schleicher
control systems

Operating Manual

CNC Programming

XCx and ProNumeric

CNC Programming Version 03/15
Article No. R4.322.2080.0 (322 38162)

Target Group

These programming instructions have been written for trained personnel with specialised knowledge. There are special requirements for the selection and training of the personnel who work on the automation system. Suitable personnel are, for example, skilled workers with an electrical training background and electrical engineers who have been trained to work with automation systems.

Applicability of these Programming Instructions

Version Hardware XX / Software XX

Previous versions of these programming instructions

11/00 08/02 07/05 02/06 07/07 02/09 04/14

Where to get operating manuals

You can download all our operating manuals free of charge from our website at *http:\\www.schleicher.berlin* or order them by writing to the following address (please quote Order No.):

Additional documentation

For commissioning the CNC	
CNC commissioning for XCx and ProNumeric	R4.322.2340.0
Schleicher Dialog	Commissioning software

Copyright by

Schleicher Electronic Berlin GmbH
Wilhelm-Kabus-Straße 21-35
10829 Berlin

Germany

Tel.: +49 30 33005 - 0
Fax: +49 30 33005 - 344
E-Mail: *info@schleicher.berlin*
Internet: *http://www.schleicher.berlin*

Contents

Document conventions

This programming manual uses the following symbols to indicate safety-related and handling warnings:

Warning!

Indicates possible injury to persons or damage to the automation system or the equipment if relevant warnings are not observed.

Specifies information on preventing a hazard.

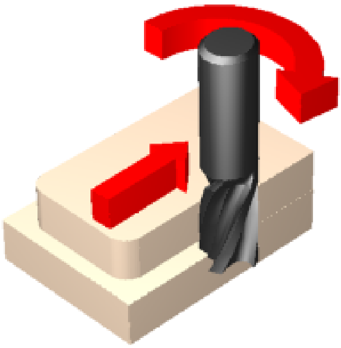
Important! or **Note!**

Highlights important information on handling the automation system or the respective part in the operating manual.

Other objects are represented as follows.

Object	Example
File names	MANUAL.DOC
Menus / Menu items	[Insert / Graphic / From file]
Paths / Directories	C:\Windows\System
Hyperlinks	*http://www.schleicher.berlin*
Program listings	`MaxTsdr_9.6   = 60` `MaxTsdr_93.75 = 60`
Keys	<Esc> <Enter> (press one after the other) <Ctrl+Alt+Del> (press all keys at the same time)
Configuration data identifiers	Q34 and Q.054
Names of shared RAM variables	cncMem.sysSect[n].flgN2P.bM345Act

1 CNC-Programming the XCx and ProNumeric



The NC-Program for the XCx and ProNumeric has been created in compliance with DIN 66025.

An NC-Program comprises records, which are made up of words. This can be called NC language.

A word in NC language consists of an address character and a sequence of digits.

Additional preparatory functions, which are not defined in DIN 66025, are indicated by address identifier $, followed by a single-digit or a two-digit number.

Special CNC functions require settings in the system parameters or in the PLC program. A note is then provided for the function in the description of the function.

The NC-Program processes one record after another. A requirement for the program to process on the XCx or the ProNumeric is that there must be a PLC user program running, because the PLC program and the NC-Program work together.

The NC-Programs are created in the Schleicher dialogue. The NC-Programs can also be created using any text editor that can save the corresponding files in ASCII format. After the NC-Programs have been created, they must be imported into the CNC controller. Importing NC-Programs can be done either with the Schleicher dialogue or with a PLC program.

1.1 Record structure

Each record consists of several words (functions) and the record delimiter (inserted automatically when you press enter). The type and number of words in a record is not fixed. The words in a record should be arranged in the following order:

N	**Record number**
G, $	**G-Word for preparatory function.** **The $ function is placed in the record according to the function**
A, B, C, D, L, O, P, U, V, W, X, Y, Z	**Designation for the axis names (in upper case)**
@A, @B, … @Z	**Designation for the axis names (with @ prefix and capital letters)** **The @ prefix means that the axis letters are interpreted as lower case.** **Thus, for example, i, j and k can be used for the axis names without colliding with interpolation parameters I, J, K.**
a, b, c, … z	**Designation for the axis names (in lower case)** **From V.09.05/3 onwards, lower case axis letters can be programmed without the @ prefix (e.g. x123.456).** **For reasons of compatibility with old NC-Programs, using lower case characters for the axes in programming is only possible when Q25 bit 3 (lower case characters allowed) is set to 1.** **When bit 3 (of Q25) is set, note the following:** **1. @X123.456 is the same as x123.456.** **2. Programming with lower case axis characters using @-prefix (e.g. @x123.456) is not allowed and leads to the error message "Illegal character (System, n nnnn)" (Error No.: 0x02110003).** **3. No automatic lower-case to upper-case conversion takes place in the Schleicher dialogue of the NC-Program editor.** **Important:** **Programming with lower-case axis characters is not possible with Schleicher COP x CNC / HBG operator panels. Here, as before, the axes must be programmed with the @ prefix.**
I.. J.. K..	**Interpolation parameters or parameters for thread pitch. These words each relate to a particular group of words for the coordinates and must be placed directly after the group.**
F	**Feed rate** **The F-Word alone serves as the feed rate for all programmed axes. For axes that do not move at feed rate (e.g. $ function) the preparatory function, then the axis coordinates and then the F-Word plus axis letter are written after the path assignments.**
FF	**Feed rate reduction**
ACC	**Acceleration**
S	**Spindle speed**
T	**Tool including compensation**
M	**Additional or switching function**
RA, RB, RD, RF	**Smoothing**
E, SE, RS, WA, WN	**Interface CNC - PLC**
R	**Arithmetic parameter**

BN, BNR, B%	**Program branch, subroutine call**
(..........)	**Comment**
\	**Record extension (see Programming subsequent records)**
/	**Fade symbol**

Important!

The words for record number, coordinate, interpolation parameter and thread pitch parameter must not be repeated in a record.

A record must not contain more than 120 characters, including spaces and record delimiter. A record can be extended using Programming subsequent records.

(see page 12)

Record number

N	Record number
Format	Nnnnnnnn (nnnnnnn = 7-digit decimal number in range 1 through 9999999)
Explanation	**The number is for locating program sections.**
Notes	**The number does not determine the order in which records are processed. You can program records with the same number as long as they are not the destination of a jump instruction.**

Example	
	N10
	N9999999

Comment

(......)	Comment
Format	(Text)
Notes	**Use only displayable 7-bit ASCII characters, without the ( ) characters.**

Example	
	N10 (this is a comment)

Programming subsequent records

If the maximum record length of 120 characters is insufficient to program all the required NC words in one record you can use a backslash (\) at the end of the record to declare the next record as a subsequent record. The record decoder then treats both records as one.
Subsequent records do not appear in the record display (monitor etc.).

Example		
	N100 G1 G90 G61 X200.002 + R9012 * 12.345 - R9100 Y145.901 -R9102 / 1.205 (KOMMENTAR) ZR9600 * 123.456 M77 SE11 WA22\	Record and subsequent record
	N1001 F3000 R34:= R20+12	This record belongs to N100
	N110 G0	This is a new record

Important!
If the record number of a subsequent record is programmed as a jump address the record decoder will identify it as a normal record ID.
Subsequent records should always have an unambiguous record number.

Example record structure	N10 G1 X100 Y5.4 F1000 ACC150 S500 M03 (comment)
N10	Record number
G1	G-Word Should be at the start of the record for reasons of clarity.
X100	Axis designation and target coordinate in mm for all axes which move in this record. Up to 4 places after the point can be programmed, depending on the interpolation fineness. The default resolution setting is 1µm or 0.001.
F1000	Feed rate in mm/min.
ACC150	Acceleration in %
S500	Speed of main spindle in r.p.m.
M03	M-Functions Switch functions whose execution is programmed in the PLC.
(......)	Comment in brackets.
\	Subsequent record character (see Programming subsequent records)

1.2 Program Structure

%1000 (Name)	**Program start**	**Program number and program name**
N10	**Sequence of records**	**The number of program records is limited only by the available memory capacity.**
N20		
N30		
.		
.		
Nnnnn M30	**Program ends with M17 and M30**	

1.2.1 Program number and program name

%	**Program number and program name**
Format	%nnnnnnnn (Name) nnnnnnnn = 8-digit decimal number in the range 1 through 99999999 (Name) = Program name max. 100 characters, the first 20 characters are displayed in the HBG.
Explanation	
Notes	**There is basically no difference between the program numbers of main programs and subroutines. But you should organize the program numbers so that the program structure is clear.**

Example		
	%1	**Program No. 1**
	%1000 (machine startup)	**Program No. 1000 Name: Machine startup**
	%99999999	**Program number 99999999 has preassigned special functions. (See Initialization program, G80 through G89 and cycle programming.)**

Note

The operating system can cope with 16 to 4096 programs.
The default setting is 256.

You can alter the setting via the user interface.

1.2.2 Program ends with M17 and M30

	Program end
Explanation	**M17 terminates a subroutine and returns to the calling NC-Program. If there is no calling program M17 has the same effect as M30.** **M30 terminates the NC-Program. Controller switches to RESET operating state.** **When Q25 bit 5 = 1, then M17 and M30 (page 107) are not required.**

1.2.3 Initialization program

%nnnnnnnn	Initialization program
Format	%nnnnnnnn nnnnnnnn = 8-digit decimal number, default setting is 00000000
Explanation	**Initialization program for setting parameters at CNC-START.**
Notes	**The initialization program runs through before the START of the main program. The program number can be freely chosen, the default setting is: 00000000.** **The program number must be entered in the configuration data of the subsystem in Q130.** **The initialization program must be closed with M17.** **If no program number is entered (Q130 = 00000000) the active CNC program is started directly at CNC start.**

Example	**%99999999**	**(Initialization)**
	N10 G11 X	**(Home to X)**
	N20 G11 Y	**(Home to Y)**
	.	**(synchronize other axes)**
	.	
	N100 F1000	**(Velocity presetting)**
	N110 M17	**(End subroutine)**

1.2.4 BN and BNR unconditional program branches

BN	**Unconditional program jump**	
Format	BNnnnn+/- nnnn = record number, +/- = search direction)	
Note	**The + symbol can be omitted.**	

Example		
	BN10-	**(Jump to record No. 10, search up to program start)**
	BN120	**(Jump to record No. 120, search down to program end)**

BNR	**Unconditional program jump parameterized**	
Format	BNRnnnn+/- nnnn = record number, +/- = search direction)	
Notes	**The + symbol can be omitted.**	

Example		
	BNR10-	**(Jump to record No. in arithmetic parameter 10, search up)**
	BNR20	**(Jump to record No. in arithmetic parameter 20, search down)**

1.2.5 B% Unconditional subroutine call

B%	**Unconditional subroutine call**
Format	B%nnnnnnnn nnnnnnnn = program number, 8-digit decimal number
Explanation	**For the subroutine call, an NC-Record is programmed without further NC-Words. Program execution continues in the called program.**
Notes	**After the subroutine call, only the number of passes can be programmed; no other commands are allowed.**

B%nnn R	**Unconditional subroutine call with loop count**
Format	B%nnnnnnnn R nnnnnnnn = program number
Explanation	**The called NC-Program is repeated by the number of times indicated in the arithmetic parameter.**
Notes	**The value in the arithmetic parameter is decremented on each repetition.** **The value must be a positive whole number when the call is made. For values ≤ 1 the subroutine will be executed once.**

B%R	**Unconditional subroutine call parameterized**
Format	B%R R = arithmetic parameter
Explanation	**The program number of the calling program is in the arithmetic parameter.**
Notes	**You can calculate and call an 8-digit program number by specifying a max. 8-digit number as offset value with a max. 7-digit R-Parameter value. The following arithmetic functions are allowed:** **B%[Offset + R[R]xyz],** **B%[Offset - R[R]xyz],** **B%[R[R]xyz + Offset],** **B%[R[R]xyz - Offset].**

Note

The nesting depth of subroutines is 4.

Subroutines may not call themselves (recursive) or a previously called NC-Program.

Important!

The program number of a subroutine must be a positive whole number.

The subroutine must end with M17 (program end).

After the end of the subroutine, program execution is continued with the next record of the calling program.

1.2.6 Conditional program executions, comparisons

If comparisons are programmed in a record the following parts of the record will only be executed if the result of comparison is "true". If the result is "not true" only the part of the record before the comparison will be executed.
With comparisons you can create conditional program jumps and subroutine calls.

Comparisons with arithmetic parameters

	Comparing arithmetic parameters
R . . < Value	**R . . . . less than value**
R . . <= Value	**R . . . . less than or equal to value**
R . . = Value	**R . . . . equal to value**
R . . <> Value	**R . . . . not equal to value**
R . . >= Value	**R . . . . greater than or equal to value**
R . . > Value	**R . . . . greater than value**
Explanation	
Notes	**Arithmetic parameters (R-Parameters) are word flags, which are used in the NC-Program to save any values.** For more on arithmetic parameters see page 112.

Example		
	N10 R1 < 10 BN100	**(If R1 < 10 jump to record 100)**

Important!

Parameter comparison is executed at the time of record decoding. Parameter changes between record decoding and record execution will be ignored.

G and $-Words and T calls will be executed regardless of the comparison.

No subroutine call B% should be programmed prior to making a comparison.

Comparisons with bit variables

E...=	**Requesting bit variables directly at the start of record execution**
Format	Ennn=1 Ennn=0 nnn 3-digit decimal number in the range 0 - 255 for global bit variables and 256-511 for system-specific bit variables.
Explanation	**E 0 = cnc.Mem.comSect.abFlg[0]** **E 127 = ...** **E 128 = ...** **E 255 = cnc.Mem.comSect.abFlg[255]** **These bit variables are also used for CNC words SE, RS, WA and WN (see page 110).**
Notes	**Comparison is executed at the time of record change from the preceding record. The following records are not decoded until the comparison has been executed.**

Example	
	N10 X100 E0 = 1 B%9000 (Program %9000 is called if E0 = 1; otherwise the CNC program continues in the next line.)

Note

Comparison is executed at the time of record change from the preceding record. The following records are not decoded until the comparison has been executed.

No subroutine call B% should be programmed prior to making a comparison.

1.2.7 Conditional skipping of parts of records

/	**Conditional skipping of following part of record**
Format	/
Explanation	**You can exclude part of a record from execution using bit variables *cncMem.comSect.flgP2N.bBlkFade* for all NC-Records or *cncMem.sysSect[n].flgP2N.bBlkFade* for subsystem n.**
Notes	**This function requires a PLC program.**

Example	
	N10 SE01 / G11 X (The part of the record following / will not be executed if the bit variable = 1.)

Important!

The bit variable is requested at the time of record decoding. Changes to the bit variables between record decoding and record execution will be ignored.

1.2.8 Loading NC-Records with R-Parameters

From SW version OS 06.26/0

Function for reducing the record-change time for NC-Records with extensive R-Parameter calculations.

"Loading" means that the marked NC-Records are calculated with the arithmetic parameters in the decoder task, during which no record-change is made in the IPO cycle. In this way the calculations can generally be processed in less time.

	Loading NC-Records with R-Parameters
Explanation	The function is enabled in the NC-Program with word G29 (update arithmetic parameters when record is being prepared). Function G29 must be programmed with R-Parameter calculations before the 1st NC-Record. Loading is deactivated with G28 (update arithmetic parameters when record is executed). NC-Records that are to be loaded must contain R-Parameter calculations only. Movements, G-Functions, jumps etc. must not be programmed. If they are programmed, this causes the loading process to be stopped.
Notes	This function requires that bit 4 = 1 must be set in Q111 (filter out NC-Records). The records with the arithmetic parameters should be programmed at the beginning of the NC-Program. The calculations are performed in the decoder task, these NC-Records are not displayed on the monitor.

