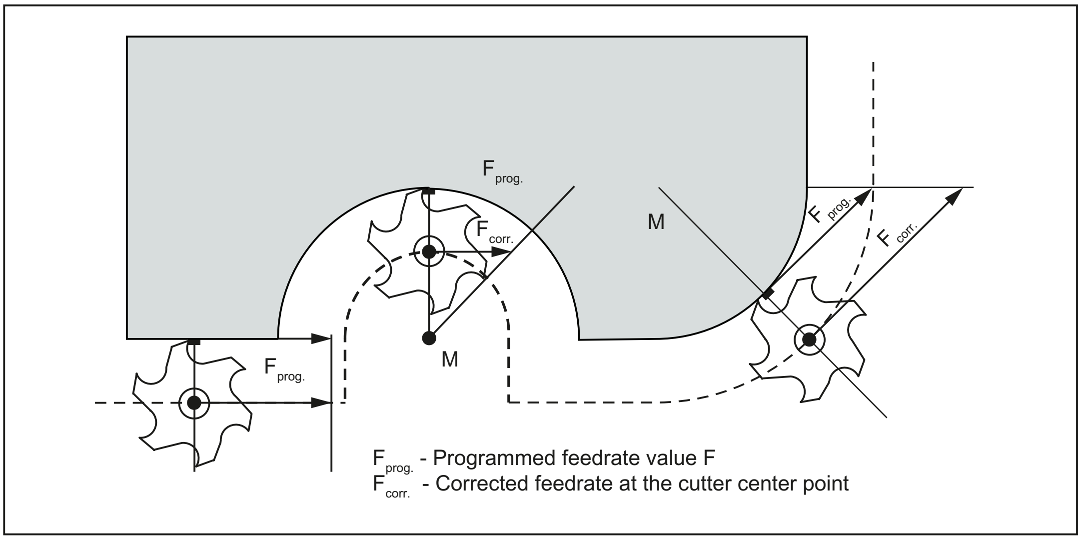
This feedrate correction (override) is not necessary for linear paths. The path velocities at the cutter center point and at the programmed contour are identical.

If you wish the programmed feedrate always to act at the cutter center point path, then disable the feedrate override. The modally acting G group that contains CFTCP/CFC (G functions) is provided for switching.

Programming

CFTCP	; Feedrate override OFF (the programmed feedrate acts at the milling cutter center point)
CFC	; Feedrate override with circle ON

See the following illustration for feedrate override G901 with internal / external machining:



Corrected feedrate

- External machining:

  $F_{corr.} = F_{prog.} (r_{cont} + r_{tool}) / r_{cont}$

- Internal machining:

  $F_{korr.} = F_{prog.} (r_{cont} - r_{tool}) / r_{cont}$

  r_{cont}: Radius of the circle contour

  r_{tool}: Tool radius

Programming example

```
N10 G42 G1 X30 Y40 F1000        ; Tool radius compensation ON
N20 CFC F350                    ; Feedrate override with circle ON
N30 G2 X50 Y40 I10 J-7 F350     ; Feed value acts on contour
N40 G3 X70 Y40 I10 J6 F300      ; Feed value acts on contour
N50 CFTCP                       ; Feedrate override OFF, programmed feedrate value
                                acts at the milling cutter center point
N60 M30
```

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

Functionality

G functions are provided for optimum adaptation to different requirements to set the traversing behavior at the block boundaries and for block advancing. Example: For example, you would like to quickly position with the axes or you would like to machine path contours over multiple blocks.

Programming

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

Exact stop G60, G9

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

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

- G601; Exact stop window fine

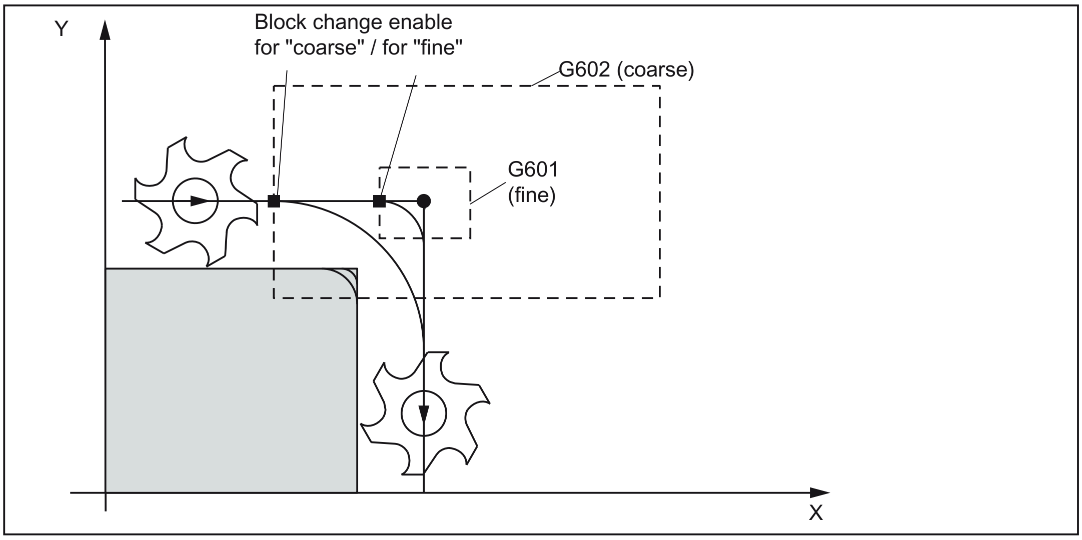
  Block advance takes place when all axes have reached the "Exact stop window fine" (value in the machine data).

- G602: Exact stop window coarse

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

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

See the following illustration for exact stop window coarse or fine, in effect for G60/G9:



Programming example

```
N5 G602                         ; Exact stop window coarse
N10 G0 G60 X20                  ; Exact stop modal
N20 X30 Y30                     ; G60 continues to act
N30 G1 G601 X50 Y50 F100        ; Exact stop window fine
N40 G64 X70 Y60                 ; Switching over to continuous-path mode
N50 G0 X90 Y90
N60 G0 G9 X95                   ; Exact stop acts only in this block
N70 G0 X100 Y100                ; Again continuous-path mode
M30
```

Note

The G9 command only generates exact stop for the block in which it is programmed; G60, however, is effective until it is canceled by G64.

Continuous-path control mode G64

The objective of the continuous-path control mode is to avoid deceleration at the block boundaries and to switch to the next block with a path velocity as constant as possible (in the case of tangential transitions). The function works with look-ahead velocity control over several blocks.

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

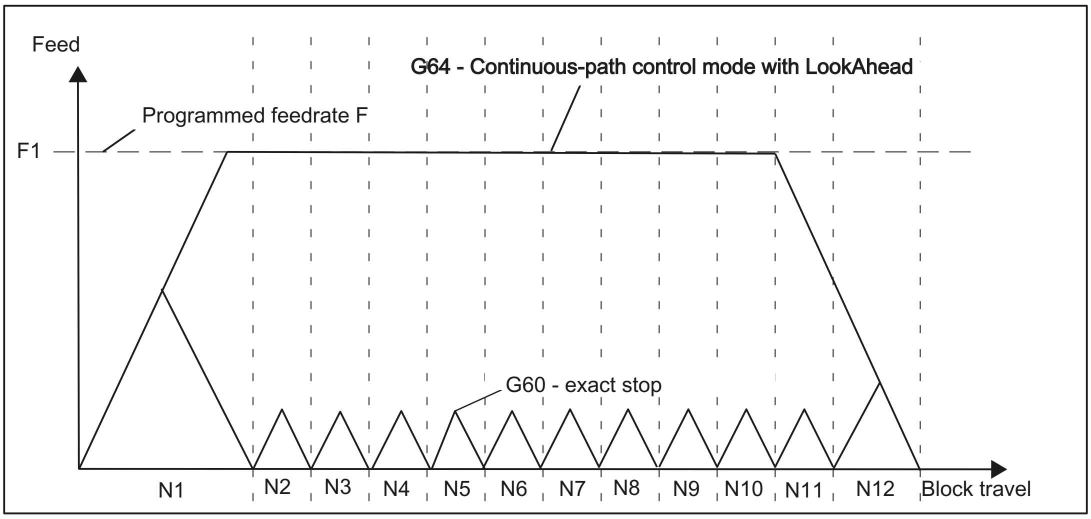
Programming example

```
N10 G64 G1 X10 Y20 F1000      ; Continuous-path mode
N20 X30 Y30                   ; Continuous-path control mode continues to be active
N30 G60 Z50                   ; Switching over to exact stop
M30
```

Look-ahead velocity control:

In the continuous-path control mode with G64, the control system automatically determines the velocity control for several NC blocks in advance. This enables acceleration and deceleration across multiple blocks with approximately tangential transitions. For paths that consist of short travels in the NC blocks, higher velocities can be achieved than without look ahead.

See the following illustration for comparison of the G60 and G64 velocity behavior:



1.3.15 Acceleration pattern: BRISK, SOFT

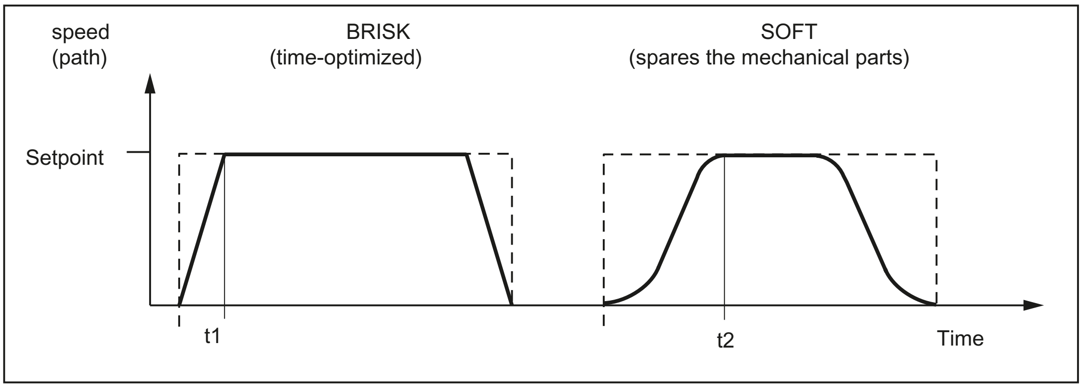
BRISK

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

SOFT

The axes of the machine accelerate along a non-linear, constant characteristic until reaching the final velocity. With this jerk-free acceleration, SOFT allows for reduced machine load. The same behavior can also be applied to braking procedures.

See the following illustration for basic course of the path velocity when using BRISK/SOFT:



Programming

BRISK	; Jerking path acceleration
SOFT	; Jerk-limited path acceleration

Programming example

```
N10 SOFT G1 X30 Z84 F650       ; Jerk-limited path acceleration
N90 BRISK X87 Z104             ; Continuing with jerking path acceleration
```

1.3.16 Dwell time: G4

Functionality

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

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

Programming

G4 F...	; Dwell time in seconds
G4 S...	; Dwell time in spindle revolutions

Programming example

```
N5 G1 F200 Z-50 S300 M3        ; Feed F; spindle speed S
N10 G4 F2.5                    ; Dwell time 2.5 seconds
N20 Z70
N30 G4 S30                     ; Dwelling 30 revolutions of the spindle, corresponds
                               at S=300 rpm and 100% speed override to: t=0.1 min
N40 X60                        ; Feed and spindle speed remain effective
M30
```

Note

G4 S.. is only possible if a controlled spindle is available (if the speed specifications are also programmed via S...).

1.4 Spindle movements

1.4.1 Gear stages

Function

Up to 5 gear stages can be configured for a spindle for speed / torque adaptation. The selection of a gear stage takes place in the program via M commands (see Section "Miscellaneous function M (Page 85)"):

- M40: Automatic gear stage selection
- M41 to M45: Gear stage 1 to 5

1.4.2 Spindle speed S, directions of rotation

Functionality

The spindle speed is programmed in revolutions per minute under the address S provided that the machine possesses a controlled spindle.

The direction of rotation and the start or end of the movement are specified via M commands (also see Section "Miscellaneous function M (Page 85)").

M3: Spindle clockwise

M4: Spindle counter-clockwise

M5: Spindle stop

Note

For integer S values, the decimal point can be omitted, e.g. S270.

Information

If you write M3 or M4 in a **block with axis movements**, the M commands become active **before** the axis movements.

Default setting: The axis movements only start once the spindle has accelerated to speed (M3, M4). M5 is also issued before the axis movement. However, there is no waiting for spindle standstill. The axis movements begin before spindle standstill.

The spindle is stopped at program end or with RESET.

At program start, spindle speed zero (S0) is in effect.

Note

Other settings can be configured via machine data.

Programming example

```
N10 G1 X70 Z20 F300 S270 M3    ; Before the axis traversing X, Z the spindle
                               accelerates to 270 rpm, clockwise
N80 S450                       ; Speed change
N170 G0 Z180 M5                ; Z movement, spindle comes to a stop
```

1.4.3 Spindle positioning: SPOS

Functionality

Requirement: The spindle must be technically designed for position control.

With the function SPOS= you can position the spindle in a specific **angular position**. The spindle is held in the position through position control.

The **speed** of the positioning procedure is defined in machine data.

With SPOS=*value* from the M3/M4 movement, the respective **direction of rotation** is maintained until the end of the positioning. When positioning from standstill, the position is approached via the shortest path. The direction results from the respective start and end position.

Exception: First movement of the spindle, i.e. if the measuring system is not yet synchronized. In this case, the direction is specified in machine data.

Other movement specifications for the spindle are possible with SPOS=ACP(...), SPOS=ACN(...), ... as for rotary axes.

The spindle movement takes place parallel to any other axis movements in the same block. This block is ended when both movements are finished.

Programming

SPOS=...	; Absolute position: 0 ... <360 degrees
SPOS=ACP(...)	; Absolute dimensions, approach position in positive direction
SPOS=ACN(...)	; Absolute dimensions, approach position in negative direction
SPOS=IC(...)	; Incremental dimensions, leading sign determines the traversal direction
SPOS=DC(...)	; Absolute dimensions, approach position directly (on the shortest path)

Programming example

```
N10 SPOS=14.3                  ; Spindle position 14.3 degrees
N80 G0 X89 Z300 SPOS=25.6      ; Positioning spindle with axis movements
                               ; This block is ended when all movements have
                               finished
N81 X200 Z300                  ; The N81 block only begins once the spindle position
                               from N80 is reached
```

1.5 Contour programming support

1.5.1 Contour definition programming

Functionality

If the end points for the contour are not directly specified in the machining drawing, it is also possible to use an angle specification ANG=... to determine the straight line. In a contour corner, you can insert the elements chamfer or rounding. The respective instruction CHR= ... or RND=... is written in the block, which leads to the corner.

The blueprint programming can be used in blocks with **G0 or G1** (linear contours).

Theoretically, any number of straight line blocks can be connected and a rounding or a chamfer can be inserted between them. Every straight line must be clearly identified by point values and/or angle values.

Programming

ANG=...	; Angle specification for defining a straight line
RND=...	; Insert rounding, value: Radius of chamfer
CHR=...	; Insert chamfer, value: Side length of the chamfer

Information

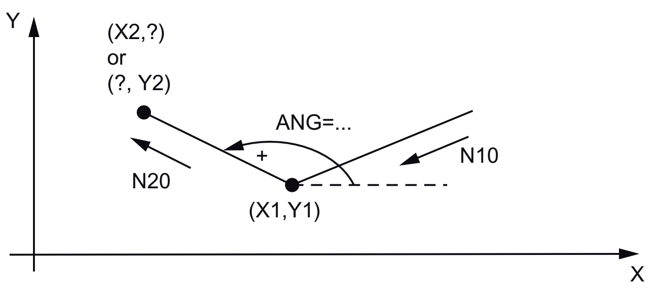
The blueprint programming function is executed in the current plane G17 to G19. It is not possible to change the plane during blueprint programming.

If radius and chamfer are programmed in one block, only the radius is inserted regardless of the programming sequence.

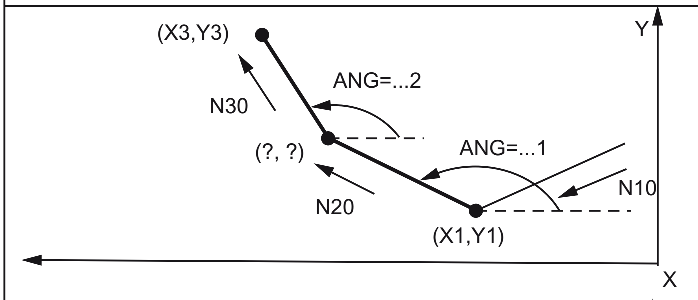
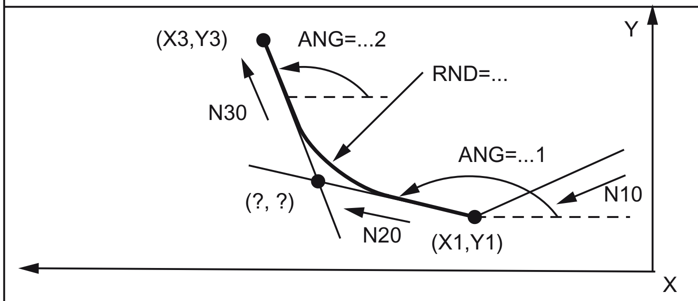
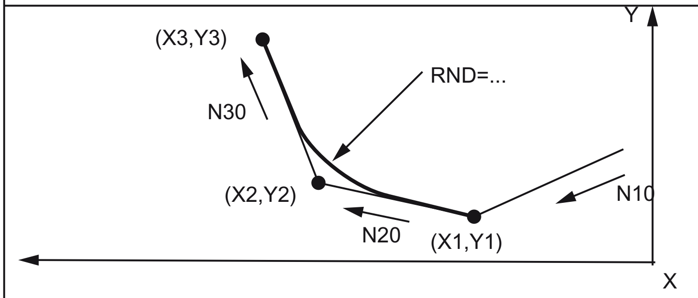
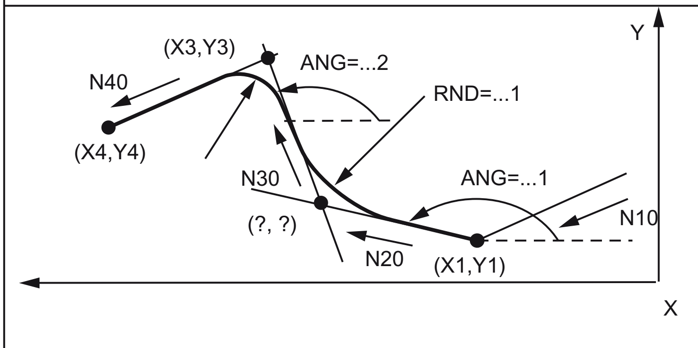
Angle ANG

If only one end point coordinate of the plane is known for a straight line or for contours across multiple blocks the cumulative end point, an angle parameter can be used for uniquely defining the straight line path. The angle is always referred to the abscissa of the current plane G17 to G19, e.g. for G17 on the X axis. Positive angles are aligned counter-clockwise.

See the following specification of an angle for determination of a straight line using the example of the G17 plane:

Contour	Programming
	End point in N20 not always known N10 G1 X1 Y1 N20 X2 ANG=... or: N10 G1 X1 Y1 N20 Y2 ANG=... The values are only examples.

See the following illustration for multiple block contours using the example of the G17 plane:

Contour	Programming
	End point in N20 unknown N10 G1 X1 Y1 N20 ANG=30 N30 X3 Y3 ANG=60 N40 M30
	End point in N20 unknown, insert rounding: N10 G1 X1 Y1 N20 ANG=30 RND=0.1 N30 X3 Y3 ANG=60 analog Inserting a chamfer: N10 G1 X1 Y1 N20 ANG=30 CHR=0.1 N30 X3 Y3 ANG=60
	End point in N20 known Inserting a rounding: N10 G1 X1 Y1 N20 X2 Y2 RND=0.5 N30 X3 Y3 analog Inserting a chamfer: N10 G1 X1 Y1 N20 X2 Y2 CHR=0.2 N30 X3 Y3
	End point in N20 unknown Inserting a rounding: N10 G1 X1 Y1 N20 ANG=30 RND=0.3 N30 X3 Y3 ANG=60 RND=0.3 N40 X4 Y4 analog Inserting a chamfer: N10 G1 X1 Y1 N20 ANG=30 CHR=0.3 N30 X3 Y3 ANG=60 CHR=0.3 N40 X2 Y4 N50 M30

1.5.2 Rounding, chamfer

Functionality

You can insert the chamfer (CHF or CHR) or rounding (RND) elements into a contour corner. If you wish to round several contour corners sequentially by the same method, use "Modal rounding" (RNDM).

You can program the feedrate for the chamfer/rounding with FRC (non-modal) or FRCM (modal). If FRC/FRCM is not programmed, the normal feedrate F is applied.

Programming

CHF=...	; Insert chamfer, value: Length of chamfer
CHR=...	; Insert chamfer, value: Side length of the chamfer
RND=...	; Insert rounding, value: Radius of chamfer
RNDM=...	; Modal rounding:
	Value >0: Radius of chamfer, modal rounding ON
	This rounding is inserted in all contour corners.
	Value = 0: Modal rounding OFF...
FRC=...	; Non-modal feedrate for chamfer/rounding
	Value >0, feedrate in mm/min (G94) or mm/rev. (G95)
FRCM=...	; Modal feedrate for chamfer/rounding:
	Value >0: Feedrate in mm/min (G94) or mm/rev. (G95),
	Modal feedrate for chamfer/rounding ON
	Value = 0: Modal feedrate for chamfer/rounding OFF
	Feedrate F applies to the chamfer/rounding.

Information

The chamfer/rounding functions are executed in the current planes G17 to G19.

The appropriate instruction CHF= ... or CHR=... or RND=... or RNDM=... is written in the block with axis movements leading to the corner.

The programmed value for chamfer and rounding is automatically reduced if the contour length of an involved block is insufficient.

No chamfer/rounding is inserted, if

- more than three blocks in the connection are programmed that do not contain any information for traversing in the plane,
- or a plane change is carried out.

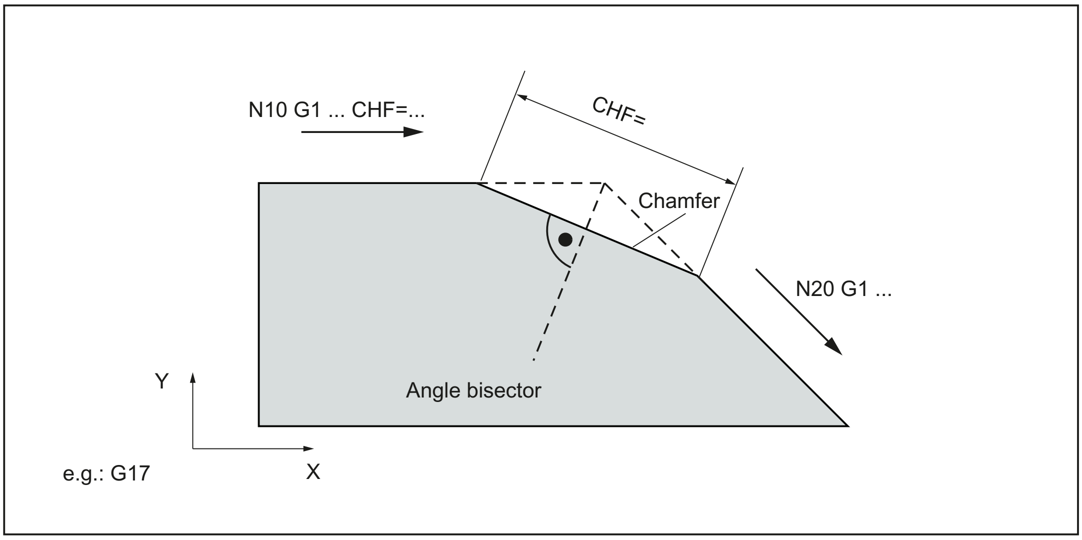
F, FRC,FRCM are not active when a chamfer is traversed with G0.

If the feedrate F is active for chamfer/rounding, it is by default the value from the block which leads away from the corner. Other settings can be configured via machine data.

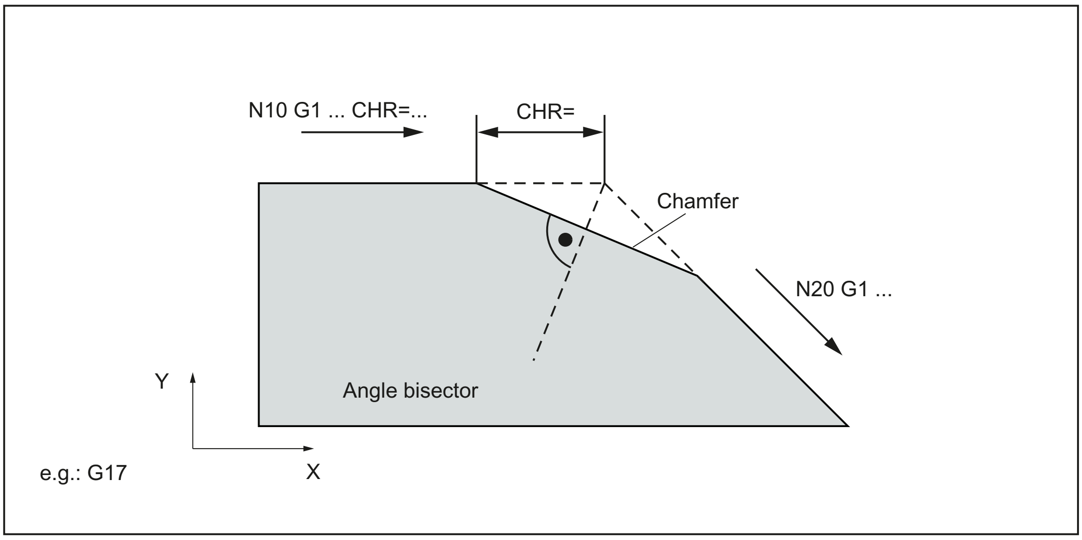
Chamfer CHF or CHR

A linear contour element is inserted between **linear and circle contours** in any combination. The edge is broken.

See the following illustration for inserting a chamfer with CHF using the example: Between two straight lines.



See the following illustration for inserting a chamfer with CHR using the example: Between two straight lines.



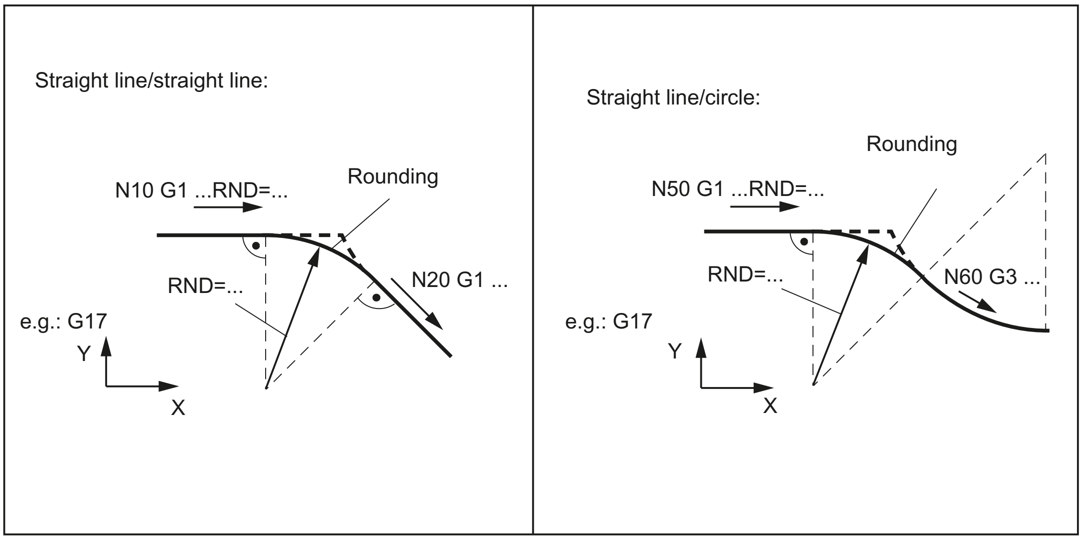
Programming examples of chamfer

```
N5 G17 G94 F300 G0 X100 Y100
N10 G1 X85 CHF=5                    ; Insert chamfer with chamfer length of 5 mm
N20 X70 Y70
N30 G0 X60 Y60
N100 G1 X50 CHR=7                   ; Insert chamfer with leg length of 7 mm
N110 X40 Y40
N200 G1 FRC=200 X30 CHR=4           ; Insert chamfer with feedrate FRC
N210 X20 Y20
M30
```

Rounding RND or RNDM

A circle contour element can be inserted with tangential connection between the **linear and circle contours** in any combination.

See the following examples for inserting roundings:



Programming examples for rounding

```
N10 G17 G94 F300 G0 X100 Y100
N20 G1 X85 RND=8                    ; Insert 1 rounding with radius 8 mm, feedrate
                                    F
N30 X70 Y70
N40 G0 X60 Y60
N50 G1 X50 FRCM= 200 RNDM=7.3       ; Modal rounding, radius 7.3 mm with special
                                    feedrate FRCM (modal)
N60 G3 X40 Y40 CR=20                ; continue inserting this rounding - to N70
N70 G1 X30 Y30 RNDM=0               ; Modal rounding OFF
N80 X20 Y20
N90 M30
```

1.6 Tool and tool offset

1.6.1 General Information

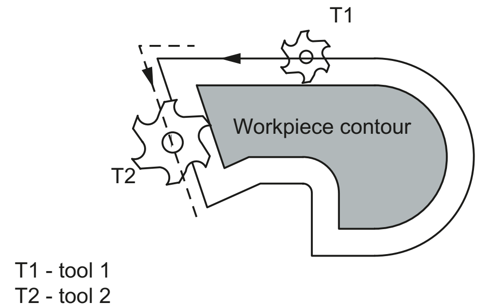
Functionality

When creating programs for machining workpieces, it is not necessary to take into account the tool length or the tool radius. You program the workpiece dimensions directly, for example following the drawing.

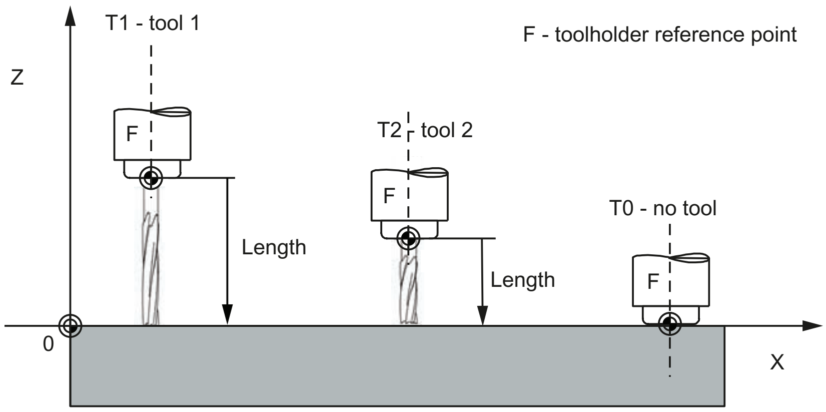
You enter the tool data separately in a special data section.

Simply call the required tool with its offset data in the program and enable the tool radius compensation if necessary. The control system performs the required path compensations based on this data to create the described workpiece.

See the following illustration for machining of a workpiece with different tool radius:



See the following illustration for approaching the workpiece position Z0 - different length compensations:



1.6.2 Tool T

Functionality

The tool selection takes place when the T word is programmed. Whether this is a **tool change** or only a **preselection**, is defined in the machine data:

- The tool change (tool call) is performed either directly using the T word or
- The change takes place after the preselection with the T word by an additional instruction **M6** (see also Section "Miscellaneous function M (Page 85)").

Note

If a certain tool was activated, it remains stored as an active tool even beyond the end of the program and after turning off / turning on the control system.

If you change a tool manually, input the change also in the control system so that the control system 'knows' the correct tool. For example, you can start a block with the new T word in MDA mode.

Programming

T... ; Tool number: 1 ... 32 000, T0 - no tool

A maximum of 64 tools can be stored in the control system.

Programming example

```
; Tool change without M6:
N10 T1                                ; Tool 1
N70 T588                              ; Tool 588
; Tool change with M6:
N10 T14                               ; Preselect tool 14
N15 M6                                ; Perform tool change; thereafter, T14 is
                                      active
```

1.6.3 Tool compensation number D

Functionality

It is possible to assign 1 to 9 data fields with different tool offset blocks (for multiple cutting edges) to a specific tool. If a special cutting tool is required, it can be programmed with D and the corresponding number.

If no D word is written, D1 takes effect automatically.

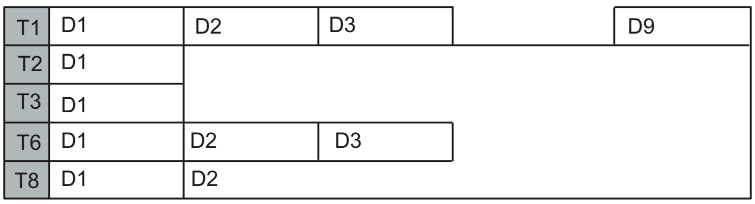
When D0 is programmed, offsets for the tool have no effect.

Programming

D...	; Tool offset number: 1 ... 9, D0: No compensations active!

A maximum of 64 data fields (D numbers) for tool offset blocks can be stored simultaneously in the control system:

See the following illustration for examples for assigning tool compensation numbers / tool:



Each tool has separate compensation blocks - max. 9.

Information

The tool length compensations are effective immediately once the tool is active - if no D number has been programmed - with the values of D1.

The offset is applied with the first programmed traverse of the respective length offset axis. Observe any active G17 to G19.

A tool radius compensation must also be activated by G41/G42.

Programming example

Tool change **without M6 command** (only with T):

```
N5 G17               ; Determines the length offset axis (here Z axis)
N10 T1               ; Tool 1 is activated with the associated D1
N11 G0 Z...          ; For G17, Z is length offset axis, the length offset
                     compensation is overlaid here
N50 T4 D2            ; Load tool 4, D2 from T4 is active
...
N70 G0 Z... D1       ; D1 for tool 4 active, only cutting edge changed
```

Tool change using the M6 command:

```
N5 G17               ; Determines the length offset axis (here Z axis)
N10 T1               ; Tool preselection
...
N15 M6               ; Tool change, T1 is active with the appropriate D1
N16 G0 Z...          ; For G17, Z is length offset axis, the length offset
                     compensation is overlaid here
...
N20 G0 Z... D2       ; D2 for tool 1 is active; for G17, Z is length offset axis,
                     the difference of the D1->D2 length offset is overlaid here
N50 T4               ; T4 tool preselection; note: T1 with D2 is still active !
...
N55 D3 M6            ; Tool change, T4 is active with the appropriate D3
...
```

Contents of a compensation memory

Enter the following in the offset memory:

- Geometrical dimensions: length, radius.

  They consist of several components (geometry, wear). The control computes the components to a certain dimension (e.g. overall length 1, total radius). The respective overall dimension becomes effective when the compensation memory is activated.

  How these values are calculated in the axes is determined by the tool type and the commands G17, G18, G19 (see following illustrations).

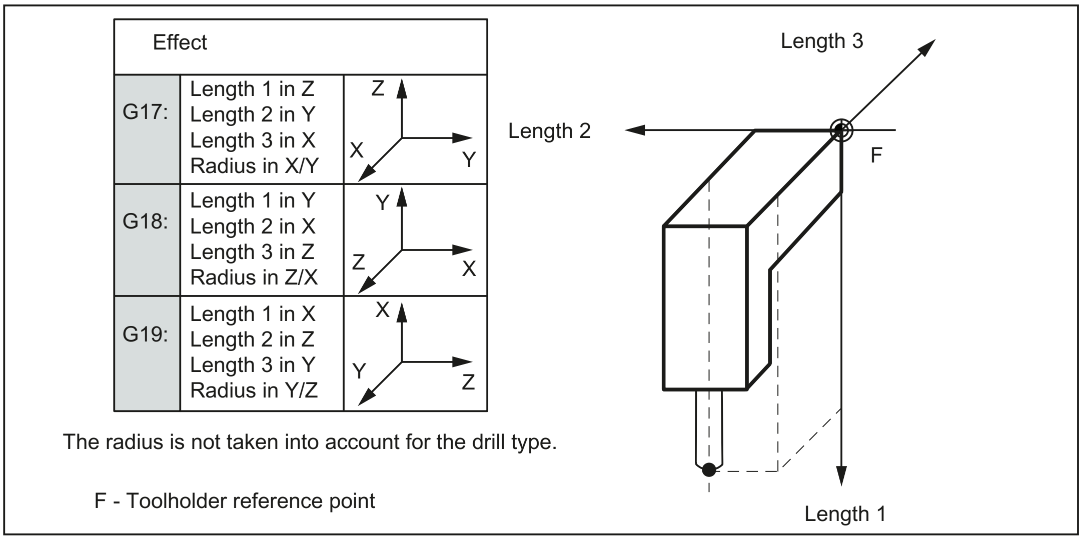
- Tool type

  The tool type (drill, cutter) defines which geometry data are necessary and how they are taken into account.

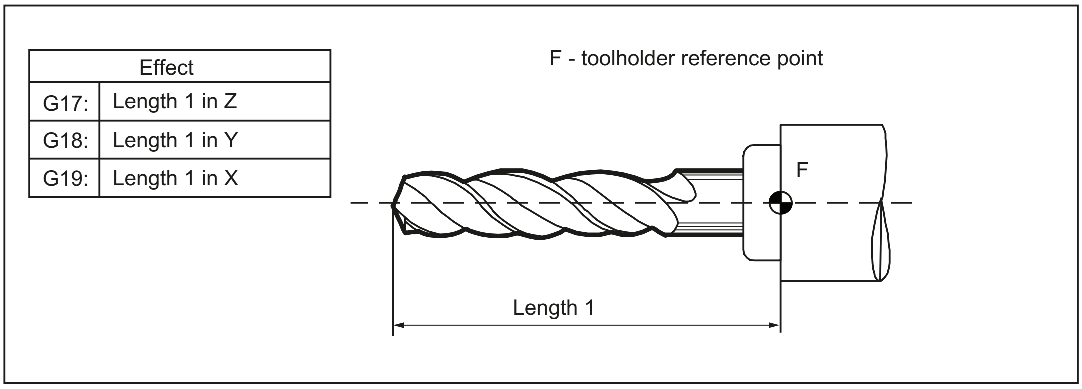
Tool special cases

For the tool types 'cutter' and 'drill', the parameters for length 2 and length 3 are only required for special cases (e.g. multi-dimensional length offset for an angle head construction).