Example	Load R-Parameters
	N10 G29
	N20 R6001:=1 R6002:=2 R6003:=3 R6004:=4 R6005:=5
	N21 R6006:=6 R6008:=8 R6009:=9
	N22 R6000:= R2*R3+R4*R5+R6
	N23 R6007:=-R8+R9*R1
	N24 R6010:= 2.5*R2+R3 R6010:=R1:R5
	N25 R1001:=1 R1002:=2 R1003:=3 R1004:=4 R1005:=5
	N26 R1006:=6 R1007:=7 R1008:=8 R1009:=9 R1010:=10
	N27 R1011:=11 R1012:=12 R1013:=13 R1014:=14 R1015:=15
	N28 R1016:=16 R1017:=17 R1018:=18 R1019:=19 R1020:=20
	N29 R1021:=21 R1022:=22 R1023:=23 R1024:=24 R1025:=25
	N30 R1026:=26 R1027:=27 R1028:=28 R1029:=29 R1030:=30
	N40 G28 G0 X50 Y50 Z50 R13 := 13 R14 := 14
	N50 R15:= 15
	N60 R16:= 16

Records N20 through N30 are loaded, from N40 onwards all R-Parameters are updated with record change.

1.2.9 Indirect programming with arithmetic parameters

The constants in a record can be replaced with arithmetic parameters. The arithmetic parameters are evaluated when the record is prepared.

Example	
	N10 GR0 XR1001 YR1002 FR1003 SER1
	N20 B% R2500
	N30 BN R10-

Axes X and Y move to the positions indicated in R1001 and R1002. The feed rate is taken from parameter R1003. The number of the G-Function is given by the content of R0 and the bit variable with the number from R1 is set. Then a jump is made to the program with the number from R2500.

Note

Only positive whole R parameter values are valid for R-Parameters that replace whole number constants (e.g. SExx, BN%xx). Integer-R-Parameters are used for this purpose (R0-R999, R2000-R5999).

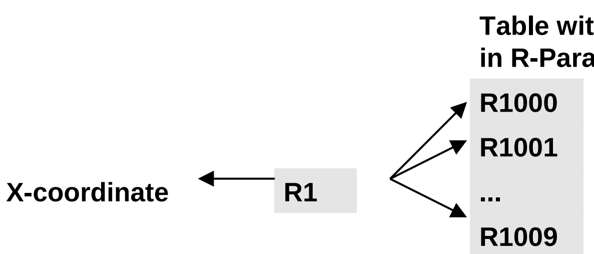
The controller operating system does not round the decimal places of real-R-Parameters!
If R0 =1,001 in the above example program execution will be aborted with error message "Ungültige G-Funktion" ["Invalid G-Function"].

$ functions cannot be parameterized.

1.2.10 Indexed programming

While they may replace a constant in indirect programming, arithmetic parameters can also be used as a pointer to another arithmetic parameter.

Example		
	N10 XRR1	**(R1 = pointer to coordinate)**
	N20 R1 := R1 + 1 R1 >= 20 R1 := 10	



Each time the described subroutine is called it moves the X axis to the next position in the table. After 10 calls it starts with the 1st position again. For the sake of clarity start initialization and constraints have been omitted.

1.3 Calculations in the record

1.3.1 Calculations

	Calculations
R0 := 100	**Assigns a constant to an arithmetic parameter**
R0 := R1	**Assigns an arithmetic parameter to another arithmetic parameter.**
R0 := -R1	**Negated assignment**
R0 := R1 + R2	**Addition**
R0 := R1 - R2	**Subtraction**
R0 := R1 * R2	**Multiplication**
R0 := R1 : R2	**Division**
R0 := ABS R1	**Absolute value of R1**
R0 := SQR R1	**Square root of the absolute value of R1**
R0 := SIN R1	**Sine of R1 in degrees**
R0 := COS R1	**Cosine of R1 in degrees**
R0 := TAN R1	**Tangent of R1 in degrees**
R0 := ATA R1	**Arc tangent of R1 in degrees**
R0 := R1 MOD R2	**Division of R1 by R2. The whole number remainder of division is entered in R0.**
Notes	**The maximum number of assignments to arithmetic parameters that can be made in one record is 8. Arithmetic parameters (R-Parameters) are variables that are used in the NC-Program for storing arbitrary values.** For more on arithmetic parameters see page 112. **If several assignments are programmed in a record they will always be executed from left to right.** **If several calculations are programmed in an assignment the calculations will always be carried out from right to left (reverse chain calculation).** **Brackets could not be set (brackets indicate comments). Example:** **R1:= R2*R3+R4*R5+R6 corresponds to R1:= R2*(R3+(R4*(R5+(R6))))** **R7:= -R8+R9*R1 corresponds to R7:= -(R8+(R9*(R1)))**

Note

In trigonometric functions the angle is specified in degrees (0 through 360). The typical error near the quadrant transitions is 1 * 10^{-5}, otherwise 1 * 10^{-6}.

1.3.2 Coordinate calculation

Axis coordinates can be calculated in the record, e.g. scale factors and offset.

Example	
	N10 X100 * R1001
	N20 Y200 + R1002

Note

Parameter calculations with negative axis coordinates plus negative parameter value are calculated as follows:

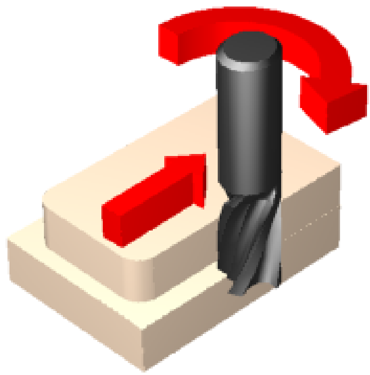
N10 X-35+ R1003 (Content R1003 = -3)
X = -(35+(-3))
X = -32

1.3.3 Constants

In all calculations arithmetic parameters can be replaced with a constant.

Example		
	N20 R1001:= 2,5 * R1002 + R1003 R1001:= R1001 : R1005	$R1001 = \frac{(R1002 + R1003) * 2.5}{R1005}$

2 Feed rate, Acceleration and Spindle Speed



2.1 Feed rate (path feed rate) in general

The feed rate depends on the mode, the selected interpolation type and the machine data pre-settings. The default setting is mm/min.

Feed rate in manual mode

In manual mode the axes are moved at the set conventional speed (set in Q.000).
With an additional overlaid rapid-feed velocity, the axes are moved at the set rapid feed velocity (set in Q.028).

Path feed rate with G0

Programmed rapid traverse. The path feed rate is calculated in such a way that the slowest axis moves at its rapid-feed velocity (set in Q.029).

Path feed rate with G1

All axes programmed within one record will be interpolated at G1 in such a way that the resulting path feed rate corresponds to the programmed feed rate F. The unit for F depends on G94 (mm/min) and G95 (mm/spindle revolution).

Example	
	N10 G1 X100 Y50 Z20 F5000

Path feed rate with G2/G3

With circular interpolation the programmed feed rate F relates to the circular path. If other axes are programmed in this record, these will be interpolated as straight lines and their velocity will be calculated in such a way that they reach their target coordinates at the same time as the circular movement.

Feed rate with G10

Point-to-point positioning in rapid traverse. Each axis moves at its rapid-feed velocity (Q.029) to the programmed coordinate.

2.2 Programming (path feed rate) F

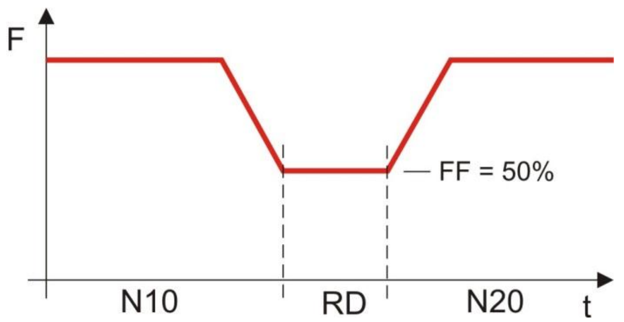
F	Feed rate (path feed rate)
Format	Fnnnnn FXnnnnn nnnnn = 5-digit decimal number X = arbitrary axis letter
Explanation	**The F-Word is used to program the feed rate (path feed rate). The valence of the word is dependent on the G-Function.**
Notes	**G93 feed in % rapid traverse** **The feed rates programmed with the F-Word are calculated as a % of the rapid traverse.** **G94 feed rate / path feed rate in mm/min.** **The feed rates/path feed rates programmed with the F-Word are calculated in mm/min.** **G94 is the default setting.** **G95 feed rate in mm/rev. of the main spindle** **The path feed rate programmed with the F-Word is calculated in mm/spindle revolution. A spindle with an actual-value system is required for G95.**

Example	
	N10 G1 X100 Y50 Z20 F5000

2.3 Programming feed rate reduction FF

FF	Feed rate reduction
Format	FFnnnnn nnnnn = 5-digit decimal number
Explanation	**Feed rate reduction when changing record, as a percentage of the programmed feed rate.**
Notes	**The feed rate reduction is effective by records in conjunction with G62/G64.**

Example	
	N10 G1 X100 RD5 FF50
	N20 Y100



2.4 Programming acceleration ACC

ACC	**Ramp type and acceleration override**
Format	**ACCtnnn** t = type of ramp: 0 = Linear ramp; 1 = Sin^2 ramp; 2 = Speed reduction prior to record change (linear); 3 = Speed reduction prior to record change (Sin^2 ramp) nnn = Acceleration override 0 - 200%
	The following applies to independent axes: Always on linear ramp **ACCXnnn** X = Axis letter nnn = Acceleration override 0 - 200%
Explanation	Acceleration is programmed as acceleration override in % of the preset acceleration value.
Notes	The programmable acceleration is self-holding, until M30 or CNC-RESET. The ramp type and the ramp override can be changed at G64. With RD-programming (record transitions with any axes), if the programmed rounding path is reduced the set velocity for the transition records is also reduced. If ACC2000+100 (2000 = ramp type linear + acceleration 100%) deceleration to record change velocity will take place before record change. Record change velocity is always the lower velocity of the two records. ACC2100 FF50% ... The ACC function cannot be applied to special functions such as G33, G63, and oscillation. Independent axes are specified with the axis letter and are programmed as follows. E.g. X-axis: ACCX50 (corresponds to ACCX0050 = ACCX1050) In the Manual mode of operation, all axes are driven with linear ramps. In Automatic mode, the ramp type is set in Q37 bit 4. 0 = Linear ramp 1 = Sin2-ramp From OS 8:40/1 Deceleration factor (adjustable with PLC program) when pressing the STOP key. - The controller always decelerates using the greatest ramp factor - E.g. when PLC factor > ACC ramp, the controller decelerates using the PLC factor.

Example		
	N10 G1 X100 Y500 F2000 ACC50	(Acceleration with 50 % linear ramp)
	N20 G1 X100 Y650 F500 ACCR1	(Acceleration value in R1)
	N30 G1 X350 Y650 F1500 ACCRR0	(Parameter no. for acceleration value in R0)

Example		
	N110 G1 G64 X10 ACC0050 F100	(Acceleration with 50 % linear ramp)
	N120 X100 ACC1100 F2000	(Acceleration with 100% sine ramp)
	N120 X150 ACC 2050	(Deceleration with linear ramp to record change velocity F100)
	N130 X250 ACC 3050 F500	(Deceleration with sine ramp to record change velocity F100)
	N140 G60 X280 ACC1100 F100	(Deceleration with sine ramp to standstill)

2.5 Programming Spindle Speed S

S	Spindle speed
Format	Snnnnn SXnnnnn nnnnn = 5-digit decimal number X = arbitrary axle letter
Explanation	**The programmed value is evaluated as the spindle speed (default setting) in r.p.m. or cutting speed m/min If there are several spindles in a subsystem, one axis can be selected as the spindle by entering an axis letter.**
Notes	**G97 is used to evaluate the speed in r.p.m.** **With G96 the S-Word is the cutting speed (circumferential speed) in m/min. The radius associated with the circumference is formed from the actual value of an axis specified with $34.** **The speed of other spindles is programmed with S"axis name".** **The value programmed in S is entered in the shared RAM variable *cncMem.axSect[n].wrdN2P.lPrgSVal*.** **Variable *cncMem.axSect[n].flgN2P.bSFctMod* is set to TRUE as the modification signal. This variable must be acknowledged by the PLC user program.** **If no axis in the subsystem is specified as spindle or rotary axis (Q.054) the content of the S-Word will be saved in the variables *cncMem.sysSect[n].wrdN2P.lSFct* for processing by a PLC program. Variable *cncMem.axSect[n].flgN2P.bSFctMod* then serves as the modification signal.**

Example	
	N10 G1 X100 Y100 S3500

3 G-Functions

According to DIN 66025 Part 2, G-Functions are CNC functions that describe the interpolation context of the NC axes. In this overview the G-Words are organized in groups.

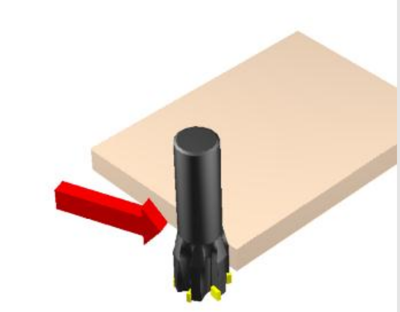
Only one function from each group can be active.

Normally the functions remain active until they are deselected by another function from the same group.

Group	Properties D = Default setting S = Active for 1 record		Meaning
1		**G0**	**Contour control in rapid feed.**
	D	**G1**	**Straight interpolation**
		G2	**Clockwise circle-helix interpolation**
		G3	**Anticlockwise circle-helix interpolation**
		G10	**Point-to-point positioning in rapid feed**
		G11	**Home to reference point**
		G12	**Clockwise spiral interpolation**
		G13	**Anticlockwise spiral interpolation**
		G25	**Online curve interpolation OCI without tangential transition**
		G26	**Online curve interpolation OCI with tangential transition**
		G27	**Freeform interpolation of CNC-Programs created offline**
		G32	**Tapping with controlled spindle**
		G33	**Thread cutting**
		G63	**Tapping without compensating chuck**
		G76	**Thread cycle**
		G77	**Tapping cycle without compensating chuck**
2	**S**	**G4**	**Dwell time**
3	**D**	**G5**	Deselection of tangential tracing
		G6	Tangential tracing with the transition radius (inner circle)
		G7	Tangential tracing with the transition radius (outer circle)
		G8	Tangential tracing without transition radius
4	**D**	**G17**	**Plane selection X-Y**
		G18	**Plane selection X-Z**
		G19	**Plane selection Y-Z**
5	**D**	**G20**	**Deselection of coordinate transformation**
		G21	**Position specified in Cartesian coordinates**
		G22	**Position specified in Cartesian coordinates**
		G23	**Position specified by the axis positions**
		G24	**Position specified by the axis positions**
6	**D**	**G28**	**Update arithmetic parameters when record is executed**
		G29	**Update arithmetic parameters when record is executed**

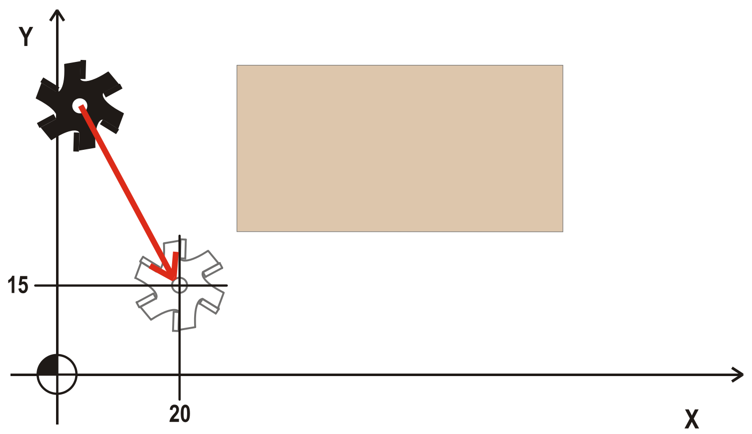
Group	Properties D = Default setting S = Active for 1 record		Meaning
7	S	G39	Interrupt record preparation
8	D	G40	Switch off tool-radius compensation
		G41 G42	Tool radius compensation left/right
		G43 G44	Tool radius compensation positive/negative
	S	G50	Tool radius compensation without transition contour
9		G45 G46	Feed rate correction
10		G52	Coordinate rotation
11	D	G53 to G59	Zero point offset
12	S	G9	Exact positioning
	D	G60	Record change after exact stop boundary reached
		G61	Record change after elimination of set-actual deviation
		G62	Record change with acceleration monitoring
		G64	Record change without loss of velocity
		G66	Synchronization of the IPO interpolation points
13	S	G67	Special function for oscillating
14		G70	Units in inches; the last used function applies
	D	G71	Units in millimetres
15		G72	Coordinate systems: Selection of reference system
		G74	Coordinate systems: Selection of compensation system
16	D	G80 to G89	Machining cycles
17	D	G90	Absolute measurements
		G91	Incremental measurements
18		G92	Reference point offset
19		G93	Specification of feed rate in % of rapid feed
	D	G94	Feed rate in mm/min (in/min)
		G95	Feed rate in mm/rev. (in/rev.)
20		G96	Constant cutting speed
	D	G97	Spindle speed given in r.p.m.
21	D	G98	Accept self-maintaining preparatory functions
		G99	Do not accept self-maintaining preparatory functions

3.1 G0 Contour control with rapid feed velocity

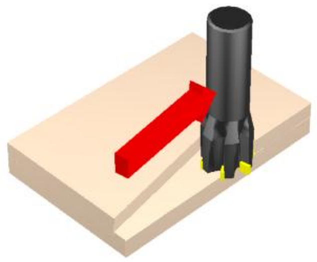
G0	**Path control with rapid-feed velocity and linear interpolation**	
Format	G0 X Y X, Y = arbitrary axis letter	
Explanation	**All axes reach the programmed end position simultaneously. The path feed rate is calculated in the controller so that the shortest positioning time is achieved without exceeding the axis-specific rapid-feed velocity (Q.029).**	
Notes	**The record change does not occur until exact position has been reached on all axes, regardless of the exact positioning level programmed with G60 to G64.** **If Q38 bit 2 = 1, the record change is made with the programmed record change function.** **The programmed feed rate F is not active but is retained and reactivated after G0.** **As long as G0 is active the shared RAM variable *cncMem.sysSect[n].flgN2P.bG0Act* is set to TRUE.**	

Note

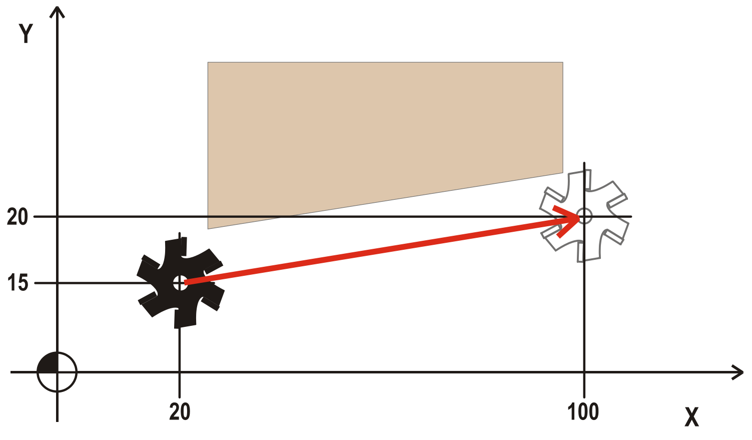
G0 is not suitable for workpiece machining.

Example	
	N10 G0 X20 Y15
	

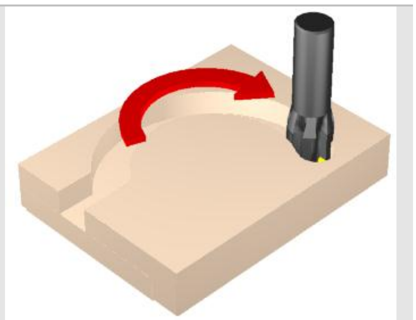
3.2 G1 Contour control with linear interpolation

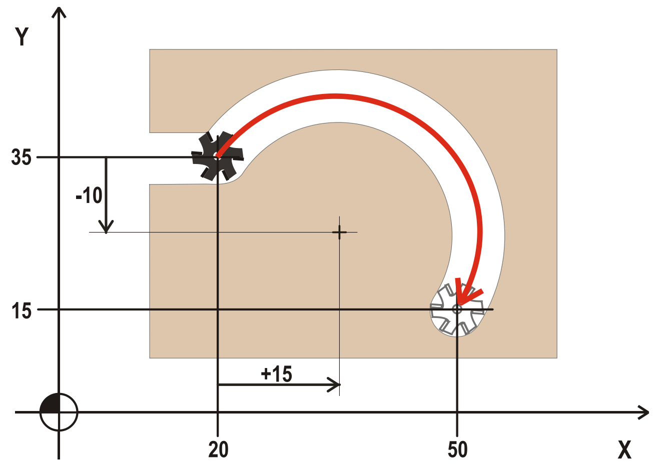
G1	**Contour control with linear interpolation**	
Format	G1 X Y F X, Y = arbitrary axis letter F = path feed rate	
Explanation	**All axes reach the programmed end position simultaneously on a straight line.** **The path feed rate is identical with the current programmed feed rate F.**	
Notes	**Linear interpolation is permissible n-dimensionally in all axes simultaneously.** **The maximum achievable path feed rate is restricted by the slowest axis so it cannot be slower than the velocity programmed in F.** **As long as G1 is active the shared RAM variable *cncMem.sysSect[n].flgN2P.bG1Act* is set to TRUE.**	

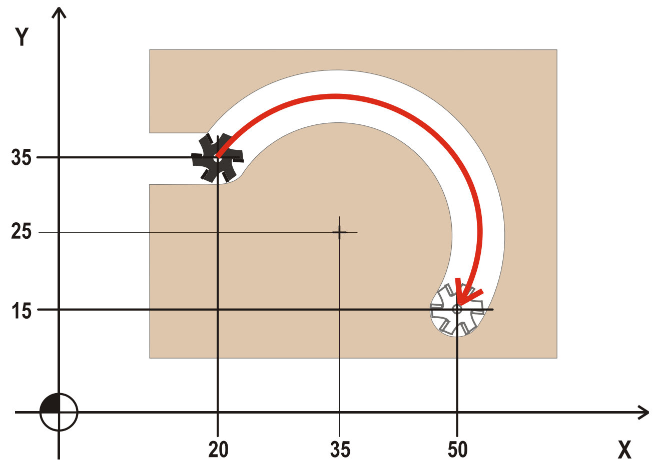
Example		
	N10 G1 X100 Y20 F1000	**(The end position is approached on a straight line at a path feed rate of 1000 mm/min.)**

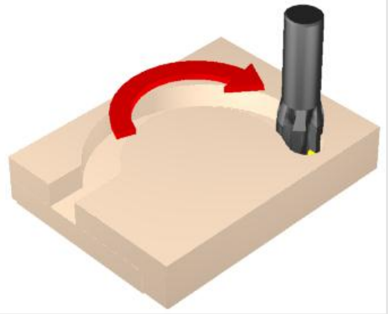


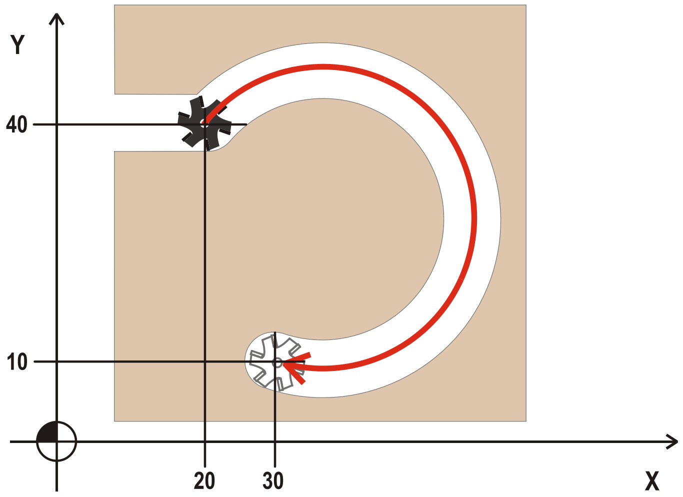
3.3 G2, G3 and RC circle and helix interpolation

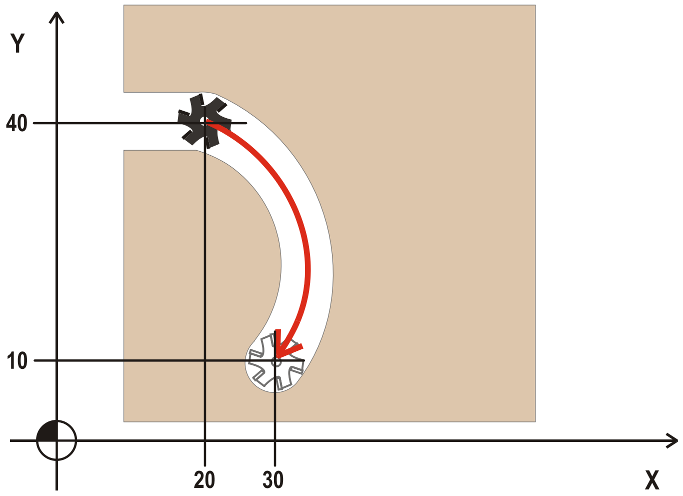
G2	**Clockwise circle and helix interpolation**	
G3	**Anticlockwise circle and helix interpolation**	
Format	G2 X Y I J F G3 X Z I K F G2 X Y RC F X, Y, Z = axis letters I, J, K = auxiliary coordinates F = path feed rate RC = radius	
Explanation	**Circular and helix interpolation with specified circle centre.**	
Notes	**The coordinate for the circle centre is programmed using auxiliary coordinates I, J, K or specified under RC. The auxiliary coordinates are assigned to the axes:** **Axes parallel to X = I** **Axes parallel to Y = J** **Axes parallel to Z = K** **The reference point of the auxiliary coordinates can be set: relative, based on the record start point (Q25 bit2=0) or absolute, based on the currently selected coordinate system (Q25 bit2=1).** **Circle interpolation can only be carried out in one plane. The circle plane must concur with the selected working plane G17/G18/G19. If this is not the case an error will be indicated and program execution will be interrupted.** **The corresponding dimensional coordinates must be assigned to the axes involved in circular interpolation. In Q.054 bits (0,1 or 2) are assigned to the axes of a spatial coordinate. If several axes are assigned to the same spatial coordinates, a choice must be made with $47.** **The end coordinates can be absolute or incremental, depending on the preparatory function (G90 or G91).** **The circle end point must achieve the precision set in Q.106, otherwise an error message will be output. Permissible deviations of the circle end position will be compensated by spiral interpolation in the circular path. If other axes are programmed as well as the circular axes, these axes will be included in the interpolation context so that they reach the end position simultaneously with the circular axes (Helix interpolation). The path feed rate programmed in F relates to the resulting spatial path (see $38).** **If the start and end positions of the circle are identical a full circle will be interpolated unless the circle centre was specified with RC.**	

Example	Auxiliary coordinates with incremental reference (Q25 bit 2=0)	
	N10 G1 X20 Y35 F1000	
	N20 G2 X50 Y15 I15 J-10	**(The absolute programmed end position is approached clockwise on a circular path at a constant path feed rate of 1000 mm/min.) The circle centre coordinates are relative, relating to the start position.)**
		

Example	Auxiliary coordinates with absolute reference (Q25 bit 2=1)	
	N10 G1 X20 Y35 F1000	
	N20 G2 X50 Y15 I35 J25	**(The absolute programmed end position is approached clockwise on a circular path at a constant path feed rate of 1000 mm/min.) The circle centre coordinates are absolute, relating to the programmed zero point.)**
		

RC	**Circular and helix with radial programming**	
Format	RCnnn RCRnnn nnn = decimal number Rnnn = arithmetic parameter	
Explanation	**Circular and helix interpolation with specified arc radius.**	
Notes	**Only the end coordinates and the radius have to be programmed:** **If RC < 0 (negative), an arc with angle at circumference > 180° will be made.** **If RC > 0 (positive), an arc with angle at circumference < 180° will be made.** **A full circle can only be programmed as two parts.**	

Example		
	N100 G1 X20 Y40 F500	
	N110 G2 X30 Y10 RC-20	**(RC negative, arc > 180°)**
		

Example	
	N100 G1 X20 Y40 F500
	N110 G2 X30 Y10 RC20 **(RC positive, arc < 180°)**
	

3.4 G4 and TI Dwell time

G4	**Dwell time**
Format	G4 F G4 R F = dwell time in seconds R = arithmetic parameter contains dwell time in seconds

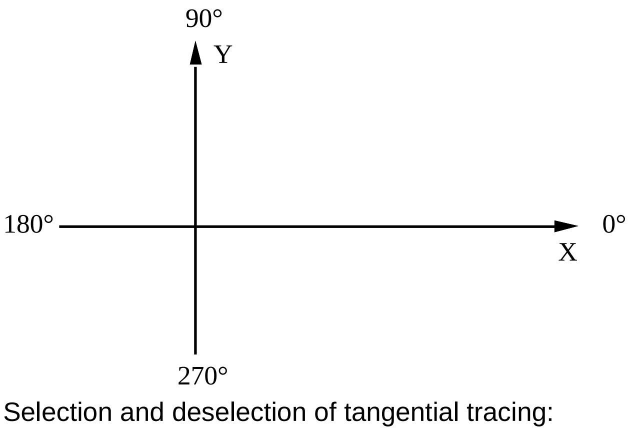
Example		
	N10 G4 F1.2	**(dwell time 1.2 seconds)**
	N10 G4 R1002	**(dwell time in R parameter R1002)**
	N10 G4 FR1002	**(dwell time in R parameter R1002)**

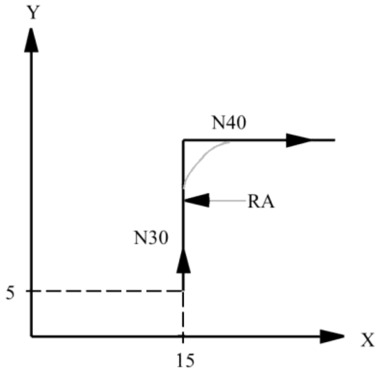
TI	**Dwell time**
Format	TI nnn TI R nnn = decimal number (integer double word 2,147,483,647), unit sec. R = arithmetic parameter, contains dwell time in seconds
Explanation	**TI can be programmed parallel to the motion**
Notes	**If a TI dwell time is programmed parallel to a movement the time will run in parallel to the movement. The record changes when both conditions have been met: target coordinate reached and time expired.** **The dwell time is effective record by record.** **Can also be programmed with G4.**