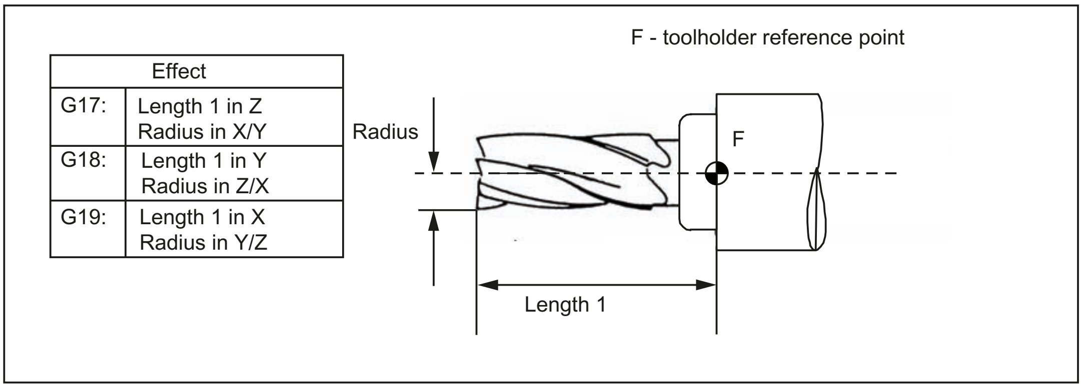
See the following illustration for effect of the tool length compensation - 3D (special case):



See the following illustration for effect of the offsets with the tool type 'drill':



See the following illustration for effect of the offsets with the tool type 'cutter':



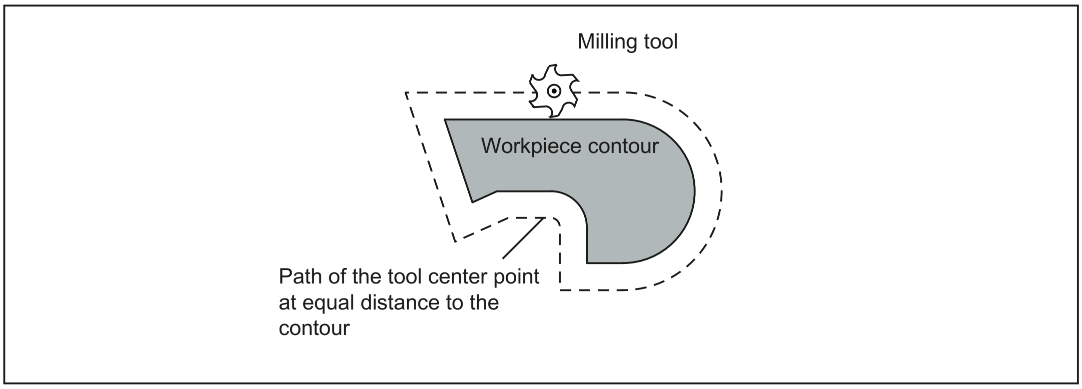
1.6.4 Selecting the tool radius compensation: G41, G42

Functionality

The control system is working with tool radius compensation in the selected plane G17 to G19.

A tool with a corresponding D number must be active. The tool radius compensation is activated by G41/G42. The control system automatically calculates the required equidistant tool paths for the programmed contour for the respective current tool radius.

See the following illustration for tool radius compensation:



Programming

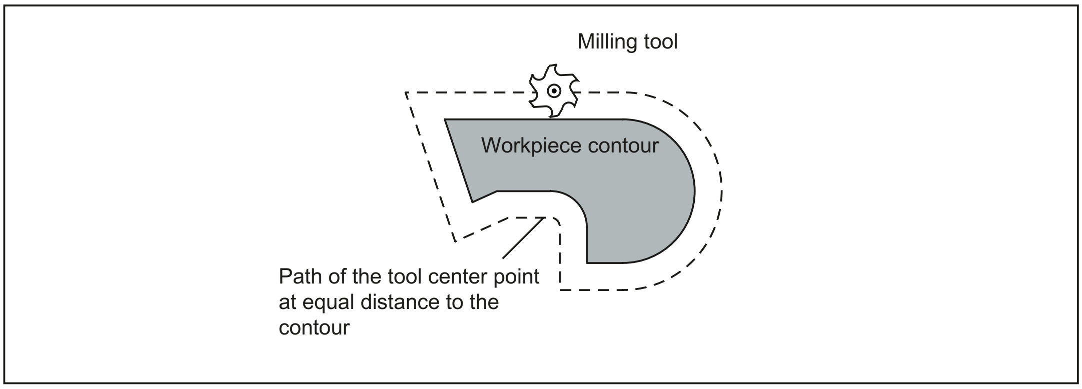
G41 X... Y...	; Tool radius compensation left of contour
G42 X... Y...	; Tool radius compensation right of contour

Note

The selection can only be made for linear interpolation (G0, G1).

Program both axes of the plane (e.g. with G17: X, Y). If you only specify one axis, the second axis is automatically completed with the last programmed value.

See the following illustration for compensation to the right/left of the contour:

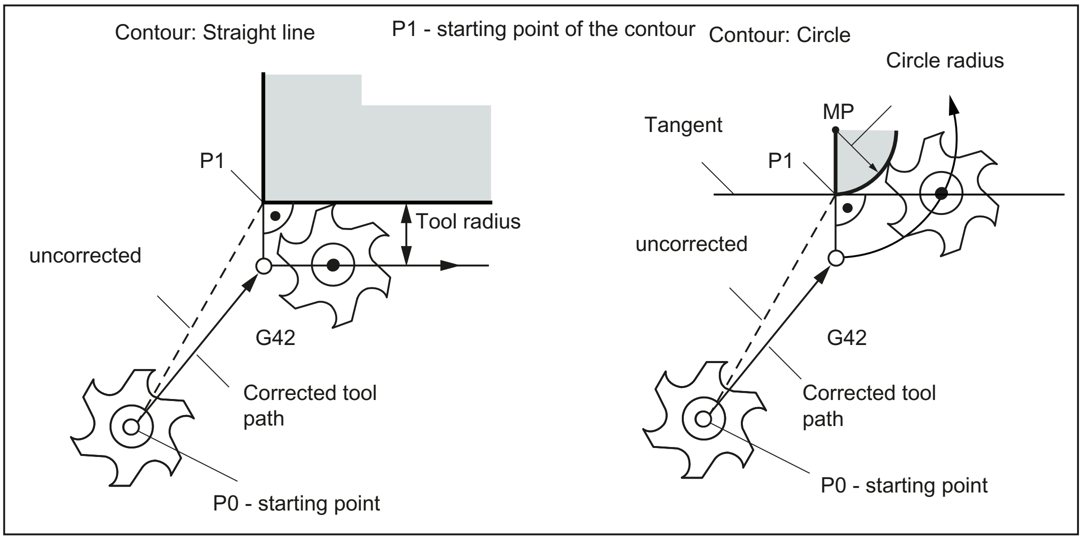


Starting the compensation

The tool travels in a straight line directly to the contour and is positioned perpendicular to the path tangent at the starting point of the contour.

Select the starting point such that a collision-free travel is ensured.

See the following illustration for start of the tool radius compensation with G42 as example:



The tool tip goes around the left of the workpiece when the tool runs clockwise using G41; the tool tip goes around the right of the workpiece when the tool runs counter-clockwise using G42.

Information

As a rule, the block with G41/G42 is followed by the block with the workpiece contour. The contour description, however, may be interrupted by 5 blocks which lie between them and do not contain any specifications for the contour path in the plane, e.g. only an M command or infeed motions.

Programming example

```
N10 T1
N20 G17 D2 F300               ; Correction number 2, feed 300 mm/min
N25 X0 Y0                     ; P0 - starting point
N30 G1 G42 X11 Y11            ; Selection right of contour, P1
N31 X20 Y20                   ; Starting contour, circle or straight line
M30
```

After the selection, it is also possible to execute blocks that contain infeed motions or M outputs:

```
N20 G1 G41 X11 Y11            ; Selection to the left of the contour
N21 Z20                       ; Infeed movement
N22 X20 Y20                   ; Starting contour, circle or straight line
```

1.6.5 Corner behavior: G450, G451

Functionality

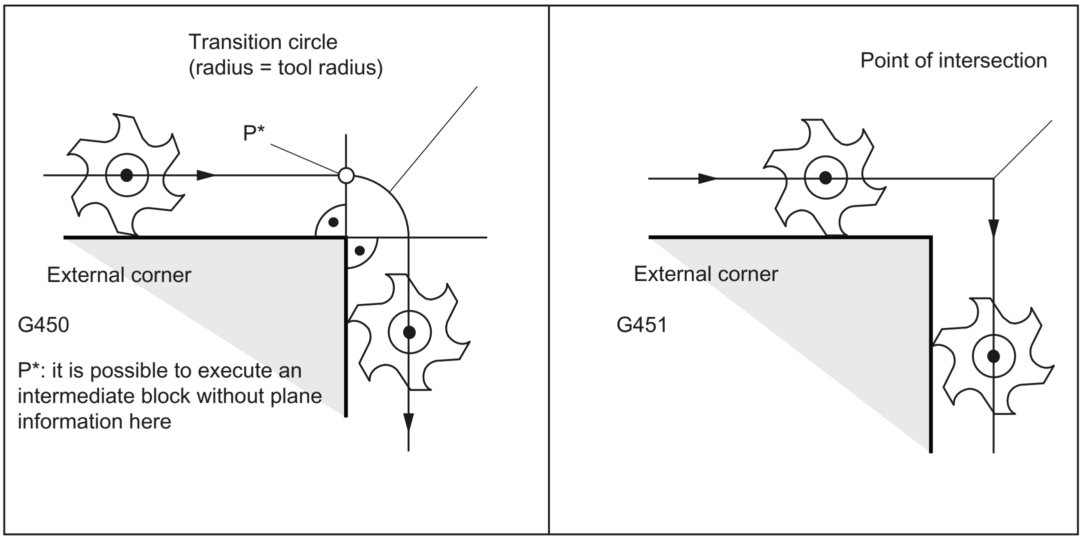
By using the functions G450 and G451, you can set the behavior for a non-continuous transition from one contour element to another contour element (corner behavior) when G41/G42 is active.

Internal and external corners are detected by the control system itself. For internal corners, the intersection of the equidistant paths is always approached.

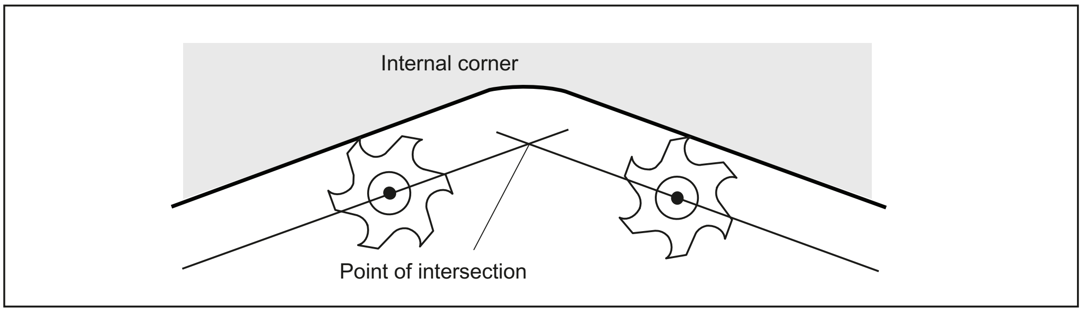
Programming

G450	; Transition circle
G451	; Point of intersection

See the following illustration for corner behavior at an external corner:



See the following illustration for corner behavior at an internal corner:



Transition circle G450

The tool center point travels around the workpiece external corner in an arc with the tool radius.

In view of the data, for example, as far as the feedrate value is concerned, the transition circle belongs to the next block containing traversing movements.

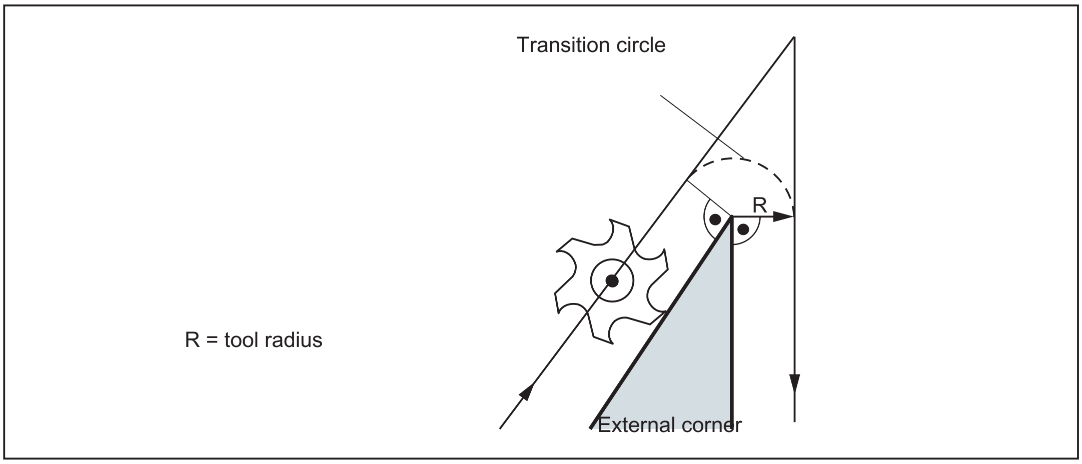
Point of intersection G451

For a G451 intersection of the equidistant paths, the point (intersection) that results from the center point paths of the tool (circle or straight line) is approached.

With acute contour angles and active point of intersection, depending on the tool radius, unnecessary idle motions could result for the tool.

In this case, the control system automatically switches to transition circle for this block if a certain set angle value (100°) is reached.

See the following illustration for acute contour angle and switching to transition circle:



1.6.6 Tool radius compensation OFF: G40

Functionality

The compensation mode (G41/G42) is deselected with G40. G40 is also the activation position at the beginning of the program.

The tool ends the **block in front of G40** in the normal position (compensation vector vertically to the tangent at the end point);

If G40 is active, the reference point is the tool center point. Subsequently, when deselected, the tool tip approaches the programmed point.

Always select the end point of the G40 block such that collision-free traversing is guaranteed!

Programming

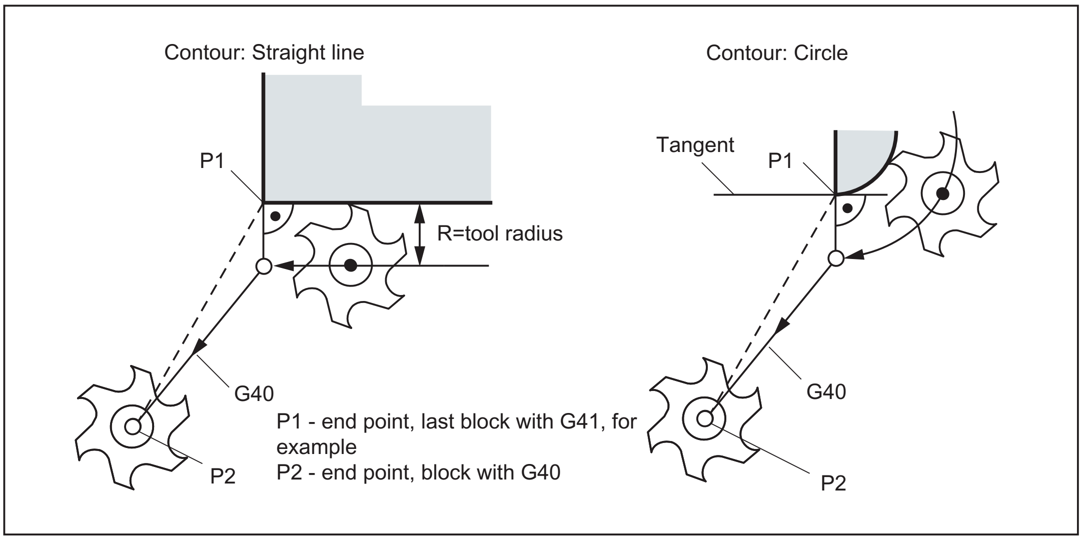
```
G40 X... Y...        ; Tool radius compensation OFF
```

Note

The compensation mode can only be deselected with linear interpolation (G0, G1).

Program both axes of the plane (e.g. with G17: X, Y). If you only specify one axis, the second axis is automatically completed with the last programmed value.

See the following illustration for quitting the tool radius compensation:



Programming example

```
N10 G0 X20 Y20 T1 D1 M3 S500
N20 G41 G1 X10 Y10 F100
N30 G2 X20 Y20 CR=20              ; Last block on the contour, circle or straight
                                  line, P1
N40 G40 G1 X10 Y10                ; Switch off tool radius compensation, P2
N50 M30
```

1.6.7 Special cases of the tool radius compensation

Repetition of the compensation

The same compensation (e.g. G41 -> G41) can be programmed once more without writing G40 between these commands.

The last block in front of the new compensation call ends with the normal position of the compensation vector at the end point. The new compensation is carried out as a compensation start (behavior as described for change in compensation direction).

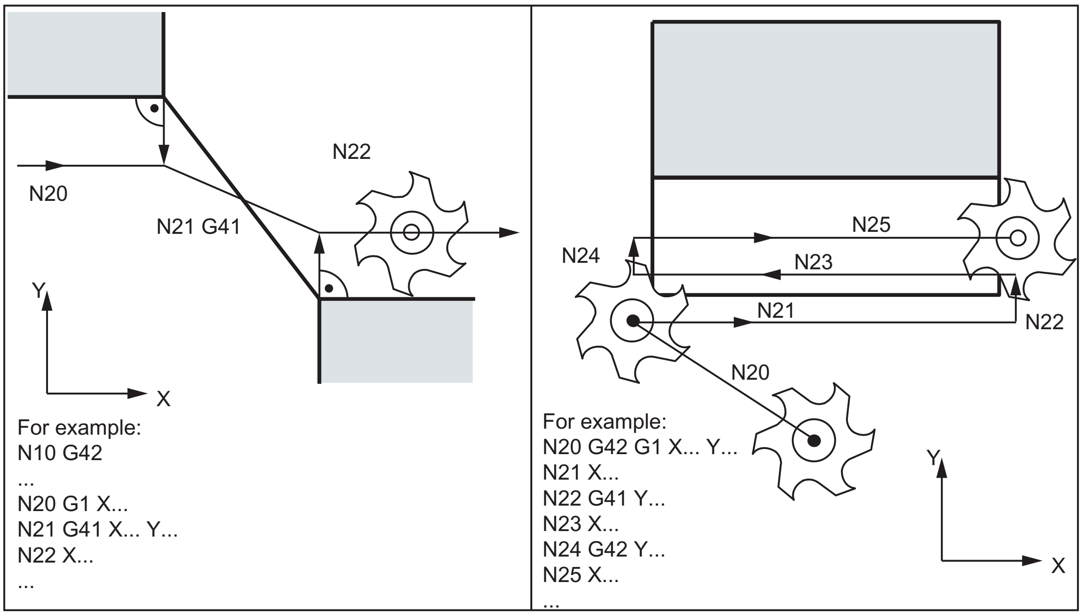
Changing the offset number

The offset number D can be changed in the compensation mode. A modified tool radius is active with effect from the block in which the new D number is programmed. Its complete modification is only achieved at the end of the block. In other words: The modification is traversed continuously over the entire block, also for circular interpolation.

Change of the compensation direction

The compensation direction G41 <-> G42 can be changed without writing G40.

The last block with the old compensation direction ends with the normal position of the compensation vector at the end point. The new compensation direction is executed as a compensation start (default setting at starting point).



Cancellation of compensation by M2

If compensation mode is canceled using M2 (end of program) without writing the command G40, the last block with coordinates of the plane (G17 to G19) will end in the normal position of the compensation vector. No compensating movement is executed. The program ends with this tool position.

Critical machining cases

When programming, pay special attention to cases where the contour travel is smaller than the tool radius;

Such cases should be avoided.

Also check over multiple blocks that the contour contains no "bottlenecks".

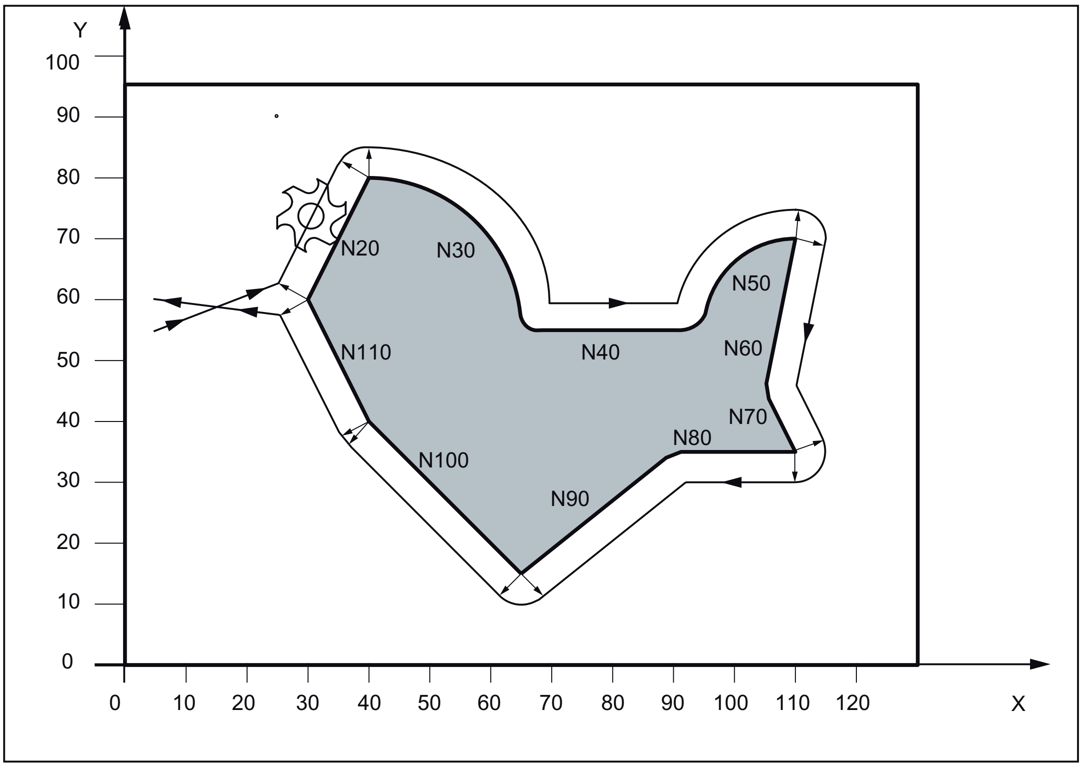
When carrying out a test/dry run, use the largest tool radius you are offered.

Acute contour angles

If very sharp outside corners occur in the contour with active G451 intersection, the control system automatically switches to transition circle. This prevents long idle motions.

1.6.8 Example of tool radius compensation

See the following illustration for example of tool radius compensation:



Programming example

N1 T1	; Tool 1 with offset D1
N5 G0 G17 G90 X5 Y55 Z50	; Approach starting point
N6 G1 Z0 F200 S80 M3	
N10 G41 G450 X30 Y60 F400	; Compensation to the left of the contour, transition circle
N20 X40 Y80	
N30 G2 X65 Y55 I0 J-25	
N40 G1 X95	
N50 G2 X110 Y70 I15 J0	
N60 G1 X105 Y45	
N70 X110 Y35	
N80 X90	
N90 X65 Y15	
N100 X40 Y40	
N110 X30 Y60	
N120 G40 X5 Y60	; Terminate compensation mode
N130 G0 Z50 M2	

1.7 Miscellaneous function M

Functionality

The miscellaneous function M initiates switching operations, such as "Coolant ON/OFF" and other functions.

Various M functions have already been assigned a fixed functionality by the CNC manufacturer. The functions not yet assigned fixed functions are reserved for free use of the machine manufacturer.

Note

An overview of the M miscellaneous functions used and reserved in the control system can be found in section "Overview of instructions".

Programming

```
M...                ;Max. 5 M functions per block
```

Effect

Activation in blocks with axis movements:

If the functions **M0, M1, M2** are contained in a block with traversing movements of the axes, these M functions become effective **after the traversing movements.**

The functions M3, M4, M5 are output to the internal interface (PLC) before the traversing movements. The axis movements only begin once the controlled spindle has ramped up for M3, M4. For M5, however, the spindle standstill is not waited for. The axis movements already begin before the spindle stops (default setting).

The remaining M functions are output to the PLC with the traversing movements.

If you would like to program an M function directly before or after an axis movement, insert a separate block with this M function.

Note

The M function interrupts the G64 continuous path mode and generates exact stop:

Programming example

```
N10 S1000
N20 X10 M3 G1 F100          ;M function in the block with axis movement, spindle
                            accelerates before the X axis movement
N30 M78 M67 M10 M12 M37     ;Max. 5 M functions in the block
M30
```

Note

In addition to M and H functions, T, D, and S functions can also be transferred to the PLC (programmable logic controller). In all, a maximum of 10 such function outputs are possible in a block.

1.8 H function

Functionality

With H functions, floating point data (REAL data type - as with arithmetic parameters, see Section "Arithmetic parameter R (Page 87)") can be transferred from the program to the PLC.

The meaning of the values for a given H function is defined by the machine manufacturer.

Programming

H0=... to H9999=... ;Max. 3 H functions per block

Programming example

```
N10 H1=1.987 H2=978.123 H3=4      ;3 H functions in block
N20 G0 X71.3 H99=-8978.234        ;With axis movements in block
N30 H5                            ;Corresponds to H0=5.0
```

Note

In addition to M and H functions, T, D, and S functions can also be transferred to the PLC (programmable logic controller). In all, a maximum of 10 function outputs of this type are possible in a part program block.

1.9 Arithmetic parameters, LUD and PLC variables

1.9.1 Arithmetic parameter R

Functionality

The arithmetic parameters are used if an NC program is not only to be valid for values assigned once, or if you must calculate values. The required values can be set or calculated by the control system during program execution.

Another possibility consists of setting the arithmetic parameter values by operator inputs. If values have been assigned to the arithmetic parameters, they can be assigned to other variable-setting NC addresses in the program.

Programming

R0=... to R299=...	;Assign values to the arithmetic parameters
R[R0]=...	;Indirect programming: Assign a value to the arithmetic parameter R, whose number can be found, e.g. in R0
X=R0	;Assign arithmetic parameters to the NC addresses, e.g. for the X axis

Value assignments

You can assign values in the following range to the R parameters:

±(0.000 0001 ... 9999 9999)
(8 decimal places, arithmetic sign, and decimal point)

The decimal point can be omitted for integer values. A plus sign can always be omitted.

Example:

R0=3.5678 R1=-37.3 R2=2 R3=-7 R4=-45678.123

Use the **exponential notation** to assign an extended range of numbers:

± (10^{-300} ... 10^{+300})

The value of the exponent is written after the **EX** characters; maximum total number of characters: 10 (including leading signs and decimal point)

Range of values for EX: -300 to +300

Example:

```
R0=-0.1EX-5            ;Meaning: R0 = -0.000 001
R1=1.874EX8            ;Meaning: R1 = 187 400 000
```

Note

There can be several assignments in one block incl. assignments of arithmetic expressions.

Assignments to other addresses

The flexibility of an NC program lies in assigning these arithmetic parameters or expressions with arithmetic parameters to other NC addresses. Values, arithmetic expressions and arithmetic parameters can be assigned to all addresses; **Exception: addresses N, G, and L**.

When assigning, write the " = " sign after the address character. It is also possible to have an assignment with a minus sign.

A separate block is required for assignments to axis addresses (traversing instructions).

Example:

```
N10 G0 X=R2           ;Assignment to X axis
```

Arithmetic operations/arithmetic functions

When operators/arithmetic functions are used, it is imperative to use the conventional mathematical notation. Machining priorities are set using the round brackets. Otherwise, multiplication and division take precedence over addition and subtraction.

Degrees are used for the trigonometrical functions.

Permitted arithmetic functions: see Section "List of instructions (Page 12)"

Programming example: Calculating with R parameters

```
N10 R1= R1+1                          ;The new R1 is calculated from the old R1
                                      plus 1
N20 R1=R2+R3 R4=R5-R6 R7=R8*R9 R10=R11/R12
N30 R13=SIN(25.3)                     ;R13 equals sine of 25.3 degrees
N40 R14=R1*R2+R3                      ; Multiplication and division take precedence
                                      over addition or subtraction R14=(R1*R2)+R3
N50 R14=R3+R2*R1                      ;Result, the same as block N40
N60 R15=SQRT(R1*R1+R2*R2)             ;Meaning:
N70 R1= -R1                           ;The new R1 is the negative old R1
```

Programming example: Assign R parameters to the axes

```
R1=40 R2=10 R3=-20 R4=-45 R5=-30
N10 G1 G90 X=R1 Z=R2 F300             ;Separate blocks (traversing blocks)
N20 Z=R3
N30 X=-R4
N40 Z= SIN(25.3)-R5                   ;With arithmetic operations
M30
```

Programming example: Indirect programming

```
N10 R1=5                      ;Assigning R1 directly value 5 (integer)
N20 G0 X R[R1]=27.123         ;Indirectly assign R5 the value 27.123
M30
```

1.9.2 Local User Data (LUD)

Functionality

The operator/programmer (user) can define his/her own variable in the program from various data types (LUD = Local User Data). These variables are only available in the program in which they were defined. The definition takes place immediately at the start of the program and can also be associated with a value assignment at the same time. Otherwise the starting value is zero.

The name of a variable can be defined by the programmer. The naming is subject to the following rules:

- A maximum of 32 characters can be used.
- It is imperative to use letters for the first two characters; the remaining characters can be either letters, underscore or digits.
- Do not use a name already used in the control system (NC addresses, keywords, names of programs, subroutines, etc.).

Programming / data types

DEF BOOL varname1	;Boolean typ, values: TRUE (=1), FALSE (=0)
DEF CHAR varname2	;Char type, 1 ASCII code character: "a", "b", ...
	;Numerical code value: 0 ... 255
DEF INT varname3	;Integer type, integer values, 32 bit value range:
	;-2 147 483 648 through +2 147 483 647 (decimal)
DEF REAL varname4	;Real type, natural number (like arithmetic parameter R),
	;Value range: ±(0.000 0001 ... 9999 9999)
	;(8 decimal places, arithmetic sign and decimal point) or
	;Exponential notation: ± (10 to power of -300 ... 10 to power of +300)
DEF STRING[string length] varname41	; STRING type, [string length]: Maximum number of characters

Each data type requires its own program line. However, several variables of the same type can be defined in one line.

Example:

```
DEF INT PVAR1, PVAR2, PVAR3=12, PVAR4      ;4 type INT variables
```

Example for STRING type with assignment:

```
DEF STRING[12] PVAR="Hello"                ; Define variable PVAR with a maximum of
                                           12 characters and assign string "Hello"
```

Fields

In addition to the individual variables, one or two-dimensional fields of variables of these data types can also be defined:

```
DEF INT PVAR5[n]                ;One-dimensional field, type INT, n: integer
DEF INT PVAR6[n,m]              ;Two-dimensional field, type INT, n, m: integer
```

Example:

```
DEF INT PVAR7[3]                ;Field with 3 elements of the type INT
```

Within the program, the individual field elements can be reached via the field index and can be treated like individual variables. The field index runs from 0 to a small number of the elements.

Example:

```
N10 PVAR7[2]=24                 ;The third field element (with index 2) is assigned
                                the value 24.
```

Value assignment for field with SET instruction:

```
N20 PVAR5[2]=SET(1,2,3)         ;After the 3rd field element, different values are
                                assigned.
```

Value assignment for field with REP instruction:

```
N20 PVAR7[4]=REP(2)             ;After field element [4] - all are assigned the same
                                value, here 2.
```

1.9.3 Reading and writing PLC variables

Functionality

To allow rapid data exchange between NC and PLC, a special data area exists in the PLC user interface with a length of 512 bytes. In this area, PLC data are compatible in data type and position offset. In the NC program, these compatible PLC variables can be read or written.

To this end, special system variables are provided:

$A_DBB[n]	;Data byte (8-bit value)
$A_DBW[n]	;Data word (16-bit value)
$A_DBD[n]	;Data double-word (32-bit value)
$A_DBR[n]	;REAL data (32-bit value)

"n" stands here for the position offset (start of data area to start of variable) in bytes

Programming example

```
R1=$A_DBR[5]            ;Reading a REAL value, offset 5 (starts at byte 5 of range)
```

Note

The reading of variables generates a preprocessing stop (internal STOPRE).

Note

Writing of PLC tags is generally limited to a maximum of three tags (elements).

Where PLC tags are to be written in rapid succession, one element will be required per write operation.

If more write operations are to be executed than there are elements available, then block transfer will be required (a preprocessing stop may need to be triggered).

Example:

```
$A_DBB[1]=1 $A_DBB[2]=2 $A_DBB[3]=3

STOPRE

$A_DBB[4]=4
```

1.10 Program jumps

1.10.1 Unconditional program jumps

Functionality

NC programs process their blocks in the sequence in which they were arranged when they were written.

The processing sequence can be changed by introducing program jumps.

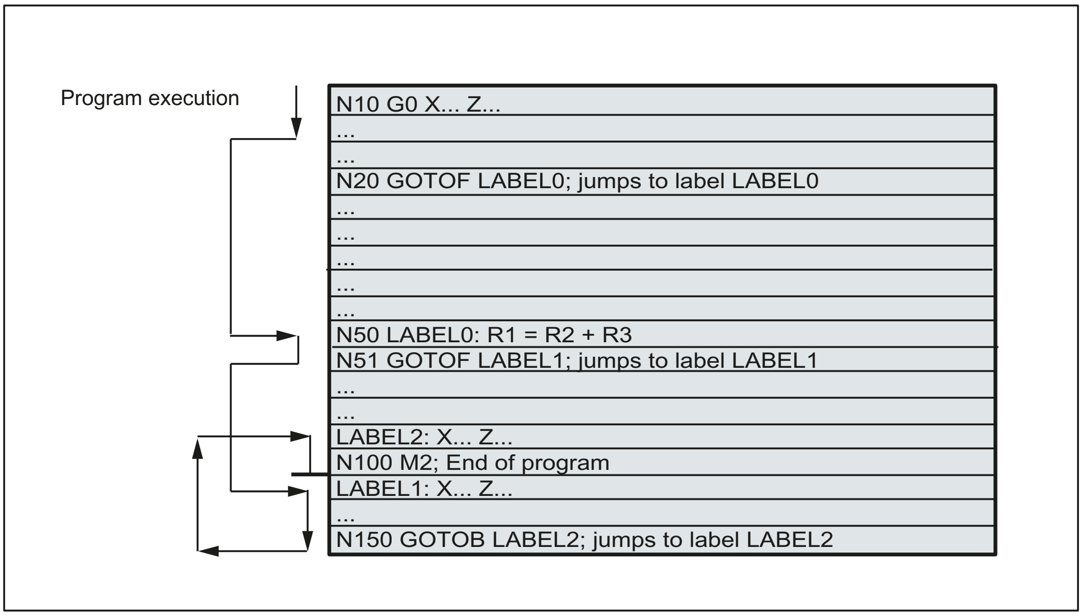
The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

The unconditional jump instruction requires a separate block.

Programming

GOTOF label	;Jump forward (in the direction of the last block of the program)
GOTOB label	;Jump backwards (in the direction of the first block of the program)
Label	;Selected string for the label (jump label) or block number

See the following illustration for an example of unconditional jumps:



1.10.2 Conditional program jumps

Functionality

Jump conditions are formulated after the **IF instruction**. If the jump condition (**value not zero**) is satisfied, the jump takes place.

The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

Conditional jump instructions require a separate block. Several conditional jump instructions can be located in the same block.

By using conditional program jumps, you can also considerably shorten the program, if necessary.

Programming

IF condition GOTOF label	;Jump forward
IF condition GOTOB label	;Jump backwards
GOTOF	;Jump direction forward (in the direction of the last block of the program)
GOTOB	;Jump direction backwards (in the direction of the first block of the program)
Label	;Selected string for the label (jump label) or block number
IF	;Introduction of the jump condition
Condition	;Arithmetic parameter, arithmetic expression for formulating the condition

Comparison operations

Operators	Meaning
= =	Equal to
< >	Not equal to
>	greater than
<	less than
> =	greater than or equal to
< =	less than or equal to

The comparison operations support formulating of a jump condition. Arithmetic expressions can also be compared.

The result of comparison operations is "satisfied" or "not satisfied." "Not satisfied" sets the value to zero.

Programming example for comparison operators

```
R1>1                           ;R1 greater than 1
1 < R1                         ;1 less than R1
R1<R2+R3                       ;R1 less than R2 plus R3
R6>=SIN( R7*R7)                ; R6 greater than or equal to SIN (R7) squared
```

Programming example

```
N10 IF R1 GOTOF LABEL1                ;If R1 is not null then go to the block
                                      having LABEL1
G0 X30 Y30
N90 LABEL1: G0 X50 Y30
N100 IF R1>1 GOTOF LABEL2             ;If R1 is greater than 1 then go to the
                                      block having LABEL2
G0 X40 Y40
N150 LABEL2: G0 X60 Y60
G0 X70 Y70
N800 LABEL3: G0 X80 Y80
G0 X100 Y100
N1000 IF R45==R7+1 GOTOB LABEL3       ;If R45 is equal to R7 plus 1 then go to the
                                      block having LABEL3
M30
Several conditional jumps in the
block:
N10 MA1: G0 X20 Y20
N15 G0 X0 Y0
N20 IF R1==1 GOTOB MA1 IF R1==2 GOTOF MA2
N30 G0 X10 Y10
N50 MA2: G0 X50 Y50
N60 M30
```

Note

The jump is executed for the first fulfilled condition.

1.10.3 Program example for jumps

Task

Approaching points on a circle segment:

Existing conditions:

Start angle: 30° in R1

Circle radius: 32 mm in R2

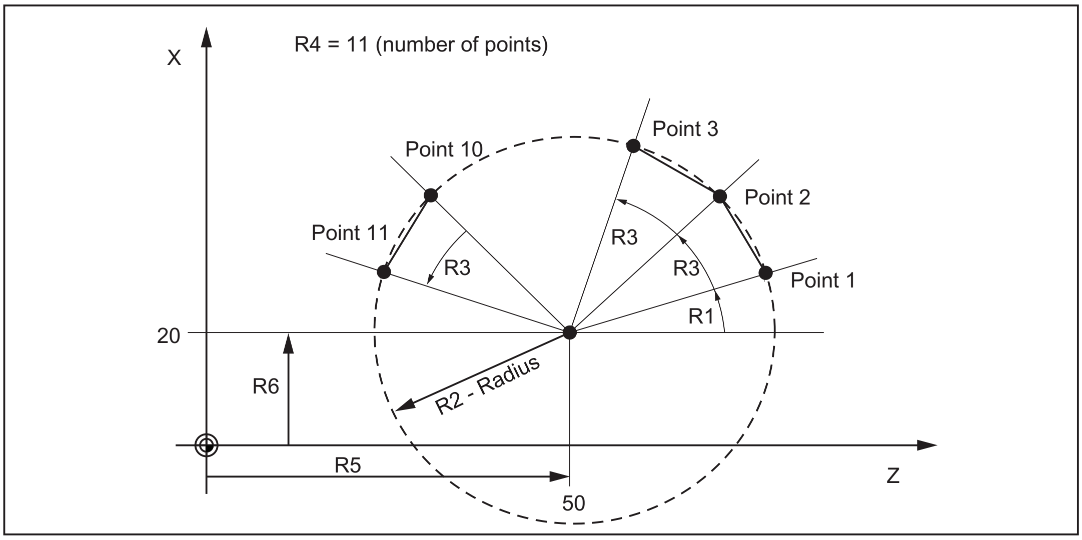
Position spacing: 10° in R3

Number of points: 11 in R4

Position of circle center in Z: 50 mm in R5

Position of circle center in X: 20 mm in R6

See the following illustration for linear approach of points on a circle segment:



Programming example

`N10 R1=30 R2=32 R3=10 R4=11 R5=50 R6=20`	`;Assignment of initial values`
`N20 MA1: G0 Z=R2*COS (R1)+R5 X=R2*SIN(R1)+R6`	`;Calculation and assignment to axis addresses`
`N30 R1=R1+R3 R4= R4-1`	
`N40 IF R4 > 0 GOTOB MA1`	
`N50 M2`	

Explanation

In block N10, the starting conditions are assigned to the corresponding arithmetic parameters. The calculation of the coordinates in X and Z and the processing takes place in N20.