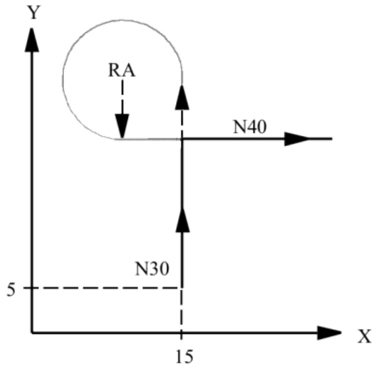
Important!
The smallest value for the TI-wait action is 0.001s.
The accuracy of this action depends on the IPO time.
A value less than the IPO time is inadmissible.
Internal sample calculation for 5.0 s: 5 * 1000 / IPO time

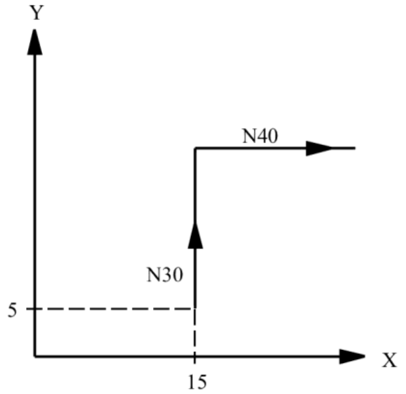
Example		
	N10 TI 2.5	**(dwell time 2.5 seconds)**
	N10 TI R1002	**(dwell time in R parameter R1002)**
	N10 G1 X0.5 F500 TI2.5	**(dwell time 2.5 seconds during G1)**

3.5 G5, G6, G7 and G8 Tangential tracing for circle and straight line

G5	**Deselection of tangential tracing**
G6	**Tangential tracing with the transition radius (inner circle)**
G7	**Tangential tracing with the transition radius (outer circle)**
G8	**Tangential tracing without transition radius**
Format	G5 ... G6 Yxx RAxx G7 Yxx RAxx G8 X X, Y, Z = axis letters RA = transition radius
Explanation	The tangential controller serves to maintain constant the angular position of a rotary axis relative to the path angle in one plane. The plane is determined by the working plane (G17 / G18 / G19 Page 46). In the following, plane XY (G17) has been selected, A is the rotary axis.
Notes	Programming the rotary axis and the inclination angle: The rotary axis, whose angle of inclination is to be kept constant, is identified with the special-path condition $35. Here, the following coordinate value will be interpreted as the inclination angle. This angle must always be specified in degrees (0-359.999). The rotary axis is not traversed in this NC-Record. When tracing is selected, the angle of inclination can be changed at any time. Example: N10 $35 A45 ($35 Page 91, A is to be traced, angle of inclination 45°) The center of rotation of the rotary axis must always lie on the contour. The reference point and the direction of rotation of the rotary axis must be adjusted to the path angle. The path angle is defined as follows:  Selection and deselection of tangential tracing: The selection is made with G6, G7 or G8. These functions are self-maintaining. For G6 and G7, it is also possible to program a self-maintaining transition radius under RA ... . The controller then adds transitional circles to resolve non-tangential transitions. When the transition radius is equal to zero, G6 and G7 work like G8: an intermediate record is generated in which the rotary axis is adjusted to the new path angle. Deselection is made with G5, whereby programming must be in the last traversing record prior to the deselection of G8.

Example	G6 Tangential tracing with the transition radius (inner circle)
	Program flow: Two contour elements are to be connected to a transitional circle ('inner circle') with a radius of 5 mm. The rotary axis A is to trace with an angle of 0° to the contour. A transition record (#N20) is to be introduced prior to the first G6 record, to adjust the rotary axis to the path angle of 90°. The end coordinate of N30 is corrected by the amount of the transition radius. A transition record (#N30) is inserted for the transition radius. The rotary axis is traced by 90°(helix interpolation).
	N10 G0 X15 Y5 (Start position)
	N20 $35 A0 (Selection A with angle 0°)
	N30 G1 G6 Y20 RA5 F500
	N40 G1 G8 X30 (Last record with G8)
	N50 G5 .... (Deselection with G5)
	
	Resulting NC-Program
	N10 G0 X15 Y5 (Start position)
	N20 $35 A0 (Selection A with angle 0°)
	#N20 G0 A90 (Setting the initial angle)
	N30 G1 G6 Y15 RA5 F500
	#N30 G2 X20 Y20 RC5 A0 (Radius + Helix)
	N40 G1 G8 X30 (Last record with G8)
	N50 G5 .... (Deselection with G5)

Example	Tangential tracing with the transition radius (outer circle)	
	Program flow: Two contour elements are to be connected to a transitional circle ('outer circle') with a radius of 5 mm. The rotary axis A is to trace with an angle of 0° to the contour. A transition record (#N20) is introduced prior to the first G7 record, to adjust the rotary axis to the path angle of 90°. The end coordinate of N30 is corrected by the amount of the transition radius. Another transition record (# N30) for the transition circle is inserted. Thereby the rotary axis is traced by +270° (Helix interpolation).	
	N10 G0 X15 Y5	(Start position)
	N20 $35 A0	(Selection A with angle 0°)
	N30 G1 G6 Y20 RA5 F500	
	N40 G1 G8 X30	(Last record with G8)
	N50 G5	(Deselection with G5)
		
	Resulting NC-Program	
	N10 G0 X15 Y5	(Start position)
	N20 $35 A0	(Selection A with angle 0°)
	#N20 G0 A90	**(Setting the initial angle)**
	N30 G1 G6 Y15 RA5 F500	
	#N30 G3 X15 Y20 RC-5 A0	**(Radius + Helix)**
	N40 G1 G8 X30	(Last record with G8)
	N50 G5	(Deselection with G5)

Example	G8 Tangential tracing without transition radius	
	Program flow: A transition record (#N20) is introduced prior to the first G8 record, to adjust the rotary axis to the path angle of 90°. Another transition record (# N30) is inserted. Thereby the rotary axis is traced by 90° in rapid traverse mode. This record is only inserted if the change in direction between N30 and N40 is more than 0.5°. If the transition is almost tangential, the new position of the rotary axis in record N40 is approached by interpolation.	
	N10 G0 X15 Y5	(Start position)
	N20 $35 A0	(Selection A with angle 0°)
	N30 G1 G8 Y20 F500	
	N40 G1 X30	(Last record with G8)
	N50 G5	(Deselection with G5)
		
	Resulting NC-Program	
	N10 G0 X15 Y5	(Start position)
	N20 $35 A0	(Selection A with angle 0°)
	#N20 G0 A90	**(Setting the initial angle)**
	N30 G1 G6 Y15 RA5 F500	
	#N30 G0 A0	**(Setting new angle)**
	N40 G1 G8 X30	(Last record with G8)
	N50 G5	(Deselection with G5)

3.6 G9, G60 Exact positioning

G9	**Exact positioning, effective record by record**
Format	G9
Explanation	**Record change occurs when the following error of all axes in the subsystem is less than the respective stop tolerance range. (set in Q.048)**
Notes	**Exact positioning with G9 is effective for just one record. In the next record the previously programmed record change condition applies.** **As long as the axis is not in exact position the shared RAM variable *cncMem.axSect[n].flgN2P.bInPos* is set to FALSE.**

G60	**Exact stop, self-maintaining**
Format	G60
Explanation	**Record change occurs when the set position has been reached and the following error of all axes in the subsystem is less than the respective stop tolerance range (Q.048).**
Notes	**As long as the axis is not in exact position the shared RAM variable *cncMem.axSect[n].flgN2P.bInPos* is set to FALSE.** **G60 is the default setting. It can be deselected with G61 or G64.**

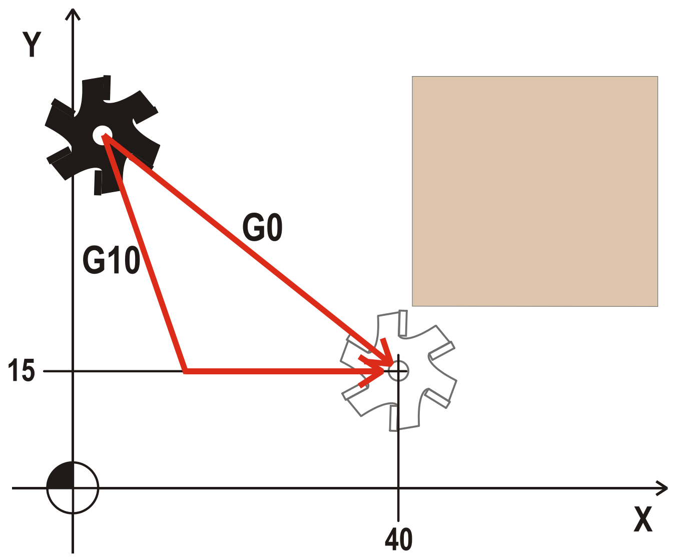
Example	
	N10 G60 G1 X1000 F1000
	N20 X2000 F500

3.7 G10 Point-to-point positioning in rapid feed mode

G10	Point-to-point positioning in rapid feed
Format	G10 X Y X, Y = arbitrary axis letter
Explanation	**Contrary to G0, all axes move at their axis-specific rapid-feed velocity, so they do not normally reach the end position simultaneously.**
Notes	**The record change occurs only after the exact stop position has been reached for all axes, regardless of the record-change function selected with G60 to G64.** **Feed rates programmed in F are retained and are reactivated when G10 is deselected.**

Note

G0 is not suitable for workpiece machining.

Example	
	N10 G10 X40 Y15
	

3.8 G11 Homing

G11	Homing
Format	G11 X X = arbitrary axis letter
Explanation	**The selected axis homes to its reference point**
Notes	**The axes are not interpolated and move at their specific velocities.** **If the axis is not yet synchronized, the system generally drives with the home position search velocities.** **If the axis is synchronized it will move to the home position coordinate at programmed velocity F or rapid feed. The velocity must not be programmed in the G11 record.** **G11 is effective record by record.** **NC-Record preparation is stopped until the NC-Record has been processed (implicit G39).**

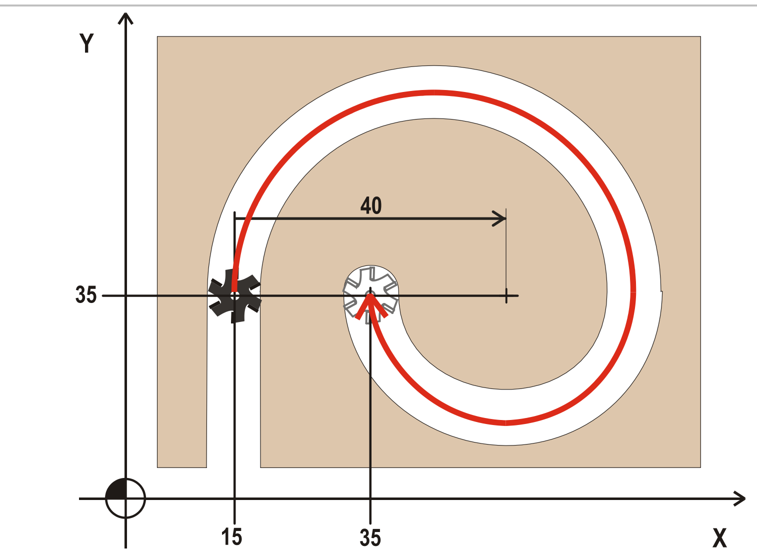
Example	
	N10 G0
	N20 G11 X
	or
	N10 G1 F1000
	N20 G11 X

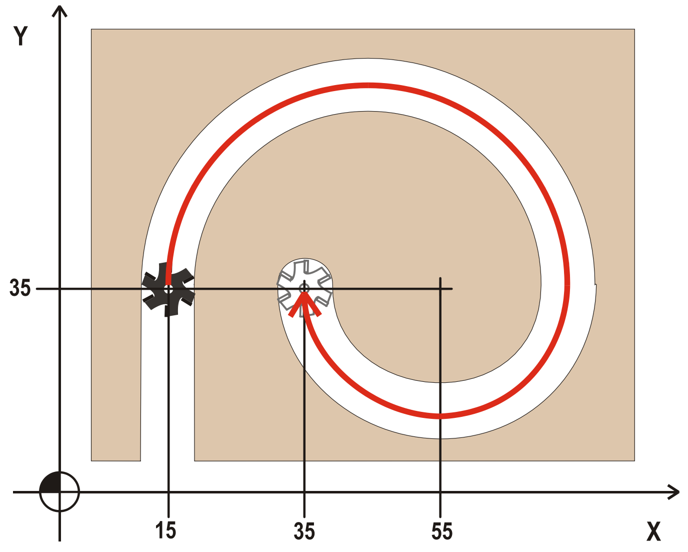
If several axes are moved in one record with G11, the reference coordinate of the axes must be specified.

Example	
	N10 G0
	N20 G11 X0 G11 Y0

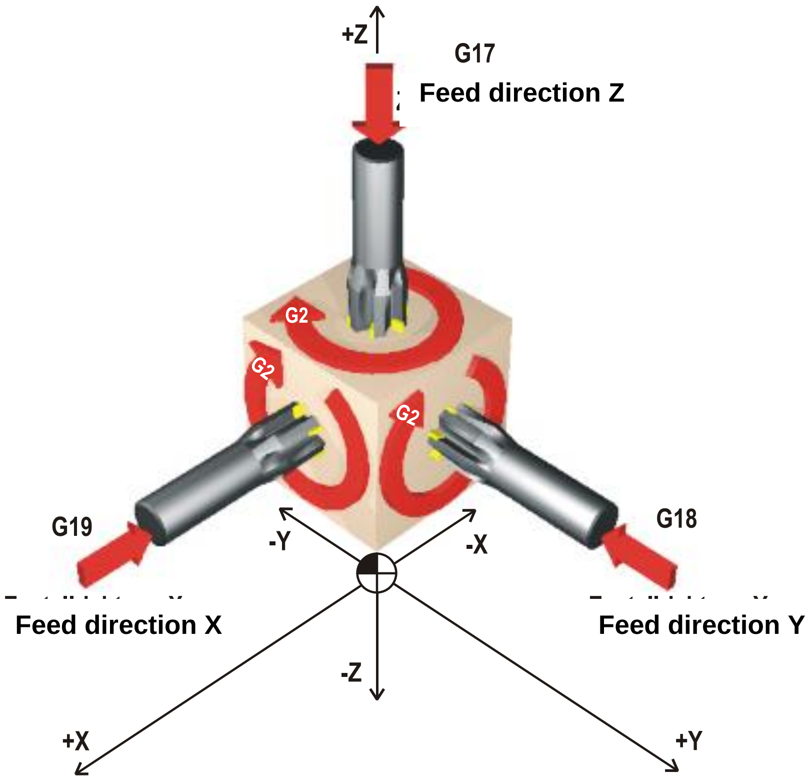
3.9 G12 and G13 Spiral interpolation

G12	**Clockwise spiral interpolation**	
Format	G12	
G13	**Anticlockwise spiral interpolation**	
Format	G13	
Explanation	**G12/G13 programming corresponds to that of G2/G3. In spiral interpolation, the difference between start radius and end radius is travelled at path angle, generating an Archimedean spiral.**	
Notes	**As in circular interpolation, the coordinates of the centre can be specified with I, J, K in absolute or relative terms (Q25 bit 2 = 1 absolute, bit 2 = 0 relative). Axes that have been programmed in addition to the spiral axes are incorporated into the interpolation context in such a way that they reach the end point (Helix interpolation) simultaneously with the spiral axes.**	

Example	Auxiliary coordinates relative to start position (Q25 bit 2=0)	
	N10 G1 X15 Y35 F1000	
	N20 G12 X35 Y35 I40 J0	**(The programmed end position is approached clockwise on a spiral path at a constant path feed rate of 1000 mm/s) The spiral centre coordinates are relative, relating to the start position.)**
		

Example	Auxiliary coordinates absolute (Q25 bit 2=1)
	N10 G1 X15 Y35 F1000
	N20 G12 X35 Y35 I55 J35 **The spiral centre coordinates are absolute, relating to the programmed zero point.)**
	

3.10 G17, G18 and G19 Selecting the work planes

G17	**Plane selection X/Y**
G18	**Plane selection X/Z**
G19	**Plane selection Y/Z**
Format	G17 G18 G19
Explanation	**G17 work plane X/Y feed direction Z** **G18 work plane X/Z feed direction Y** **G19 work plane Y/Z feed direction X** 
Notes	**Machine axes are assigned to spatial coordinates in the configuration parameters Q.054 bits 0 – 2.** **If several axes in one subsystem are assigned to the same dimensional coordinate you can select with $47.** **Plane selection defines the plane for the following functions:** **Circular and helix interpolation (in the figure the direction of rotation is shown for G2/G12)** **Tool length and radius compensation** **G17 is the default setting.**

3.11 G20 through G24 Functions for coordinate transformations

G20	**Deselection of transformation**
G21	**Position specified in cartesian coordinates** PTP drive movement
G22	**Position specified in cartesian coordinates** CP drive movement
G23	**Position specified by the axis positions** PTP drive movement
G24	**Position specified by the axis positions** CP drive movement
Format	G20 G21 X Y Z G22 X Y Z G23 X Y Z G24 X Y Z X, Y, Z = arbitrary axis letters
Explanation	**Software option "06 CNC Coordinate Systems" is required for this function. These G-Words are described in the following manual: "Coordinate transformation, Article No. R4.322.1390.0 (322 140 05)".**

Note

Software option "06 CNC Coordinate systems" also makes G-Words G72 and G74 Page 72 available.

3.12 G25 and G26 Online curve interpolation OCI

G25	**Online curve interpolation OCI without tangential transition**
G26	**Online curve interpolation OCI with tangential transition**
Format	G25 X Y Z G26 X Y Z X, Y, Z = arbitrary axis letters
Explanation	**Contour control for smooth, stepless paths.**
Notes	**You can use the select/deselect condition to specify whether there will be a tangential transition to preceding or following programmed paths. The online curve interpolator requires interpolation points in the form of CNC-Records. G1 or G2/G3 can be programmed at any point to generate sharp corners or straight lines. OCI can be used simultaneously with all axes in the NC subsystem, so you can generate three-dimensional curves of unlimited complexity.**

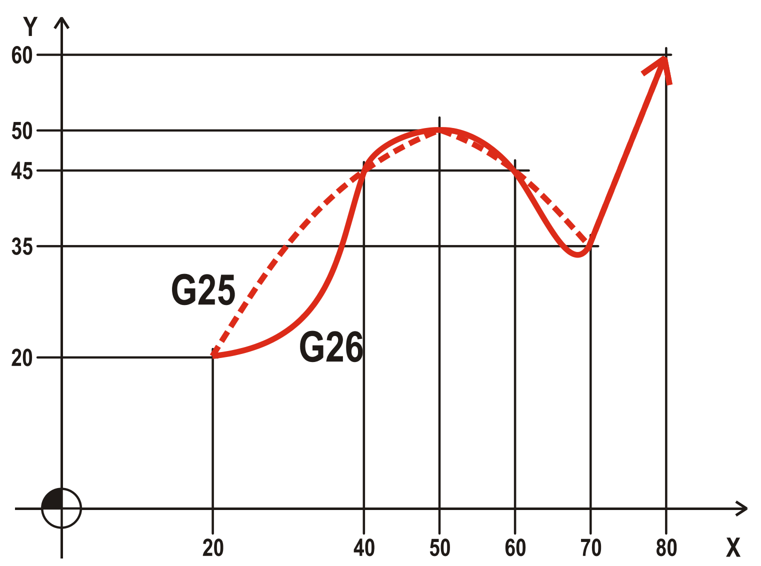
Note

After OCI has been deselected by any G-Function (G0/G1....) at least 2 motion records must follow for OCI to operate.

G39 and NC-Functions containing an implicit G39 are not permissible in the OCI.

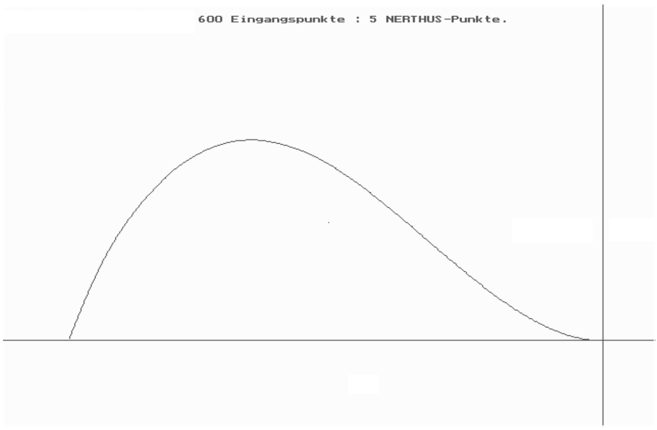
Tool path compensation with G40 through G44 is not permissible in connection with OCI.

Example	G25 OCI without tangential transition.	G26 OCI with tangential transition.
	N10 G1 X20 Y20	N10 G1 X20 Y20
	N20 G25 X40 Y45	N20 G26 X40 Y45
	N30 X50 Y50	N30 X50 Y50
	N40 X60 Y45	N40 X60 Y45
	N50 X70 Y35	N50 X70 Y35
	N60 G1 X80 Y60	N60 G1 X80 Y60
	N70 X125 Y65	N70 X125 Y65



3.13 G27 Freeform interpolation

G27	Freeform interpolation of CNC-Programs created offline
Format	G27
Explanation	**Contour control based on NERTHUS* interpolation point reduction** ***NERTHUS is a Schleicher software product**
Notes	**This function requires offline programming (e.g. CAM system) with subsequent processing by NERTHUS software.** **The NERTHUS software prepares the table of axis coordinates for the G27 function.** **The table can include up to 6 axes of a freeform. It is used to create the CNC subroutine (reduced with NERTHUS), which may not be altered in the CNC controller.** **If a contour correction is required, offline programming and preparation with the NERTHUS software has to be repeated.** **Calculations of compensation and transformation must be done during offline programming.** **Zero point offsets are permissible.** **Please refer to the NERTHUS software manual to ensure correct operation.**

Example	A CNC program created from initially 600 points after processing with the NERTHUS software
	%1 **N0 G1 G90 X-37.937 Y.169** **N1 G27 G64 X-31.16 Y12.503 IX7.1 IY18.858 JX17.964 JY18.858** **N2 X-24.067 Y15.399 JX10.845 JY-1.575** **N3 X-13.617 Y8.802 JX15.977 JY-14.615** **N4 G61 X0 Y0 JX20.819 JY-.099** **N5 M17**
	

3.14 G28 and G29 Update of arithmetic parameters (R-Parameters)

G28	Update arithmetic parameters when record is executed
Format	G28
Explanation	**R-Parameters programmed in the CNC-Record are updated when the corresponding CNC-Record is executed.**
Notes	**G28 is the default setting.**

G29	Update arithmetic parameters when record is executed
Format	G29
Explanation	**R-Parameters programmed in the CNC-Record are updated when the CNC-Record is prepared in the record decoder.**
Notes	**The time for Reading/Writing R-Parameters is therefore undefined. When there are a large number of intermediate buffers and extensive parameter calculations, preparing records can be faster with this function.**

Important!

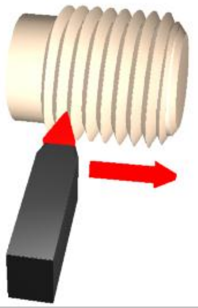
When G29 is used with NC-Start/-Stop or in a single record, the R-Parameters are calculated and entered repeatedly. For example, piece counters then deliver incorrect results (the pieces are counted more than once).

3.15 G32 Tapping with controlled spindle

G32	Tapping with controlled spindle
Format	G32 Z I Z = arbitrary axis letter I = pitch of thread
Explanation	**In contrast to tapping with** Fehler! Verweisquelle konnte nicht gefunden werden. **In this function the spindle is interpolated with the lead axis. This requires a position-controlled spindle.** **The thread pitch I can be positive (tapping with M3) or negative (tapping with M4). I is only programmed in the first G32 record.** **G32 is especially suited for blind holes, because the exact thread depth is achieved.**
Notes	**G32 must be called when the spindle (M5) is stationary.** **The lead axis must be specified with $33 before G32 is called.** **The speed of the spindle must be programmed in S.** **M3, M4 and M5 must not be used.** **Single record mode, speed override and the stop key are not locked. All other modes are locked.** **A thread can also be programmed with several G32 records. Record change conditions G60, G61 or G64 apply. This makes it possible, for example, to output an M-Function during tapping.**

Example	#	
	N10 $33 Z S2000 M5	**Lead axis, 2000 r.p.m., spindle stop**
	N20 G0 Z200 (C90)	**Start position (possibly also for spindle)**
	N30 G32 Z190 I2	**Thread with M3, pitch 1 mm**
	N40 Z200	**Z back, spindle reversed**
	N50 G0	**Continue with G0**

3.16 G33 Thread cutting single record

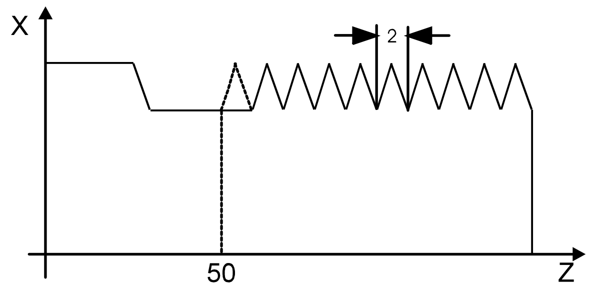
G33	**Thread cutting single record** 
Format	G33 X Z K X, Z = axis letters I, J, K = auxiliary coordinates
Explanation	
Notes	**Function G33 requires a spindle with a positioning transducer. The spindle can be operated as a controlled, uncontrolled or PLC-controlled spindle.** **Before G33 is called:** **the direction of rotation of the spindle and the speed must be programmed,** **the lead axis must be declared with the $33 function.** **Record change must not occur until the spindle is turning in the programmed direction. Right-hand or left-hand thread is decided by the direction of rotation of the spindle and the travel direction. A later alteration is not possible.** **If the direction of rotation of the spindle changes the axis returns to the record start position and stops there.** **Depending on the lead axis, the thread pitch is programmed with an auxiliary coordinate I, J, K (X = I, Y = J, Z = K).** **Cycle G76 is available for thread cutting.**

Note

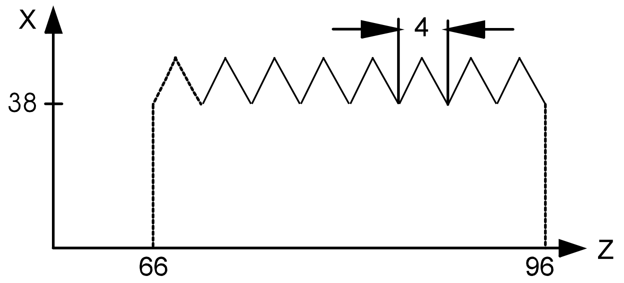
Interlocks with G33:

- **Override is set to 100%.**
- **Stop key is locked.**
- **In single record mode stop is not until after the last G33 record.**
- **A change in the mode of operation is not possible until after the last G63 record.**

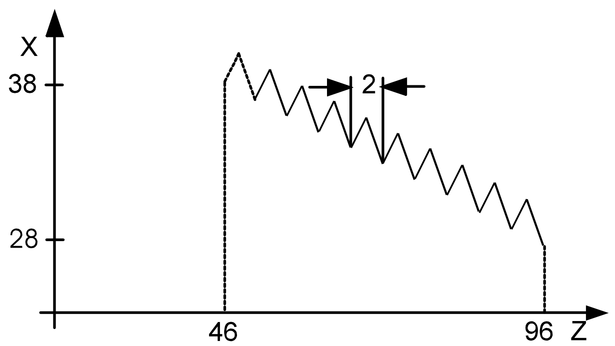
Example	Cylindrical thread	
	N10 M03 S700	**(Spindle on, speed 700 r.p.m.)**
	N20 $33 Z	**(Z is the lead axis)**
	N30 G33 Z50 K2	**(Pitch is 2 mm)**



Example	Cylindrical thread (a chaser)	
	N10 M03 S700	**(Spindle on, speed 700 r.p.m.)**
	N20 $33 Z	**(Z is the lead axis)**
	N30 G0 X38 Z96	**(Home to start position)**
	N40 G91 G33 Z-30 K4	**(Thread length 30 mm with 4 mm pitch)**
	N50 G0 G90 X35 Z98	**(Move to end position)**
	N60 M05	**(Spindle off)**



Example	Conical thread (a chaser)	
	N10 M03 S900	**(Spindle on, speed 900 r.p.m.)**
	N20 $33 Z	**(Z is the lead axis)**
	N30 G0 X28 Z96	**(Home to start position)**
	N40 G91 G33 X10 Z-50 K2	**(Cone 10x50, pitch 2 mm)**
	N50 G0 G90 X40 Z98	**(Move to end position)**
	N60 M05	**(Switch spindle off)**

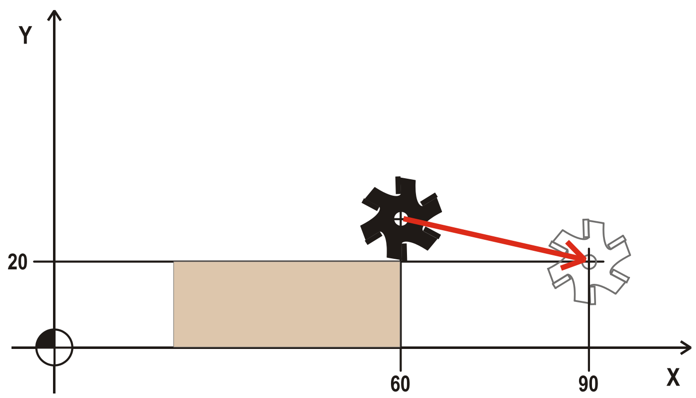
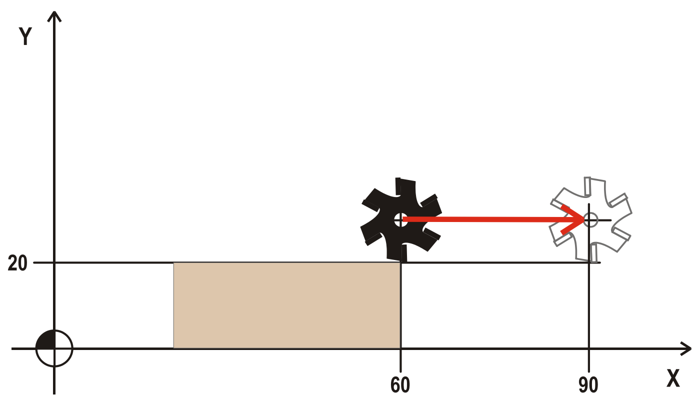


3.17 G39 Stop record preparation

G39	Stop record preparation
Format	G39
Explanation	**Record preparation (decoding) stops until the buffer is empty and the last prepared record has been processed.**
Notes	**G39 is activated automatically in the following functions:** **G11 Homing** **E1 = 0 or 1 communication flag comparison** **Read/write Q parameter by record** **Change NC axes between NC subsystems** **$1 Stopping an axis** **$25 Switch off follow-up operation** **$28 Reintegrate axis in record change** **$32 when Q37 bit 1 = 1** **$40 Oscillation off**

3.18 G40 Switch off tool radius compensation

G40	Switch off tool-radius compensation
Format	G40 [X Y Z F] X, Y, Z = arbitrary axis letters F = path feed rate
Explanation	**Tool radius compensation is switched off**
Notes	**If G40 is programmed with a motion the tool radius compensation is activated on the path. If G40 is programmed without a motion the tool centre becomes the actual position of the axis. The tool radius compensation is also switched off with M30.**

Example	G40 with motion
	N30 G41 X... F500
	N40 G41 X60 F500
	N50 G40 X90 F500
	
Example	G40 without motion
	N30 G41 X... F500
	N40 G41 X60 F500
	N50 G40
	N60 G1 X90
	

3.19 T-Word tool selection for tool compensation

T	**Tool selection**
Format	Tnn **nn = number of Tool data memory, 2-digit decimal number**
Explanation	**The Tool data memory nn is selected and activated.** **Working with the tool data memory requires a PLC program to confirm the tool call or tool change. No record change takes place without a PLC program.**
Notes	**Tool data, stored in Tool data memory, is taken into consideration in the travel instruction. It remains valid until another tool is selected or tool compensation is switched off with T0.** **The number of the selected tool data memory is continuously displayed in shared RAM variable *cncMem.sysSect[n].wrdN2P.lToolMem*.** **The T-function call is indicated by a change-signal in the coupler memory variables *cncMem.sysSect [n].flgN2P.bTFctMod*.**

Example		
	N10 G1 X100 Y50 T01	**Tool 1 selected**
	N20 G0 X0 Y0 T0	**Tool compensation switched off**

Important!