In block N30, R1 is incremented by the clearance angle R3, and R4 is decremented by 1.

If R4 > 0, N20 is executed again; otherwise, N50 with End of program.

1.10.4 Jump destination for program jumps

Functionality

A **label** or a **block number** serve to mark blocks as jump destinations for program jumps. Program jumps can be used to branch to the program sequence.

Labels can be freely selected, but must contain a minimum of 2 and a maximum of 8 letters or numbers of which the **first two characters must be letters** or underscore characters.

Labels that are in the block that serves as the jump destination are **ended by a colon**. They are always at the start of a block. If a block number is also present, the label is located **after the block number**.

Labels must be unique within a program.

Programming example

```
N10 LABEL1: G1 X20 F100        ;LABEL1 is the label, jump destination
N20 G0 X10 Y10
TR789: G0 X10 Z20              ;TR789 is the label, jump destination
N30 G0 X30 Z30                 - No block number existing
N100 G0 X40 Z40                ;Block number can be jump target
M30
```

1.11 Subroutine technique

1.11.1 General information

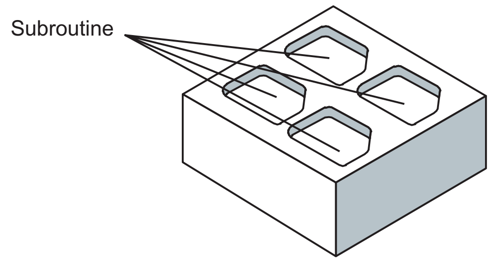
Application

Basically, there is no difference between a main program and a subroutine.

Frequently recurring machining sequences are stored in subroutines, e.g certain contour shapes. These subroutines are called at the appropriate locations in the main program and then executed.

One form of a subroutine is the **machining cycle**. The machining cycles contain generally valid machining cases (e.g. drilling, tapping, groove cutting, etc.). By assigning values via included transfer parameters, you can adapt the subroutine to your specific application.

See the following illustration for example for using a subroutine for a workpiece four times:



Set-up

The structure of a subroutine is identical to that of a main program (see Section "Program structure (Page 7)"). Like main programs, subroutines contain **M2 - end of program** in the last block of the program sequence. This means a return to the program level where the subroutine was called from.

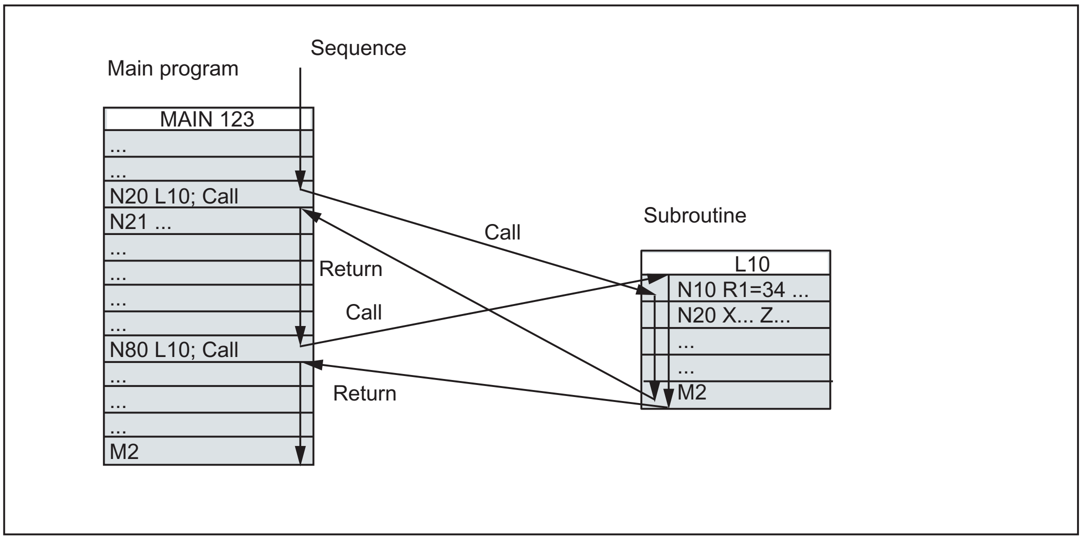
End of program

The end instruction **RET** can also be used instead of the M2 program end in the subroutine.

RET must be programmed in a separate block.

The RET instruction is used when G64 continuous-path mode is not to be interrupted by a return. With M2, G64 is interrupted and exact stop is initiated.

See the following illustration for example of sequence when calling a subroutine twice:



Subroutine name

The program is given a unique name allowing it to be selected from several subroutines. When you create the program, the program name may be freely selected, provided the following conventions are observed.

The same rules apply as for the names of main programs.

Example: **LRAHMEN7**

It is also possible to use the address word **L...** in subroutines. The value can have 7 decimal places (integers only).

Note

With address L, leading zeros are meaningful for differentiation.

Example: **L128** is not **L0128** or **L00128**.

These are three different subroutines.

Note

The subroutine name **LL6** is reserved for tool change.

Subroutine call

Subroutines are called in a program (main or subroutine) with their names. To do this, a separate block is required.

Example:

```
N10 L785                        ; Call subroutine L785
N20 LRAHMEN7                    ; Call subroutine LRAHMEN7
```

Program repetition P...

If a subroutine is to be executed several times in succession, write the number of times it is to be executed in the block of the call after the subroutine name under the **address P**. A maximum of **9,999 cycles** are possible (P1 ... P9999).

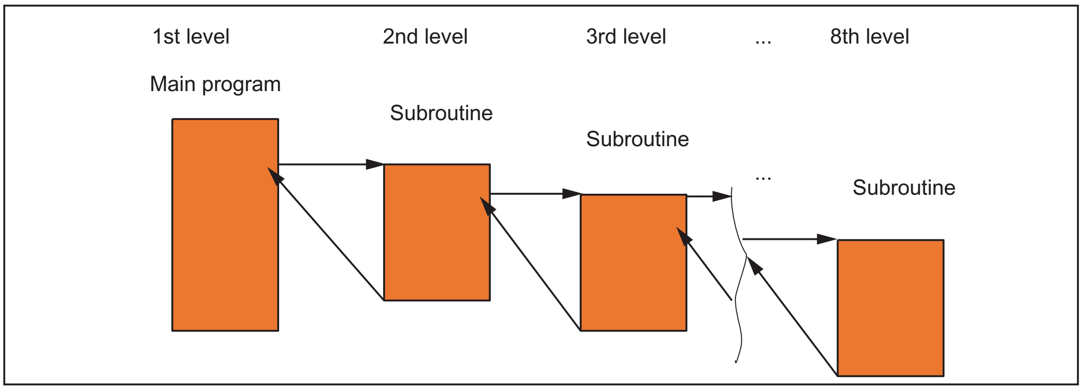
Example:

```
N10 L785 P3                     ; Call subroutine L785, 3 cycles
```

Nesting depth

Subroutines can also be called from a subroutine, not only from a main program. In total, up to **8 program levels** are available for this type of nested call, including the main program level.

See the following illustration for execution with 8 program levels:



Information

Modal G functions can be changed in the subroutine, e.g. G90 -> G91. When returning to the calling program, ensure that all modal functions are set the way you need them to be.

Please make sure that the values of your arithmetic parameters used in upper program levels are not inadvertently changed in lower program levels.

When working with SIEMENS cycles, up to 4 program levels are needed.

1.11.2 Calling machining cycles

Functionality

Cycles are technology subroutines realizing a certain machining process generally, for example, drilling or milling. Adaptation to the particular problem is performed directly via supply parameters/values when calling the respective cycle.

Programming example

```
N10 DEF REAL RTP, RFP, SDIS, DP, DTB
N20 G18 X100 Z100 G0
N30 M3 S100 F100
N40 G17 X0
N50 CYCLE83(110, 90, 0, -80, 0, -10, 0, 0,    ; Call of cycle 83, transfer values
0, 0, 1, 0)                                   directly, separate block
N60 G0 X100 Z100
N70 RTP=100 RFP= 95.5 SDIS=1, DP=-5, DTB=3    ; Set transfer parameters for cycle 82
N80 CYCLE82(RTP, RFP, SDIS, DP, , DTB)        ; Call of cycle 82, separate block
N90 M30
```

1.11.3 Modal subroutine call

Functionality

The subroutine in the block containing MCALL is called automatically after each successive block containing a path motion. The call acts until the next MCALL is called.

The modal call of the subroutine which contains MCALL or quitting of the call requires a separate block.

MCALL is advantageous, for example, when producing drill patterns.

Programming example

Application example: Drilling a row of holes

```
N10 MCALL CYCLE82(100, 0, 1, -10, 2)          ; Drilling cycle 82
N20 HOLES1(1, 2, 45, 2, 2, 1)                 ; Cycle for row of holes; after each
                                              approach of the hole position,
                                              CYCLE82(...) will be called with the
                                              transfer parameters
N30 MCALL                                     ; Modal call of CYCLE82(...) completed
N40 M30
```

1.11.4 Execute external subroutine (EXTCALL)

Function

With the `EXTCALL` command, you can reload and execute programs stored on an external USB memory sticker.

Machine data

The following machine data is used for the `EXTCALL` command:

- MD10132 $MN_MMC_CMD_TIMEOUT

  Monitoring time for the command in part program
- MD18362 $MN_MM_EXT_PROG_NUM

  Number of program levels that can be processed simultaneously from external
- SD42700 $SC_EXT_PROGRAM_PATH

  Program path for external subroutine call

Note

When using SD42700 $SC_EXT_PROGRAM_PATH, all subprograms called with EXCALL are searched under this path.

Programming with path specification in SD42700 EXT_PROGRAM_PATH

`EXTCALL` ("<program name>")

Parameter

EXTCALL	; Keyword for subroutine call
<program name>	; Constant/variable of STRING type
Example:	
EXTCALL ("RECTANGULAR POCKET")	

Programming without path specification in SD42700 EXT_PROGRAM_PATH

`EXTCALL` ("<path\program name>")

Parameter

EXTCALL	; Keyword for subroutine call
<Path\program name>	; Constant/variable of STRING type

Example:

EXTCALL ("D:\EXTERNE_UP\RECHTECKTASCHE")

Note

External subroutines must not contain jump statements such as `GOTOF`, `GOTOB`, `CASE`, `FOR`, `LOOP`, `WHILE`, or `REPEAT`.

`IF-ELSE-ENDIF` constructions are possible.

Subroutine calls and nested `EXTCALL` calls may be used.

RESET, POWER ON

RESET and POWER ON cause external subroutine calls to be interrupted and the associated load memory to be erased.

Example

Processing of external customer USB memory sticker

The "Main.mpf" main program is stored in NC memory and is selected for execution:

```
N010 PROC MAIN
N020 G0 X10 Y10
N030 EXTCALL ("D:\EXTERNE_UP\BOHRUNG")
N040 G0 X20 Y20
N050 M30
```

The "BOHRUNG.SPF" subprogram to be reloaded is located on the USB memory sticker.

```
N010 PROC BOHRUNG
N020 G1 F1000
N030 X=10 Z=10
N040 G0 X50 Y50
N050 M17
```

1.12 Timers and workpiece counters

1.12.1 Runtime timer

Functionality

The timers are prepared as system variables ($A...) that can be used for monitoring the technological processes in the program or only in the display.

These timers are read-only. There are timers that are always active. Others can be deactivated via machine data.

Timers - always active

- $AN_SETUP_TIME

  Time since the last control power-up with default values (in minutes)

  It is automatically reset in the case of a "Control power-up with default values".

- $AN_POWERON_TIME

  Time since the last control power-up (in minutes)

  It is reset to zero automatically with each power-up of the control system.

Timers that can be deactivated

The following timers are activated via machine data (default setting).

The start is timer-specific. Each active run-time measurement is automatically interrupted in the stopped program state or for feedrate-override-zero.

The behavior of the activated timers for active dry run feedrate and program testing can be specified using machine data.

- $AC_OPERATING_TIME

  Total execution time in seconds of NC programs in "AUTO" mode

  In "AUTO" mode, the runtimes of all programs between program start and end are summed up. The timer is zeroed with each power-up of the control system.

- $AC_CYCLE_TIME

  Runtime of the selected NC program (in seconds)

  The runtime between program start and end is measured in the selected NC program. The timer is reset with the start of a new NC program.

- $AC_CUTTING_TIME

  Tool action time (in seconds)

  The runtime of the path axes is measured in all NC programs between program start and end without rapid traverse active and with the tool active (default setting).

  The measurement is interrupted when a dwell time is active.

  The timer is automatically set to zero with each power-up of the control system.

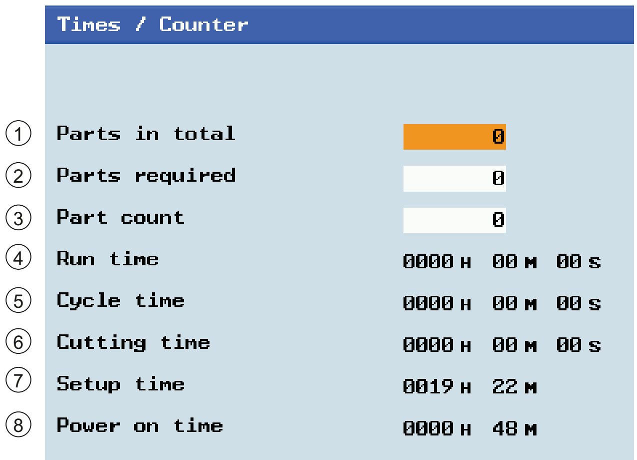
Programming example

```
N10 IF $AC_CUTTING_TIME>=R10 GOTOF WZZEIT              ; Tool operation time limit
                                                         value?
G0 X20 Y20
N80 WZZEIT:G0 X30 Y30
N90 MSG("Tool action time: Limit value reached")
N100 M0
M30
```

Display

The content of the active system variables is visible on the window opened through the following key operations:

Window display:



① = $AC_TOTAL_PARTS

② = $AC_REQUIRED_PARTS

③ =$AC_ACTUAL_PARTS

$AC_SPECIAL_PARTS is not available for display.

④ = $AC_OPERATING_TIME

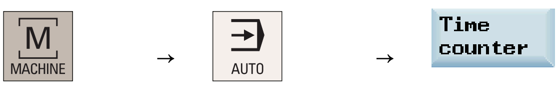
⑤ = $AC_CYCLE_TIME

⑥ = $AC_CUTTING_TIME

⑦ = $AN_SETUP_TIME

⑧ = $AN_POWERON_TIME

You can also view the time counter information through the following operating area:



1.12.2 Workpiece counter

Functionality

The "Workpiece counter" function provides counters for counting workpieces.

These counters exist as system variables with write and read access from the program or via operator input (observe the protection level for writing!).

Machine data can be used to control counter activation, counter reset timing and the counting algorithm.

Counters

- $AC_REQUIRED_PARTS

  Number of workpieces required (workpiece setpoint)

  In this counter you can define the number of workpieces at which the actual workpiece counter $AC_ACTUAL_PARTS is reset to zero.

  The generation of the display alarm 21800 "Workpiece setpoint reached" can be activated via machine data.

- $AC_TOTAL_PARTS

  Total number of workpieces produced (total actual)

  The counter specifies the total number of all workpieces produced since the start time.

  The counter is automatically set to zero upon every booting of the control system.

- $AC_ACTUAL_PARTS

  Number of actual workpieces (actual)

  This counter registers the number of all workpieces produced since the starting time. When the workpiece setpoint is reached ($AC_REQUIRED_PARTS, value greater than zero), the counter is automatically zeroed.

- $AC_SPECIAL_PARTS

  Number of workpieces specified by the user

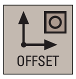
  This counter allows users to make a workpiece counting in accordance with their own definition. Alarm output can be defined for the case of identity with $AC_REQUIRED_PARTS (workpiece target). Users must reset the counter themselves.

Programming example

```
N10 IF $AC_TOTAL_PARTS==R15 GOTOF SIST                  ; Count reached?
G0 X20 Y20
N80 SIST: G0 X30 Y30
N90 MSG("Workpiece setpoint reached")
N100 M0
```

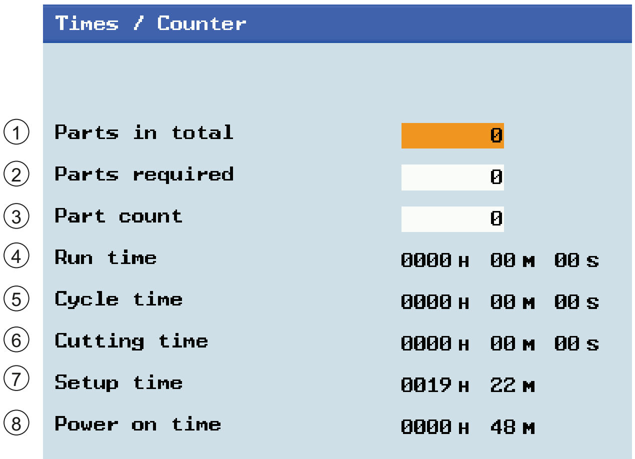
Display

The content of the active system variables is visible on the window opened through the following key operations:



→ Sett. data → Time counter

Window display:

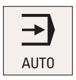


① = $AC_TOTAL_PARTS
② = $AC_REQUIRED_PARTS
③ =$AC_ACTUAL_PARTS

$AC_SPECIAL_PARTS is not available for display.

④ = $AC_OPERATING_TIME
⑤ = $AC_CYCLE_TIME
⑥ = $AC_CUTTING_TIME
⑦ = $AN_SETUP_TIME
⑧ = $AN_POWERON_TIME

You can also select whether to activate the workpiece counter function through the following operating area:

→


→ Time counter

1.13 Smooth approach and retraction

Functionality

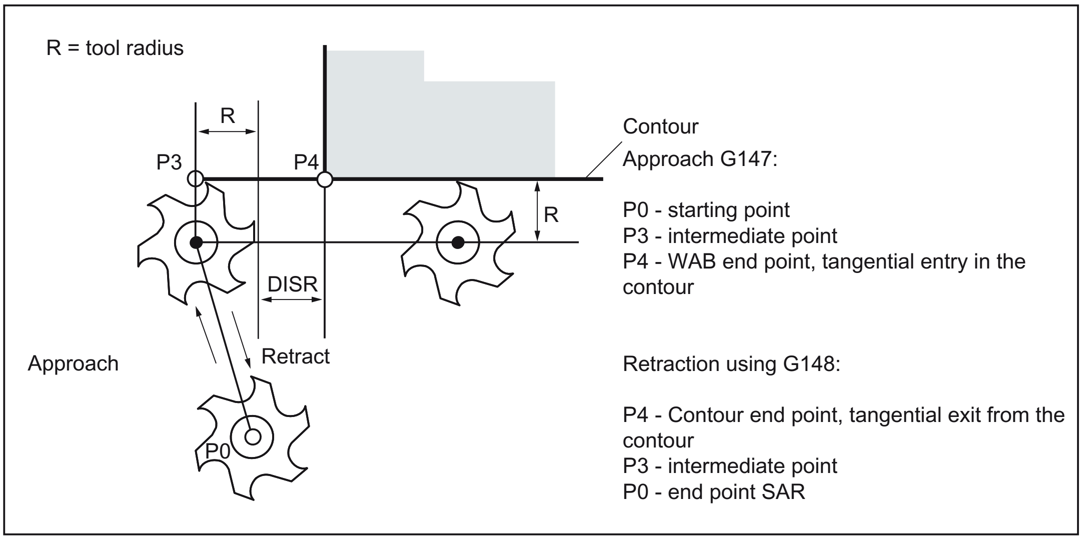
The function "Smooth approach and retraction" (SPR) is intended to approach the beginning of a contour tangentially ("smooth"), to a large degree independently of the position of the starting point. The control system will calculate the intermediate points and will generate the required traversing blocks. This function is used preferably in conjunction with the tool radius compensation (TRC). The G41 and G42 commands determine the approach/retraction direction to the left or right of the contour.

The approach/retraction path (straight line, quarter or semi-circle) is selected using a group of G commands. To parameterize this path (circle radius, length, approach straight line), special addresses can be used; this also applies to the feedrate of the infeed motion. The infeed motion can additionally be controlled via another G group.

Programming

G147	; Approach with a straight line
G148	; Retraction with a straight line
G247	; Approach with a quadrant
G248	; Retraction with a quadrant
G347	; Approach with a semi-circle
G348	; Retraction with a semi-circle
G340	; Approach and retraction in space (basic setting)
G341	; Approach and retraction in the plane
DISR=...	; Approach and retraction with straight lines (G147/G148): Distance of the cutter edge from the start or end point of the contour
	; Approach and retraction along circles (G247, G347/G248, G348): Radius of the tool center point path
DISCL=...	; Distance of the end point for the fast infeed motion from the machining plane (safety clearance)
FAD=...	; Speed of the slow infeed motion
	The programmed value acts according to the active command of the G group 15 (feed: G94, G95)

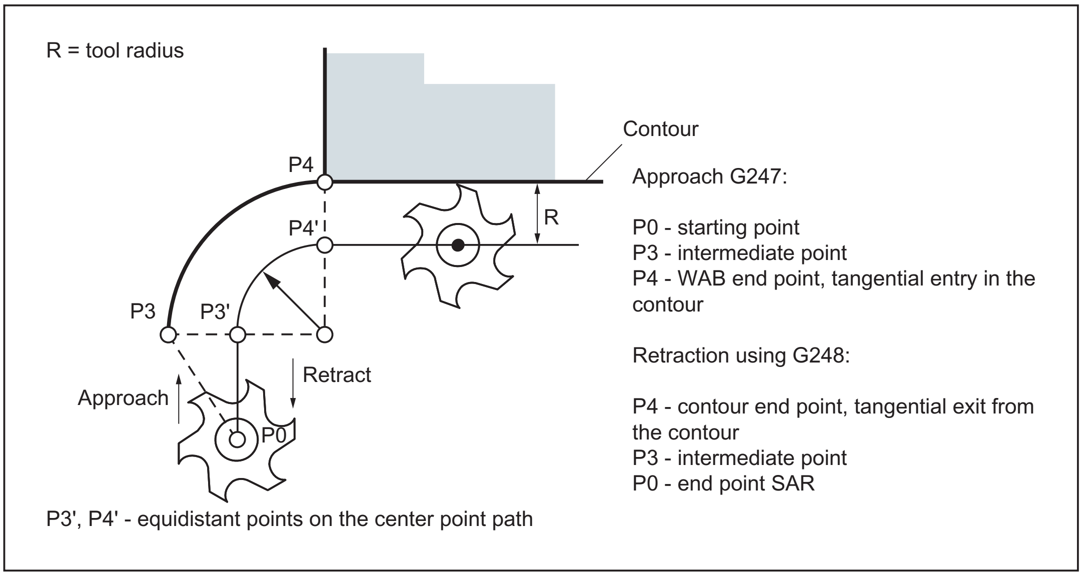
See the following illustration for approaching along a straight line using the example of G42 or retraction using G41 and completion with G40:



Programming example: Approach/retraction along a straight line in a plane

```
N10 T1 G17                              ; Activate tool, X/Y plane
N20 G0 X20 Y20                          ; Approach P0
N30 G42 G147 DISR=8 F600 X4 Y4          ; Approach, point P4 programmed
N40 G1 X40                              ; Continue in the contour
N50 Y12
N100 G41 G1 X15 Y15
N110 X4 Y4                              ; P4 - contour end point
N120 G40 G148 DISR=8 F700 X8 Y8         ; Retraction, point P0 programmed
M30
```

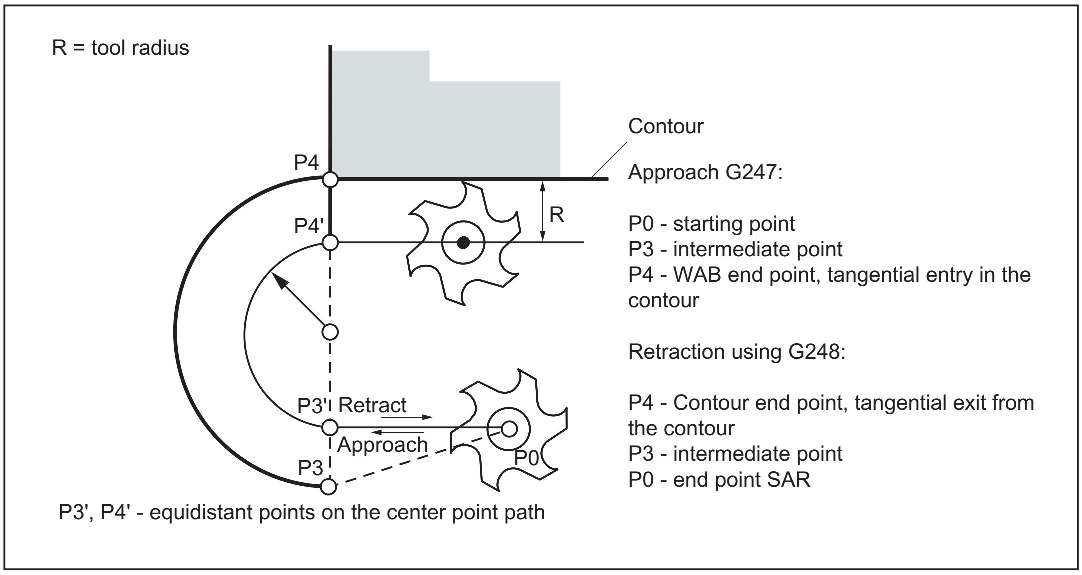
See the following illustration for approaching along a quadrant using the example of G42 or retraction using G41 and completion with G40:



Programming example: Approach/retraction along a quarter in a plane

```
N10 T1 D1 G17                          ; Activate tool, X/Y plane
N20 G0 X20 Y20                         ; Approach P0
N30 G42 G247 DISR=20 F600 X4 Y4        ; Approach, point P4 programmed
N40 G1 X40                             ; Continue in the contour
N50 Y12
N60 G41 G1 X15 Y15
N70 X4 Y4                              ; P4 - contour end point
N80 G40 G248 DISR=20 F700 X8 Y8        ; Retraction, point P0 programmed
N90 M30
```

See the following illustration for approaching along a semi-circle using the example of G42 or retraction using G41 and completion with G40:



Note

Make sure that a positive radius is entered for the tool radius. Otherwise, the directions for G41, G42 will be changed.

Controlling the infeed motion using DISCL and G340, G341

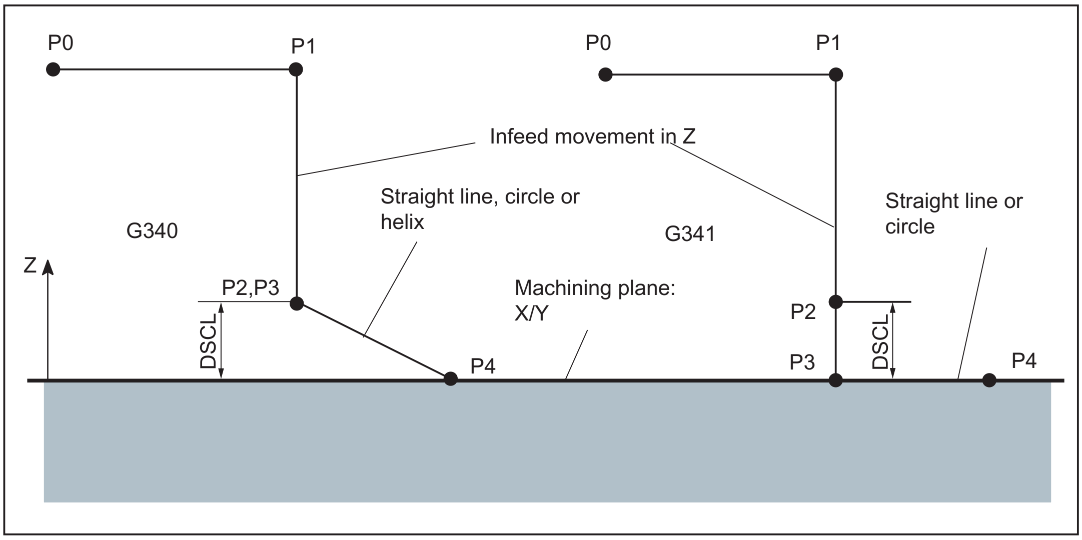
DISCL=... specifies the distance of point P2 from the machining plane (see following figure).

In the case DISCL=0, the following will apply:

- With G340: The whole approach motion consists only of two blocks (P1, P2 and P3 are identical). The approach contour is generated from P3 to P4.
- With G341: The whole approach motion consists only of three blocks (P2 and P3 are identical). If P0 and P4 are located in the same plane, only two blocks will result (there will be no infeed motion from P1 to P3).

It is monitored that the point defined by DISCL lies between P1 and P3, i.e. with all motions that possess a component which runs vertically to the machining plane, this component must have the same sign. If a reversal of the direction is detected, a tolerance of 0.01 mm is permitted.

See the following sequence of the approach motion dependent on G340 / G341 (example with G17):



Programming example: Approach along a semi-circle with infeed

N10 T1 D1 G17 G90 G94	; Activate tool, X/Y plane
N20 G0 X0 Y0 Z30	; Approach P0
N30 G41 G347 G340 DISCL=3 DISR=13 Z=0 F500	
	; Approach along a semi-circle with radius: 13 mm,
	; Safety clearance to the plane: 3 mm
N40 G1 X40 Y-10	
N50 G40 X20 Y20	
N60 M30	

alternatively N30 / N40:

```
N30 G41 G347 G340 DISCL=3 DISR=13 X40 Y-10 Z0 F500
```

or

```
N30 G41 G347 G340 DISCL=3 DISR=13 F500
N40 G1 X40 Y-10 Z0
```

Explanation with regard to N30 / N40:

By using G0 (from N20), the point P1 (starting point of the semi-circle, corrected by the tool radius) is approached in the plane Z=30, then lowering to the depth (P2, P3) with Z=3 (DISCL). The contour is reached at point X40 Y-10 in the depth Z=0 (P4) along a helix curve at a feedrate of 500 mm/min.

Approach and retraction velocities

- Velocity of the previous block (e.g. G0):

  All motions from P0 up to P2 are executed at this speed, i.e. the motion parallel to the machining plane and the part of the infeed motion up to the safety clearance DISCL.

- Programmed feedrate F:

  This feedrate is active from P3 or P2 if FAD is not programmed. If no F word is programmed in the SAR block, the velocity of the previous block will act.

- Programming using FAD:

  Specify the feedrate for

  - G341: Infeed motion vertically to the machining plane from P2 to P3
  - G340: from point P2 or P3 to P4

    If FAD is not programmed, this part of the contour is traversed at the speed which is active modally from the preceding block, in the event that no F command defining the speed is programmed in the SAR block.

- **During retraction**, the roles of the modally effective feedrate from the previous block and the feedrate programmed in the SAR block are changed, i.e. the actual retraction contour is traversed using the old feedrate, and a new velocity programmed using the F word will apply correspondingly from P2 to P0.

Programming example: Approach along a quadrant, infeed using G341 and FAD

```
N10 T1 D1 G17 G90 G94                     ; Activate tool, X/Y plane
N20 G0 X0 Y0 Z30                          ; Approach P0
N30 G41 G341 G247 DISCL=5 DISR=13 FAD=500 X40 Y-10 Z=0 F800
N40 G1 X50
N50 G40 G1 X20 Y20
N60 M30
```

Explanation with regard to N30:

By using G0 (from N20), the point P1 (starting point of the quadrant, corrected by the tool radius) is approached in the plane Z=30, then lowering to the depth (P2) with Z=5 (DISCL). Using a feedrate of FAD=500 mm/min, it is lowered to a depth of Z=0 (P3) (G341). Then, the contour is approached at point X40,Y-10 along a quadrant in the plane (P4) using F=800 mm/min.

Intermediate blocks

A maximum of five blocks **without** moving the geometry axes can be inserted between an SAR block and the next traversing block.

Information

Programming when retracting:

- With an SAR block with a geometry axis programmed, the contour ends at P2. The positions on the axes that constitute the machining plane result from the retraction contour. The axis component perpendicular to this is defined by DISCL. With DISCL=0, the motion will run completely in the plane.
- If in the SAR block only the axis is programmed vertically to the machining plane, the contour will end at P1. The positions of the remaining axes will result, as described above. If the SAR block is also the TRC disable block, an additional path from P1 to P0 is inserted such that no motion results at the end of the contour when disabling the TRC.
- If only one axis on the machining plane is programmed, the missing second axis is modally added from its last position in the previous block.

Cycles 2

2.1 Overview of cycles

Cycles are generally applicable technology subroutines that can be used to carry out a specific machining process, such as drilling of a thread (tapping) or milling of a pocket. These cycles are adapted to individual tasks by parameter assignment.

Drilling cycle, drilling pattern cycles and milling cycles

The following standard cycles can be carried out using the SINUMERIK 808D control system:

- Drilling cycles

  CYCLE81: Drilling, centering

  CYCLE82: Drilling, counterboring

  CYCLE83: Deep-hole drilling

  CYCLE84: Rigid tapping

  CYCLE840: Tapping with compensating chuck

  CYCLE85: Reaming 1

  CYCLE86: Boring

  CYCLE87: Drilling with stop 1

  CYCLE88: Drilling with stop 2

  CYCLE89: Reaming 2

- Drilling pattern cycles

  HOLES1: Row of holes

  HOLES2: Circle of holes

  CYCLE802: Arbitrary positions

- Milling cycles

  CYCLE71: Face milling

  CYCLE72: Contour milling

  CYCLE76: Milling the rectangular spigot

  CYCLE77: Circular spigot milling

  LONGHOLE: Elongated hole

SLOT1: Groove milling pattern on a circle

SLOT2: Circumferential groove milling pattern

POCKET3: Rectangular pocket milling (with any milling tool)

POCKET4: Circular pocket milling (with any milling tool)

CYCLE90: Thread milling

CYCLE832: High speed settings

2.2 Programming cycles

Call and return conditions

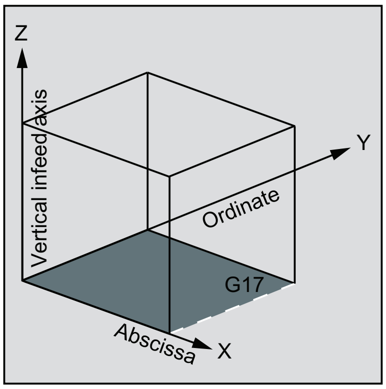
The G functions effective prior to the cycle call and the programmable offsets remain active beyond the cycle.

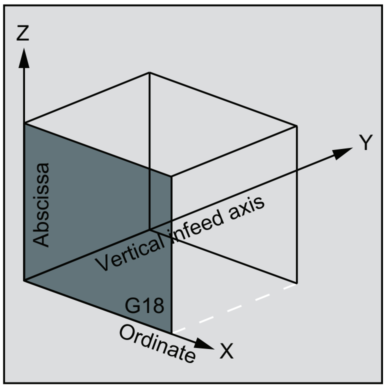
The machining level (G17, G18, G19) must be defined before calling the cycle. A cycle operates in the current plane with:

- First axis of the plane (abscissa)
- Second axis of the plane (ordinate)
- Drilling axis/infeed axis, third axis, standing vertically to the plane (vertical infeed axis)

With drilling cycles, the drilling operation is carried out in the axis standing vertically to the current plane. In milling, the depth infeed is carried out in this axis.

See the following illustrations for plane and axis assignment:





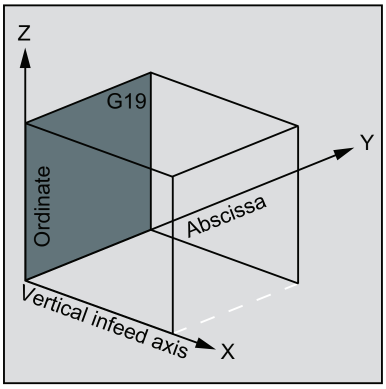


Table 2- 1 Plane and axis assignment

Command	Plane (abscissa/ordinate)	Vertical infeed axis
G17	X/Y	Z
G18	Z/X	Y
G19	Y/Z	X

Messages output during execution of a cycle

During various cycles, messages that refer to the state of machining are displayed on the screen of the control system during program execution.

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

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

Block display during execution of a cycle

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

Cycle call and parameter list

The defining parameters for the cycles can be transferred via the parameter list when the cycle is called.

Note

Cycle calls must always be programmed in a separate block.

Basic instructions with regard to the assignment of standard cycle parameters

Each defining parameter of a cycle has a certain data type. The parameter being used must be specified when the cycle is called. In this parameter list, the following parameters can be transferred:

- R parameters (only numerical values)
- Constants

If R parameters are used in the parameter list, they must first be assigned values in the calling program. Proceed as follows to call the cycles:

- With an incomplete parameter list

  or

- By omitting parameters.

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

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

If when calling the cycle the parameter list contains more entries than parameters are defined in the cycle, the general NC alarm 12340 "Too many parameters" is displayed and the cycle is not executed.

Note

Axis-specific and channel-specific machine data of the spindle must be configured.

Cycle call

The individual methods for writing a cycle are shown in the programming examples provided for the individual cycles.

Simulation of cycles

Programs with cycle calls can be tested first in simulation.

During simulation, the traversing movements of the cycle are visualized on the screen.

2.3 Graphical cycle support in the program editor

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

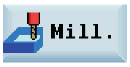
Function

The cycle support consists of three components:

1. Cycle selection
2. Input screens for parameter assignment
3. Help screen for each cycle (is to be found in the input screen).

Operating the cycle support

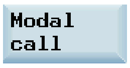
To add a cycle call to the program, proceed as described below:



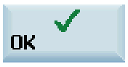
1. Select a cycle type with the corresponding horizontal softkey to open the lower-level vertical softkey bar until the desired input screen form with the help display appears on the screen.
2. Enter the values directly (numerical values) or indirectly (R parameters, for example, R27, or expressions consisting of R parameters, for example, R27 + 10).

   If numerical values are entered, the control system automatically performs a check to see whether the value lies within the permitted range.

3. Use this key to select values for some parameters that may have only a few values for selection.



4. For drilling cycles, it is also possible to call a cycle modally with this key. To deselect the modal call, press the softkey below:



5. Press this softkey to confirm your input. To cancel the input, press the softkey below:

Recompiling

Recompiling of program codes serves to make modifications to an existing program using the cycle support.

Position the cursor on the line to be modified and press this softkey. This reopens the input screen from which the program piece has been created, and you can modify and accept the values.

2.4 Drilling cycles

2.4.1 General information

Drilling cycles are motional sequences specified according to DIN 66025 for drilling, boring, tapping, etc.

They are called in the form of a subroutine with a defined name and a parameter list.

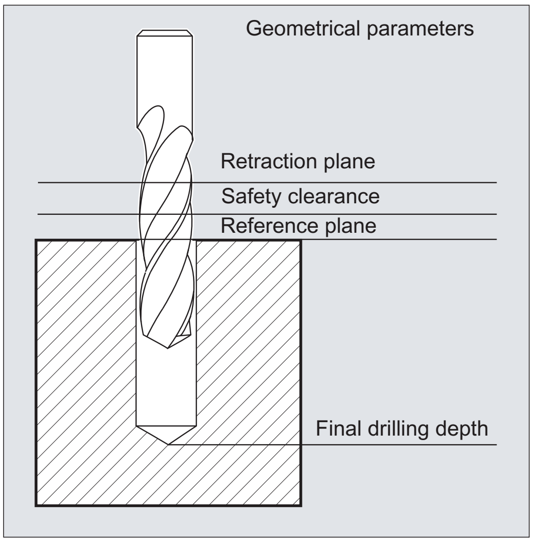
The drilling cycles can be modal, that is, they are executed at the end of each block containing motion commands. Further cycles created by the user can also be called modally.

There are two types of parameters:

- Geometrical parameters
- Machining parameters

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

See the following illustration for drilling, centering - CYCLE81:



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

2.4.2 Requirements

Call and return conditions

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

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

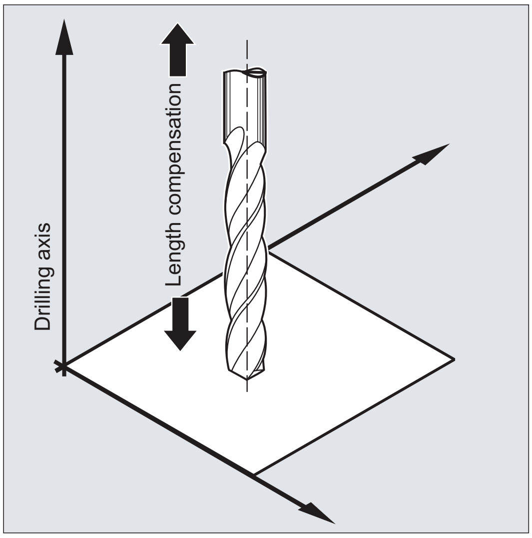
The G functions and the current data record active before the cycle was called remain active beyond the cycle.

Plane definition

In the case of drilling cycles, it is generally assumed that the current workpiece coordinate system in which the machining operation is to be performed is to be defined by selecting plane G17, G18 or G19 and activating a programmable offset. The drilling axis is always the axis of this coordinate system which stands vertically to the current plane.

A tool length compensation must be selected before the cycle is called. Its effect is always perpendicular to the selected plane and remains active even after the end of the cycle.

See the following illustration for length compensation:



Dwell time programming

The parameters for dwell times in the drilling cycles are always assigned to the F word and must therefore be assigned with values in seconds. Any deviations from this procedure must be expressly stated.

2.4.3 Drilling, centering - CYCLE81

Programming

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

Parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Final drilling depth (absolute)
DPR	REAL	Final drilling depth relative to the reference plane (enter without sign)

Function

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

Sequence

Position reached prior to cycle start:

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

The cycle creates the following sequence of motions:

Approach of the reference plane brought forward by the safety clearance by using G0

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

Explanation of the parameters

RFP and RTP (reference plane and retraction plane)

Normally, reference plane (RFP) and retraction plane (RTP) have different values. The cycle assumes that the retraction plane precedes the reference plane. This means that the distance from the retraction plane to the final drilling depth is larger than the distance from the reference plane to the final drilling depth.

SDIS (safety clearance)

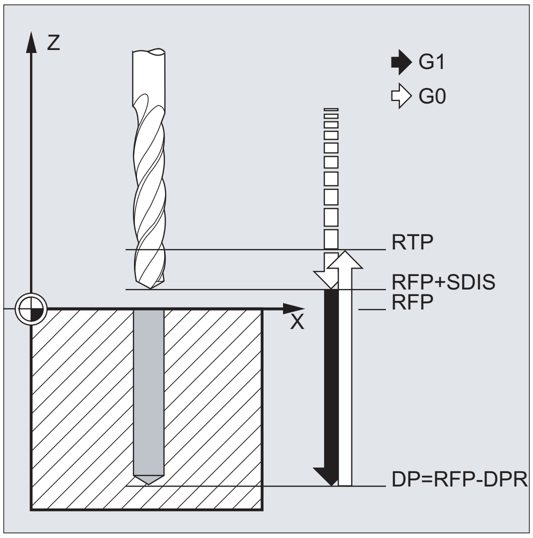
The safety clearance (SDIS) acts with reference to the reference plane. This is brought forward by the safety clearance.

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

DP and DPR (final drilling depth)

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

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



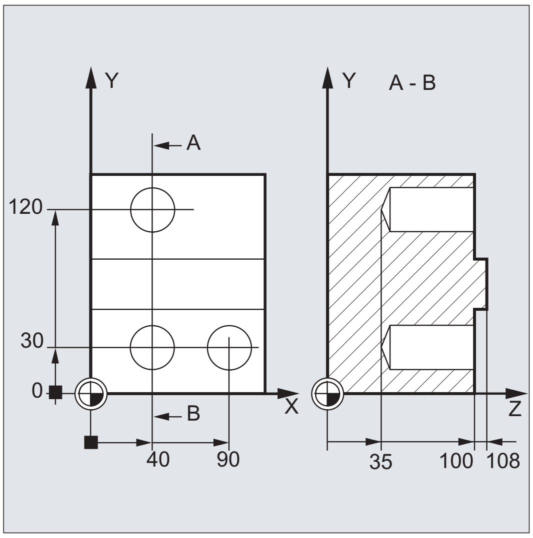
Note

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

If the values for reference and retraction planes are identical, a relative depth specification is not permitted. The error message 61101 "Reference plane defined incorrectly" is output and the cycle is not executed. This error message is also output if the retraction plane is located after the reference plane, i.e. its distance to the final drilling depth is smaller.

Programming example: Drilling_centering

This program produces three drill holes using the CYCLE81 drilling cycle. The drilling axis is always the Z axis.



`N10 G0 G17 G90 F200 S300 M3`	; Specification of technology values
`N20 D3 T3 Z110`	; Approach retraction plane
`N30 X40 Y120`	; Approach of the first drilling position
`N40 CYCLE81(110, 100, 2, 35,)`	; Cycle call with absolute final drilling depth, safety clearance and incomplete parameter list
`N50 Y30`	; Approach next drilling position
`N60 CYCLE81(110, 102, , 35,)`	; Cycle call without safety clearance
`N70 G0 G90 F180 S300 M03`	; Specification of technology values
`N80 X90`	; Approach next position
`N90 CYCLE81(110, 100, 2, , 65,)`	; Cycle call with relative final drilling depth and safety clearance
`N100 M02`	; End of program

2.4.4 Drilling, counterboring - CYCLE82

Programming

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

Parameters

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

Function

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

Sequence

Position reached prior to cycle start:

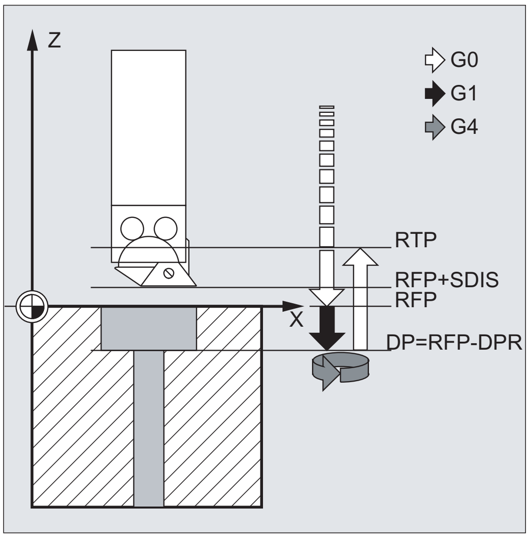
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the final drilling depth with the feedrate (G1) programmed prior to the cycle call
- Dwell time at final drilling depth
- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".



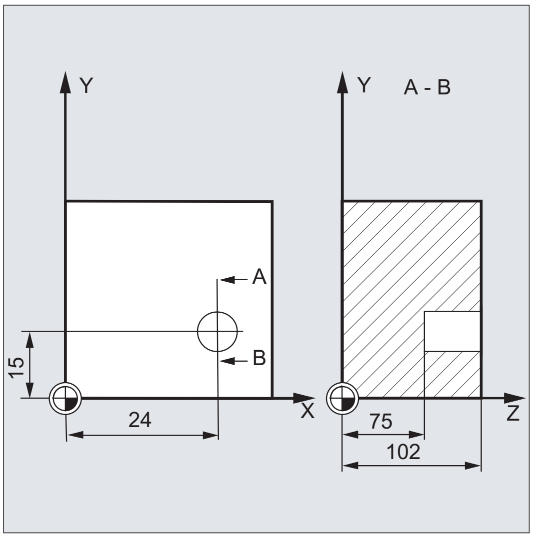
DTB (dwell time)

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

Programming example1: Drilling_counterboring

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

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



```
N10 G0 G17 G90 F200 S300 M3              ; Specification of technology
                                         values
N20 D1 T10 Z110                          ; Approach retraction plane
N30 X24 Y15                              ; Approach drilling position
N40 CYCLE82 (110, 102, 4, 75, , 2)       ; Cycle call with absolute final
                                         drilling depth and safety
                                         clearance
N50 M02                                  ; End of program
```

Programming example2: Drilling_counterboring

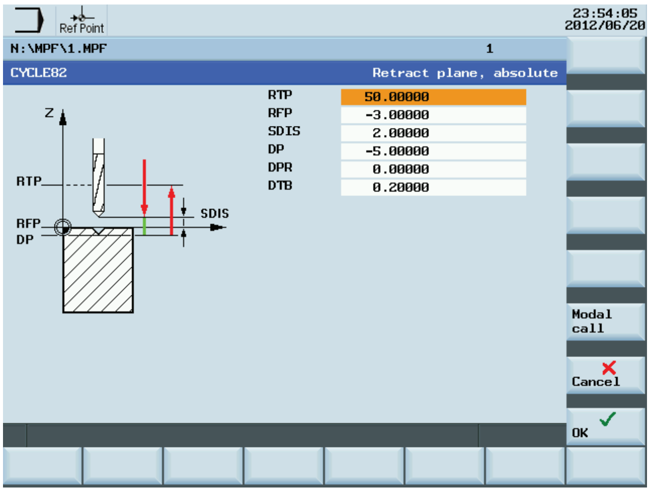
Proceed through the following steps:

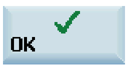
1. Select the desired operating area.

2. Open the vertical softkey bar for available drilling cycles.

3. Press this softkey from the vertical softkey bar.

4. Press this softkey to open the window for CYCLE82. Parameterize the cycle as desired.





5. Confirm your settings with this softkey. The cycle is then automatically transferred to the program editor as a separate block.

2.4.5 Deep-hole drilling - CYCLE83

Programming

CYCLE83 (RTP, RFP, SDIS, DP, DPR, FDEP, FDPR, DAM, DTB, DTS, FRF, VARI, AXN, MDEP, VRT, DTD, DIS1)

Parameters

Parameter	Data type	Description	
RTP	REAL	Retraction plane (absolute)	
RFP	REAL	Reference plane (absolute)	
SDIS	REAL	Safety clearance (enter without sign)	
DP	REAL	Final drilling depth (absolute)	
DPR	REAL	Final drilling depth relative to the reference plane (enter without sign)	
FDEP	REAL	First drilling depth (absolute)	
FDPR	REAL	First drilling depth relative to the reference plane (enter without sign)	
DAM	REAL	Amount of degression (enter without sign)	
		Values:	>0: degression as value <0: degression factor =0: no degression
DTB	REAL	Dwell time at drilling depth (chip breakage)	
		Values:	>0: in seconds <0: in revolutions
DTS	REAL	Dwell time at starting point and for chip removal	
		Values:	>0: in seconds <0: in revolutions
FRF	REAL	Feedrate factor for the first drilling depth (enter without sign) Range of values: 0.001 ... 1	
VARI	INT	Machining type: Chip breakage=0, Chip removal=1	
AXN	INT	Tool axis	
		Values:	1: 1st geometrical axis 2: 2nd geometrical axis 3: 3rd geometrical axis
MDEP	REAL	Minimum drilling depth (only in connection with degression factor)	
VRT	REAL	Variable retraction value for chip breakage (VARI=0)	
		Values:	>0: if traction value =0: retraction value 1mm set

Parameter	Data type	Description	
DTD	REAL	Dwell time at final drilling depth	
		Values:	>0: in seconds <0: in revolutions =0: value same as DTB
DIS1	REAL	Programmable limit distance for reinsertion in the drill hole (for chip removal VARI=1)	
		Values:	>0: programmable value applies =0: automatic calculation

Function

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

Deep hole drilling is performed with a depth infeed of a maximum definable depth executed several times, increasing gradually until the final drilling depth is reached.

The drill can either be retracted to the reference plane + safety clearance after every infeed depth for swarf removal or retracted in each case by 1 mm for chip breaking.

Sequence

Position reached prior to cycle start:

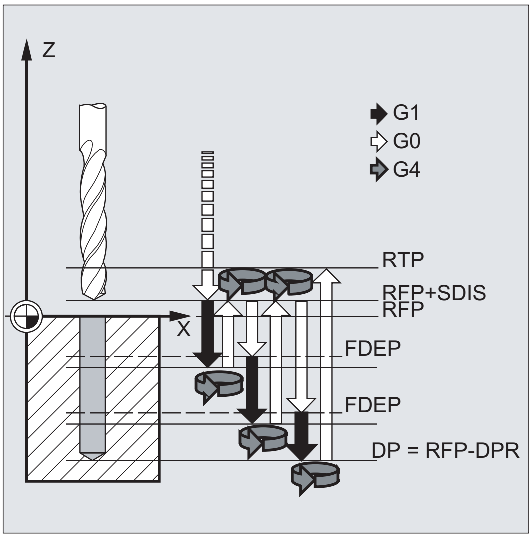
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence:

Deep hole drilling with chip removal (VARI=1)

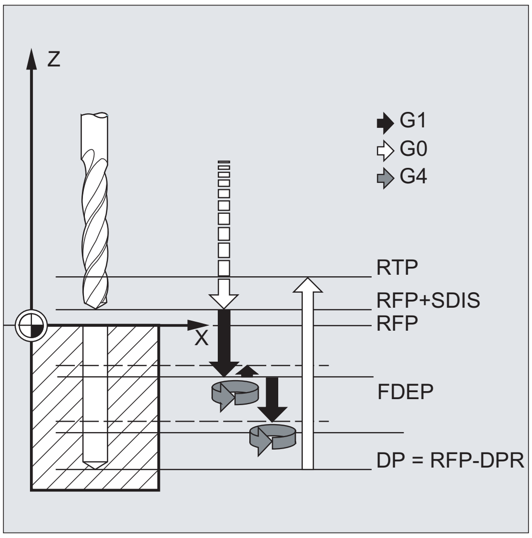
- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the first drilling depth with G1, the feedrate for which is derived from the feedrate defined with the program call which is subject to parameter FRF (feedrate factor)
- Dwell time at final drilling depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance for swarf removal by using G0
- Dwell time at the starting point (parameter DTS)
- Approach of the drilling depth last reached, reduced by anticipation distance by using G0
- Traversing to the next drilling depth with G1 (sequence of motions is continued until the final drilling depth is reached)
- Retraction to the retraction plane with G0

See the following illustration for parameters for CYCLE83:



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

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the first drilling depth with G1, the feedrate for which is derived from the feedrate defined with the program call which is subject to parameter FRF (feedrate factor)
- Dwell time at final drilling depth (parameter DTB)
- Retraction by 1 mm from the current drilling depth with G1 and the feedrate programmed in the calling program (for chip breaking)
- Traversing to the next drilling depth with G1 and the programmed feedrate (sequence of motions is continued until the final drilling depth is reached)
- Retraction to the retraction plane with G0



Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".

Interrelation of the DP (or DPR), FDEP (or FDPR) and DAM parameters

The intermediate drilling depth is calculated in the cycle on the basis of final drilling depth, first drilling depth and amount of degression as follows:

- In the first step, the depth parameterized with the first drilling depth is traversed as long as it does not exceed the total drilling depth
- From the second drilling depth on, the drilling stroke is obtained by subtracting the amount of degression from the stroke of the last drilling depth, provided that the latter is greater than the programmed amount of degression
- The next drilling strokes correspond to the amount of degression, as long as the remaining depth is greater than twice the amount of degression
- The last two drilling strokes are divided and traversed equally and are therefore always greater than half of the amount of degression
- If the value for the first drilling depth is incompatible with the total depth, the error message 61107 "First drilling depth defined incorrectly" is output and the cycle is not executed

The FDPR parameter has the same effect in the cycle as the DPR parameter. If the values for the reference and retraction planes are identical, the first drilling depth can be defined as a relative value.

If the first drilling depth is programmed larger than the final drilling depth, the final drilling depth is never exceeded. The cycle will reduce the first drilling depth automatically as far as the final drilling depth is reached when drilling only once, and will therefore drill only once.

DTB (dwell time)

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

DTS (dwell time)

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

FRF (feedrate factor)

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

VARI (machining type)

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

Note

The anticipation distance is calculated internally in the cycle as follows:

- If the drilling depth is 30 mm, the value of the anticipation distance is always 0.6 mm.
- For larger drilling depths, the formula drilling depth / 50 is used (maximum value 7 mm).

AXN (tool axis)

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

The identifiers have the following meanings:

`AXN=1`	First axis of the current plane
`AXN=2`	Second axis of the current plane
`AXN=3`	Third axis of the current plane

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

```
G18

AXN=1
```

MDEP (minimum drilling depth)

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

VRT (variable retraction value for chip breakage with VARI=0)

You can program the retraction path for chip breaking.

DTD (dwell time at final drilling depth)

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

DIS1 (programmable limit distance for VARI=1)

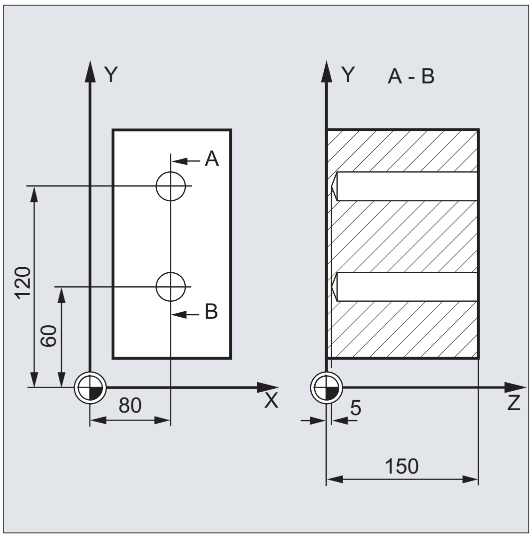
The limit distance after re-insertion in the hole can be programmed.

The limit distance is calculated within the cycle as follows:

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

Programming example1: Deep-hole drilling

This program executes the cycle CYCLE83 at the positions X80 Y120 and X80 Y60 in the XY plane. The first drill hole is drilled with a dwell time zero and machining type chip breaking. The final drilling depth and the first drilling depth are entered as absolute values. In the second cycle call, a dwell time of 1 s is programmed. Machining type chip removal is selected, the final drilling depth is relative to the reference plane. The drilling axis in both cases is the Z axis.



N10 G0 G17 G90 F50 S500 M4	; Specification of technology values
N20 D1 T12	; Approach retraction plane
N30 Z155	
N40 X80 Y120	; Approach first drilling position
N50 CYCLE83(20,0,3,-15,,-6,,1,1,1,1,0,3,4,3,1,2)	; Call of cycle; depth parameters with absolute values
N60 X80 Y60	; Approach next drilling position
N70 CYCLE83(20,0,3,-15,,-6,,1,1,1,1,0,3,4,3,1,2)	; Cycle call with relative data for final drilling depth and first drilling depth; the safety clearance is 1 mm and the feedrate factor is 0.5
N80 M02	; End of program

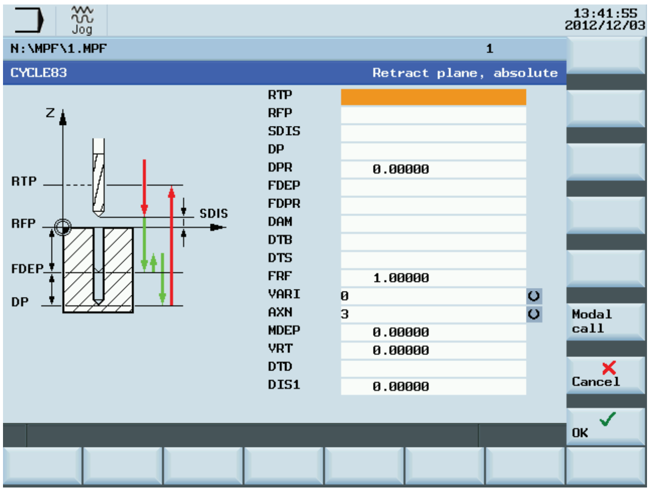
Programming example 2: Deep-hole drilling

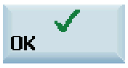
Proceed through the following steps:

1. Select the desired operating area.

2. Open the vertical softkey bar for available drilling cycles.

3. Press this softkey to open the window for CYCLE83. Parameterize the cycle as desired.





4. Confirm your settings with this softkey. The cycle is then automatically transferred to the program editor as a separate block.

2.4.6 Rigid tapping - CYCLE84

Programming

CYCLE84 (RTP, RFP, SDIS, DP, DPR, DTB, SDAC, MPIT, PIT, POSS, SST, SST1, AXN, 0, 0, VARI, DAM, VRT)

Parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Final drilling depth (absolute)
DPR	REAL	Final drilling depth relative to the reference plane (enter without sign)
DTB	REAL	Dwell time at thread depth (chip breakage)
SDAC	INT	Direction of rotation after end of cycle Values: 3, 4 or 5 (for M3, M4 or M5)
MPIT	REAL	Thread lead as a thread size (signed): Range of values 3 (for M3) to 48 (for M48); the sign determines the direction of rotation in the thread
PIT	REAL	Thread lead as a value (signed) Range of values: 0.001 ... 2000.000 mm); the sign determines the direction of rotation in the thread
POSS	REAL	Spindle position for oriented spindle stop in the cycle (in degrees)
SST	REAL	Speed for tapping
SST1	REAL	Speed for retraction
AXN	INT	Tool axis Values [1]: 1: 1st axis of the current plane 2: 2nd axis of the current plane 3: 3rd axis of the current plane
PSYS	INT	Internal parameter; only the default value 0 is possible
PSYS	INT	Internal parameter; only the default value 0 is possible
VARI	INT	Machining type Values: 0: Tapping in one pass 1: Deep-hole tapping with chip breakage 2: Deep-hole tapping with chip removal
DAM	REAL	Incremental drilling depth value range: 0 <= Max. value
VRT	REAL	Variable retraction value for chip breakage value range: 0 <= Max. value

[1] The definition of the 1st, 2nd, and 3rd axes depends upon the current plane selected.

Function

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

CYCLE84 can be used to make tapped holes without compensating chuck. For tapping with compensating chuck, a separate cycle CYCLE840 is provided.

Note

CYCLE84 can be used if the spindle to be used for the boring operation is technically able to be operated in the position-controlled spindle operation.

Sequence

Position reached prior to cycle start:

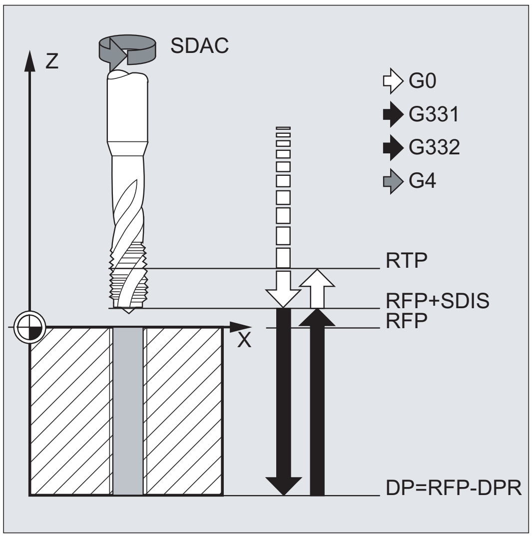
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Oriented spindle stop (value in the parameter POSS) and switching the spindle to axis mode
- Tapping to final drilling depth and speed SST
- Dwell time at thread depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance, speed SST1 and direction reversal
- Retraction to the retraction plane with G0; spindle mode is reinitiated by reprogramming the spindle speed active before the cycle was called and the direction of rotation programmed under SDAC

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".



DTB (dwell time)

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

SDAC (direction of rotation after end of cycle)

Under SDAC, the direction of rotation after end of cycle is programmed.

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

MPIT and PIT (thread lead as a thread size and as a value)

The value for the thread lead can be defined either as the thread size (for metric threads between M3 and M48 only) or as a value (distance from one thread turn to the next as a numerical value). Any parameters not required are omitted in the call or assigned the value zero.

RH or LH threads are defined by the sign of the lead parameters:

- Positive value → right (same as M3)
- Negative value → left (same as M4)

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

POSS (spindle position)

Before tapping, the spindle is stopped with orientation in the cycle by using the command SPOS and switched to position control.

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

SST (speed)

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

SST1 (retraction speed)

The speed for retraction from the tapped hole is programmed under SST1.

If this parameter is assigned the value zero, retraction is carried out at the speed programmed under SST.

AXN (tool axis)

The identifiers have the following meanings:

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

For example, to machine a center hole (in Z) in the G17 plane, you program:

```
G17

AXN=3
```

Deep-hole tapping: VARI, DAM, VRT

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

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

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

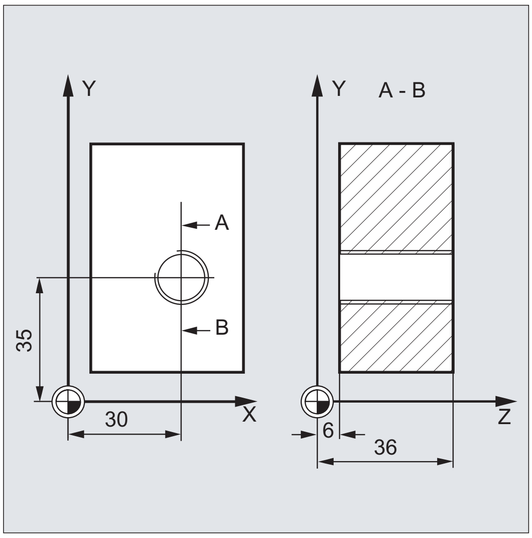
- The programmed incremental drilling depth is executed in each step until the rest up to the final drilling depth is less than (<) 2 x DAM
- The remaining drilling depth is halved and executed in two steps. Thus, the minimum drilling depth is not smaller than DAM / 2.

Note

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

Programming example1: Rigid tapping

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



`N10 G0 G90 T11 D1`	; Specification of technology values
`N20 G17 X30 Y35 Z40`	; Approach drilling position
`N30 CYCLE84(20,0,3,-15,,1,3,6,,0,500,500,3,0,0,0,5,0)`	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 M02`	; End of program

Programming example 2: Rigid tapping

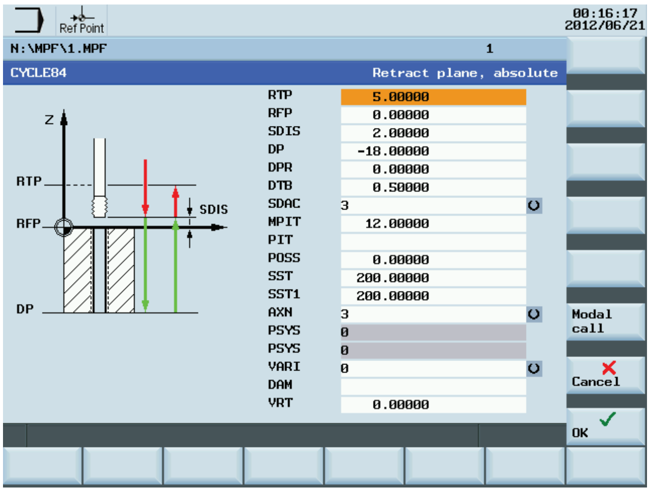
Proceed through the following steps:

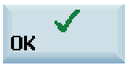
1. Select the desired operating area.

2. Open the vertical softkey bar for available drilling cycles.

3. Press this softkey from the vertical softkey bar.

4. Press this softkey to open the window for CYCLE84. Parameterize the cycle as desired.





5. Confirm your settings with this softkey. The cycle is then automatically transferred to the program editor as a separate block.

2.4.7 Tapping with compensating chuck - CYCLE840

Programming

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

Parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Final drilling depth (absolute)
DPR	REAL	Final drilling depth relative to the reference plane (enter without sign)
DTB	REAL	Dwell time at thread depth (chip breakage)
SDR	INT	Direction of rotation for retraction Values: 0 (automatic direction reversal), 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, 1 = without encoder
MPIT	REAL	Thread lead as a thread size (signed): Range of values 3 (for M3) to 48 (for M48)
PST	REAL	Thread lead as a value (signed) Range of values: 0.001 ... 2000.000 mm
AXN	INT	Tool axis
		Values[1]: 1: 1st axis of the current plane 2: 2nd axis of the current plane 3: 3rd axis of the current plane

[1] The definition of the 1st, 2nd, and 3rd axes depends upon the current plane selected.

Function

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

This cycle is used to program tapping with the compensating chuck:

- Without encoder
- With encoder.

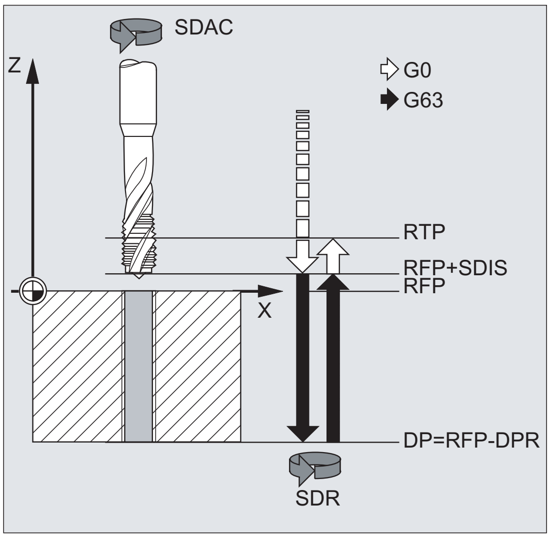
Sequence

Tapping with compensating chuck without encoder

Position reached prior to cycle start:

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

The cycle creates the following sequence of motions:



- Approach of the reference plane brought forward by the safety clearance by using G0
- Tapping to the final drilling depth
- Dwell time at tapping depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance
- Retraction to the retraction plane with G0

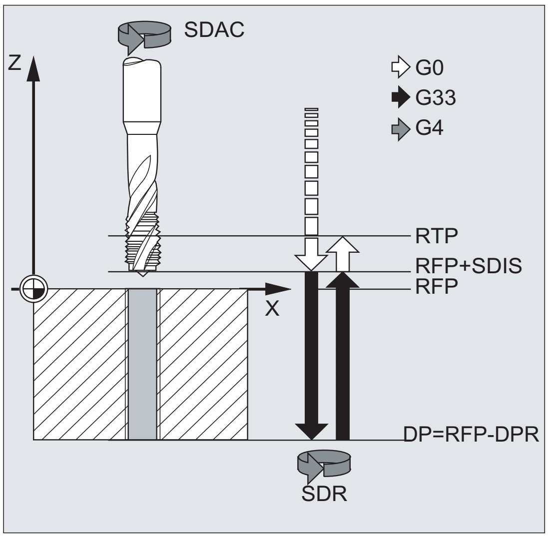
Sequence of operations

Tapping with compensating chuck with encoder

Position reached prior to cycle start:

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

The cycle creates the following sequence of motions:



- Approach of the reference plane brought forward by the safety clearance by using G0
- Tapping to the final drilling depth
- Dwell time at thread depth (parameter DTB)
- Retraction to the reference plane brought forward by the safety clearance
- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".

DTB (dwell time)

The dwell time must be programmed in seconds.

SDR (direction of rotation for retraction)

SDR=0 must be set if the spindle direction is to reverse automatically.

If the machine data is defined such that no encoder is set (in this case, machine data MD30200 $MA_NUM_ENCS is 0), the parameter must be assigned the value 3 or 4 for the direction of rotation; otherwise, alarm 61202 "No spindle direction programmed" is output and the cycle is aborted.

SDAC (direction of rotation)

Because the cycle can also be called modally (see Section "Graphical cycle support in the program editor (Page 118)"), it requires a direction of rotation for tapping further threaded holes. This is programmed in parameter SDAC and corresponds to the direction of rotation programmed before the first call in the higher-level program. If SDR=0, the value assigned to SDAC has no meaning in the cycle and can be omitted in the parameterization.

ENC (tapping)

If tapping is to be performed without encoder although an encoder exists, parameter ENC must be assigned value 1.

If, however, no encoder is installed and the parameter is assigned the value 0, it is ignored in the cycle.

MPIT and PIT (thread lead as a thread size and as a value)

The parameter for the lead is only relevant if tapping is performed with encoder. The cycle calculates the feedrate from the spindle speed and the lead.

The value for the thread lead can be defined either as the thread size (for metric threads between M3 and M48 only) or as a value (distance from one thread turn to the next as a numerical value). Any parameters not required are omitted in the call or assigned the value zero.

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

Note

Depending on the settings in machine data MD30200 $MA_NUM_ENCS, the cycle selects whether tapping is to be performed with or without encoder.

The direction of rotation for the spindle must be programmed with M3 or M4.

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

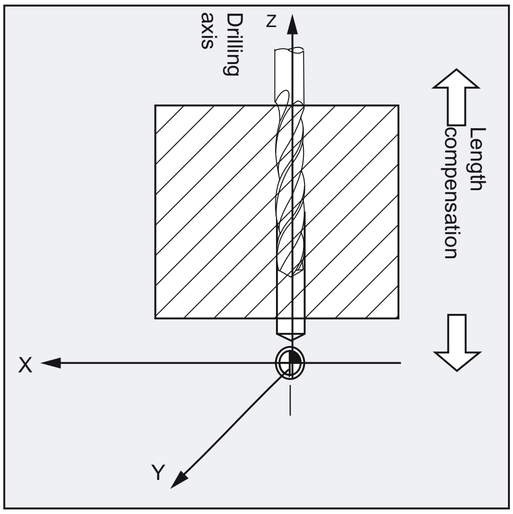
A longer compensating chuck is usually required for tapping without encoder.

AXN (tool axis)

The following figure presents the options for the drilling axes to be selected.

With G17:

- AXN=1; Corresponds to X
- AXN=2; Corresponds to Y
- AXN=3; Corresponds to Z



Using AXN (number of the drilling axis) to program the drilling axis enables the drilling axis to be directly programmed.

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

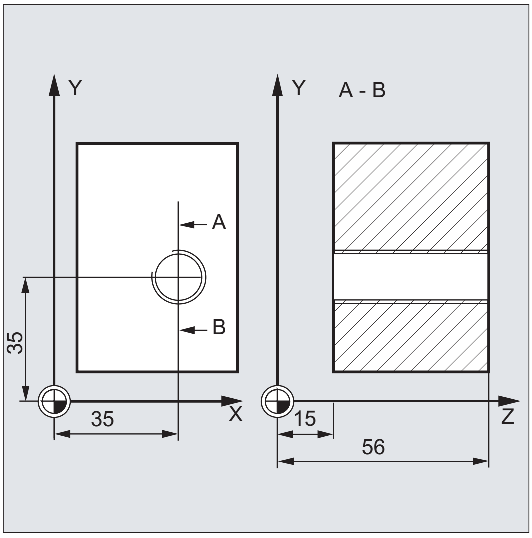
For example, to machine a hole in the G17 plane with Z axis, you program:

G17

AXN=3

Programming example: Tapping without encoder

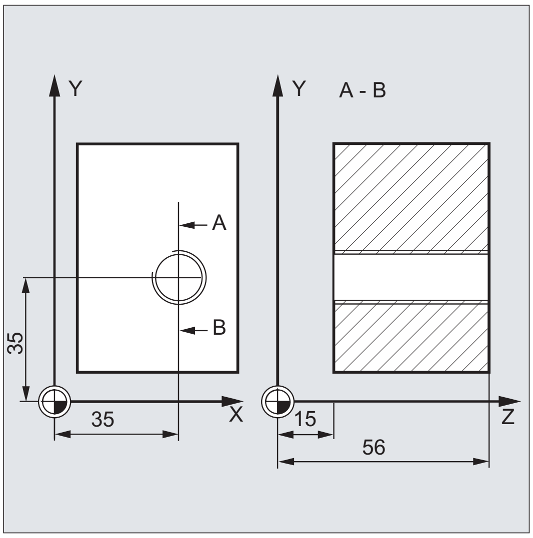
In this program, a thread is tapped without encoder at position X35 Y35 in the XY plane; the tapping axis is the Z axis. The parameters SDR and SDAC for the direction of rotation must be assigned; parameter ENC is assigned the value 1, the value for the depth is the absolute value. Lead parameter PIT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




```
N10 G90 G0 T11 D1 S500 M3                ; Specification of technology
                                         values
N20 G17 X35 Y35 Z60                      ; Approach drilling position
N30 G1 F200                              ; Setting the path feedrate
N40 CYCLE840(20,0,3,-15,,1,3,4,1,6,,3)   Cycle call, dwell time 1 s,
                                         direction of rotation for
                                         retraction M4, direction of
                                         rotation after cycle M3, no
                                         safety clearance, parameters MPIT
                                         and PIT have been omitted
N50 M02                                  ; End of program
```

Programming example: Tapping with encoder

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



`N10 G90 G0 T11 D1 S500 M4`	`; Specification of technology values`
`N20 G17 X35 Y35 Z60`	`; Approach drilling position`
`N30 CYCLE840(20,0,3,-15,,1,3,4,1,6,,3)`	`; Cycle call, without safety clearance, with absolute depth specification`
`N40 M02`	`; End of program`

2.4.8 Reaming 1 - CYCLE85

Programming

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

Parameters

Table 2- 2 CYCLE85 parameters

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

Function

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

The inward and outward movement is performed at the feedrate assigned to FFR and RFF respectively.

Sequence

Position reached prior to cycle start:

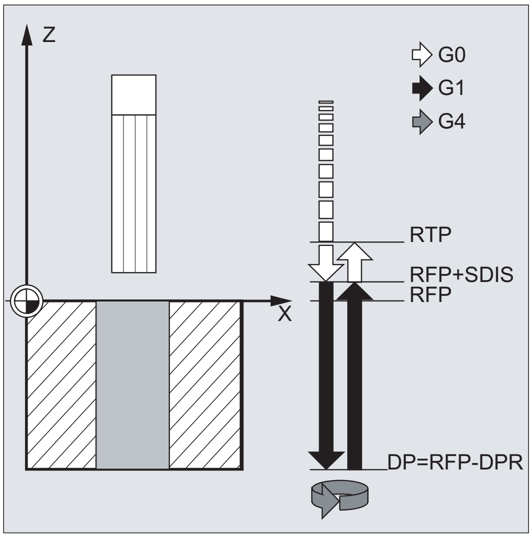
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to the final drilling depth with G1 and at the feedrate programmed under the parameter FFR
- Dwell time at final drilling depth
- Retraction to the reference plane brought forward by the safety clearance with G1 and the retraction feedrate defined under the parameter RFF
- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".



DTB (dwell time)

The dwell time to the final drilling depth is programmed under DTB in seconds.

FFR (feedrate)

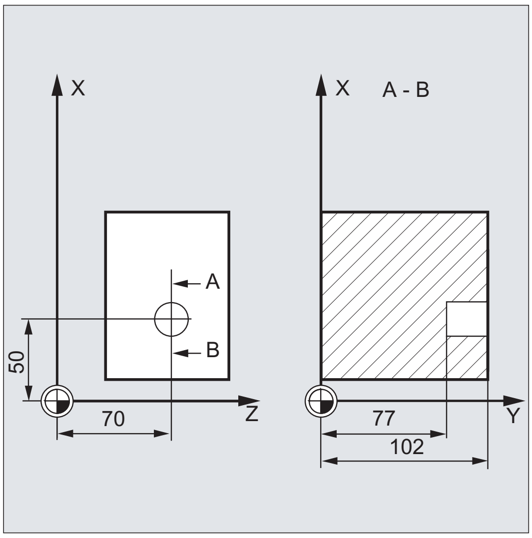
The feedrate value programmed under FFR is active in drilling.