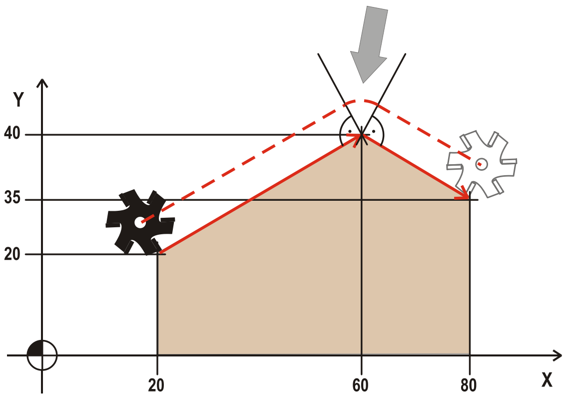
The tool is regarded as deselected after M30 or a program abort through RESET. When starting or restarting an NC-Program, the tools must be selected before processing can be started.

The actual value display is corrected accordingly.

Important!

If an NC-Program is to be executed in several subsystems, ensure that the Tool data memory with corresponding tool data is entered in each subsystem when you select tools.

3.20 G41/G42 Tool radius compensation

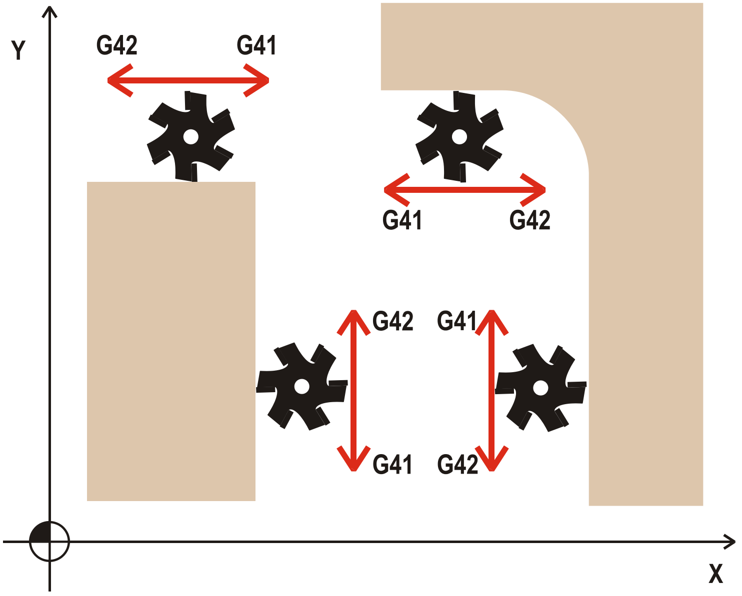
G41 **G42**	**Tool radius compensation (WRK) left of contour** **Tool radius compensation (WRK) right of contour**
Format	G41 X Y Z G42 X Y Z F X, Y, Z = arbitrary axis letters F = path feed rate
Explanation	**With functions G41 and G42 you can carry out tool path compensation regardless of the tool data.**
Notes	**You can compensate tool radius WRK (default setting) or tool nose radius SRK. To activate the tool nose radius compensation, a compensation quadrant must be selected and entered into the Tool data memory. See Quadrant assignment in the annex.** **Tool selection** **Before WRK a tool must be selected with the T word. Tool compensation data for the tool must be stored in the corresponding Tool data memory.** **Plane selection** **A machining plane for the WRK must be selected using instructions G17, G18, G19. The machining plane cannot be altered while WRK is active.** **Approach** **When a WRK has been selected the tool radius is activated in the first positioning record. The selection must be made outside the machining contour, and the approach path must be clear. See also Approach and departure strategies in the annex.** **Compensation is parallel to the contour. The axes are moved so that the tool centre is perpendicular to the programmed contour.** **Depending on the programmed contour, transition radii may be inserted by the record decoder.**  **The transition radius is a separate record, displayed under the number of the preceding record. These records are not taken over into the NC-Program, instead they are only saved in the buffer. G50 works without the insertion of interim records.** **The tool centre is always displayed in the actual and set value displays.** **Feed rate calculation can be switched with G45 or G46.**

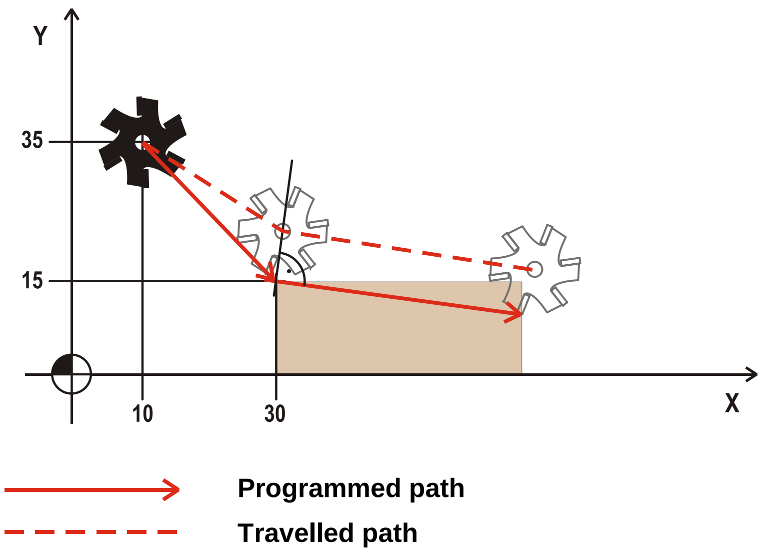
Important!

When tool radius compensation is active the following restrictions must be considered.

- **When there are several sequential NC-Records without drive motion, the program may stop without an error message. In this case the number of NC-Records without drive motion must be reduced.**
- **Sequential NC-Records must not contain identical coordinates (Error 0x21300005).**
- **The programmed radius of the workpiece contour must be greater than the tool radius.**
- **Where there are interior corners, ensure that the tool can drive into the corner (Error 0x21300003).**
- **Tool and tool memory cannot be changed.**
- **The machining plane cannot be altered.**
- **G39 or a function resulting in an implicit G39 must not be used.**

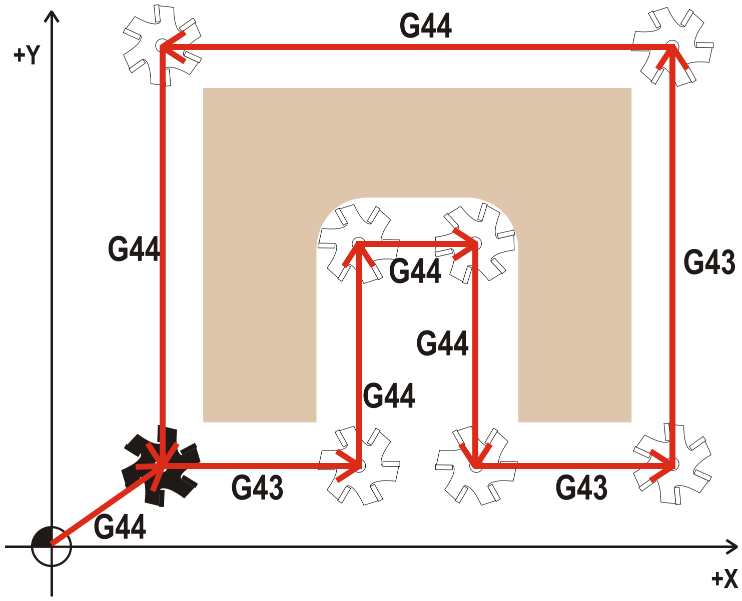
You may have to deselect tool radius compensation with G40.

Example	G41 and G42
	

Example	G41 approach
	N10 T01 G41 X30 Y15 F500
	

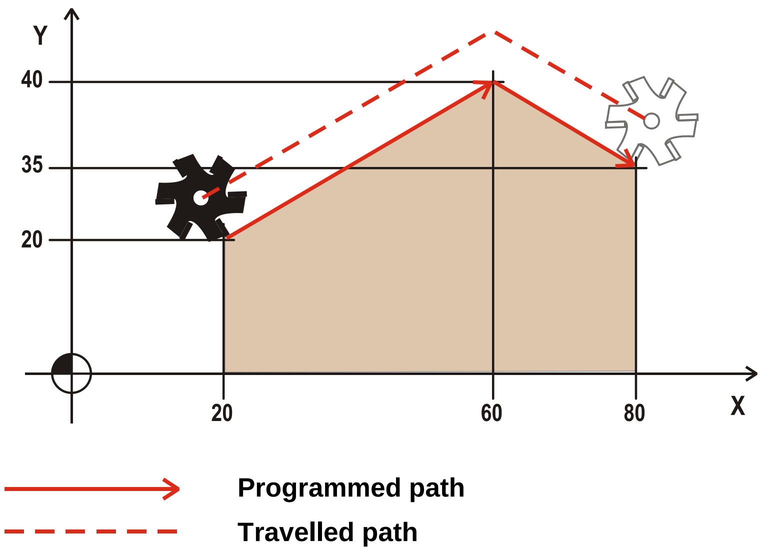
3.21 G43 / G44 Tool radius compensation, positive/negative

G43 **G44**	**Tool radius compensation positive** **Tool radius compensation negative**
Format	G43 G44
Explanation	**Tool radius compensation parallel to coordinate axes**
Notes	**Tool selection, plane selection and restrictions in programming are identical to functions G41 and G42.**

Example	
	

3.22 G50 Tool radius compensation without transition contour

G50	**Tool radius compensation between straights without transition radius**
Format	G50
Explanation	**No transition radius is inserted at a straight-straight transition on an outside corner.**
Notes	**The start and end coordinates are recalculated.** **G50 is effective record by record.**

Example	
	N60 G41 X20 Y20
	N60 G50 X60 Y40
	N70 X80 Y35
	

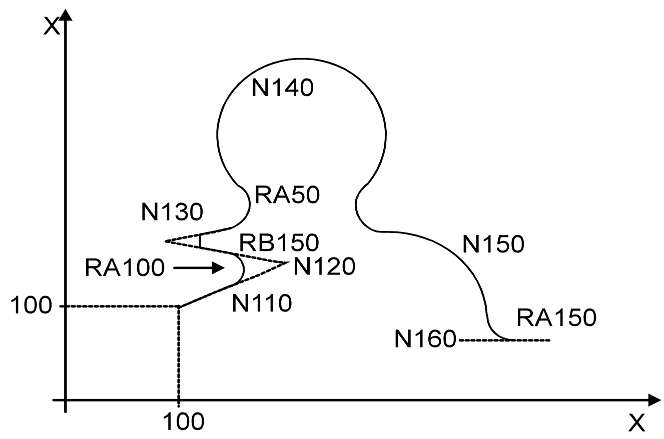
3.23 G45/G46 Path feed rate compensation

G45 **G46**	**Switch path feed rate compensation off** **Switch on path feed rate compensation**
Format	G45 G46
Explanation	**Path feed rate is calculated on the programmed contour, not in relation to the tool centre.** **The resulting velocity is restricted to the range 50 % to 200 % of programmed velocity. E.g. transition radii on outside corners are executed at 200 %.**
Notes	**G46 is effective only with active tool radius compensation.** **G45 is the default setting.**

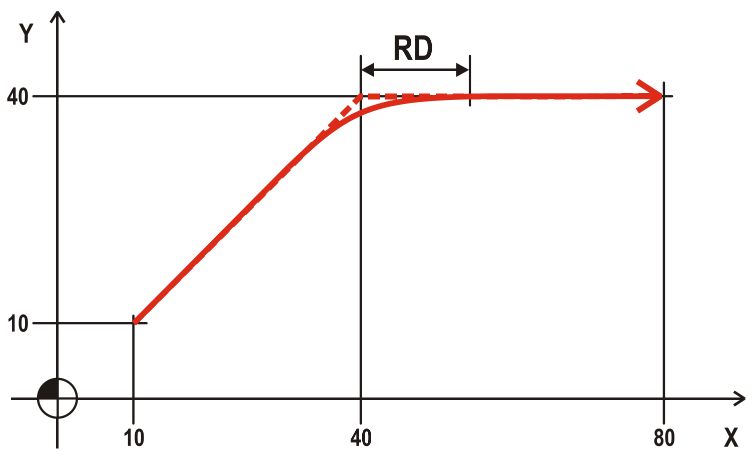
3.24 Smoothing RA, RB, RD, RF

RA	**Smoothing with transition radius between arc and straight line**
RB	**Smoothing with chamfer between straight lines**
Format	RAnnnn RBnnnn nnnn = decimal number, radius / chamfer length
Notes	**For RA and RB a working plane must be selected (e.g. G18 for XZ plane).** **G39 or a function resulting in an implicit G39 must not be programmed immediately after or in the following record.**

Example	
	N100 G1 F100 X1000 Y100
	N110 X900 Y250 RA100
	N130 X300 Y550 RB150
	N130 X700 Y650 RA50
	N140 G2 X1020 Y650 RC-330 RA100
	N150 X1370 Y100 RC330 RA150
	N160 G1 X1700



RD	Smoothing with parabola between straight lines
Format	RDnn
Explanation	**A parabola is inserted in the straight-straight transition.**
Notes	**Applies to any axes, between two straight lines (G0/G1) without velocity reduction (G64).** **The parameter indicates the distance from the start and end positions of the inserted parabola to the vertex.** **If RD = 0 RD will not be executed.** **If the value of RD is greater than 40 % of the path length of one of the two NC-Records, RD will be limited to 40 % of the path length of the shorter record.** **From version OS06/40.0:** **However, the smoothing path is now limited only to the total path length of the number of NC-Records set in Q109. The limitation is flexible.** **The path feed rate can be specified as percent of the path feed rate programmed in F using FFnnn. See Example 2. When ramp type 2000 (ACC2100) is simultaneously selected, this path feed rate will be achieved at the start of the transition record.** **G39 or a function resulting in an implicit G39 must not be programmed immediately after or in the following record.**

Example 1	
	N10 G0 X10 Y10
	N20 G1 G64 X40 Y40 RD20 F1000
	N30 X80
	

Example 2	Influencing velocity with FF	
	N10 G1 X0 Y0 F2000	Start position, path feed rate 2000 mm/min
	N20 G64 X20 Y100 RD20 ACC2100 FF40	Smoothing with RD, path feed rate 40 %
	N30 X40 Y0	Continue with 100 %

RF	Axis-specific smoothing with soft acceleration
Format	RFxnn x = axis letter, nn = feed rate at which maximum acceleration is reached.
Explanation	The RF function is self-maintaining. Deselection is done with RFx 0 or with RF 0 for all axes. The RF value can be greater than the programmed feed rate. Then this axis does not reach the possible acceleration. The movement will be softer. If the feed rate is programmed to be greater than the RF value of an axis, the permissible acceleration of this axis is exceeded (G64) or the feed rate is automatically reduced (G62).
Notes	The RF function must only be used in conjunction with G1. As long as RF is effective on an axis, RA, RB and RD cannot be used. When a robot transformation is activated, the RF function cannot be used. The RF function should be activated only for those axes where it is needed, because it also uses additional computing time for those axes that are not moved. The RF function is only effective with G62 and G64. When G61 and G64 are alternately programmed, the RF values are self-maintaining. With G9 and G39 the RF function for this record is suppressed.