RFF (retraction feedrate)

The feedrate value programmed under RFF is active when retracting from the hole to the reference plane + safety clearance.

Programming example: First drilling

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




```
N10 T11 D1
G1 F200 M3 S200
N20 G18 Z70 X50 Y105                    ; Approach drilling position
N30 CYCLE85(105, 102, 2, , 25, , 300, 450)   ; Cycle call, no dwell time
                                        programmed
N40 M02                                 ; End of program
```

2.4.9 Boring - CYCLE86

Programming

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

Parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Final drilling depth (absolute)
DPR	REAL	Final drilling depth relative to the reference plane (enter without sign)
DTB	REAL	Dwell time at final drilling depth (chip breakage)
SDIR	INT	Direction of rotation Values: 3 (for M3), 4 (for M4)
RPA	REAL	Retraction path along the first axis of the plane (incremental, enter with sign)
RPO	REAL	Retraction path along the second axis of the plane (incremental, enter with sign)
RPAP	REAL	Retraction path along the drilling axis (incremental, enter with sign)
POSS	REAL	Spindle position for oriented spindle stop in the cycle (in degrees)

Function

The cycle supports the boring of holes with a boring bar.

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

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

Sequence

Position reached prior to cycle start:

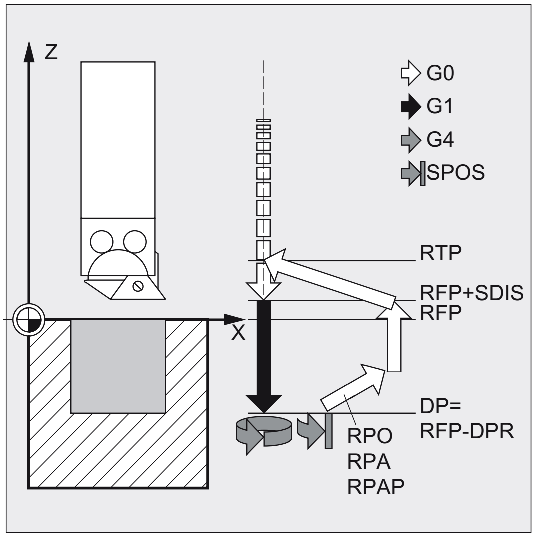
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time to final drilling depth
- Oriented spindle stop at the spindle position programmed under POSS
- Traverse retraction path in up to three axes with G0
- Retraction in the drilling axis to the reference plane brought forward by the safety distance by using G0
- Retraction to the retraction plane with G0 (initial drilling position in both axes of the plane)

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".



DTB (dwell time)

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

SDIR (direction of rotation)

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

RPA (retraction path along the first axis)

Use this parameter to define a retraction movement along the first axis (abscissa), which is executed after the final drilling depth has been reached and oriented spindle stop has been performed.

RPO (retraction path along the second axis)

Use this parameter to define a retraction movement along the second axis (ordinate), which is executed after the final drilling depth has been reached and oriented spindle stop has been performed.

RPAP (retraction path along the drilling axis)

You use this parameter to define a retraction movement along the drilling axis, which is executed after the final drilling axis has been reached and oriented spindle stop has been performed.

POSS (spindle position)

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

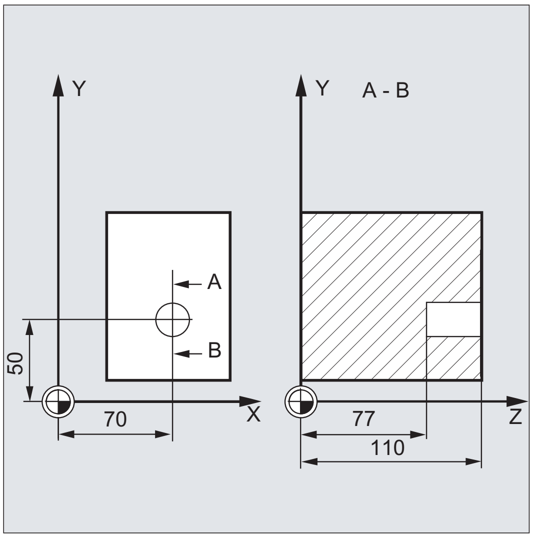
Note

It is possible to stop the active spindle with orientation. The angular value is programmed using a transfer parameter.

CYCLE86 can be used if the spindle to be used for the drilling operation is technically able to execute the SPOS command.

Programming example: Second drilling

CYCLE86 is called at position X70 Y50 in the XY plane. The drilling axis is the Z axis. The final drilling depth is programmed as an absolute value; no safety clearance is specified. The dwell time at the final drilling depth is 2 sec. The top edge of the workpiece is positioned at Z110. In the cycle, the spindle is to rotate with M3 and to stop at 45 degrees.



```
N10 G0 G17 G90 F200 S300 M3                   ; Specification of technology
                                              values
N20 T11 D1 Z112                               ; Approach retraction plane
N30 X70 Y50                                   ; Approach drilling position
N40 CYCLE86(112, 110, , 77, 0, 2, 3, -1, -1, 1, ; Cycle call with absolute
45)                                           drilling depth
N50 M02                                       ; End of program
```

2.4.10 Boring with stop 1 - CYCLE87

Programming

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

Parameters

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

Function

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

During drilling 3, a spindle stop without orientation M5 is generated after reaching the final drilling depth, followed by a programmed stop M0. Pressing the following key continues the retraction movement at rapid traverse until the retraction plane is reached:

Sequence

Position reached prior to cycle start:

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

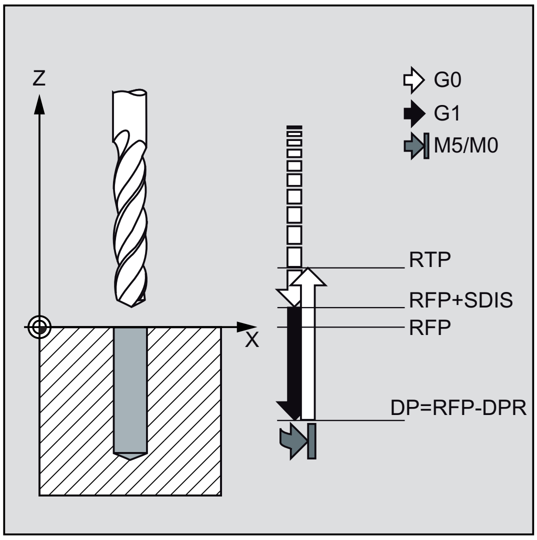
The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Spindle stop with M5
- Press the following key:

- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".



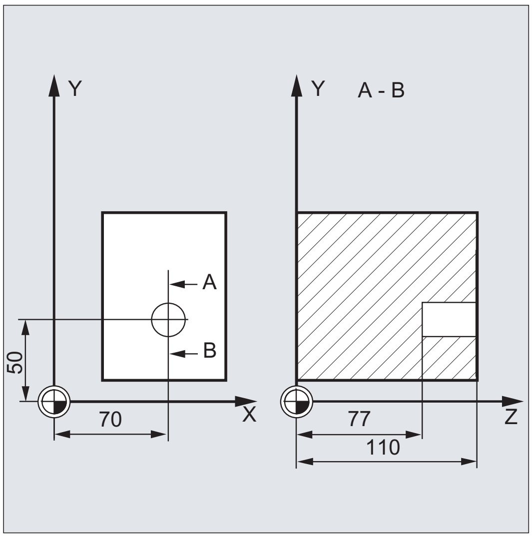
SDIR (direction of rotation)

This parameter determines the direction of rotation with which the drilling operation is carried out in the cycle.

If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is generated and the cycle is aborted.

Programming example: Third drilling

CYCLE87 is called at position X70 Y50 in the XY plane. The drilling axis is the Z axis. The final drilling depth is specified as an absolute value. The safety clearance is 2 mm.




```
DEF REAL DP, SDIS                        ; Definition of parameters
N10 DP=77 SDIS=2                         ; Value assignments
N20 G0 G17 G90 F200 S300                 ; Specification of technology
                                         values
N30 D3 T3 Z113                           ; Approach retraction plane
N40 X70 Y50                              ; Approach drilling position
N50 CYCLE87 (113, 110, 2, -10, , 3)      ; Cycle call with programmed
                                         direction of rotation of spindle
                                         M3
N60 M02                                  ; End of program
```

2.4.11 Drilling with stop 2 - CYCLE88

Programming

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

Parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Final drilling depth (absolute)
DPR	REAL	Final drilling depth relative to the reference plane (enter without sign)
DTB	REAL	Dwell time at final drilling depth (chip breakage)
SDIR	INT	Direction of rotation Values: 3 (for M3), 4 (for M4)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. When drilling with stop, a spindle stop without orientation M5 and a programmed stop M0 are generated when the final drilling depth is reached. Pressing the following key traverses the outward movement at rapid traverse until the retraction plane is reached:

Sequence

Position reached prior to cycle start:

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

The cycle creates the following sequence of motions:

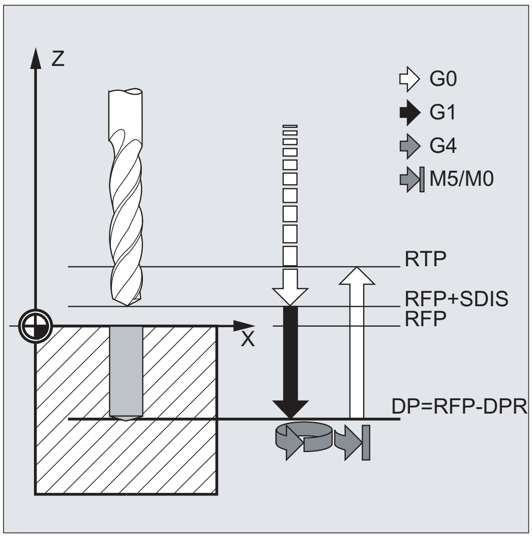
- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time at final drilling depth
- Spindle and program stop with M5 M0. After program stop, press the following key:

- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".



DTB (dwell time)

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

SDIR (direction of rotation)

The programmed direction of rotation is active for the distance to be traversed to the final drilling depth.

If values other than 3 or 4 (M3/M4) are generated, alarm 61102 "No spindle direction programmed" is generated and the cycle is aborted.

Programming example: Fourth drilling

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

M4 is active in the cycle.

```
N10 G17 G90 F100 S450                  ; Specification of technology values
N20 G0 X80 Y90 Z105                    ; Approach drilling position
N30 CYCLE88 (105, 102, 3, , 72, 3, 4)  ; Cycle call with programmed spindle direction M4
N40 M02                                ; End of program
```

2.4.12 Reaming 2 - CYCLE89

Programming

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

Parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Final drilling depth (absolute)
DPR	REAL	Final drilling depth relative to the reference plane (enter without sign)
DTB	REAL	Dwell time at final drilling depth (chip breakage)

Function

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

Sequence

Position reached prior to cycle start:

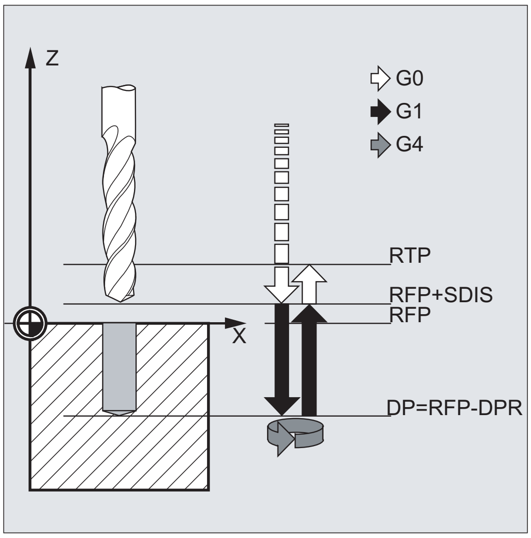
The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

- Approach of the reference plane brought forward by the safety clearance by using G0
- Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call
- Dwell time to final drilling depth
- Retraction up to the reference plane brought forward by the safety clearance using G1 and the same feedrate value
- Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".

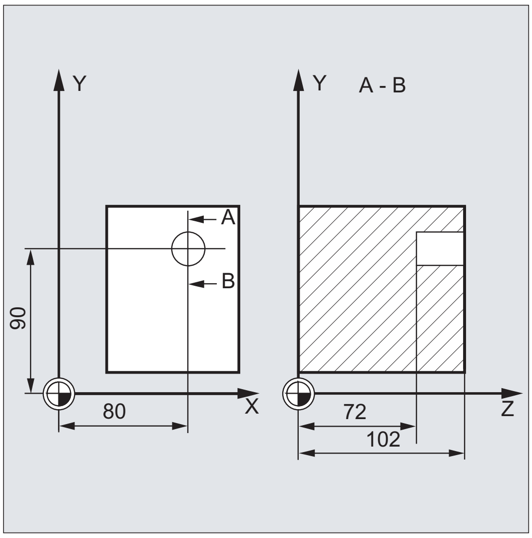


DTB (dwell time)

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

Programming example: Fifth drilling

At X80 Y90 in the XY plane, the drilling cycle CYCLE89 is called with a safety clearance of 5 mm and specification of the final drilling depth as an absolute value. The drilling axis is the Z axis.



```
DEF REAL RFP, RTP, DP, DTB                    ; Definition of parameters
RFP=102 RTP=107 DP=72 DTB=3                   ; Value assignments
N10 G90 G17 F100 S450 M4                      ; Specification of technology
                                              values
N20 G0 X80 Y90 Z107                           ; Approach drilling position
N30 CYCLE89 (RTP, RFP, 5, DP, , DTB)          ; Cycle call
N40 M02                                       ; End of program
```

2.5 Drilling pattern cycles

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

2.5.1 Requirements

Drilling pattern cycles without drilling cycle call

Drilling pattern cycles can also be used for other applications without prior modal call of a drilling cycle 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.

To acknowledge the error message, press the following key:

To continue the program execution, press the following key:

The drilling pattern cycle will then approach each of the positions calculated from the input data one after the other without calling a subroutine at these points.

Behavior when quantity parameter is zero

The number of holes in a 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 issued and the cycle is aborted.

Checks in case of limited ranges of input values

Generally, there are no plausibility checks for defining parameters in the drilling pattern cycles.

2.5.2 Row of holes - HOLES1

Programming

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

Parameters

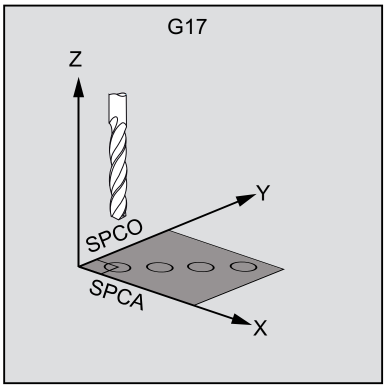
Parameter	Data type	Description
SPCA	REAL	First axis of the plane (abscissa) of a reference point on the straight line (absolute)
SPCO	REAL	Second axis of the plane (ordinate) of this reference point (absolute)
STA1	REAL	Angle to the first axis of the plane (abscissa) Range of values: –180<STA1≤180 degrees
FDIS	REAL	Distance from the first hole to the reference point (enter without sign)
DBH	REAL	Distance between the holes (enter without sign)
NUM	INT	Number of holes

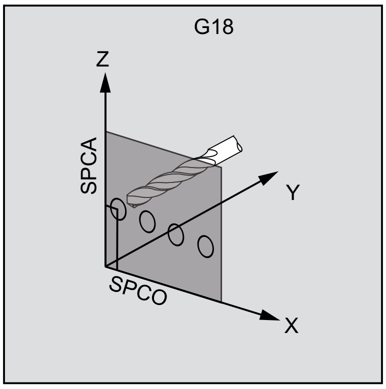
Function

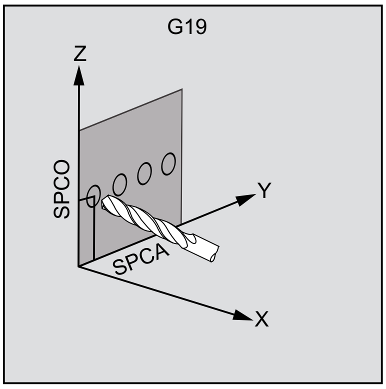
This cycle can be used to produce a row of holes, i.e. a number of holes arranged along a straight line, or a grid of holes. The type of hole is determined by the drilling cycle that has already been called modally.

Sequence

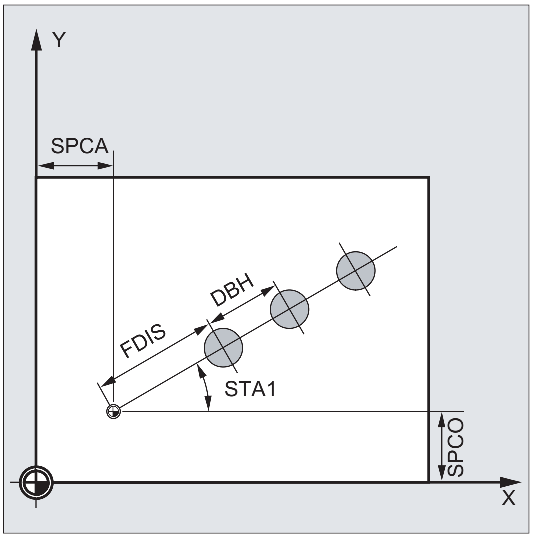
To avoid unnecessary travel, the cycle calculates whether the row of holes is machined starting from the first hole or the last hole from the actual position of the plane axes and the geometry of the row of holes. The drilling positions are then approached one after the other at rapid traverse.







Explanation of the parameters



SPCA and SPCO (reference point on the first axis of the plane and of the second axis of the plane)

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

STA1 (angle)

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

FDIS and DBH (distance)

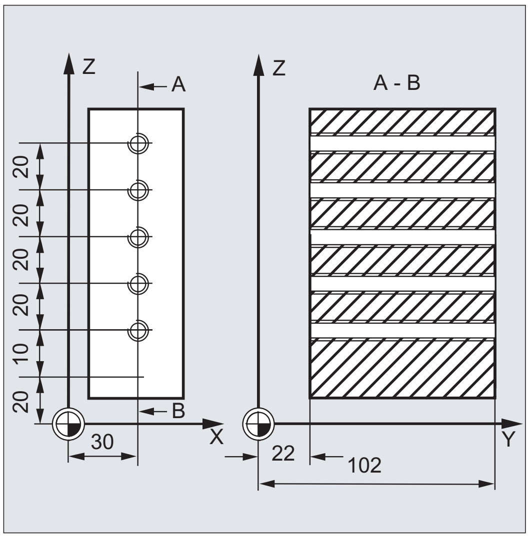
The distance of the first hole and the reference point defined under SPCA and SPCO is programmed with FDIS. The parameter DBH contains the distance between any two holes.

NUM (number)

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

Programming example: Row of holes

Use this program to machine a row of holes consisting of five tapped holes arranged parallel to the Z axis of the ZX plane and which have a distance of 20 mm one to another. The starting point of the row of holes is at Z20 and X30 whereby the first hole has a distance of 10 mm from this point. The geometry of the row of holes is described by the cycle HOLES1. First, drilling is carried out using CYCLE82, and then tapping is performed using CYCLE84 (tapping without compensating chuck). The holes are 80 mm in depth (difference between reference plane and final drilling depth).

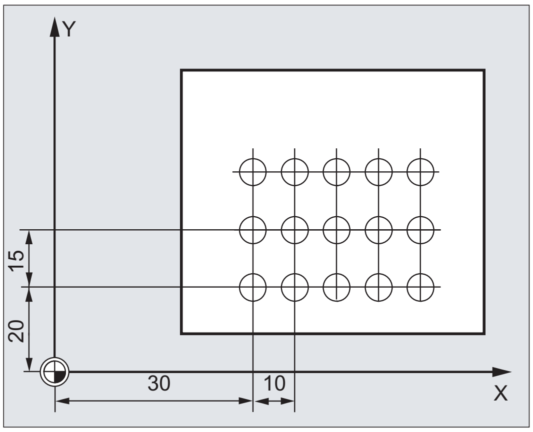



```
N10 G90 F30 S500 M3 T10 D1                    ; Specification of the
                                              technological values for the
                                              machining step
N20 G17 G90 X20 Z105 Y30                      ; Approach start position
N30 MCALL CYCLE82(105, 102, 2, 22, 0, 1)      ; Modal call of drilling cycle
N40 HOLES1(20, 30, 0, 10, 20, 5)              ; Call of row-of-holes cycle; the
                                              cycle starts with the first hole;
                                              only the drill positions are
                                              approached in this cycle
N50 MCALL                                     ; Deselect modal call
...                                           ; Change tool
N60 G90 G0 X30 Z110 Y105                      ; Approach position next to 5th
                                              hole
N70 MCALL CYCLE84(105, 102, 2, 22, 0, , 3, ,  ; Modal call of the tapping cycle
4.2, ,300, )
N80 HOLES1(20, 30, 0, 10, 20, 5)              ; Call of row of holes cycle
                                              starting with the fifth hole in
                                              the row
N90 MCALL                                     ; Deselect modal call
N100 M02                                      ; End of program
```

Programming example: Grid of holes

Use this program to machine a grid of holes consisting of five rows with five holes each, which are arranged in the XY plane, with a spacing of 10 mm between them. The starting point of the grid is at X30 Y20.

The example uses R parameters as transfer parameters for the cycle.



R10=102	; Reference plane
R11=105	; Retraction plane
R12=2	; Safety clearance
R13=75	; Drilling depth
R14=30	; Reference point for the row of holes in the first axis of the plane
R15=20	; Reference point for the row of holes in the second axis of the plane
R16=0	; Starting angle
R17=10	; Distance from first hole to reference point
R18=10	; Distance between the holes
R19=5	; Number of holes per row
R20=5	; Number of rows
R21=0	; Row counter
R22=10	; Distance between the rows
N10 G90 F300 S500 M3 T10 D1	: Specification of the technological values
N20 G17 G0 X=R14 Y=R15 Z105	; Approach starting position
N30 MCALL CYCLE82(R11, R10, R12, R13, 0, 1)	; Modal call of drilling cycle
N40 LABEL1:	; Call of row of holes cycle
N41 HOLES1(R14, R15, R16, R17, R18, R19)	
N50 R15=R15+R22	; Calculate y value for the next line
N60 R21=R21+1	; Increment line counter
N70 IF R21<R20 GOTOB LABEL1	; Return to LABEL1 if the condition is fulfilled
N80 MCALL	; Deselect modal call
N90 G90 G0 X30 Y20 Z105	; Approach starting position
N100 M02	; End of program

2.5.3 Circle of holes - HOLES2

Programming

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

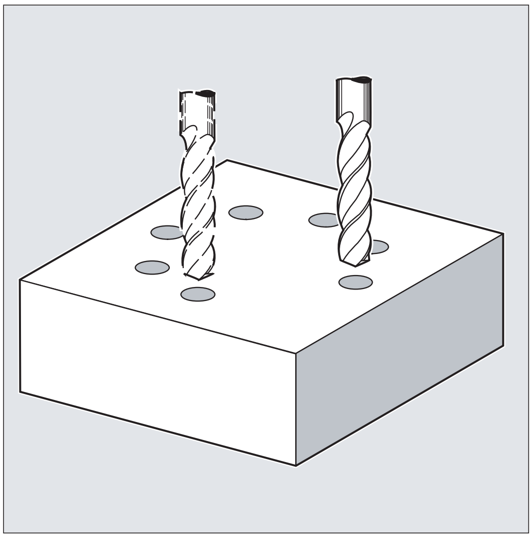
Parameters

Parameter	Data type	Description
CPA	REAL	Center point of circle of holes (absolute), first axis of the plane
CPO	REAL	Center point of circle of holes (absolute), second axis of the plane
RAD	REAL	Radius of circle of holes (enter without sign)
STA1	REAL	Starting angle Range of values: –180<STA1≤180 degrees
INDA	REAL	Incrementing angle
NUM	INT	Number of holes

Function

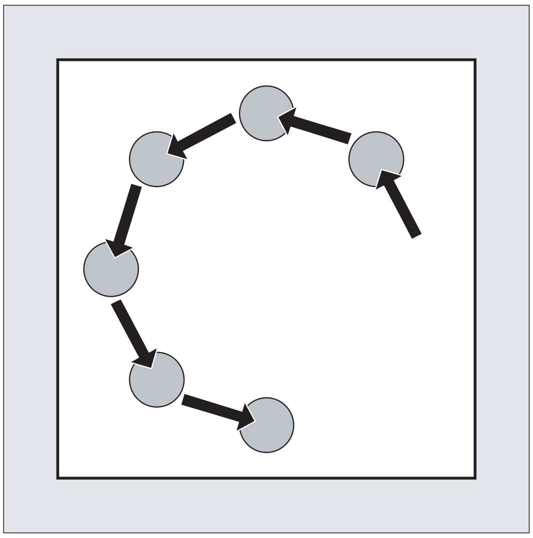
Use this cycle to machine a circle of holes. The machining plane must be defined before the cycle is called.

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

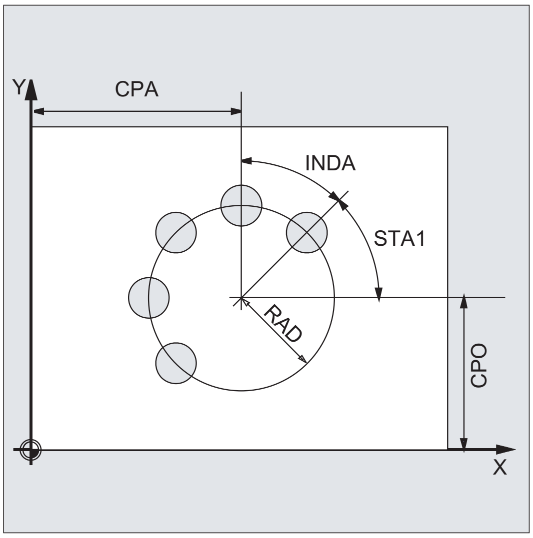


Sequence

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



Explanation of the parameters



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

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

STA1 and INDA (starting and incremental angle)

These parameters define the arrangement of the holes on the circle of holes.

The STA1 parameter defines the angle of rotation between the positive direction of the first axis (abscissa) in the workpiece coordinate system active before the cycle was called and the first hole. The INDA parameter contains the angle of rotation from one hole to the next.

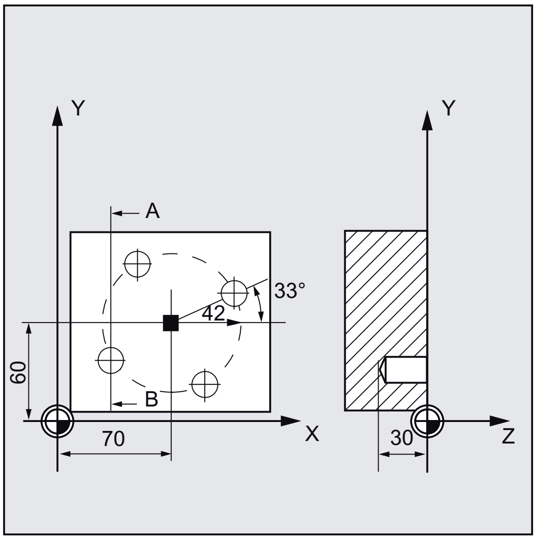
If the INDA parameter is assigned the value zero, the indexing angle is calculated internally from the number of holes which are positioned equally in a circle.

NUM (number)

The NUM parameter defines the number of holes.

Programming example1: Circle of holes

The program uses CYCLE82 to produce four holes having a depth of 30 mm. The final drilling depth is specified as a relative value to the reference plane. The circle is defined by the center point X70 Y60 and the radius 42 mm in the XY plane. The starting angle is 33 degrees. The safety clearance in drilling axis Z is 2 mm.




```
N10 G90 F140 S170 M3 T10 D1              ; Specification of technology
                                         values
N20 G17 G0 X50 Y45 Z2                    ; Approach starting position
N30 MCALL CYCLE82(2, 0, 2, , 30, 0)      ; Modal call of the drilling
                                         cycle, without dwell time, DP is
                                         not programmed
N40 HOLES2 (70, 60, 42, 33, 0, 4)        ; Call of the circle-of-holes
                                         cycle; the incremental angle is
                                         calculated in the cycle since the
                                         parameter INDA has been omitted
N50 MCALL                                ; Deselect modal call
N60 M02                                  ; End of program
```

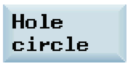
Programming example 1: Circle of holes

Proceed through the following steps:

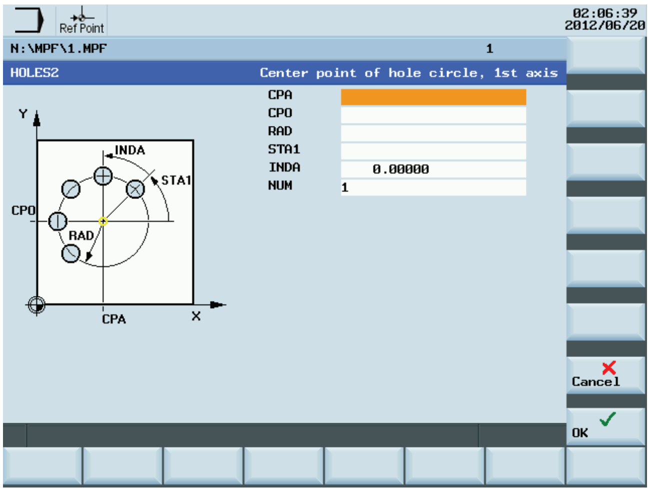
1. Select the desired operating area.

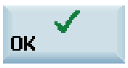
2. Open the vertical softkey bar for available drilling cycles.

3. Press this softkey from the vertical softkey bar.



4. Press this softkey to open the window for this cycle. Parameterize the cycle as desired.





5. Confirm your settings with this softkey. The cycle is then automatically transferred to the program editor as a separate block.

2.5.4 Arbitrary positions - CYCLE802

Programming

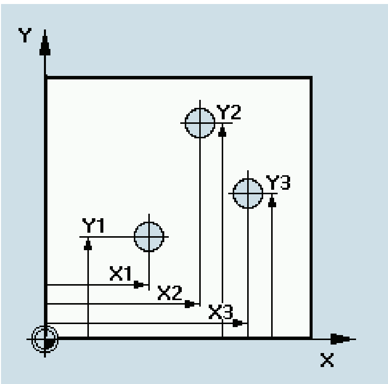
CYCLE802 (111111111, 111111111, X0, Y0, X1, Y1, X2, Y2, X3, Y3, X4, Y4)

Parameters

Parameter	Data type	Description
PSYS	INT	Internal parameter, only the default value 111111111 is possible
PSYS	INT	Internal parameter, only the default value 111111111 is possible
X0	REAL	First position in the X axis
Y0	REAL	First position in the Y axis
X1	REAL	Second position in the X axis
Y1	REAL	Second position in the Y axis
X2	REAL	Third position in the X axis
Y2	REAL	Third position in the Y axis
X3	REAL	Fourth position in the X axis
Y3	REAL	Fourth position in the Y axis
X4	REAL	Fifth position in the X axis
Y4	REAL	Fifth position in the Y axis

Function

This cycle allows you to freely program positions, i.e., rectangular or polar. Individual positions are approached in the order in which you program them.



Sequence

The drilling tool in the program traverses all programmed positions in the order in which you program them. Machining of the positions always starts at the reference point. If the position pattern consists of only one position, the tool is retracted to the retraction plane after machining.

Explanation of the parameters

X0, Y0...X4, Y4

All positions will be programmed absolutely.

Programming example:

Drilling in G17 at the Positions

```
X20 Y20
X40 Y25
X30 Y40
N10 G90 G17                               ; Absolute dimension data X/Y plane
N20 T10                                   ; Selects the tool
N30 M06                                   ; Tool change
S800 M3                                   ; Spindle speed clockwise rotation of the
                                          spindle
M08 F140                                  ; Feedrate Coolant on
G0 X0 Y0 Z20                              ; Approach starting position
MCALL CYCLE82 (2, 0, 2, -5, 5, 0)         ; Modal call of the drilling
N40 CYCLE802 (111111111, 111111111, 20,   ; call cycle positions
20, 40, 25, 30, 40)
N50 MCALL                                 ; Deselect modal call
N60 M30                                   ; End of the program
```

2.6 Milling cycles

2.6.1 Requirements

Call and return conditions

Milling cycles are programmed independently of the particular axis name.

Before you call the milling cycles, a tool compensation must be activated.

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

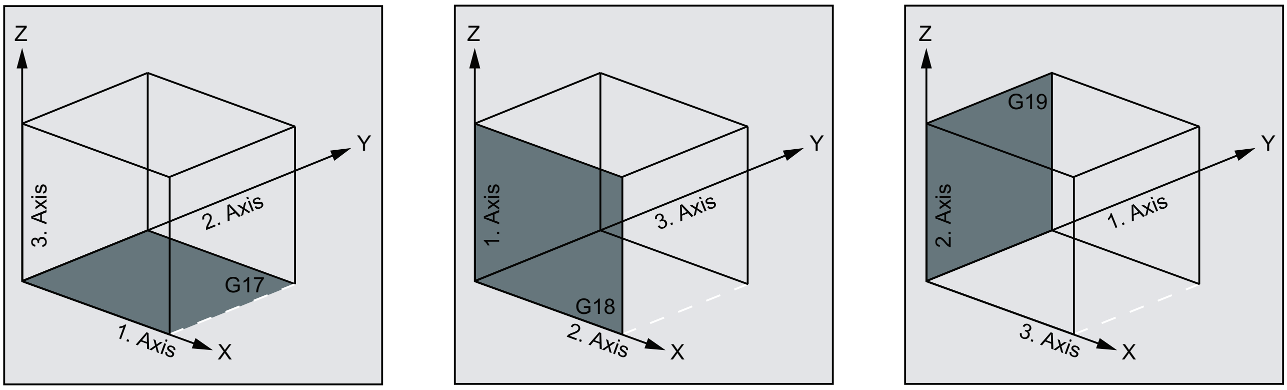
The center point coordinates for the milling pattern or the pocket to be machined are programmed in a rectangular coordinate system.

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

Plane definition

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

See the following illustration for plane and axis assignment:



Messages with regard to the machining state

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

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

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

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

2.6.2 Face milling - CYCLE71

Programming

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

Parameters

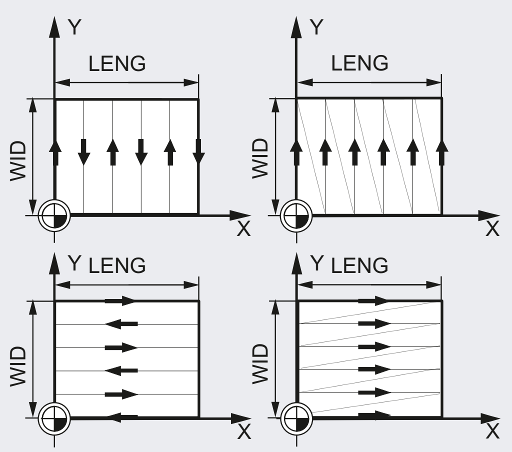
Parameter	Data type	Description
_RTP	REAL	Retraction plane (absolute)
_RFP	REAL	Reference plane (absolute)
_SDIS	REAL	Safety clearance (to be added to the reference plane; enter without sign)
_DP	REAL	Depth (absolute)
_PA	REAL	Starting point (absolute), first axis of the plane
_PO	REAL	Starting point (absolute), second axis of the plane
_LENG	REAL	Rectangle length along the first axis, incremental. The corner from which the dimension starts results from the sign.
_WID	REAL	Rectangle length along the second axis, incremental. The corner from which the dimension starts results from the sign.
_STA	REAL	Angle between the longitudinal axis of the rectangle and the first axis of the plane (abscissa, enter without sign). Range of values: 0° ≤ STA < 180°
_MID	REAL	Maximum infeed depth (enter without sign)
_MIDA	REAL	Maximum infeed width during solid machining in the plane as a value (enter without sign)
_FDP	REAL	Retraction travel in the finishing direction (incremental, enter without sign)
_FALD	REAL	Finishing dimension in the depth (incremental, enter without sign)
_FFP1	REAL	Feedrate for surface machining
_VARI	INT	Machining type (enter without sign) UNITS DIGIT Values: 1 roughing, 2 finishing TENS DIGIT: Values: 1: parallel to the first axis of the plane, in one direction, 2: parallel to the second axis of the plane, in one direction, 3: parallel to the first axis of the plane, with alternating direction 4: parallel to the second axis of the plane, with alternating direction
_FDP1	REAL	Overrun travel in the direction of the plane infeed (incremental, enter without sign)

Function

Use CYCLE71 to mill any rectangular surface. The cycle differentiates between roughing (machining the surface in several steps until reaching the final machining allowance) and finishing (milling the end face in one step). The maximum infeed in width and depth can be specified.

The cycle operates without cutter radius compensation. The depth infeed is performed in the open.

See the following illustration for possible face milling strategies:



Possible solid machining strategies for face milling

Sequence

Position reached prior to cycle start:

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

The cycle creates the following sequence of motions:

- G0 is applied to approach the infeed point at the current position level. The reference plane, brought forward by the safety distance, is then also approached with G0 to this position. Then, also with G0, feeding to the machining plane. G0 is possible since infeed in the open is possible.

  There are several roughing strategies (paraxial in one direction or back and forth).
- Sequence of motions when roughing:

  Face milling can be performed in several planes based on the programmed values _DP, _MID and _FALD. Machining is carried out from the top downward, i.e. one plane each is removed and then the next depth infeed is carried out in the open (_FDP parameters). The traversing paths for solid machining in the plane depend on the values of the parameters _LENG, _WID, _MIDA, _FDP, _FDP1 and the cutter radius of the active tool.

  The first path to be milled is always traversed such that the infeed depth exactly corresponds to _MIDA, ensuring that no width infeed larger than the maximum possible width infeed occurs. The tool center point therefore does not always travel exactly on the edge (only if _MIDA = cutter radius). The dimension by which the tool traverses outside the edge is always equal to the cutter diameter - _MIDA even if only one surface cut is performed, i. e. area width + overrun is less than _MIDA. The other paths for width infeed are calculated internally so as to produce a uniform path width (<= _MIDA).

- Sequence of motions when finishing:

  When finishing, the surface is milled in the plane once. This means that the finishing allowance when roughing has to be selected also such that the residual depth can be removed with the finishing tool in one step.

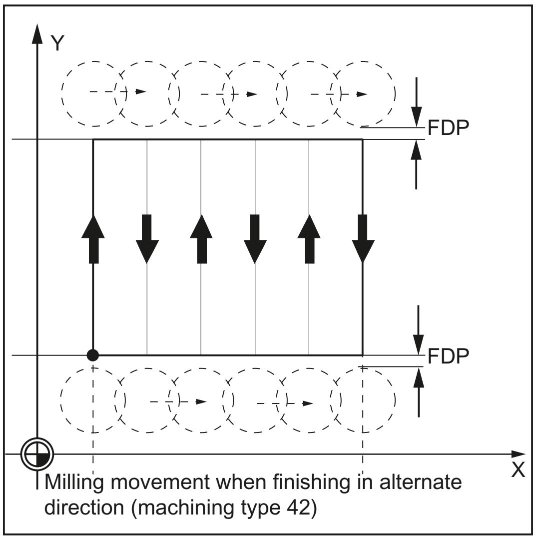
  After each surface milling pass in the plane, the tool will retract. The retraction travel is programmed under the parameter _FDP.

  Machining in one direction stops at the final machining allowance + safety distance and the next starting point is approached in rapid traverse.

  When roughing in one direction, the tool will retract by the calculated infeed depth + safety clearance. The depth infeed is performed at the same point as in roughing.

  After finishing has been completed, the tool retracts from the last position reached to the retraction plane _RTP.

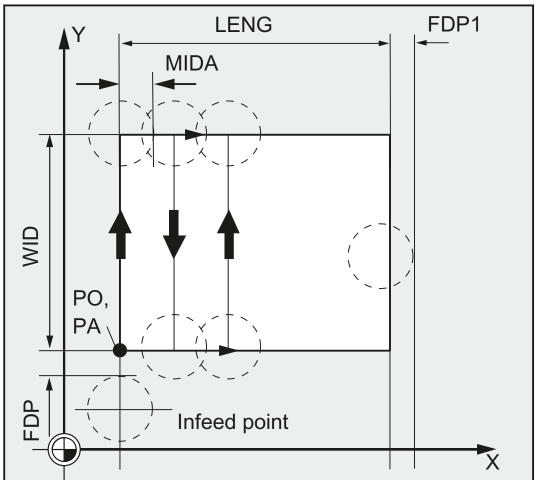
  See the following illustration for milling movement:



Explanation of the parameters

For an explanation of the parameters _RTP, _RFP, and _SDIS, refer to Section "Drilling, centering - CYCLE81 (Page 122)".

For an explanation of the parameters _STA, _MID, and _FFP1, refer to Section "Milling a rectangular pocket - POCKET3 (Page 218)".



Roughing with MIDA larger than milling radius
(machining type 41)

_DP (depth)

The depth can be specified as an absolute value (_DP) to the reference plane.

_PA, _PO (starting point)

Use the parameters _PA and _PO to define the starting point of the area in the axes of the plane.

_LENG, _WID (length)

Use the parameters _LENG and _WID to define the length and width of a rectangle in the plane. The position of the rectangle, with reference to _PA and _PO, results from the sign.

_MIDA (max. infeed width)

Use this parameter to define the maximum infeed width when machining in a plane. Analogously to the known calculation method for the infeed depth (equal distribution of the total depth with maximum possible value), the width is distributed equally, maximally with the value programmed under _MIDA.

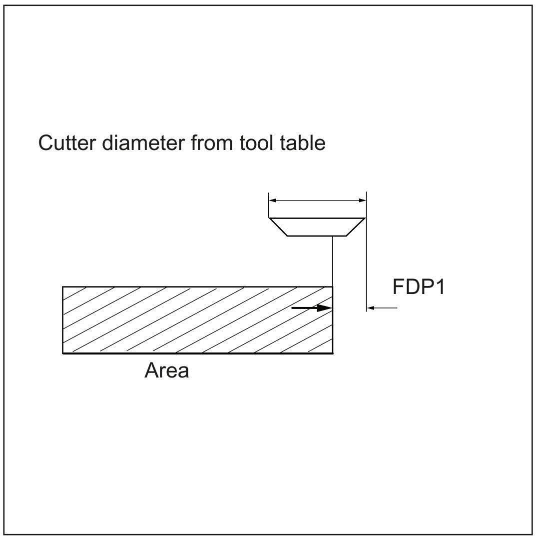
If this parameter is not programmed or has value 0, the cycle will internally use 80% of the milling tool diameter as the maximum infeed width.

_FDP (retraction travel)

Use this parameter to define the dimension for the retraction travel in the plane. This parameter should reasonably always have a value greater than zero.

_FDP1 (overrun travel)

Use this parameter to specify an overrun travel in the direction of the plane infeed (_MIDA). Thus, it is possible to compensate the difference between the current cutter radius and the tool nose radius (e.g. cutter radius or cutting tips arranged at an angle). The last milling cutter center point path therefore always results as _LENG (or _WID) + _FDP1 - tool radius (from the compensation table).



_FALD (finishing allowance)

When roughing, a finishing allowance in the depth is taken into account which is programmed under this parameter.

The residual material remained as the finishing allowance must always be specified for finishing to ensure that the tool can be retracted and then fed to the starting point of the next cut without collision.

If > 0, the parameter is ignored for finishing.

_VARI (machining type)

Use the parameter _VARI to define the machining type.

Possible values are:

- Units digit:

  1=roughing to finishing allowance

  2=finishing
- Tens digit:

  1=parallel to the first axis of the plane; unidirectional

  2=parallel to the second axis of the plane; unidirectional

  3=parallel to the first axis of the plane; with alternating direction

  4=parallel to the second axis of the plane; with alternating direction

If a different value is programmed for the parameter _VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

Note

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

Programming example: Face milling

Parameters for the cycle call:

Parameter	Description	Value
_RTP	Retraction plane	10 mm
_RFP	Reference plane	0 mm
_SDIS	Safety clearance	2 mm
_DP	Milling depth	-11 mm
_PA	Starting point of the rectangle	X = 100 mm
_PO	Starting point of the rectangle	Y = 100 mm
_LENG	Rectangle dimensions	X = +60 mm
_WID	Rectangle dimensions	Y = +40 mm
_STA	Angle of rotation in the plane	10 degrees
_MID	Maximum infeed depth	6 mm
_MIDA	Maximum infeed width	10 mm
_FDP	Retraction at the end of the milling path	5 mm
_FALD	Finishing allowance in depth	No finishing allowance
_FFP1	Feedrate in the plane	4000 mm/min
_VARI	Machining type	31 (Roughing parallel to the X axis with alternating direction)
_FDP1	Overrun on last cut as determined by the cutting edge geometry	2 mm

A milling cutter with 10 mm radius is used.

```
N10 T2 D2
N20 G17 G0 G90 G54 G94 F2000 X0 Y0 Z20           ; Approach start position
N30 CYCLE71(10, 0, 2, -11, 100, 100, 60, 40, 10,  ; Cycle call
6, 10, 5, 0, 4000, 31, 2)
N40 G0 G90 X0 Y0
N50 M02                                           ; End of program
```

2.6.3 Contour milling - CYCLE72

Programming

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

Parameters

Parameter	Data type	Description
_KNAME	STRING	Name of contour subroutine
_RTP	REAL	Retraction plane (absolute)
_RFP	REAL	Reference plane (absolute)
_SDIS	REAL	Safety clearance (to be added to the reference plane; enter without sign)
_DP	REAL	Depth (absolute)
_MID	REAL	Maximum infeed depth (incremental; enter without sign)
_FAL	REAL	Finishing allowance at the edge contour (enter without sign)
_FALD	REAL	Finishing allowance at the base (incremental, enter without sign)
_FFP1	REAL	Feedrate for surface machining
_FFD	REAL	Feedrate for depth infeed (enter without sign)
_VARI	INT	Machining type (enter without sign) UNITS DIGIT Values: 1: roughing, 2: finishing TENS DIGIT: Values: 0: intermediate travel with G0, 1 intermediate travel with G1 HUNDREDS DIGIT Values: 0: Retraction at the end of contour to _RTP 1: Retraction at the end of contour to _RFP + _SDIS 2: Retraction by _SDIS at the end of contour 3: No retraction at the end of contour
_RL	INT	Traveling around the contour either centrally, to the right or to the left (with G40, G41 or G42; enter without sign) Values: 40: G40 (approach and return, straight line only) 41: G41 42: G42

Parameter	Data type	Description
_AS1	INT	Specification of the approach direction/path: (enter without sign) UNITS DIGIT: Values: 1: Straight tangential line 2: Quadrant 3: Semi-circle TENS DIGIT: Values: 0: Approach to the contour in the plane 1: Approach to the contour in a spatial path
_LP1	REAL	Length of the approach travel (with straight-line) or radius of the approach arc (with circle) (enter without sign)
The following parameters can be selected as options:		
_FF3	REAL	Retraction feedrate and feedrate for intermediate positions in the plane (in the open)
_AS2	INT	Specification of the retraction direction/path: (enter without sign) UNITS DIGIT: Values: 1: Straight tangential line 2: Quadrant 3: Semi-circle TENS DIGIT: Values: 0: Retraction from the contour in the plane 1: Retraction from the contour in a spatial path
_LP2	REAL	Length of the retraction travel (with straight-line) or radius of the retraction arc (with circle) (enter without sign)

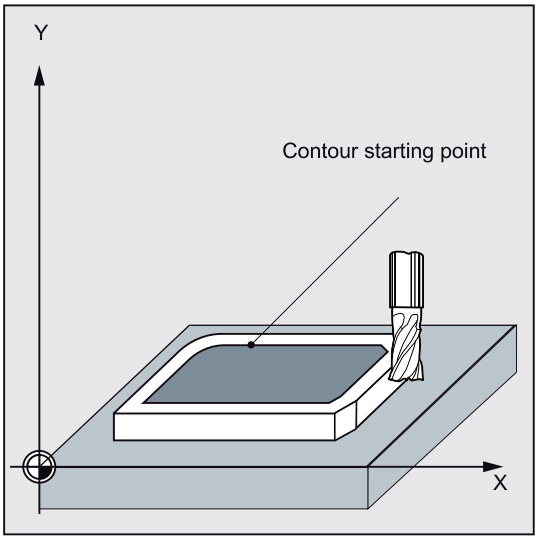
Function

Use CYCLE72 to mill along any contour defined in a subroutine. The cycle operates with or without cutter radius compensation.

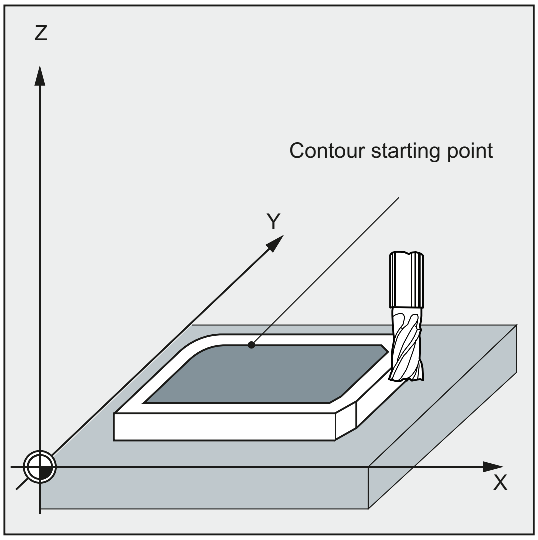
It is not imperative that the contour is closed. Internal or external machining is defined via the position of the cutter radius compensation (centrally, left or right to the contour).

The contour must be programmed in the direction as it is to be milled and must consist of a minimum of two contour blocks (start and end point), since the contour subroutine is called directly internally in the cycle.

See the following illustration for path milling 1:



See the following illustration for path milling 2:



Functions of the cycle

- Selection of roughing (single-pass traversing parallel to contour, taking into account a finishing allowance, if necessary at several depths until the finishing allowance is reached) and finishing (single-pass traversing along the final contour if necessary at several depths)
- Smooth approach to and retraction from the contour either tangentially or radially (quadrant or semi-circle)
- Programmable depth infeeds
- Intermediate motions either at rapid traverse rate or at feedrate

Sequence

Position reached prior to cycle start:

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

The cycle generates the following sequence of motions when roughing:

The depth infeeds are distributed equally with the maximum possible value of the specified parameters.

- Traversing to the starting point for first milling with G0/G1 (and FF3). This point is calculated internally in the control system and depends on the following:
  - Contour starting point (first point in the subroutine),
  - Direction of the contour at the starting point,
  - Approach mode and its parameters
  - Tool radius

    The cutter radius compensation is activated in this block.
- Depth infeed to the first or next machining depth plus programmed safety clearance with G0/G1. The first machining depth results from the following data:
  - Total depth
  - Finishing allowance
  - The maximum possible depth infeed
- Approach of the contour vertically with depth infeed _FFD and then in the plane at the programmed feedrate _FFP1 or 3D with the feedrate programmed under _FAD according to the programming for smooth approach
- Milling along the contour with G40/G41/G42
- Smooth retraction from the contour with G1 while continuing feed for the surface machining by the retraction amount
- Retraction with G0 /G1 (and feedrate for intermediate paths _FF3), depending on the programming
- Retraction to the depth infeed point with G0/G1 (and _FF3).
- This sequence is repeated on the next machining plane up to finishing allowance in the depth.

A completion of roughing, the tool stands above the point (calculated internally in the control system) of retraction from the contour at the height of the retraction plane.

The cycle generates the following sequence of motions when finishing:

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

Smooth approach and retraction of the contour is carried out according to the existing parameters. The appropriate path is calculated internally in the control system.

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

Note

Contour programming

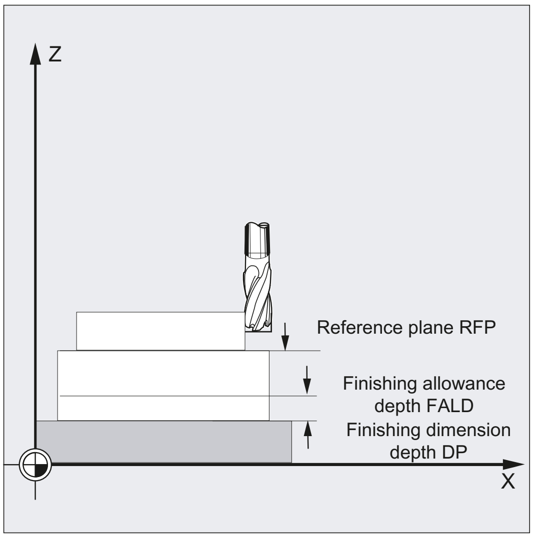
When programming the contour, observe the following:

- No programmable offset may be selected in the subroutine prior to the first programmed position.
- The first block of the contour subroutine is a straight line block containing G90 / G0 or G90 / G1 and defines the start of the contour.
- The starting condition of the contour is the first position in the machining plane which is programmed in the contour subroutine.
- The cutter radius compensation is selected/deselected by the higher-level cycle; therefore, no G40, G41, G42 is programmed in the contour subroutine.

Explanation of the parameters

For an explanation of the parameters _RTP, _RFP, and _SDIS, refer to Section "Drilling, centering - CYCLE81 (Page 122)".

For an explanation of the parameters _MID, _FAL, _FALD, _FFP1, _FFD, and _DP, refer to Section "Milling a rectangular pocket - POCKET3 (Page 218)".



_KNAME (name)

The contour to be milled is programmed completely in a subroutine. _KNAME defines the name of the contour subroutine.

- Defining the contour as a subroutine

  _KNAME = name of the subroutine

  – If the subroutine already exists, specify a name, and then continue.

  – If the subroutine does not yet exist, specify a name and then press the following softkey:

    New file

    A program with the entered name is created and the program automatically jumps to the contour editor.

  – Use the following softkey to confirm your input and return to the screen form for this cycle.

    Tech interface

- Defining the contour as a section of the called program

  KNAME = name of the starting label: name of the end label

  Input:

  – If the contour is not yet described, specify the name of the starting label and press the following softkey. If the controu is already described (name of starting label: name of end label), directly press the following softkey:

    Attach contour

    The control system automatically creates starting and end labels from the name you have entered and the program jumps to the contour editor.

  – Use the following softkey to confirm your input and return to the screen form for this cycle:

    Tech interface

Examples:

`_KNAME="CONTOUR_1"`	The milling contour is the complete program CONTOUR_1.
`_KNAME="START:END"`	The milling contour is defined as a section in the calling program, which starts from the block containing label START to the block containing label END.

_LP1, _LP2 (length, radius)

Use the parameter _LP1 to program the approach travel or approach radius (distance from the tool external edge to the contour starting point), and the parameter _LP2 to program the retraction travel or retraction radius (distance from the tool external edge to the contour end point).

Parameters _LP1 and _LP2 must be set to >0. In the case of zero, error 61116 "Approach or retraction path=0" is output.

Note

When using G40, the approach or retraction travel is the distance from the tool center point to the start or end point of the contour.

_VARI (machining type)

Use the parameter _VARI to define the machining type.

If a different value is programmed for the parameter _VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

_RL (bypassing the contour)

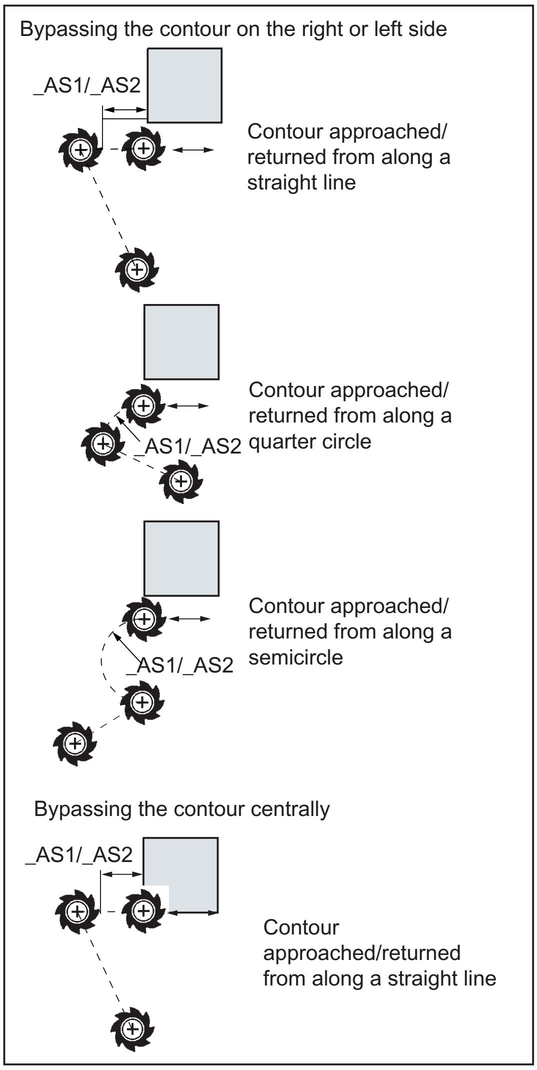
With the parameter _RL, you program the traveling around the contour centrally, to the right or to the left with G40, G41 or G42.

_AS1, _AS2 (approach direction/path, retraction direction/path)

Use the parameter _AS1 to program the specification of the approach path and _AS2 to program that of the retraction path. If _AS2 is not programmed, then the behavior of the retraction path is analogous to that of the approach path.

Smooth approach of the contour along a spatial path (helix or straight line) should only be programmed if the tool is not yet being used or is suitable for this type of approach.

See the following illustration for _AS1/_AS2:



In the case of central (G40), approach and retraction is only possible along a straight line.

_FF3 (retraction feedrate)

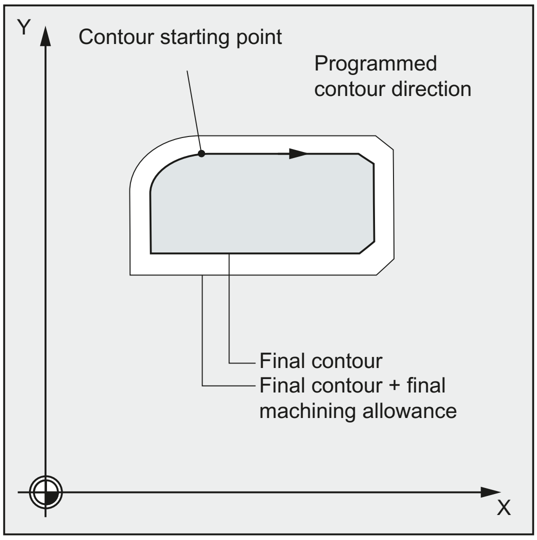
Use the parameter _FF3 to define a retraction feedrate for intermediate positions in the plane (in the open) if the intermediate motions are to be carried out with feedrate (G01). If no feedrate value is programmed, the intermediate motions with G01 are carried out at surface feedrate.

Note

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

Programming example 1: Milling around a closed contour externally

This program is used to mill the contour shown in the diagram below.



Parameters for the cycle call:

Parameter	Description	Value
_RTP	Retraction plane	250 mm
_RFP	Reference plane	200 mm
_SDIS	Safety clearance	3 mm
_DP	Infeed depth	175 mm
_MID	Maximum infeed depth	10 mm
_FALD	Finishing allowance in depth	1.5 mm
_FFD	Feedrate depth infeed	400 mm/min
_FAL	Finishing allowance in the plane	1 mm
_FFP1	Feedrate in the plane	800 mm/min
_VARI	Machining type	111 (Roughing up to finishing allowance; intermediate paths with G1, for intermediate paths retraction in Z to _RFP + _SDIS)
Parameters for approach:		
_RL	G41 - left of the contour, i.e. external machining	41
_LP1	Approach and retraction in a quadrant in the plane	20 mm radius
_FF3	Retraction feedrate	1000 mm/min

N10 T3 D1	; T3: Milling cutter with radius 7
N20 S500 M3 F3000	; Program feedrate and spindle speed
N30 G17 G0 G90 X100 Y200 Z250 G94	; Approach start position
N40 CYCLE72("EX72CONTOUR", 250, 200, 3, 175, 10,1, 1.5, 800, 400, 111, 41, 2, 20, 1000, 2, 20)	; Cycle call
N50 X100 Y200	
N60 M2	; End of program
EX72CONTOUR.SPF	; Subroutine for contour milling (for example)
N100 G1 G90 X150 Y160	; Starting point of contour
N110 X230 CHF=10	
N120 Y80 CHF=10	
N130 X125	
N140 Y135	
N150 G2 X150 Y160 CR=25	
N160 M2	

Programming example 2: Milling around a closed contour externally

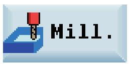
With this program, the same contour is milled as in example 1. The difference is that the contour programming is now in the calling program.

N10 T3 D1	; T3: Milling cutter with radius 7
N20 S500 M3 F3000	; Program feedrate and spindle speed
N30 G17 G0 G90 X100 Y200 Z250 G94	; Approach start position
N40 CYCLE72 ("PIECE245:PIECE245E", 250, 200, 3, 175, 10,1, 1.5, 800, 400, 11, 41, 2, 20, 1000, 2, 20)	; Cycle call
N50 X100 Y200	
N60 M2	
N70 PIECE245:	; Contour
N80 G1 G90 X150 Y160	
N90 X230 CHF=10	
N100 Y80 CHF=10	
N110 X125	
N120 Y135	
N130 G2 X150 Y160 CR=25	
N140 PIECE245E:	; End of contour
N150 M2	

Programming example 3

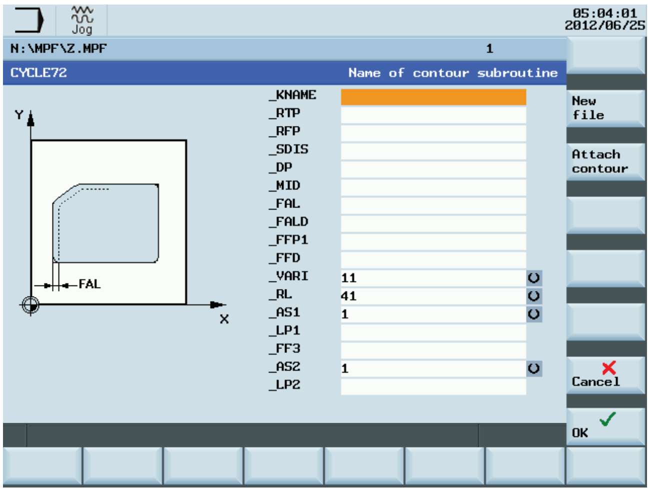
Proceed through the following steps:

1. Select the desired operating area.



2. Open the vertical softkey bar for available milling cycles.

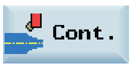
3. Press this softkey to open the window for CYCLE72. Enter a name in the first input field.



4. Press one of the following two softkeys . The program automatically jumps to the program editor screen form.

If you desire to edit and store the contour in a subroutine, press this softkey.

If you desire to edit and store the contour as a section of a main program, press this softkey.



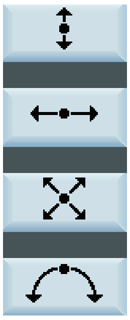
5. Press this softkey to open the contour editor. Parameterize the contour elements step by step.

   Initially you define a contour starting point and select how to approach the starting point.

   Note:

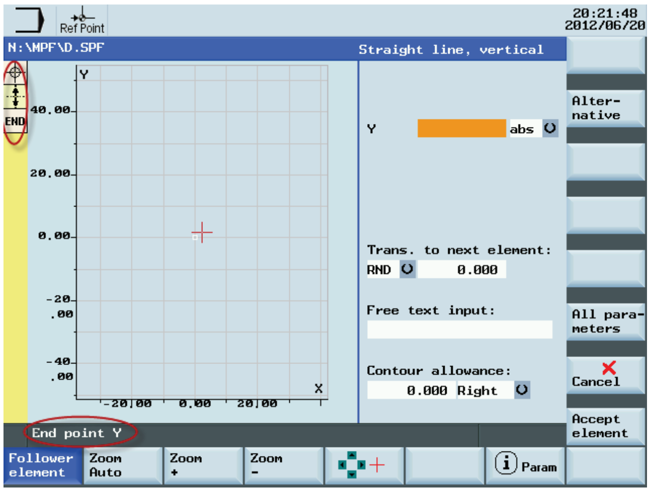
   Steps 5 to 10 below describe basic steps for contour element edits. For more information about programming in the contour editor, refer to the SINUMERIK 808D Programming and Operating Manual Milling (Part 1).

6. Press this softkey to confirm the settings.



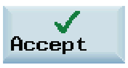
7. Select a desired machining direction and shape with the corersponding softkey. Specify the corresponding coordinates according to the drawings.

   The selected direction appears on the top left of the screen and the corresponding descriptive text is given in the information line at the bottom of the screen.



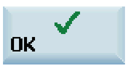
8. Press this softkey to confirm the settings.

9. Select different elements to define the contour until you complete the contour programming.



10. Press this softkey to store the contour information.

11. Press this softkey to return to the screen form for CYCLE72. Parameterize the cycle technology data as desired.



12. Confirm your settings with this softkey. The cycle is then automatically transferred to the program editor.

    Note:

    The cycle program created as a section of the main program must be stored after the M30 command.

13. If you desire to recompile the cycle, press this softkey.

2.6.4 Milling a rectangular spigot - CYCLE76

Programming

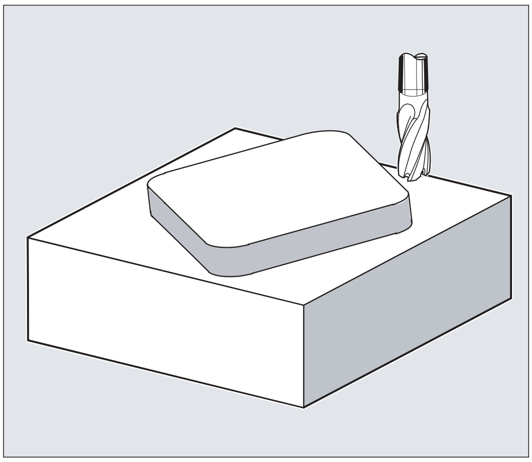
CYCLE76 (RTP, RFP, SDIS, DP, DPR, LENG, WID, CRAD, PA, PO, STA, MID, FAL, FALD, FFP1, FFD, CDIR, VARI, AP1, AP2)

Parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Final drilling depth (absolute)
DPR	REAL	Final drilling depth relative to the reference plane (enter without sign)
LENG	REAL	Spigot length
WID	REAL	Spigot width
CRAD	REAL	Spigot corner radius (enter without sign)
PA	REAL	Reference point of spigot, abscissa (absolute)
PO	REAL	Reference point of spigot, ordinate (absolute)
STA	REAL	Angle between longitudinal axis and first axis of plane
MID	REAL	Maximum depth infeed (incremental; enter without sign)
FAL	REAL	Final machining allowance at the margin contour (incremental)
FALD	REAL	Finishing allowance at the base (incremental, enter without sign)
FFP1	REAL	Feedrate on contour
FFD	REAL	Feedrate for depth infeed
CDIR	INT	Milling direction (enter without sign) Values: 0: Down-cut milling 1: Conventional milling 2: With G2 (independent of spindle direction) 3: With G3
VARI	INT	Machining type Values: 1: Roughing to final machining allowance 2: Finishing (allowance X/Y/Z=0)
AP1	REAL	Length of blank spigot
AP2	REAL	Width of blank spigot

Function

Use this cycle to machine rectangular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always carried out in the position upstream of the semi-circle style approach to the contour.



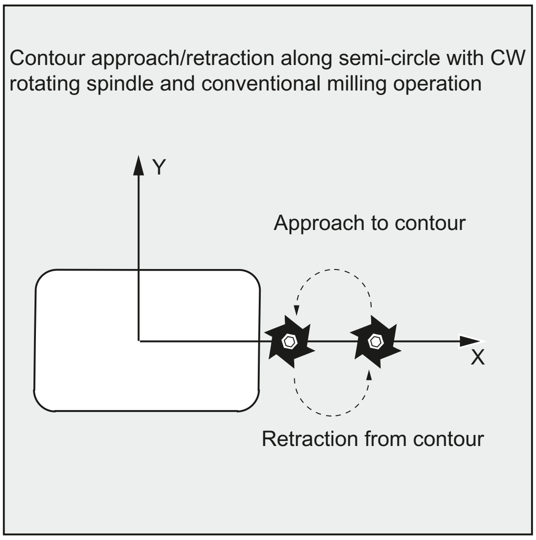
Sequence

Position reached prior to cycle start:

The starting point is a position in the positive range of the abscissa with the approach semi-circle and the programmed raw dimension on the abscissa end taken into account.

Sequence of motions when roughing (VARI=1):

- Approach/retraction from contour:



The retraction plane (RTP) is approached at rapid traverse rate to then be able to position to the starting point in the machining plane at this height. The starting point is defined with reference to 0 degrees of the abscissa.

The tool is fed to the safety clearance (SDIS) at rapid traverse with subsequent traversing to the machining depth at feedrate. To approach the spigot contour, the tool travels along a semi-circular path.

The milling direction can be determined either as up-cut milling or down-cut milling with reference to the spindle direction.

If the spigot is bypassed once, the contour is left along a semi-circle in the plane, and the tool is fed to the next machining depth.

The contour is then reapproached along a semi-circle and the spigot traversed once. This process is repeated until the programmed spigot depth is reached. Then, the retraction plane (RTP) is approached at rapid traverse rate.

- Depth infeed:
  - Feeding to the safety clearance
  - Insertion to machining depth

The first machining depth is calculated from the total depth, finishing allowance, and the maximum possible depth infeed.

Sequence of motions when finishing (VARI=2):

Depending on the set parameters FAL and FALD, finishing is either carried out at the surface contour or at the base or both together. The approach strategy corresponds to the motions in the plane as with roughing.

Explanation of the parameters

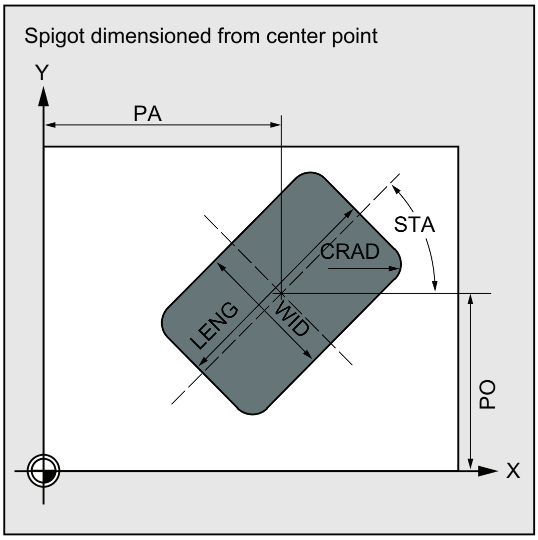
For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".

For an explanation of the parameters MID, FAL, FALD, FFP1, and FFD, refer to Section "Milling a rectangular pocket - POCKET3 (Page 218)".

LENG, WID and CRAD (spigot length, spigot width and corner radius)

Use the parameters LENG, WID and CRAD to define the form of a slot in the plane.

The spigot is always dimensioned from the center. The length (LENG) always refers to the abscissa (with a plane angle of 0 degrees).



PA, PO (reference point)

Use the parameters PA and PO to define the reference point of the spigot along the abscissa and the ordinate.

This is the spigot center point.

STA (angle)

STA specifies the angle between the first axis of the plane (abscissa) and the longitudinal axis of the spigot.

CDIR (milling direction)

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

Using the CDIR parameter, the milling direction can be programmed directly with "2 for G2" and "3 for G3", or alternatively with "synchronous milling" or "conventional milling".

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

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

VARI (machining type)

Use the parameter VARI to define the machining type.

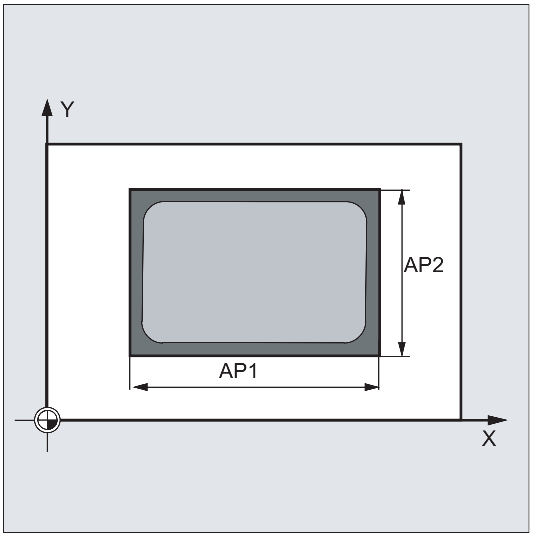
Possible values are:

- 1=roughing
- 2=finishing

AP1, AP2 (blank dimensions)

When machining the spigot, it is possible to take into account blank dimensions (e.g. when machining precast parts).

The basic sizes for the length and width (AP1 and AP2) are programmed without sign and their symmetrical positions around the spigot center are computed in the cycle. The internally calculated radius of the approach semi-circle depends on this dimension.



Note

A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is canceled and alarm 61009 "Active tool number=0" is output.

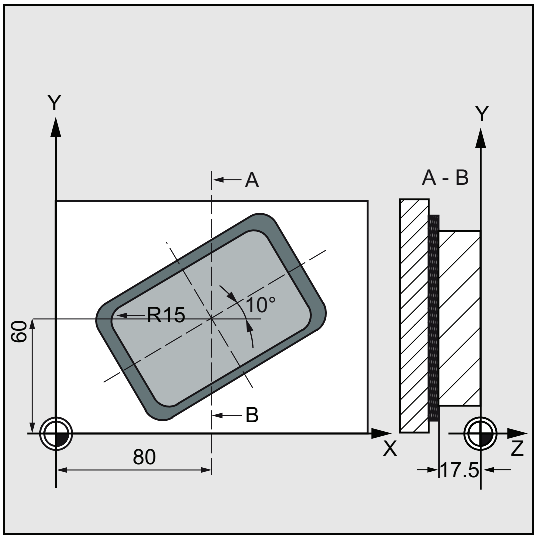
Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

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

Programming example: Spigot

Use this program to machine in the XY plane a spigot that is 60 mm long, 40 mm wide and has 15 mm corner radius. The spigot has an angle of 10 degrees relative to the X axis and is premanufactured with a length allowance of 80 mm and a width allowance of 50 mm.

See the following programming example for rectangular spigot:




```
N10 G90 G0 G17 X100 Y100 T20 D1 S3000 M3          ; Specification of technology
                                                  values
N11 M6
N30 CYCLE76 (10, 0, 2, -17.5, , 60, 40, 15, 80,   ; Cycle call
60, 10, 11, , , 900, 800, 0, 1, 80, 50)
N40 M30                                           ; End of program
```

2.6.5 Milling a circular spigot - CYCLE77

Programming

CYCLE77 (RTP, RFP, SDIS, DP, DPR, PRAD, PA, PO, MID, FAL, FALD, FFP1, FFD, CDIR, VARI, AP1)

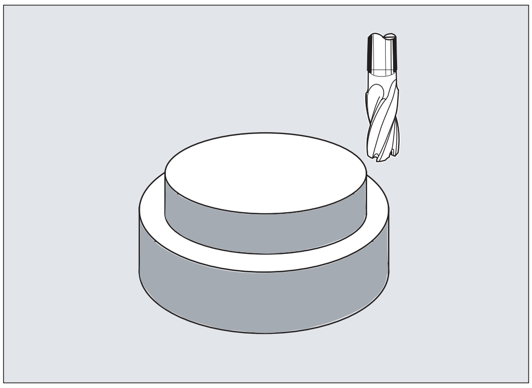
Parameters

The following input parameters are always required:

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Depth (absolute)
DPR	REAL	Depth relative to the reference plane (enter without sign)
PRAD	REAL	Spigot diameter (enter without sign)
PA	REAL	Center point of spigot, abscissa (absolute)
PO	REAL	Center point of spigot, ordinate (absolute)
MID	REAL	Maximum depth infeed (incremental; enter without sign)
FAL	REAL	Final machining allowance at the margin contour (incremental)
FALD	REAL	Finishing allowance at the base (incremental, enter without sign)
FFP1	REAL	Feedrate on contour
FFD	REAL	Feedrate for depth infeed (or spatial infeed)
CDIR	INT	Milling direction (enter without sign) Values: 0: Down-cut milling 1: Conventional milling 2: With G2 (independent of spindle direction) 3: With G3
VARI	INT	Machining type Values: 1: Roughing to final machining allowance 2: Finishing (allowance X/Y/Z=0)
AP1	REAL	Length of blank spigot

Function

Use this cycle to machine circular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always performed in the position before the semi-circular approach to the contour.



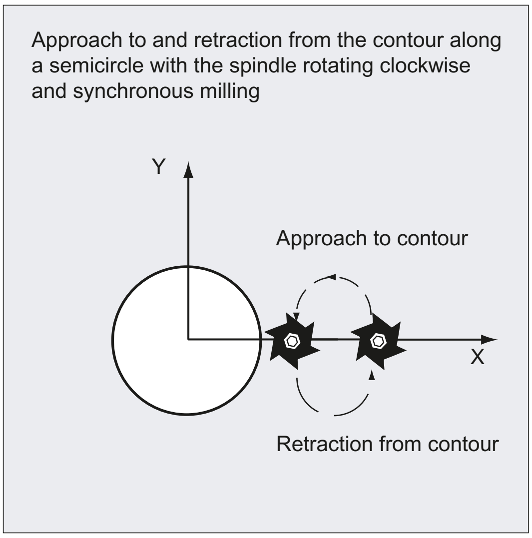
Sequence

Position reached prior to cycle start:

The starting point is a position in the positive range of the abscissa with the approach semi-circle and the programmed raw dimension taken into account.

Sequence of motions when roughing (VARI=1):

- Approach/retraction from contour:



The retraction plane (RTP) is approached at rapid traverse rate to then be able to position at this height to the starting point in the machining plane. The starting point is defined with reference to 0 degrees of the axis of the abscissa.

The tool is fed to the safety clearance (SDIS) at rapid traverse with subsequent traversing to the machining depth at feedrate. To approach the spigot contour, the tool is approached along a semi-circular path using the programmed blank spigot.

The milling direction can be determined either as up-cut milling or down-cut milling with reference to the spindle direction.

If the spigot is bypassed once, the contour is left along a semi-circle in the plane, and the tool is fed to the next machining depth.