Example		
	N100 G64 RFZ2000 F2000	
	N500 C20 Z5	
	N600 C20 Z15	
	N700 G61 C20 Z12	(RF N700 N800 not effective)
	N800 G64 C20 Z10	(RF N800 N900 effective)
	N900 G9 C20	(RF N900 N1000 not effective)
	N1000 C20 Z15	(RF N1000 N1100 effective)
	N1100 G61 C20	

3.25 G52 Coordinate rotation

G52	Coordinate rotation
Format	G52 Xnn Ynn Inn I is the angle of rotation in radians J, K specify the angle of rotation in degrees
Explanation	**The coordinate rotation can be used to adapt the coordinate system of the workpiece to that of the machine. This rotation then takes place in the plane selected with G17 / G18 / G19.**
Notes	**The centre of rotation is the workpiece zero point, which is determined by the G54 through G57 displacement. This point can be displaced again when coordinate rotation is called.** **G52 X.... Y.... I....** **'X' and 'Y' indicate the position of the centre of rotation relative to the workpiece zero point. The angle of rotation α is programmed under 'I', 'J' or 'K'.** **G52 I0 can be used to deselect coordinate rotation in the NC-Program. M30 or RESET can also be used to deselect coordinate rotation.** **The coordinate rotation is not taken into consideration in the actual value display. After coordinate rotation the first position must be approached with a straight (G0, G1).** **When tool nose radius compensation (SRK) is selected, coordinate rotation must not be changed.**

Example	%1 (tilted 45 degrees)	
	N10 G0 X0 Y0	
	N20 G54 X0 Y0	(Selection zero-point offset X25 Y10)
	N30 G52 J90	(Rotation selection)
	N40 X5 Y5	
	N50 X20	
	N60 Y15	
	N70 X5 Y20	
	N80 Y5	
	N90 X0 Y0	
	N100 G52 X0 I0	(Rotation deselection)
	N110 G53 X0 Y0	(Deselection zero-point offset)
	N120 M30	

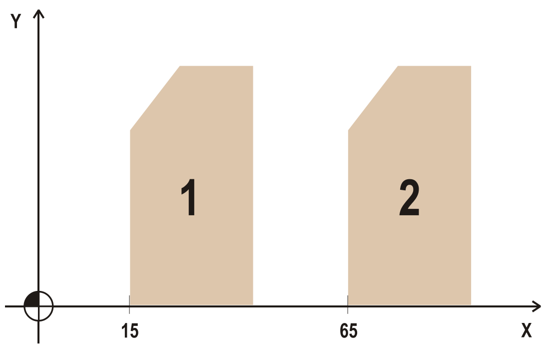
3.26 G53 through G59 Zero point offset

G54	Zero point offset 1 (parameters starting R10001)
G55	Zero point offset 2 (parameters starting R10101)
G56	Zero point offset 3 (parameters starting R10201)
G57	Zero point offset 4 (parameters starting R10301)
G58	Zero point offset 5 (parameters starting R10401)
G59	Zero point offset 6 (parameters starting R10501)
G53	Deselect zero point offset
Explanation	There are 6 zero point offsets (G54 through G59), which are normally used to describe the workpiece zero point. With each zero point offset the zero point can be displaced for all axes simultaneously.
Notes	The zero point offset values are saved on R-Parameters. They can be written and read by the NC-Program. The R-Parameters have fixed assignments to the G-Words and the axes. (See also arithmetic parameters Zero point offsets R10001 through R10564 Zero point offsets R10001 through R10564) Example for G54: R10001 = 1st axis, R10002 = 2nd axis, ... R10064 = 64th axis Functions G54 through G59 cancel each other. Functions G54 through G59 and G92 are executed simultaneously. Zero point offsets G54 - G59 and reference point offset G92 are deselected with G53. Deselection with G53 is effective record by record, self-maintaining deselection can be set (Q38 bit 6 = 1). Zero point offset is also deselected with M30 .

Important!

If zero point offset is selected with motions the zero point offset is taken into consideration for the target coordinates.
If a zero point offset is selected without motion only the displayed values for axis set and actual positions will be converted.

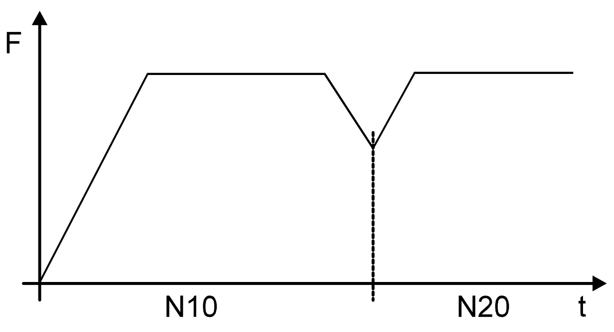
Example	Specification: X-axis is first axis	
	N10 G54 X0 G0 B% 4711	(Processing part 1 R10001=15)
	N20 G55 X0 G0 B% 4711	(Processing part 2 R10101=65)
	N30 G53 X0 G0	(Deselect with motion)



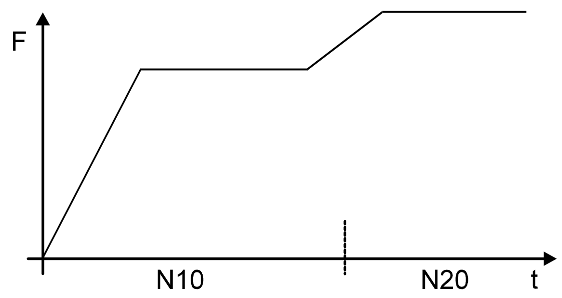
3.27 G61, G64 Smoothing

G61	Smoothing
Format	G61
Explanation	**Record change occurs when the set position is reached (set-actual deviation = 0).**
Notes	**The axes follow the position values from the controller, displaced by the following error. Record change occurs regardless of following error when the set position of each axis is equal to the programmed coordinate.** **G64 can be deselected with G60 or G64 or overwritten record-by-record by G9.**

Example	#
	N10 G61 G1 X1000 F1000
	N20 X2000



G64	Smoothing without loss of velocity
Format	G64
Explanation	**The record change takes place without a braking ramp when the difference between set and actual is = 0. Any residual path of the Interpolator is taken over into the next record, so that there is no loss of speed.**
Notes	**With G64 there is also a record change with the FF-programmed reduced speeds.** **If G64 is selected, waitinG-Functions (WA, WN, TI) should not be used because they prevent acceleration monitoring.** **Record change is executed independently of PLC enable for M and T functions.** **G64 can be deselected with G60 or G61 or overwritten record-by-record by G9.**

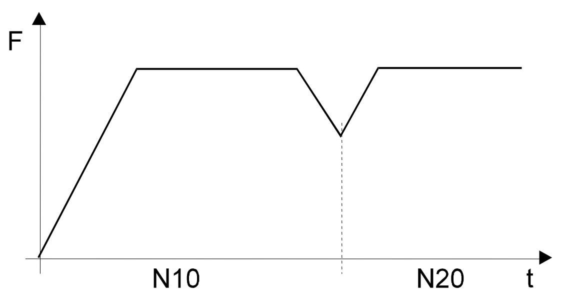
Example	#
	N10 G64 G1 X1000 F1000
	N20 X2000 F1200
	

Note

On non-tangential contour transitions (e.g. angle between 2 consecutive straight lines > 7°) G64 may act like G61 due to the acceleration monitoring.

3.28 G62 Record-change with acceleration monitoring

G62	Record change with acceleration monitoring
Format	G62
Explanation	**The record change takes place when the difference between set and actual is = 0. Any residual path of the Interpolator is taken over into the next record, so that there is no loss of speed.**
Notes	**At the same time with G62, acceleration monitoring is activated.** **From SW release OS06.26/0:** **G62 with jerk limitation is active at setting Q38, bit 4 = 1; the jerk is monitored when driving with the Sin^2 ramp.** **With OCI (G25/G26), Q38 bit 4 must be set! Otherwise, the maximum speed is exceeded.** **This will reduce the path speed by an amount necessary so that none of the participating axes exceeds the maximum acceleration value set in Q.024 through Q.027 or the acceleration values specified (and possibly reduced) with ACC. This function applies to all preset and programmed ramp functions (ACC0100, ACC1100, ACC2100, ACC3100).** **This applies both to discontinuous transitions (corners) and to transitions with RD (between G0 or G1 records) and to small arcs (G02/G03/intermediate records of SRK). The velocity is reduced so that acceleration values are not violated.** **When G64 is selected, no wait functions (WA, WN, TI) should not be used, because in this case no acceleration monitoring can take place.** **Record change is executed independently of PLC enable for M, H and T functions.** **Subroutine calls and returns are possible without loss of velocity.** **Preconditions:** **B%xxx or M17 programmed in preceding positioning record.** **No robot transformation active.** **G64 can be deselected with G60, G61 or G64 or overwritten record-by-record by G9.**

Example	
	N10 G62 G1 X500 Y200 F1000
	N20 X700 Y500
	

3.29 G63 Tapping without compensating chuck

G63	Tapping without compensating chuck as single record
Format	G63
Explanation	
Notes	**Function G63 requires a spindle with a path measuring transducer. The spindle can be operated as a controlled or uncontrolled spindle. Before G63 is called in the record, the lead axis must be declared via the $33 function.** **Before the first G63 record is called, the direction of rotation and speed of the spindle must be programmed. Record change must not occur until the spindle is turning in the programmed direction (M bit acknowledge). When G63 is active right-hand or left-hand thread is decided by the direction of rotation of the spindle and the travel direction. A later alteration is not possible.** **If the direction of rotation of the spindle changes the axis returns to the record start position and stops there.** **Thread pitch is programmed using the auxiliary coordinates I, J, K.** **With a controlled spindle, G32 can be used for interpolating in place of G63.**

Example	**#**	
	N10 G0 $33 Z0 M00 M03 S500	
	N20 G63 Z200 I2 M03	
	N30 Z220 M05	**Reversing record: The programmed thread depth must not be reached. Record change occurs when spindle stops.**
	N40 Z20 M04	
	N50 Z0 M05 BN20-	

Important!

- **Locking with G63:**
- **Override is set to 100%.**
- **Stop key is locked.**
- **In single record mode, a stop is not made until after the last G63 record.**

Mode change is not possible until after the last G63 record.

With G63 and NC-reset the spindle is stopped and the NC-Program is deselected. Gear coupling remains engaged and all modes apart from automatic are locked. G63 and thread pitch remain self-maintaining. Spindle is set to M05 S00.
Deselect G63 locking:
If an error occurs and the return program is not possible, e.g. borer is broken off, G63 locking can be cancelled by programming G0 or G1. After G0 resp. G1 have been processed, the 'Reset' key must be pressed.

Note

After programming new programs with G63 or program cycles with G77, it is recommended to do a trial program run without a workpiece.
If the "Thread Error" message appears at the thread return point the program will stop. The "Thread Error" message appears if the spindle cannot stop within the calculated distance.
In this case the thread depth must be corrected in the program.

3.30 G66 Synchronization of the IPO interpolation points

G66	Synchronization of the IPO interpolation points
Format	G66
Explanation	**G66 is used for correcting the speed over multiple records so that the record endpoint is reached in the IPO cycle. In this way it is possible to avoid beats with program loops without halting.**
Notes	**G66 should be programmed only once.** **When the axes are halted in the program loop, G66 is superfluous.**

3.31 G67 Special function for oscillating

G67	Special function for oscillating
Format	G67
Explanation	**Influences reversing behaviour of oscillating axis or first feed axis.**
Notes	**G67 is effective record-by-record and has no effect without $40 through $44.** **Reversing behaviour without G67.** **The oscillation axis remains at the reversal point until the respective infeed axis has completed its infeed increment and the programmed precise-halt condition is met. Infeed begins when the oscillating axis is at the reversal point and in turn it meets the precise-halt condition.** **The same conditions apply for the first infeed axis if a second infeed axis is programmed.** **Reversing behaviour with G67** **The oscillation axis initiates the infeed process by reaching the reversal point and meeting the programmed precise-halt condition, but changes direction before infeed has been completed. The same conditions apply for the first feed axis if a second feed axis is programmed.**

3.32 G70 and G71 inch/metric switching

G70	**Dimensions in inches**
Format	G70

G71	**Dimensions in mm**
Format	G71
Notes	Inch/mm switching relates only to the programmed coordinates. Zero point offsets, tool compensations and system parameters are not converted. They are always interpreted according to the machine data setting. The program sequence figure in the monitor displays set and actual values and set-actual deviation in the selected system of units (mm or inches). Actual values and coordinates are displayed in the selected system of units. The internal parameters are saved in floating point format, but displayed in the IPO resolution, i.e. with G70 in inches, with G71 in mm. G71 is the default setting The conversion for the feed rate F, S is set at Q25 bit 4 = 1.

Important!

G70/G71 is self-maintaining even through controller on/off.