The contour is then reapproached along a semi-circle and the spigot traversed once. This process is repeated until the programmed spigot depth is reached.

Then, the retraction plane (RTP) is approached at rapid traverse rate.

- Depth infeed:
  - Feeding to the safety clearance
  - Insertion to machining depth

The first machining depth is calculated from the total depth, finishing allowance, and the maximum possible depth infeed.

Sequence of motions when finishing (VARI=2):

According to the set parameters FAL and FALD, either finishing is carried out at the surface contour or at the base or both together. The approach strategy corresponds to the motions in the plane as with roughing.

Explanation of the parameters

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122)".

For an explanation of the parameters MID, FAL, FALD, FFP1, and FFD, refer to Section "Milling a rectangular pocket - POCKET3 (Page 218)".

PRAD (diameter of spigot)

Enter the diameter without sign.

PA, PO (spigot center point)

Use the parameters PA and PO to define the reference point of the spigot.

CDIR (milling direction)

Use this parameter to specify the machining direction for the spigot. Using the parameter CDIR, the milling direction can be programmed directly with "2 for G2" and "3 for G3", or alternatively with "synchronous milling" or "conventional milling".

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

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

VARI (machining type)

Use the parameter VARI to define the machining type. Possible values are:

- 1=roughing
- 2=finishing

AP1 (diameter of blank spigot)

Use this parameter to define the blank dimension of the spigot (without sign). The internally calculated radius of the approach semi-circle depends on this dimension.

Note

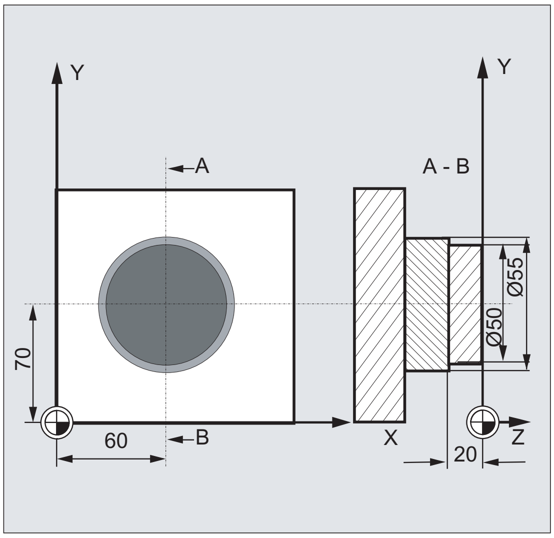
A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is canceled and alarm 61009 "Active tool number=0" is output. Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

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

Programming example: Circular spigot

Machining a spigot from a blank with a diameter of 55 mm and a maximum infeed of 10 mm per cut; specification of a final machining allowance for subsequent finishing of the spigot surface. The whole machining is performed with reverse rotation.

See the following programming exmaple for circular spigot:



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	; Change tool
N40 S2400 M3	; Specification of technology values
N50 CYCLE77 (10, 0, 3, -20, , 50, 60, 70, 10, 0, 0, 800, 800, 1, 2, 55)	; Call finishing cycle
N40 M30	; End of program

2.6.6 Long holes located on a circle - LONGHOLE

Programming

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

Parameters

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

Note

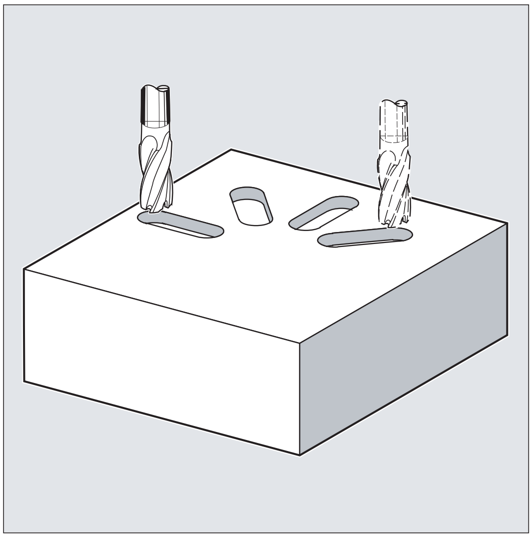
The cycle requires a milling cutter with an "end tooth cutting across center" (DIN844).

Function

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

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

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



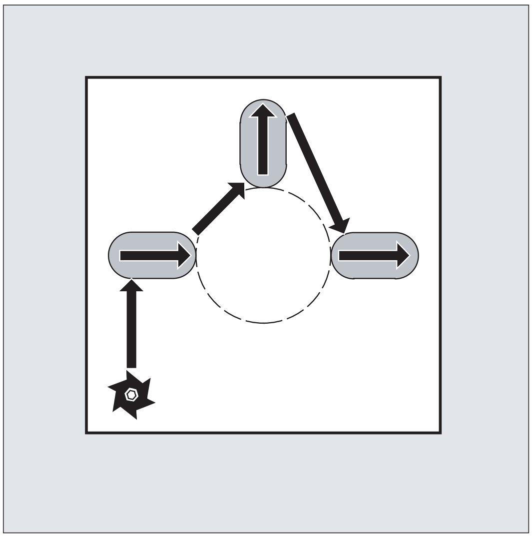
Sequence

Position reached prior to cycle start:

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

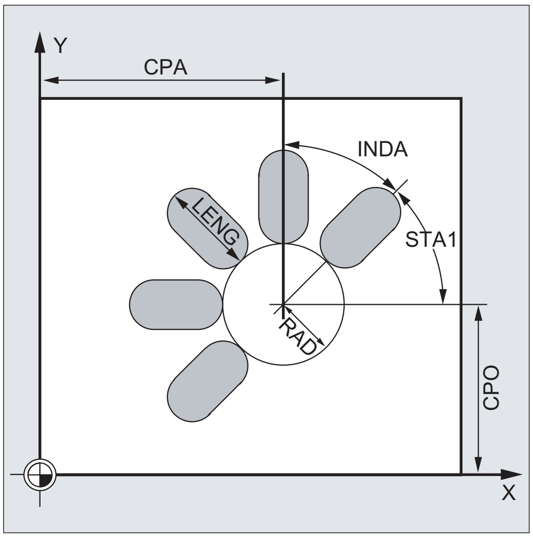
The cycle creates the following sequence of motions:

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



Explanation of the parameters

For an explanation of the parameters RTP, RFP, and SDIS, refer to Section "Drilling, centering - CYCLE81 (Page 122)".



DP and DPR (long hole depth)

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

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

NUM (number)

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

LENG (long hole length)

The length of the long hole is programmed under LENG.

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

MID (infeed depth)

Use this parameter to define the maximum infeed depth.

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

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

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

FFD and FFP1 (feedrate for depth and surface)

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

CPA, CPO and RAD (center point and radius)

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

STA1 and INDA (starting and incremental angle)

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

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

Note

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

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

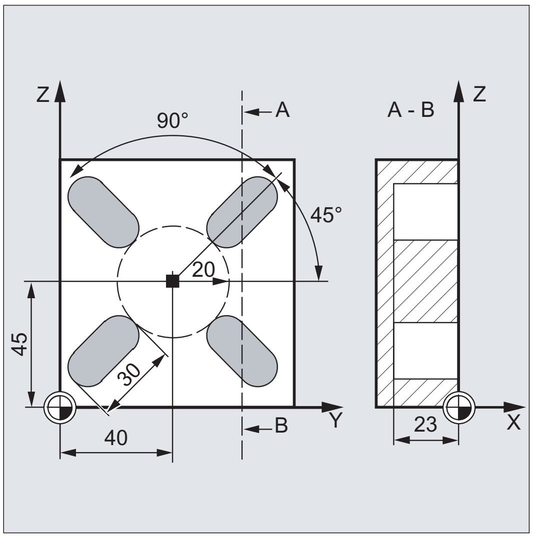
During the cycle, the workpiece coordinate system is offset and rotated. The values in the workpiece coordinate system are shown on the actual value display such that the longitudinal axis of the long hole being machined is positioned on the first axis of the current machining plane.

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

Programming example: Machining slots

By using this program, you can machine four slots of the length 30 mm and the relative depth 23 mm (difference between the reference plane and the slot root), which are arranged on a circle with the center point Y40 Z45 and the radius 20 mm in the YZ plane. The starting angle is 45 degrees, the incremental angle is 90 degrees. The maximum infeed depth is 6 mm, the safety clearance 1 mm.

See the following programming example for machining slots:




```
N10 G19 G90 D9 T10 S600 M3                   ; Specification of the
                                             technological values
N20 G0 Y50 Z25 X5                            ; Approach starting position
N30 LONGHOLE (5, 0, 1, , 23, 4, 30, 40, 45, 20,   ; Cycle call
45, 90, 100 , 320, 6)
N40 M02                                      ; End of program
```

2.6.7 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, DP1)

Parameter

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Slot depth (absolute)
DPR	REAL	Slot depth relative to the reference plane (enter without sign)
NUM	INT	Number of slots
LENG	REAL	Slot length (enter without sign)
WID	REAL	Slot width (enter without sign)
CPA	REAL	Center point of circle (absolute), first axis of the plane
CPO	REAL	Center point of circle (absolute), second axis of the plane
RAD	REAL	Radius of the circle (enter without sign)
STA1	REAL	Starting angle
INDA	REAL	Incrementing angle
FFD	REAL	Feedrate for depth infeed
FFP1	REAL	Feedrate for surface machining
MID	REAL	Maximum infeed depth for one infeed (enter without sign)
CDIR	INT	Milling direction for machining the slot Values: 2 (for G2), 3 (for G3)
FAL	REAL	Finishing allowance at the slot edge (enter without sign)
VARI	INT	Machining type Values: 0 = complete machining, 1 = roughing, 2 = finishing
MIDF	REAL	Maximum infeed depth for finishing
FFP2	REAL	Feedrate for finishing
SSF	REAL	Speed when finishing
FALD	REAL	Finishing allowance at the slot base (enter without sign)
STA2	REAL	Maximum insertion angle for oscillation movement
DP1	REAL	Insertion depth per revolution for helix (incremental)

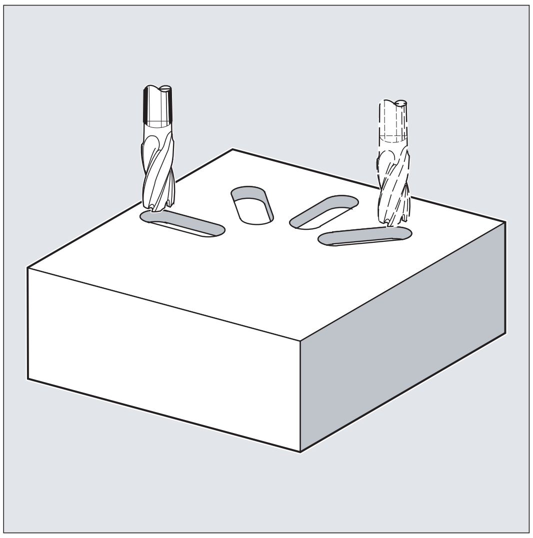
Note

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

Function

The cycle SLOT1 is a combined roughing-finishing cycle.

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



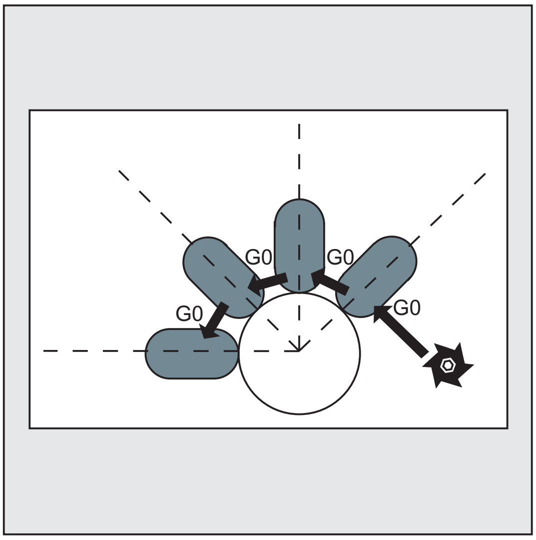
Sequence

Position reached prior to cycle start:

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

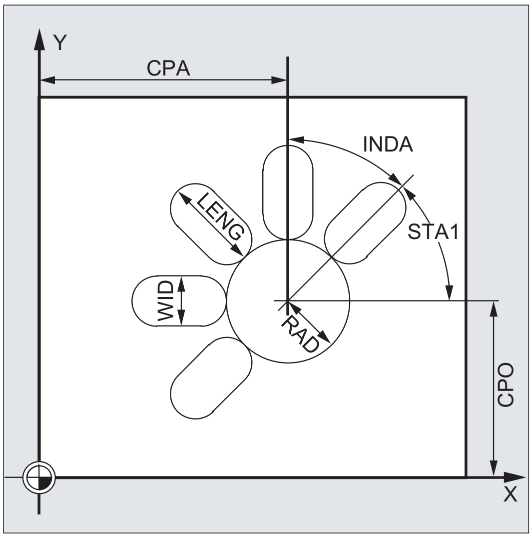
The cycle creates the following sequence of motions:

- Approach of the position at the beginning of the cycle indicated in the SLOT1 sequence illustration with G0.
- Complete machining of a slot is carried out in the following steps:
  - Approach of the reference plane brought forward by the safety clearance by using G0
  - Infeed to the next machining depth with G1 and with feedrate value FFD
  - Solid machining of the slot to the finishing allowance at the slot edge with feedrate value FFP1. Then finishing with feedrate value FFP2 and spindle speed SSF along the contour according to the machining direction programmed under CDIR.
  - The depth infeed is always carried out at the same position in the machining plane until the end depth of the slot is reached.
- Retract tool to the retraction plane and move to the next slot with G0.
- After the last slot has been machined, the tool is moved with G0 to the end position in the machining plane, which is specified in the diagram below, and the cycle is ended.



Explanation of the parameters

For an explanation of the parameters RTP, RFP, and SDIS, refer to Section "Drilling, centering - CYCLE81 (Page 122)".



DP and DPR (slot depth)

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

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

NUM (number)

Use the parameter NUM to specify the number of slots.

LENG and WID (slot length and slot width)

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

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

CPA, CPO and RAD (center point and radius)

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

STA1 and INDA (starting and incremental angle)

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

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

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

FFD and FFP1 (feedrate for depth and surface)

The feedrate FFD is active for all infeed movements perpendicular to the machining plane.

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

MID (infeed depth)

Use this parameter to define the maximum infeed depth.

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

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

The depth infeed commences at the reference plane moved forward by the safety clearance.

CDIR (milling direction)

Use this parameter to specify the machining direction for the groove. Possible values are:

- "2" for G2
- "3" for G3

If the parameter is set to an illegal value, then the message "Wrong milling direction, G3 will be generated" will be displayed in the message line. In this case, the cycle is continued and G3 is automatically generated.

FAL (finishing allowance)

Use this parameter to program a finishing allowance at the slot edge. FAL does not influence the depth infeed.

If the value of FAL is greater than allowed for the specified width and the milling cutter used, FAL is automatically reduced to the maximum possible value. In the case of roughing, milling is performed with a reciprocating movement and depth infeed at both end points of the slot.

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

Use the parameter VARI to define the machining type.

Possible values are:

- 0=complete machining in two parts
  - Solid machining of the slot (SLOT1, SLOT2) to the finishing allowance is performed at the spindle speed programmed before the cycle was called and with feedrate FFP1. Depth infeed is defined with MID.
  - Solid machining of the remaining finishing allowance is carried out at the spindle speed defined via SSF and the feedrate FFP2. Depth infeed is defined with MIDF.

    If MIDF=0, the infeed is performed right to the final depth.
  - If FFP2 is not programmed, feedrate FFP1 is active. This also applies analogously if SSF is not specified, i.e. the speed programmed prior to the cycle call will apply.
- 1=Roughing

  The groove (SLOT1, SLOT2) is solid-machined up to the finishing allowance at the speed programmed before the cycle call and at the feedrate FFP1. The depth infeed is programmed via MID.
- 2=Finishing

  The cycle requires that the slot (SLOT1, SLOT2) is already machined to a residual finishing allowance and that it is only necessary to machine the final finishing allowance. If FFP2 and SSF are not programmed, the feedrate FFP1 or the speed programmed before the cycle call is active. Depth infeed is defined with MIDF.

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

FALD (finishing allowance at slot edge)

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

DP1

Use the parameter DP1 to define the infeed depth when inserting to the helical path.

STA2 (insertion angle)

Use the STA2 parameter to define the radius of the helical path (relative to the tool center point path) or the maximum insertion angle for the reciprocating motion.

- Vertical insertion

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

- Insertion oscillation on center axis of slot

  It means that the milling center point on a straight line oscillating back and forth is inserted at an angle until it has reached the nearest current depth. The maximum insertion angle is programmed under STA2, and the length of the oscillation path is calculated from LENG-WID. The oscillating depth infeed ends at the same point as with vertical depth infeed motions; the starting point in the plane is calculated accordingly. The roughing operation begins in the plane once the current depth is reached. The feedrate is programmed under FFD.

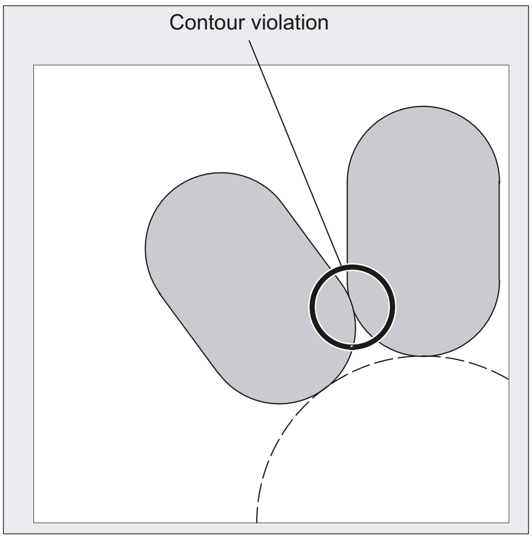
Note

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

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

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

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



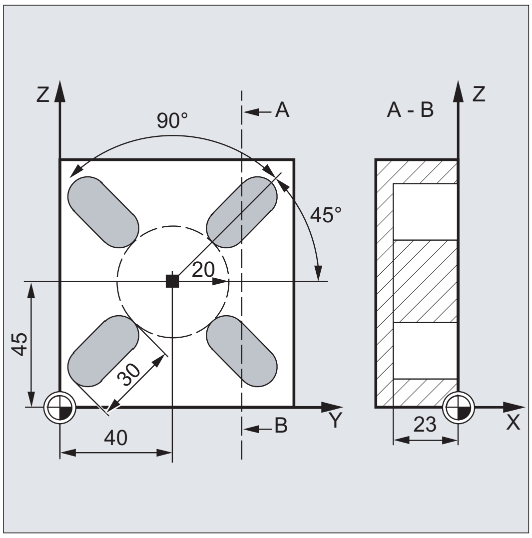
Programming example: Grooves

Four slots are milled.

The slots have the following dimensions: Length 30 mm, width 15 mm and depth 23 mm. The safety clearance is 1 mm, the final machining allowance is 0.5 mm, the milling direction is G2, the maximum infeed in the depth is 6 mm.

The slot is to be machined completely. Infeed during finishing is to be performed directly to the pocket depth and the same feedrate and speed are to be used.

See the following programming example for grooves:



`N10 G17 G90 T1 D1 S600 M3`	`; Specification of technology values`
`N20 G0 X20 Y50 Z5`	`; Approach starting position`
`N30 SLOT1(5, 0, 1, -23, , 4, 30, 15, 40, 45, 20, 45, 90, 100, 320, 6, 2, 0.5, 0, , 0, )`	`; Cycle call, VARI, MIDF, FFP2 and SSF parameters omitted`
`N40 M02`	`; End of program`

2.6.8 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

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Slot depth (absolute)
DPR	REAL	Slot depth relative to the reference plane (enter without sign)
NUM	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 (absolute), first axis of the plane
CPO	REAL	Center point of circle (absolute), second axis of the plane
RAD	REAL	Radius of the circle (enter without sign)
STA1	REAL	Starting angle
INDA	REAL	Incrementing angle
FFD	REAL	Feedrate for depth infeed
FFP1	REAL	Feedrate for surface machining
MID	REAL	Maximum infeed depth for one infeed (enter without sign)
CDIR	INT	Milling direction for machining the circumferential slot Values: 2 (for G2), 3 (for G3)
FAL	REAL	Finishing allowance at the slot edge (enter without sign)
VARI	INT	Machining type Values: 0 = complete machining, 1 = roughing, 2 = finishing
MIDF	REAL	Maximum infeed depth for finishing
FFP2	REAL	Feedrate for finishing
SSF	REAL	Speed when finishing
FFCP	REAL	Feedrate for intermediate positioning on a circular path, in mm/min

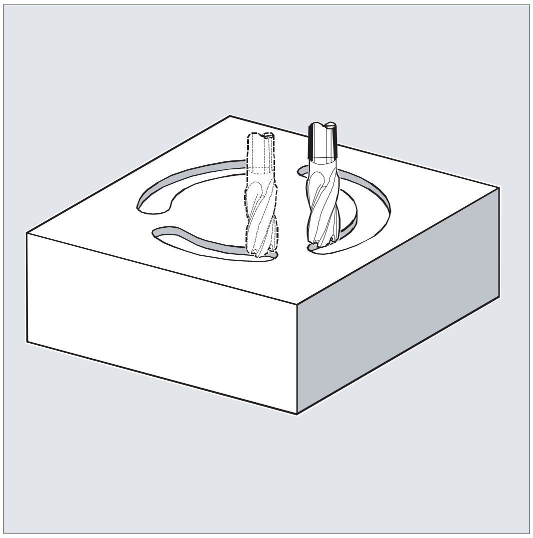
Note

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

Function

The cycle SLOT2 is a combined roughing-finishing cycle.

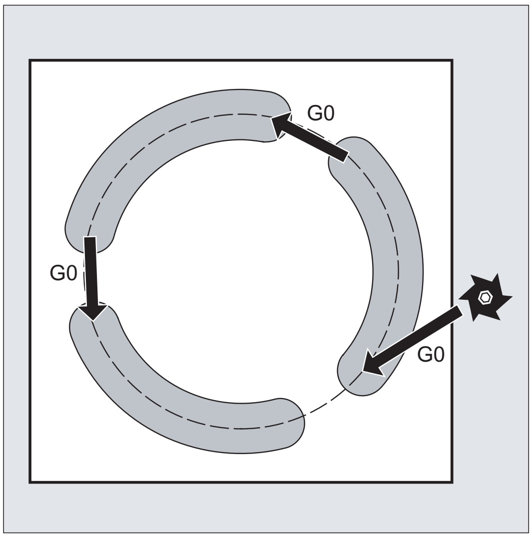
Use this cycle to machine circumferential slots arranged on a circle.



Sequence

Position reached prior to cycle start:

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



The cycle creates the following sequence of motions:

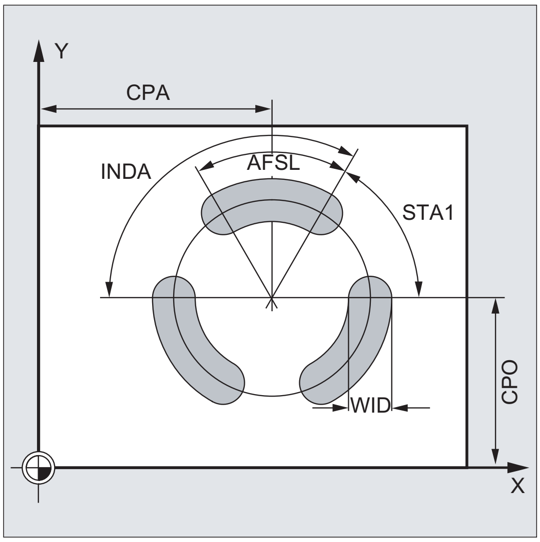
- G0 is used to approach the position specified in the diagram below at cycle start.
- The steps when machining a circumferential slot are the same as when machining an elongated hole.
- After a circumferential slot is machined completely, the tool is retracted to the retraction plane and the next slot is machined with G0.

- After the last slot has been machined, the tool is moved with G0 to the end position in the machining plane, which is specified in the diagram below, and the cycle is ended.

Explanation of the parameters

For an explanation of the parameters RTP, RFP, and SDIS, refer to Section "Drilling, centering - CYCLE81 (Page 122)".

For an explanation of the parameters DP, DPR, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, and SSF, refer to Section "Slots on a circle - SLOT1 (Page 205)".



NUM (number)

Use the parameter NUM to specify the number of slots.

AFSL and WID (angle and circumferential slot width)

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

CPA, CPO and RAD (center point and radius)

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

FFCP

Use the parameter FFCP to program a special feedrate for intermediate positioning on circular path.

STA1 and INDA (starting and incremental angle)

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

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

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

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

Note

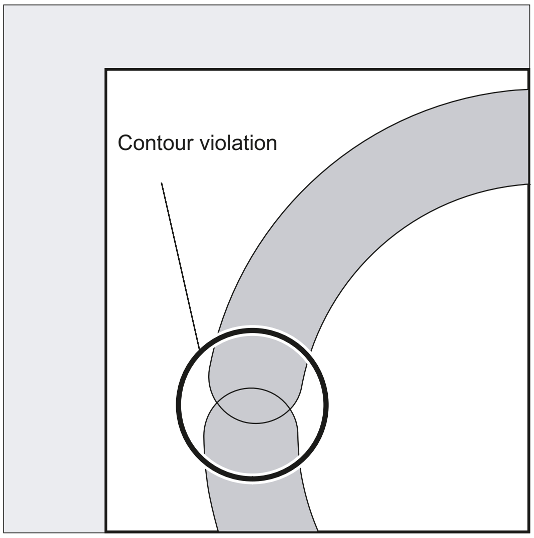
A tool compensation must be programmed before the cycle is called. Otherwise, the cycle is aborted and alarm 61000 "No tool compensation active" is output.

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

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

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

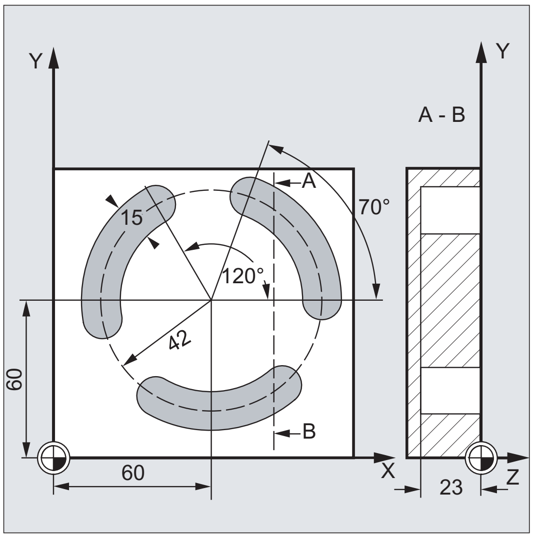
After the cycle has been completed, the workpiece coordinate system is in the same position again as it was before the cycle was called.



Programming example1: Slots2

Use this program to machine three circumferential slots arranged at a circle with center point X60 Y60 and radius 42 mm in the XY plane. The circumferential slots have the following dimensions: Width 15 mm, angle for slot length 70 degrees, depth 23 mm. The initial angle is 0 degree, the incremental angle is 120 degrees. The slot contours are machined to a final machining allowance of 0.5 mm, the safety clearance in infeed axis Z is 2 mm, the maximum depth infeed is 6 mm. The slots are to be completely machined. Speed and feedrate are to be the same when finishing. The infeed when finishing is to be performed to slot depth.

See the following programming example for circumferential slot:

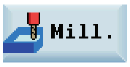


`N10 G17 G90 T1 D1 S600 M3`	`; Specification of technology values`
`N20 G0 X60 Y60 Z5`	`; Approach starting position`
`N30 SLOT2(2, 0, 2, -23, , 3, 70, 15, 60, 60, 42, , 120, 100, 300, 6, 2, 0.5, 0, , 0, )`	`; Cycle call` `Reference plane+SDIS=retraction plane means: Lowering in the infeed axis with G0 to reference plane+SDIS no longer applicable, parameters VAR, MIDF, FFP2 and SSF omitted`
`N40 M02`	`; End of program`

Programming example 2: Slots2

Proceed through the following steps:

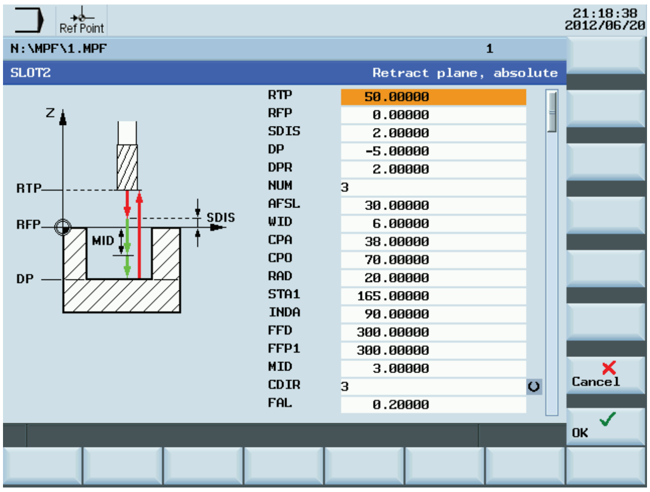
1. Select the desired operating area.

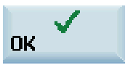


2. Open the vertical softkey bar for available milling cycles.

3. Press this softkey from the vertical softkey bar.

4. Press this softkey to open the window for SLOT2. Parameterize the cycle as desired.





5. Confirm your settings with this softkey. The cycle is then automatically transferred to the program editor.

2.6.9 Milling a rectangular pocket - POCKET3

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

Parameter	Data type	Description
_RTP	REAL	Retraction plane (absolute)
_RFP	REAL	Reference plane (absolute)
_SDIS	REAL	Safety clearance (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	Reference point for the pocket (absolute), first axis of the plane
_PO	REAL	Reference point for the pocket (absolute), second axis of the plane
_STA	REAL	Angle between the pocket longitudinal axis and the first axis of the plane (enter without sign); Range of values: 0° ≤ STA < 180°
_MID	REAL	Maximum infeed depth (enter without sign)
_FAL	REAL	Finishing allowance at the pocket edge (enter without sign)
_FALD	REAL	Finishing allowance at the base (enter without sign)
_FFP1	REAL	Feedrate for surface machining
_FFD	REAL	Feedrate for depth infeed
_CDIR	INT	Milling direction: (enter without sign) Values: 0: Down-cut milling (in the spindle direction) 1: Conventional milling 2: With G2 (independent of spindle direction) 3: With G3
_VARI	INT	Machining type UNITS DIGIT Values: 1: roughing, 2: finishing TENS DIGIT: Values: 0: Perpendicular to the pocket center with G0 1: Perpendicular to the pocket center with G1 2: Along a helix 3: Oscillation along the pocket longitudinal axis

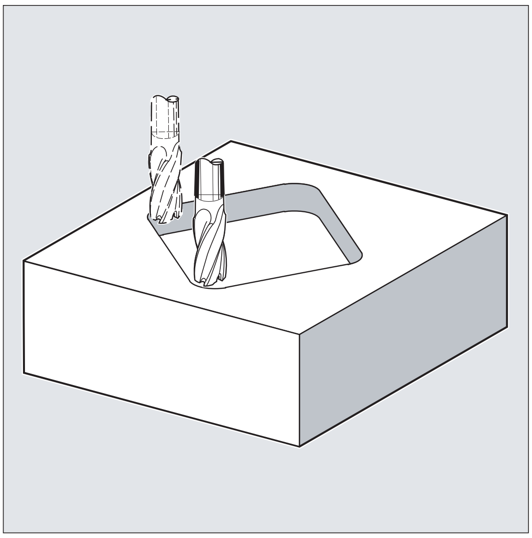
Parameter	Data type	Description
The other parameters can be selected as options. They define the insertion strategy and the overlap for solid machining (to be entered without sign):		
_MIDA	REAL	Maximum infeed width as a value in solid machining in the plane
_AP1	REAL	Blank dimension of pocket length
_AP2	REAL	Blank dimension of pocket width
_AD	REAL	Blank pocket depth dimension from reference plane
_RAD1	REAL	Radius of the helical path on insertion (relative to the tool center point path) or maximum insertion angle for reciprocating motion
_DP1	REAL	Insertion depth per 360° revolution on insertion along helical path

Function

The cycle can be used for roughing and finishing. For finishing, a face cutter is required.

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

- The milling direction can be determined either by using a G command (G2/G3) or from the spindle direction as synchronous or up-cut milling.
- For solid machining, the maximum infeed width in the plane can be programmed.
- Finishing allowance also for the pocket base
- There are three different insertion strategies:
  - vertically to the pocket center
  - along a helical path around the pocket center
  - oscillating at the pocket central axis
- Shorter approach paths in the plane for finishing
- Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).



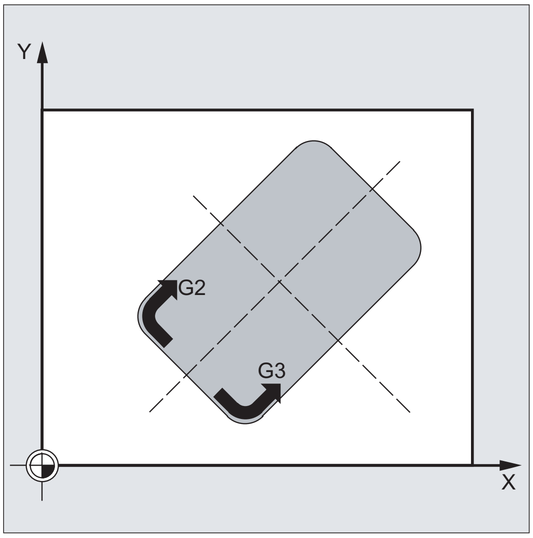
Sequence

Position reached prior to cycle start:

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

Sequence of motions when roughing:

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



Sequence of motions when finishing:

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

- Finishing on the edge

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

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

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

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

- Finishing on the base

  During finishing on the base, the machine performs G0 towards the pocket center until reaching a distance equal to pocket depth + finishing allowance + safety clearance. From this point onwards, the tool is always fed in vertically at the depth (since a tool with a front cutting edge is used for base finishing).

  The base surface of the pocket is machined once.

Insertion strategies

- Inserting vertically to the pocket center means that the current infeed depth calculated internally in the cycle (≤ maximum infeed depth programmed under _MID) is executed in a block containing G0 or G1.
- Insertion at a helical path means that the cutter center point traverses along the helical path determined by the radius _RAD1 and the depth per revolution _DP1. The feedrate is also programmed under _FFD. The direction of rotation of this helical path corresponds to the direction of rotation with which the pocket will be machined.

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

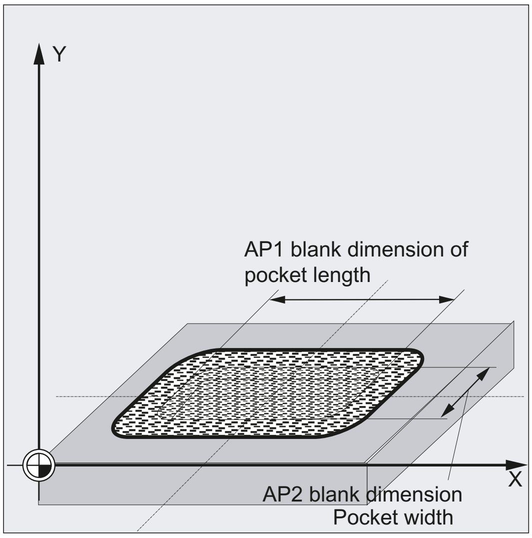
  If the current depth required for an infeed (this can be several revolutions on the helical path) is reached, a full circle is still executed to eliminate the inclined path of insertion.

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

  The starting point of the described helical path is at the longitudinal axis of the pocket in "plus direction" and is approached with G1.
- Insertion with oscillation to the central axis of the pocket means that the cutter center point is inserted oscillating on a straight line until it reaches the next current depth. The maximum immersion angle is programmed under _RAD1, and the length of the oscillation travel is calculated in the cycle. If the current depth is reached, the travel is executed once more without depth infeed in order to eliminate the inclined insertion path. The feedrate is programmed under _FFD.

Taking into account the blank dimensions

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



The basic sizes for the length and width (_AP1 and _AP2) are programmed without sign and their symmetrical positions around the pocket center point are computed in the cycle. You define the part of the pocket which is no longer to be machined by solid machining. The blank dimension for the depth (_AD) is also programmed without sign and taken into account by the reference plane in the direction of the pocket depth.

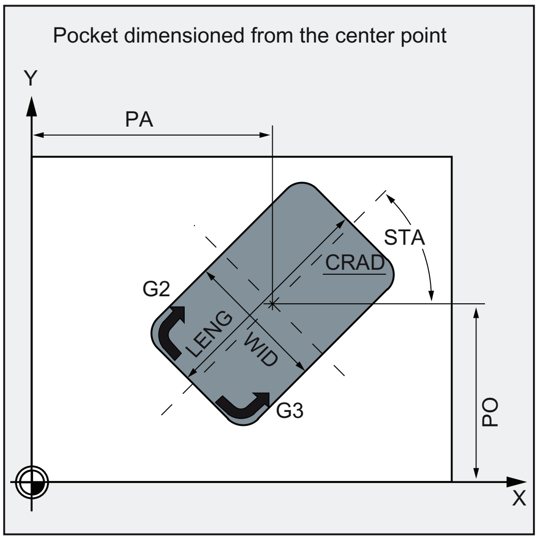
The depth infeed when taking into account blank dimensions is carried out according to the programmed type (helical path, reciprocating, vertically). If the cycle detects that there is space enough in the pocket center because of the given blank contour and the radius of the active tool, the infeed is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

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

Explanation of the parameters

For an explanation of the parameters _RTP, _RFP, and _SDIS, refer to Section "Drilling, centering - CYCLE81 (Page 122)".

For an explanation of the _DP parameter, refer to Section "Long holes located on a circle - LONGHOLE (Page 200)".



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

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

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

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

_PA, _PO (reference point)

Use the parameters _PA and _PO to define the reference point of the pocket in the axes of the plane. This is the pocket center point.

_STA (angle)

_STA indicates the angle between the first axis of the plane (abscissa) and the longitudinal axis of the pocket.

_MID (infeed depth)

Use this parameter to define the maximum infeed depth when roughing.

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

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

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

_FAL (finishing allowance at edge)

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

If the final machining allowance ≥ tool diameter, the pocket will not necessarily be machined completely. The message "Caution: final machining allowance ≥ tool diameter" appears; the cycle, however, is continued.

_FALD (finishing allowance at the base)

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

_FFD and _FFP1 (feedrate for depth and surface)

The feedrate _FFD is effective when inserting into the material.

The feedrate _FFP1 is active for all movements in the plane traversed at feedrate when machining.

_CDIR (milling direction)

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

Using the parameter _CDIR, the milling direction can be programmed directly with "2 for G2" and "3 for G3", or alternatively with "synchronous milling" or "conventional milling".

Synchronized operation or reverse rotation are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

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

_VARI (machining type)

Use the parameter VARI to define the machining type.

Possible values are:

Units digit:

- 1=roughing
- 2=finishing

Tens digit (infeed):

- 0=vertically to pocket center with G0
- 1=vertically to pocket center with G1
- 2=along a helical path
- 3=oscillating to pocket length axis

If a different value is programmed for the parameter _VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

_MIDA (max. infeed width)

Use this parameter to define the maximum infeed width when solid machining in a plane. Analogously to the known calculation method for the infeed depth (equal distribution of the total depth with maximum possible value) the width is distributed equally, maximally with the value programmed under _MIDA.

If this parameter is not programmed or has value 0, the cycle will internally use 80% of the milling tool diameter as the maximum infeed width.

Note

Applies if the calculated width infeed from edge machining is recalculated when reaching the full pocket in the depth; otherwise the width infeed calculated at the beginning is kept for the whole cycle.

_AP1, _AP2, _AD (blank dimensions)

Use the parameters _AP1, _AP2 and _AD to define the blank dimensions (incremental) of the pocket in the plane and in the depth.

_RAD1 (radius)

Use the _RAD1 parameter to define the radius of the helical path (relative to the tool center point path) or the maximum insertion angle for the reciprocating motion.

_DP1 (insertion depth)

Use the parameter _DP1 to define the infeed depth when inserting to the helical path.

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

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

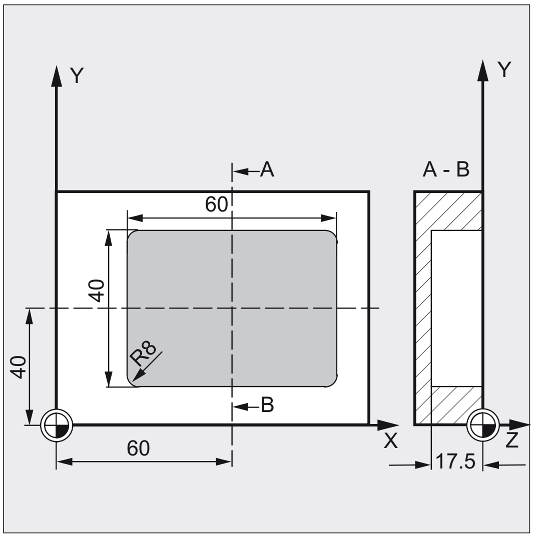
Programming example: Pocket

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

The machining direction results from the direction of rotation of the spindle in the case of down-cut milling. A milling cutter with 5 mm radius is used.

Merely a rough machining operation is to be carried out.

See the following programming example for rectangular pocket:



N10 G90 T1 D1 S600 M4	; Specification of technology values
N20 G17 G0 X60 Y40 Z5	; Approach starting position
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 M02	; End of program

2.6.10 Milling a circular pocket - POCKET4

Programming

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

Parameters

Parameter	Data type	Description
_RTP	REAL	Retraction plane (absolute)
_RFP	REAL	Reference plane (absolute)
_SDIS	REAL	Safety clearance (to be added to the reference plane; enter without sign)
_DP	REAL	Pocket depth (absolute)
_PRAD	REAL	Pocket radius
_PA	REAL	Starting point (absolute), first axis of the plane
_PO	REAL	Starting point (absolute), second axis of the plane
_MID	REAL	Maximum infeed depth (enter without sign)
_FAL	REAL	Finishing allowance at the pocket edge (enter without sign)
_FALD	REAL	Finishing allowance at the base (enter without sign)
_FFP1	REAL	Feedrate for surface machining
_FFD	REAL	Feedrate for depth infeed
_CDIR	INT	Milling direction: (enter without sign) Values: 0: Down-cut milling (in the spindle direction) 1: Conventional milling 2: With G2 (independent of spindle direction) 3: With G3
_VARI	INT	Machining type UNITS DIGIT Values: 1: roughing, 2: finishing TENS DIGIT: Values: 0: Perpendicular to the pocket center with G0 1: Perpendicular to the pocket center with G1 2: Along a helix
The other parameters can be selected as options. They define the insertion strategy and the overlap for solid machining (to be entered without sign):		
_MIDA	REAL	Maximum infeed width as a value in solid machining in the plane
_AP1	REAL	Pocket radius blank dimension
_AD	REAL	Blank pocket depth dimension from reference plane
_RAD1	REAL	Radius of the helical path during insertion (relative to the tool center point path)
_DP1	REAL	Insertion depth per 360° revolution on insertion along helical path

Function

Use this cycle to machine circular pockets in the machining plane. For finishing, a face cutter is required.

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

- The milling direction can be determined either using a G command (G2/G3) or from the spindle direction as synchronous or up-cut milling.
- For solid machining, the maximum infeed width in the plane can be programmed.
- Finishing allowance also for the pocket base.
- Two different insertion strategies:
  - vertically to the pocket center
  - 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 blank dimension at the base (optimum machining of preformed pockets possible).
- _MIDA is recalculated during edge machining.

Sequence

Position reached prior to cycle start:

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

Motion sequence when roughing (_VARI=X1):

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

Sequence of motions when finishing:

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

- Finishing on the edge

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

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

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

- Finishing on the base

  During finishing on the base, the machine performs G0 towards the pocket center until reaching a distance equal to pocket depth + finishing allowance + safety clearance. From this point onwards, the tool is always fed in **vertically** at the depth (since a tool with a front cutting edge is used for base finishing).

  The base surface of the pocket is machined once.

Insertion strategies

Refer to Section "Milling a rectangular pocket - POCKET3 (Page 218)".

Taking into account the blank dimensions

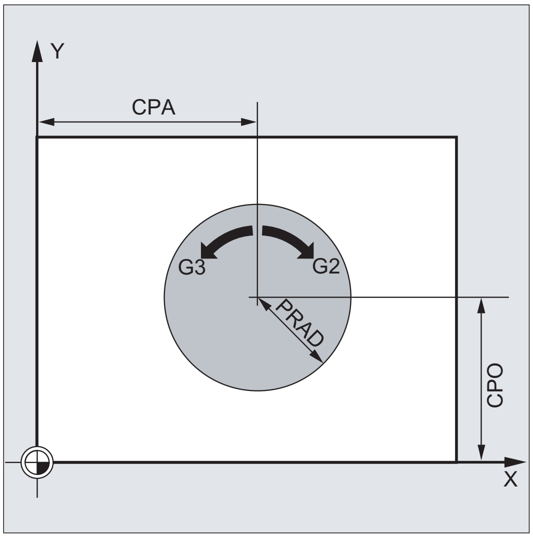
During solid machining of the pockets, it is possible to take into account blank dimensions (e.g. when machining precast parts).

With circular pockets, the blank dimension _AP1 is also a circle (with a smaller radius than the pocket radius).

Explanation of the parameters

For an explanation of the parameters _RTP, _RFP, and _SDIS, refer to Section "Drilling, centering - CYCLE81 (Page 122)".

For an explanation of the parameters _DP, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _MIDA, _AP1, _AD, _RAD1, and _DP1, refer to Section "Milling a rectangular pocket - POCKET3 (Page 218)".



_PRAD (pocket radius)

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

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

_PA, _PO (pocket center point)

Use the parameters _PA and _PO to define the pocket center point. Circular pockets are always dimensioned across the center.

_VARI (machining type)

Use the parameter _VARI to define the machining type.

Possible values are:

Units digit:

- 1=roughing
- 2=finishing

Tens digit (infeed):

- 0=vertically to pocket center with G0
- 1=vertically to pocket center with G1
- 2=along a helical path

If a different value is programmed for the parameter _VARI, the cycle is aborted after output of alarm 61002 "Machining type defined incorrectly".

Note

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

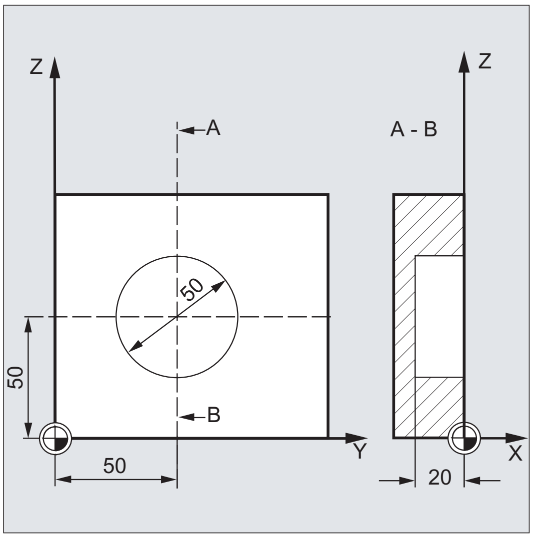
Internally in the cycle, a new current workpiece coordinate system is used which influences the actual value display. The zero point of this coordinate system is to be found in the pocket center point.

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

Programming example: Circular pocket

With this program, you can machine a circular pocket in the YZ plane. The center point is determined by Y50 Z50. The infeed axis for the depth infeed is the X axis. Neither finishing dimension nor safety clearance is specified. The pocket is machined with down-cut milling. Infeed is performed along a helical path.

A milling cutter with 10 mm radius is used. See the following programming example for circular pocket:



N10 G17 G90 G0 S650 M3 T1 D1	; Specification of technology values
N20 X50 Y50	; Approach starting position
N30 POCKET4(3, 0, 0, -20, 25, 50, 60, 6, 0, 0, 200, 100, 1, 21, 0, 0, 0, 2, 3)	; Cycle call Parameters FAL and FALD are omitted
N40 M02	; End of program

2.6.11 Thread milling - CYCLE90

Programming

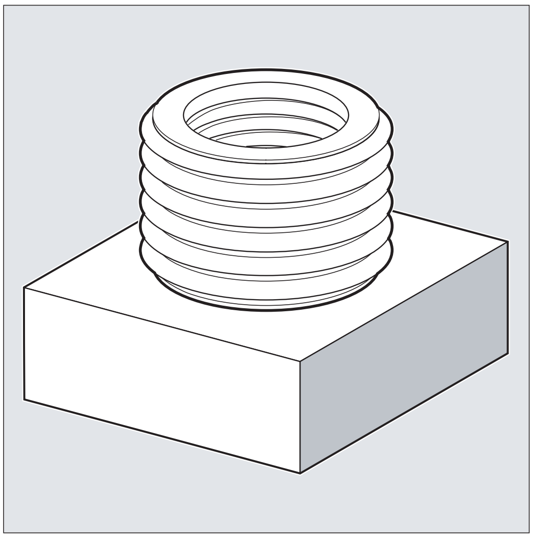
CYCLE90 (RTP, RFP, SDIS, DP, DPR, DIATH, KDIAM, PIT, FFR, CDIR, TYPTH, CPA, CPO)

Parameters

Parameter	Data type	Description
RTP	REAL	Retraction plane (absolute)
RFP	REAL	Reference plane (absolute)
SDIS	REAL	Safety clearance (enter without sign)
DP	REAL	Final drilling depth (absolute)
DPR	REAL	Final drilling depth relative to the reference plane (enter without sign)
DIATH	REAL	Nominal diameter, outer diameter of the thread
KDIAM	REAL	Core diameter, internal diameter of the thread
PST	REAL	Thread pitch; value range: 0.001 ... 2000.000 mm
FFR	REAL	Feedrate for thread milling (enter without sign)
CDIR	INT	Direction of rotation for thread milling Values: 2 (for thread milling with G2), 3 (for thread milling with G3)
TYPTH	INT	Thread type Values: 0=internal thread, 1=external thread
CPA	REAL	Center point of circle, abscissa (absolute)
CPO	REAL	Center point of circle, ordinate (absolute)

Function

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



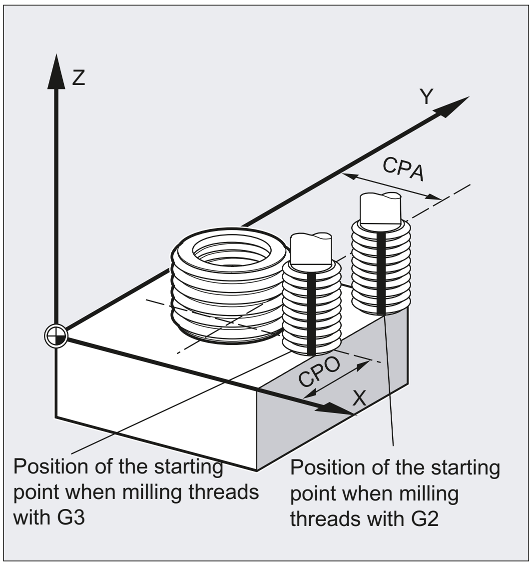
Sequence for external thread

Position reached prior to cycle start:

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

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

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



The cycle creates the following sequence of motions:

- Positioning on the starting point using G0 at the height of the retraction plane in the applicate of the current plane
- Infeed to the reference plane brought forward by the safety clearance for swarf removal, using G0
- Approach motion to the thread diameter along a circle path opposite to the direction G2/G3 programmed under CDIR
- Thread milling along a helix path using G2/G3 and the feedrate value FFR
- Retraction motion along a circle path in the opposite direction of rotation G2/G3 at the reduced feedrate FFR
- Retraction to the retraction plane along the applicate using G0

Sequence for internal thread

Position reached prior to cycle start:

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

The cycle creates the following sequence of motions:

- Positioning on the center point using G0 at the height of the retraction plane in the applicate of the current plane
- Infeed to the reference plane brought forward by the safety clearance for swarf removal, using G0
- Approach to an approach circle calculated internally in the cycle using G1 and the reduced feedrate FFR
- Approach motion to the thread diameter along a circle path according to the direction G2/G3 programmed under CDIR
- Thread milling along a helix path using G2/G3 and the feedrate value FFR
- Retraction motion along a circle path in the same direction of rotation at the reduced feedrate FFR
- Retraction to the center point of the thread using G0
- Retraction to the retraction plane along the applicate using G0

Thread from bottom to top

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

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

Programming example (thread from bottom to top)

A thread with a pitch of 3 mm is to start from -20 and to be milled to 0. The retraction plane is at 8.

```
N10 G17 X100 Y100 S300 M3 T1 D1 F1000
N20 Z8
N30 CYCLE90 (8, -20, 0, -60, 0, 46, 40, 3, 800,
3, 0, 50, 50)
N40 M2
```

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

Overshooting in the direction of the thread length

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

The overshoot is calculated as follows:

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

Δz: Overshoot, internal

p: Pitch

WR: Tool radius

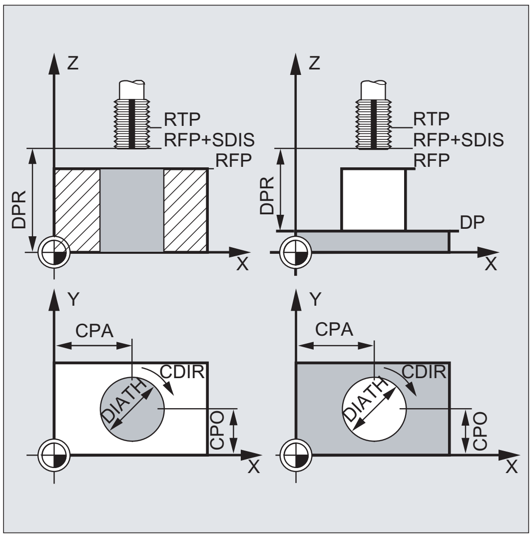
DIATH: External diameter of the thread

RDIFF: Radius difference for travel-out circle

For internal threads, RDIFF = DIATH/2 - WR; for external threads, RDIFF = DIATH/2 + WR.

Explanation of the parameters

For an explanation of the parameters RTP, RFP, SDIS, DP, and DPR, refer to Section "Drilling, centering - CYCLE81 (Page 122) ".



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

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

FFR (feedrate)

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

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

CDIR (direction of rotation)

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

If the parameter has an illegal value, the following message will appear:

"Wrong milling direction; G3 is generated".

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

TYPTH (thread type)

The parameter TYPTH is used to define whether you want to machine an external or an internal thread.

CPA and CPO (center point)

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

Note

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

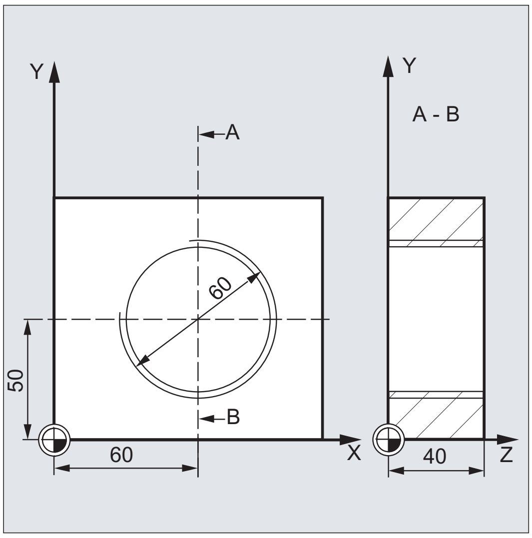
If the tool radius=0 or negative, the cycle is also aborted and this alarm is issued.

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

Programming example: Internal thread

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

See the following programming example for internal thread:



`DEF REAL RTP=48, RFP=40, SDIS=5, DP=0, DPR=40, DIATH=60, KDIAM=50`	; Definition of the variable with value assignments
`DEF REAL PIT=2, FFR=500, CPA=60,CPO=50`	
`DEF INT CDIR=2, TYPTH=0`	
`N10 G90 G0 G17 X0 Y0 Z80 S200 M3`	; Approach starting position
`N20 T5 D1`	; Specification of technology values
`N30 CYCLE90 (RTP, RFP, SDIS, DP, DPR, DIATH, KDIAM, PIT, FFR, CDIR, TYPTH, CPA, CPO)`	; Cycle call
`N40 G0 G90 Z100`	; Approach position after cycle
`N50 M02`	; End of program

2.6.12 High speed settings - CYCLE832

Programming

CYCLE832 (TOL, TOLM, 1)

Parameters

Parameter	Data type	Description
TOL	REAL	Tolerance of machining axes
TOLM	INT	Machining type selection 0: Deselect 1: Finishing 2: Semi-finishing 3: Roughing
PSYS	INT	Internal parameter, only the default value 1 is possible

Function

Use CYCLE832 to machine free-form surfaces that involve high requirements for velocity, precision and surface quality.

This cycle function groups together the important G codes, machine data and setting data that are required for high-speed cutting machining.

Explanation of the parameters

TOL (Tolerance)

This refers to the tolerance of axes involved in machining. The tolerance value is written to the relevant machine or setting data depending on the G codes.

TOLM (Machining types)

This parameter determines which technological machining type is to be used.

2.7 Error messages and error handling

2.7.1 General Information

If error conditions are detected in the cycles, an alarm is generated and the execution of the cycle is aborted.

Furthermore, the cycles display their messages in the message line of the control system. These messages do not interrupt the program execution.

The errors with their reactions and the messages in the message line of the control system are described in conjunction with the individual cycles.

2.7.2 Error handling in the cycles

If error conditions are detected in the cycles, an alarm is generated and the machining is aborted.

Alarms with numbers between 61000 and 62999 generated in the cycles. This range of numbers, in turn, is divided again with regard to alarm responses and cancel criteria.

The error text that is displayed together with the alarm number gives you more detailed information on the error cause.

Alarm number	Clearing criterion	Alarm response
61000 ... 61999	NC_RESET	Block preparation in the NC is aborted
62000 ... 62999	Clear key	The block preparation is interrupted; the cycle can be continued with the following key after the alarm has been cleared: 

2.7.3 Overview of cycle alarms

The error numbers are classified as follows:

6	_	X	_	_

- X=0 General cycle alarms
- X=1 Alarms generated by the drilling, drilling pattern and milling cycles

2.7.4 Messages in the cycles

The cycles display their messages in the message line of the control system. These messages do not interrupt the program execution.

Messages provide information with regard to a certain behavior of the cycles and with regard to the progress of machining and are usually kept beyond a machining step or until the end of the cycle. An example of messages is as follows:

"Depth: according to the value for the relative depth" from all drilling cycles.

3 Typical milling program

Blank data

Blank material: Cube aluminum

Blank length: 100 mm

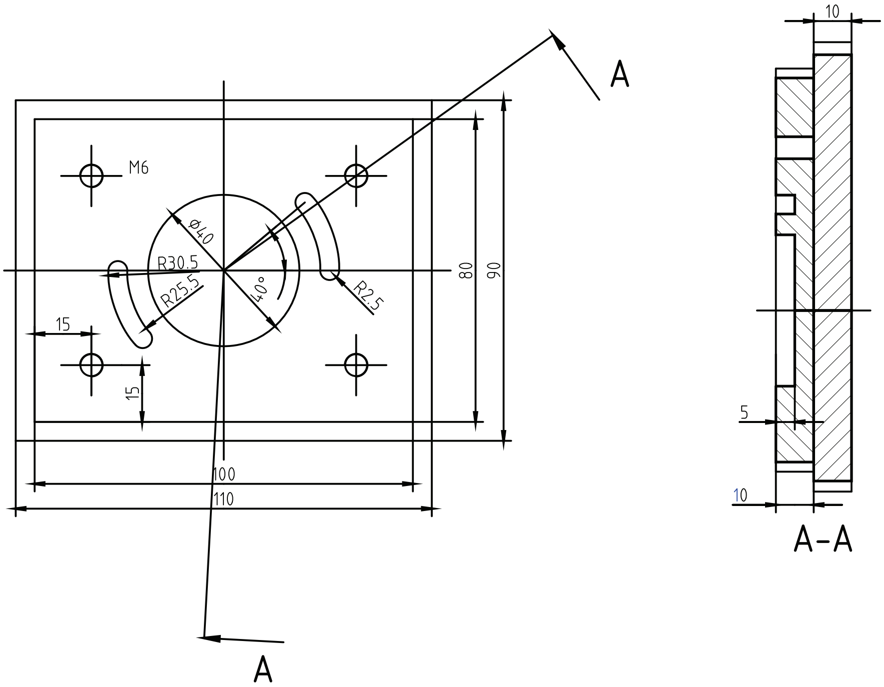
Blank width: 80 mm

Blank height: 60 mm (machining length: 46 mm; clamping length: 10 mm)

Required tools

T1, T2, T3, T4, T5, T6, T11, T14, T20

Programming example1



T1

M06

G54G90

S4000M3

CYCLE71(20.00000, 2.00000, 2.00000, 0.00000, -50.00000, -40.00000, 100.00000, 80.00000, ,5.00000, 30.00000, ,0.20000, 1500.00000, 31,)

CYCLE71(20.00000, 2.00000, 2.00000, 0.00000, -50.00000, -40.00000, 100.00000, 80.00000, ,2.00000, 30.00000, ,0.20000, 1500.00000, 12,)

```
T2
M06
S4000M3
CYCLE76( 20.00000, 0.00000, 2.00000, -10.00000, ,90.00000, 70.00000, 1.00000, 0.00000,
0.00000, ,3.00000, 0.50000, ,1200.00000, 1000.00000, 0, 1, 100.00000, 80.00000)
POCKET4( 20.00000, 0.00000, 2.00000, -5.00000, 20.00000, 0.00000, 0.00000, 2.00000,
0.50000, 0.20000, 1000.00000, 200.00000, 0, 21, 5.00000, , ,2.00000, 2.00000)

T3
M06
M8
S5000M3
CYCLE76( 20.00000, 0.00000, 2.00000, -10.00000, ,90.00000, 70.00000, 1.00000, 0.00000,
0.00000, ,12.00000, 0.50000, ,1000.00000, 1000.00000, 0, 2, 100.00000, 80.00000)
POCKET4( 20.00000, 0.00000, 2.00000, -5.00000, 20.00000, 0.00000, 0.00000, 6.00000,
0.50000, 0.20000, 1000.00000, 1000.00000, 0, 12, 5.00000, , ,2.00000, 2.00000)
T20
M06
S4000M3
M8
SLOT2( 20.00000, 0.00000, 2.00000, -5.00000, ,2, 40.00000, 5.00000, 0.00000, 0.00000,
28.00000, 0.00000, 180.00000, 300.00000, 500.00000, 2.00000, 3, 0.10000, 0, 5.00000,
500.00000, 5000.00000, 500.00000)
T11
M06
S1200M3
MCALL CYCLE83( 20.00000, 0.00000, 2.00000, -10.00000, 0.00000, -5.00000, 5.00000,
1.00000, 0.10000, ,1.00000, 0, 3, 2.00000, 1.00000, 0.10000, 1.00000)
X-35Y-25
X35Y-25
X-35Y25
X35Y25
MCALL
T14
M06
M05
```

MCALL CYCLE84(20.00000, 0.00000, 2.00000, -8.00000, 0.00000, 0.10000, 5, ,1.00000, 0.00000, 600.00000, 800.00000, 3, 0, 0, 1, 3.00000, 1.00000)

X-35Y-25

X35Y-25

X-35Y25

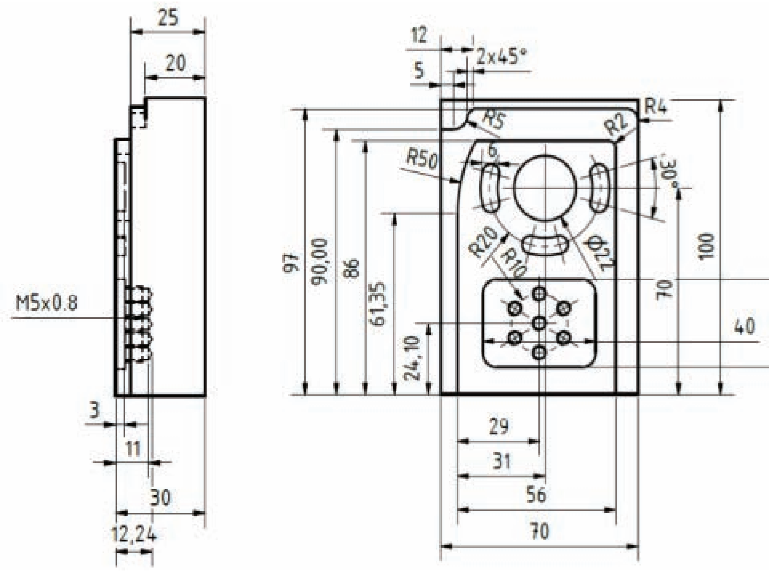
X35Y25

MCALL

G0Z100

M30

Programming example2



N5 G17 G90 G54 G71

N10 SUPA G00 Z300 D0

N15 SUPA G00 X300 Y300

N20 T1 D1

N25 MSG ("Please change to Tool No 1")

N30 M05 M09 M00

N35S4000 M3

N40 CYCLE71 (50.00000, 2.00000, 2.00000, 0.00000, 0.00000, 0.00000, 70.00000, 100.00000, 0.00000, 2.00000, 40.00000, 2.00000, 0.20000, 500.00000, 41, 5.00000)

N45 S4500 M3

```
N50 CYCLE71(50,2,2,0,0,0,70,100,0,2,40,2,0.2,300,22,5)

N55 SUPA G00 Z300 D0
N60 SUPA G00 X300 Y300
N65 T3 D1
N70 MSG ("Please change to Tool No 3")
N75 M05 M09 M00

N80 S5000 M3 G94 F300
N85 G00 X-6 Y92
N90 G00 Z2
N95 G01 F300 Z-10
N100 G41 Y 90
N105 G01 X10 RND=5
N110 G01 Y97 CHR=2
N115 G01 X70 RND=4
N120 G01 Y90
N125 G01 G40 X80
N130 G00 Z50

N135 SUPA G00 Z300 D0
N140 SUPA G00 X300 Y300
N145 T4 D1
N150 MSG("Please change to Tool No 4")
N155 M05 M09 M00

N160 S5000 M3
N165 POCKET4 ( 50.00000, 0.00000, 2.00000, -5.00000, 22.00000, 38.00000, 70.00000,
2.50000, 0.20000, 0.20000, 300.00000, 250.00000, 0, 21, 10.00000, 0.00000, 5.00000,
2.00000, 0.50000)
N170 S5500 M3
N175 POCKET4 ( 50.00000, 0.00000, 2.00000, -5.00000, 22.00000, 38.00000, 70.00000,
2.50000, 0.20000, 0.20000, 250.00000, 250.00000, 0, 22, 10.00000, 0.00000, 5.00000,
2.00000, 0.50000)

N180 SUPA G00 Z300 D0
```

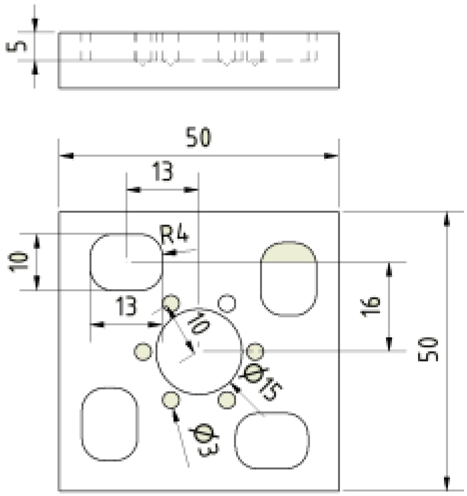
```
N185 SUPA G00 X300 Y300
N190 T5 D1
N195 MSG("Please change to Tool No 5")
N200 M05 M09 M00

N205 S7000 M3
N210 SLOT2( 50.00000, 0.00000, 2.00000, -5.00000, 2.00000, 3, 30.00000, 6.00000,
38.00000, 70.00000, 20.00000, 165.00000, 90.00000, 300.00000, 300.00000, 3.00000, 3,
0.20000, 2000, 5.00000, 250.00000, 8000.00000, )

N215 SUPA G00 Z300 D0
N220 SUPA G00 X300 Y300
N225 T2 D1
N230 MSG("Please change to Tool No 2")
N235 M05 M09 M00

N240 S5000 M3
N245 CYCLE72( "CONT1:CONT1_E", 50.00000, 0.00000, 2.00000, -5.00000, 5.00000,
0.00000, 0.00000, 300.00000, 100.00000, 111, 41, 12, 3.00000, 300.00000, 12, 3.00000)

N250 SUPA G00 Z300 D0
N255 SUPA G00 X300 Y300
N260 T2 D1
N265 MSG("Please change to Tool No 2")
N270 M05 M09 M00

N275 S6500 M3
N280 POCKET3( 50.00000, 0.00000, 1.00000, -3.00000, 40.00000, 30.00000, 6.00000,
36.00000, 24.10000, 15.00000, 3.00000, 0.10000, 0.10000, 300.00000, 300.00000, 0, 11,
12.00000, 8.00000, 3.00000, 15.00000, 0.00000, 2.00000)
N285 POCKET3( 50.00000, 0.00000, 1.00000, -3.00000, 40.00000, 30.00000, 6.00000,
36.00000, 24.10000, 15.00000, 3.00000, 0.10000, 0.10000, 300.00000, 300.00000, 0, 12,
12.00000, 8.00000, 3.00000, 15.00000, 0.00000, 2.00000)

N290 SUPA G00 Z300 D0
N295 SUPA G00 X300 Y300
N300 T6 D1
```

```
N305 MSG("Please change to Tool No 6")
N310 M05 M09 M00

N315 S6000 M3
N320 G00 Z50 X36 Y24.1
N325 MCALL CYCLE82( 50.00000, -3.00000, 2.00000, -5.00000, 0.00000, 0.20000)
N330 HOLES2( 36.00000, 24.10000, 10.00000, 90.00000, 60.00000, 6)
N335 X36 Y24.1
N340 MCALL ; Modal Call OFF

N345 SUPA G00 Z300 D0
N350 SUPA G00 X300 Y300
N355 T7 D1
N360 MSG("Please change to Tool No 7")
N365 M05 M09 M00

N370 S6000 M3
N375 MCALL CYCLE83( 50.00000, -3.00000, 1.00000, ,9.24000, ,5.00000, 90.00000,
0.70000, 0.50000, 1.00000, 0, 0, 5.00000, 1.40000, 0.60000, 1.60000)

N380 HOLES2( 36.00000, 24.10000, 10.00000, 90.00000, 60.00000, 6)
N385 X36 Y24.1
N390 MCALL ; Modal call Off

N395 SUPA G00 Z300 D0
N400 SUPA G00 X300 Y300
N405 T8 D1
N410 MSG("Please change to Tool No 8")
N415 M05 M09 M00

N420 S500 M3
N425 MCALL CYCLE84( 50.00000, -3.00000, 2.00000, ,6.00000, 0.70000, 5, ,2.00000,
5.00000, 5.00000, 5.00000, 0, 1, 0, 0, 5.00000, 1.40000)
N430 HOLES2( 36.00000, 24.10000, 10.00000, 90.00000, 60.00000, 6)
N435 X36 Y24.1
N440 MCALL ; Modal call Off
```

```
N445 SUPA G00 Z500 D0
N450 SUPA G00 X500 Y500;---------------Move to the Change position Ready to start next program or repeat ------------------------
N455 M30

;*************CONTOUR************
CONT1:
;#7__DlgK contour definition begin - Don't change!;*GP*;*RO*;*HD*
G17 G90 DIAMOF;*GP*
G0 X7 Y0 ;*GP*
G1 Y61.35 ;*GP*
G2 X13.499 Y86 I=AC(57) J=AC(61.35) ;*GP*
G1 X63 RND=2 ;*GP*
Y0 ;*GP*
;CON,0,0.0000,4,4,MST:0,0,AX:X,Y,I,J,TRANS:1;*GP*;*RO*;*HD*
;S,EX:7,EY:0;*GP*;*RO*;*HD*
;F,LFASE:0;*GP*;*RO*;*HD*
;LU,EY:61.35;*GP*;*RO*;*HD*
;ACW,DIA:210/0,EY:86,AT:0,RAD:50;*GP*;*RO*;*HD*
;LR,EX:63;*GP*;*RO*;*HD*
;R,RROUND:2;*GP*;*RO*;*HD*
;LD,EY:0;*GP*;*RO*;*HD*
;#End contour definition end - Don't change!;*GP*;*RO*;*HD*

CONT1_E:;************* CONTOUR ENDS ************
```

Milling part program example 1

The following is a milling demo part program.



N10 G17 G90 G54 G60 ROT

N20 T1 D1; FACEMILL

N30 M6

N40 S4000 M3 M8

N50 G0 X-40 Y0

N60 G0 Z2

N70 CYCLE71(50.00000, 1.00000, 2.00000, 0.00000, -25.00000, -25.00000, 50.00000, 50.00000, 0.00000, 1.00000, , ,0.00000, 400.00000, 11,)

N80 S4500

N90 CYCLE71(50.00000, 1.00000, 2.00000, 0.00000, -25.00000, -25.00000, 50.00000, 50.00000, 0.00000, 1.00000, , ,0.00000, 400.00000, 32,)

N100 G0 Z100

N110 T2 D1 ; ENDMILL D8

N120 M6

N130 S4000 M3

N140 M8 G0 X-13 Y16

N150 G0 Z2

_ANF:

N160 POCKET3(50.00000, 0.00000, 2.00000, -5.00000, 13.00000, 10.00000, 4.00000, -13.00000, 16.00000, 0.00000, 5.00000, 0.10000, 0.10000, 300.00000, 200.00000, 2, 11, 2.50000, , , ,2.00000, 2.00000)

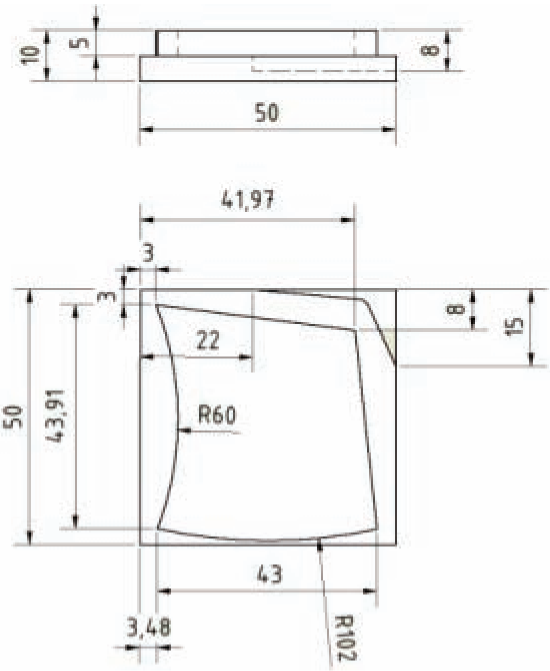
AROT Z90

_END:

```
REPEAT _ANF _END P=3
ROT
S4500 M3
_ANF1:
N160 POCKET3( 50.00000, 0.00000, 2.00000, -5.00000, 13.00000, 10.00000, 4.00000, -13.00000, 16.00000, 0.00000, 2.50000, 0.10000, 0.10000, 300.00000, 200.00000, 2, 2, 2.50000, , , ,2.00000, 2.00000)

AROT Z90
_END1:
REPEAT _ANF1 _END1 P=3
ROT

G0 X0 Y0
POCKET4( 50.00000, 0.00000, 2.00000, -5.00000, 7.50000, 0.00000, 0.00000, 2.50000, 0.10000, 0.10000, 300.00000, 200.00000, 0, 21, 2.00000, , ,4.00000, 1.00000)
S4500 M3
POCKET4( 50.00000, 0.00000, 2.00000, -5.00000, 7.50000, 0.00000, 0.00000, 5.00000, 0.10000, 0.10000, 300.00000, 200.00000, 0, 12, 2.00000, , ,4.00000, 1.00000)
G0 Z100
T3 D1 ;DRILL D3
M6
S5000 M3
G0 X0 Y0
MCALL CYCLE81( 50.00000, 0.00000, 2.00000, -5.00000, 0.00000)
HOLES2( 0.00000, 0.00000, 10.00000, 45.00000, 60.00000, 6)
MCALL

M30
```

Milling part program example 2

The following is another milling demo part program:



G17 G90 G60 G54

T1 D1 ;FACEMILL D50

M6

S3500 M3

G0 X0 Y0

G0 Z2

CYCLE71(50.00000, 1.00000, 2.00000, 0.00000, 0.00000, 0.00000, 50.00000, -50.00000, ,1.00000, 40.00000, ,0.10000, 300.00000, 11,)

S4000 M3

CYCLE71(50.00000, 0.10000, 2.00000, 0.00000, 0.00000, 0.00000, 50.00000, -50.00000, ,1.00000, 40.00000, ,0.00000, 250.00000, 32,)

T2 D2 ;ENDMILL

M6

S3500 M6

CYCLE72("CON1:CON1_E", 50.00000, 0.00000, 2.00000, -5.00000, 2.00000, 0.10000, 0.10000, 300.00000, 300.00000, 11, 42, 1, 4.00000, 300.00000, 1, 4.00000)

T4 D1 ;ENDMILL D10

M6

S4000 M3

```
G0 X55 Y-15
G0 Z2
G1 F300 Z-8
G42 G1 Y-15 X50
G1 X44 Y-2 RND=2
G1 Y0 X 22
G40 Y30
M30

;*************CONTOUR************
CON1:

;#7__DlgK contour definition begin - Don't change!;*GP*;*RO*;*HD*
G17 G90 DIAMOF;*GP*
G0 X3 Y3 ;*GP*
G2 X3.27 Y-40.91 I=AC(-52.703) J=AC(-19.298) ;*GP*
G3 X46.27 Y-47 I=AC(38.745) J=AC(54.722) ;*GP*
G1 X42 Y-8 ;*GP*
X3 Y3 ;*GP*
;CON,0,0.0000,4,4,MST:0,0,AX:X,Y,I,J;*GP*;*RO*;*HD*
;S,EX:3,EY:3;*GP*;*RO*;*HD*
;ACW,DIA:0/35,EX:3.27,DEY:-43.91,RAD:60;*GP*;*RO*;*HD*
;ACCW,DIA:0/35,DEX:43,EY:-47,RAD:102;*GP*;*RO*;*HD*
;LA,EX:42,EY:-8;*GP*;*RO*;*HD*
;LA,EX:3,EY:3;*GP*;*RO*;*HD*
;#End contour definition end - Don't change!;*GP*;*RO*;*HD*

CON1_E:;************* CONTOUR ENDS ************
```

Index