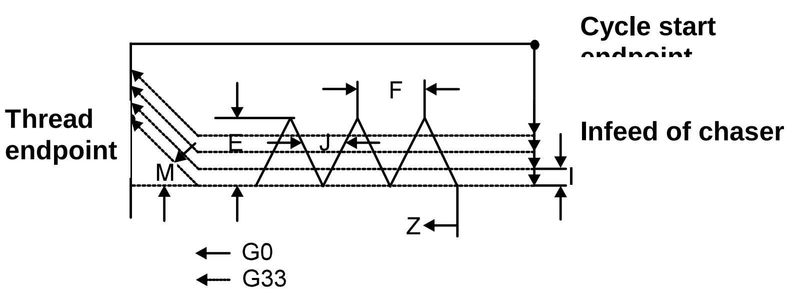
3.33 G72 and G74 Functions for coordinate systems

G72 **G74**	**Coordinate systems: Selection of reference system** **Coordinate systems: Selection of compensation system**
Format	G72 FMn G74 FMn FM = System selection, n = System number
Explanation	**Software option "06 CNC Coordinate Systems" is required for this function. These G-words are described in the following manual: "Coordinate systems, Article No. 322.153.86)".**

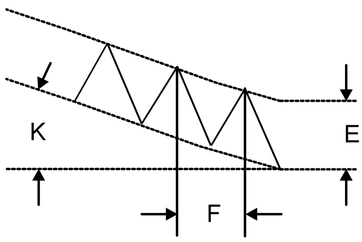
Note

G-Words G20 through G24 on page 47 are available for editing the coordinate transformation.

3.34 G76 Thread cutting cycle

G76	Thread cutting cycle	
Format	G76 Z X Z, X = axis letters	
Explanation	Thread cutting in cycle	
Notes	The syntax for thread cycle is programmed in 2 records. Here an example for thread in Z direction: N.. X.. Z.. $33 Z X and Z define the start and end position of the cycle (outside the workpiece). $33 Z defines the pitch in Z direction N... G76 X.. Z.. F.. E.. H.. [I..] [J..] [K..] [M..] X.. Z.. Thread end point F Pitch of thread E Depth of thread H Number of chasers I (opt.) Final machining allowance J (opt.) Thread angle in degrees K (opt.) Conical angle in degrees M (opt.) Overrun angle in degrees Parameters I-M are optional. If not programmed default values are: I = 0.0 J = 0.0 K = 0.0 M = 0.0 External thread if cycle start (X) > thread end position (X) Internal thread if cycle start (X) < thread end position (X) Active preparatory function at cycle end is G0 All parameters can be parameterized with R-Parameters	
Example	**%1**	
	N10 T01 M03 S700	**(Switch spindle on, speed 700 r.p.m., select tool 1)**
	N20 $33 Z	**(Z is the lead axis)**
	N30 G0 X38 Z0	**(Home to start position)**
	N40 G76 X20 Z-50 F2.5 E5 H5 I0.5	**(Thread depth 5 mm, 2.5 mm pitch)**
	N70 M17	**(End subroutine)**
		

Example	Conical thread	
	%1	
	N10 T01 M03 S700	**(Switch spindle on, speed 700 r.p.m., select tool 1)**
	N20 $33 Z	**(Z is the lead axis)**
	N30 G0 X38 Z0	**(Home to start position)**
	N40 G76 X20 Z-50 F2.5 E5 H5 I0.5 K20	**(Thread conical 20 degrees)**
	N70 M17	**(End subroutine)**



Note

If J > 0 then:
The infeed takes place in the direction of half the thread angle, so that from the second cut only one cutting edge is engaged.

$$\Delta Infeed = CurrentDepth \bullet \tan\left(\frac{J}{2}\right)$$

The above formula ensures constant-volume cut segmentation, so that cutting forces are as constant as possible.

If final machining allowance I = 0 (default setting) a non-cutting pass will be carried out.

Cut segmentation

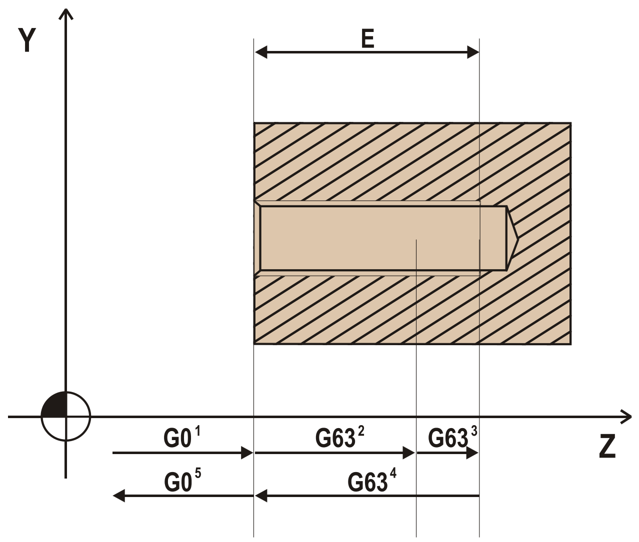
$$CurrentDepth = \frac{E-1}{\sqrt{H-1}} \bullet \sqrt{CurrentCutNumber}$$

$$= \text{E bei } (\text{H} = 1) \text{ or last cut}$$

3.35 G77 Tapping without compensating chuck cycle

G77	Tapping without compensating chuck cycle
Format	see note below
Explanation	**G77 controls the complete G63 sequence**
Notes	**G77 contains the following working steps:** **Z sets the cycle starting point** **(starting point outside of workpiece)** **G77 Z.. E.. [J..] F.. [S..] [TI..]** **Z.. Thread start point** **E Thread depth** **F Thread pitch.** **J (opt.) Chamfer angle** **S (opt.) Return speed** **TI (opt.) Dwell time when reversing** **Active preparatory function at cycle end is G0.** **All parameters can also be parameterized with R-Parameters** **The syntax for thread cycle is programmed in 2 records.**

Example	
	N10 G0 Z10 M4 S250
	N20 G77 Z13 E25 J0 F2 S200 TI0,5
	N30 G0 Z10
	N40 M5 M17



1) Approach with G0 up to acceleration path before start of thread
2) Turn on G63 drag compensation
Thread tapping G63 up to deceleration path spindle
3) Thread tapping G63 with spindle decelerating
Record change when spindle stops. The thread depth must not be reached. (Error message thread error)
4) Return to thread start at return speed and
5) Deceleration at starting point

Note

After programming new programs with G63 or program cycles with G77 we recommend running the program without the workpiece.
If the "Thread Error" message appears at the thread return point the program will stop. The "Thread Error" message appears if the spindle cannot stop within the calculated distance.
In this case the thread depth must be corrected in the program.

3.36 G80 through G89 Machining cycles G80 through G89

G80	**Cancel machining cycle**	
G81	**Machining cycle 1**	**" %99999981"**
G82	**Machining cycle 2**	**" %99999982"**
G83	**Machining cycle 3**	**" %99999983"**
G83	**Machining cycle 4**	**" %99999984"**
G85	**Machining cycle 5**	**" %99999985"**
G86	**Machining cycle 6**	**" %99999986"**
G87	**Machining cycle 7**	**" %99999987"**
G88	**Machining cycle 8**	**" %99999988"**
G89	**Machining cycle 9**	**" %99999989"**
Format	G81	
Explanation	**A machining cycle is carried out after execution of each record containing a motion.**	
Notes	**The machining cycle is programmed as a subroutine under the corresponding program number (%99999981 - %99999989). The machining cycle call is self-maintaining. I.e. once programmed the machining cycle is executed after each motion record until it is overwritten by calling another machining cycle or cancelled by programming G80.**	

Example		
	N100 G00 X50 Y50 G81	**Call machining cycle %99999981, execute after reaching programmed position.**
	N110 X100 Y100	**Execute %99999981 after reaching the programmed position.**
	N130 Y120 G80	**Cancel programmed machining cycle, %99999981 is <u>no longer</u> executed.**

3.37 G90, G91 Measurements absolute / incremental

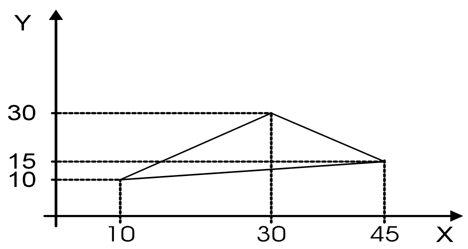
Axis-specific measurements can be programmed with $90/$91.

G90	Absolute measurements
Format	G90
Explanation	**All measurements relate in absolute terms to the current zero point.**
Notes	**If no zero point offset is active, this is the zero point defined by the reference point coordinate. It can be altered by zero point offset G54 through G59, G92 or with Zero point overlays R10601 through R10664.** **G90 is the default setting.** **This function can also be programmed for individual axes (see $90)**

Example	
	N10 G0 G90 X10 Y10
	N20 G1 X30 Y30 F1000
	N30 X45 Y15
	N40 X10 Y10

G91	Incremental measurements
Format	G91
Explanation	**The programmed value corresponds to the distance to be travelled.**
Notes	**The auxiliary coordinates (I, J, K) for circle programming are not affected by G90/G91. The setting in the configuration parameter always applies.** **Programming G91 can be disabled by the setting in Q25 bit 0=1.** **This function can also be programmed for individual axes (see 91)**

Example	
	N10 G0 G90 X10 Y10
	N20 G1 G91 X20 Y20 F1000
	N30 X15 Y-15
	N40 X-35 Y-5



3.38 G92 Reference point offset

G92	Reference point offset
Format	G92 X Y **X, Y arbitrary axis letters**
Explanation	**With G92 you can set the reference point for individual axes.**
Notes	**In the rest of the program all axis coordinates used will relate the coordinates set with G92.** **The difference between actual value and reference point offset is entered in R-Parameters and can be read by the NC-Program. (See arithmetic parameters R10701 through R10764 Reference point offset.)** **Reference point offset is inactive as long as G53 is active.** **The actual value memory is deleted with M30 or RESET.** **Functions G54 through G59 and G92 are executed simultaneously.**

Example		
	N10 G00 X100 Y7.5	
	N20 G92 X0 Y100	**X-axis is at 100 and is set to zero.** **Y-axis is at 7.5 and is set to 100.**

3.39 G93, G94, G95 Evaluation of F-Word

G93	Feed rate/ as a % of rapid traverse
Format	G93 F
Explanation	**The feed rates programmed with the F-Word are calculated as a % of the rapid traverse.**
Notes	**G93 is effective in all interpolation types.** **In G25/G26 (OCI) calculating the rapid-feed velocity will be accurate only in conjunction with the new G62 function (Q38, bit 4 = 1) (from V.06.26/0).** **N10 G1 G93 F50 has the same function as N10 G0 FPTP50 in the coordinate transformation.** **G93/G94/G95 mutually deselect each other alternately. G94 is the default setting.** **Example see G95**

G94	Feed rate/path feed rate in mm/min
Format	G94 F
Explanation	**The feed rates/path feed rates programmed with the F-Word are calculated in mm/min.**
Notes	**G93/G94/G95 mutually deselect each other alternately. G94 is the default setting.** **Example see G95**

G95	Feed rate in mm per revolution of main spindle
Format	G95 F
Explanation	**The path feed rate programmed with the F-Word is interpreted as mm/revolution of the main spindle. The resulting path feed rate in mm/min is the product of speed (S) and feed rate (F).**
Notes	**This feed rate evaluation mode requires a spindle with an actual value system.** **G93/G94/G95 mutually deselect each other alternately. G94 is the default setting.**

Example		
	N10 G1 X10 F500 M3 S1000	**(X-axis moves at 500 mm/min, spindle speed 1000 r.p.m.)**
	N20 G95 X30 F1.5	**(G95 X-axis moves at 1.5 mm per spindle revolution. Resulting path feed rate is 1500 mm/min)**
	N30 G94 X40 F500	**(G94 feed rate in mm/min again)**
	N50 M5 M17	

3.40 G96, G97 Evaluation of S-Word

G96	Constant cutting speed
Format	G96 S
Explanation	**The S-Word is interpreted as the circumferential speed in m/min. The radius associated with the circumference is formed from the actual value of an axis specified with $34. A radius offset can be set for these axes in configuration parameter Q.019. The current radius is given by actual position – tool compensation – Q.019.**
Notes	**In no radius axis is specified, the radius will be taken from parameter Q.019 of the spindle. The cutting speed can also be set in m/sec (Q38 bit 1=1). See also Programming Spindle Speed S.**

G97	Spindle speed in r.p.m.
Format	G97 S
Explanation	**The S-Word is interpreted as a constant speed in r.p.m.**
Notes	**Programming a constant spindle speed can be locked by a setting of Q25 bit 1=1. See also Programming Spindle Speed S.**

3.41 G98, G99 Self-maintaining preparatory functions in subroutines

When a subroutine is called the self-maintaining preparatory functions remain effective. If self-maintaining preparatory functions are programmed in s subroutine you can use G98 and G99 to decide whether they will remain effective after a return, or whether the previously valid preparatory functions will be restored.

G98	**Use self-maintaining preparatory functions programmed in the subroutine after return to main program**
Format	G98
Notes	**The preparatory conditions activated in the main program will not be restored after return from subroutine. For the sake of clarity G98 should only be programmed in main programs.** **G98 is the default setting.**

G99	**Do not use self-maintaining preparatory functions programmed in the subroutine after return to main program**
Format	G99
Notes	**The preparatory conditions activated in the main program will be restored after return from subroutine. For the sake of clarity G99 should only be programmed in main programs.**

Example		
	N10 G0 X100 G99	
	N20 X200	
	N30 B%9000	**Subroutine number**
	N40 X220	**Axis moves with G0 and G90**
	N50 M30	

	#9000	
	N10 G1 G91 X10 F100	**Self-maintaining preparatory functions G1/G91 are not effective after return to main program.**
	N20 M17	

4 $ Functions

The $ functions are additional preparatory functions for expanding the standard preparatory functions.

The additional path functions are arranged in function groups.

Only one function from each group can be active.

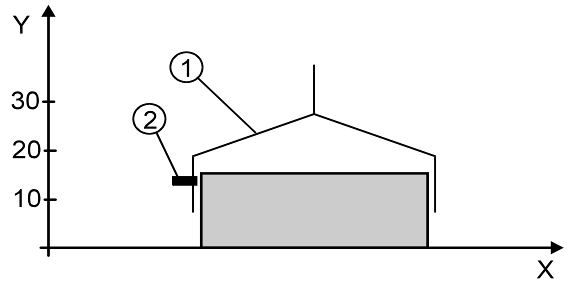
Normally the functions remain active until they are deselected by another function from the same group.

In individual cases a function is active for only one record (property = S). Some functions are default settings (property = D).

Group	Properties D = Default setting S = Active for 1 record		Meaning
1	**S**	**$1**	**Stop axis motion**
	S	**$53 - $54**	**Abort motion**
2	**S**	**$20**	**Handwheel enable for velocity superposition**
	S	**$21**	**Handwheel enable for path superposition**
3		**$23**	**Switch on internal tracing operation**
		$24	**Switch on tracing operation**
		$25	**Switch off tracing operation**
4		**$26**	**Independent. Switch on axis with individual feed rate**
		$27	**Independent. Switch off axis with individual feed rate**
		$28	**Independent. Incorporate axis in record change**
		$29	**Independent. Do not incorporate axis**
5		**$31**	**Switch on synchronous operation**
		$32	**Switch off synchronous operation**
6		**$33**	**Lead axis for thread cutting**
7		**$34**	**Radius axis for v = constant**
8		**$37**	**Path length calculation**
		$38	**Switch on contouring axis in IPO context**
		$39	**Switch off contouring axis in IPO context**
9		**$40**	**Switch oscillation off**
		$41	**Switch on oscillation with continuous infeed**
		$42	**Switch on oscillation with infeed on both sides**
		$43	**Switch on oscillation with infeed right**
		$44	**Switch on oscillation with infeed left**
10		**$47**	**Alternative machining plane**
11		**$48**	**Give back system axis**
12		**$65 / $66**	**Joint configuration for coordinate transformation**
13		**$70**	**Spline interpolation deselection**
		$71	**Spline interpolation selection**
14		**$90**	**Absolute measurements**
		91	**Incremental measurements**

4.1 $1 Stop axis motion without ramp

$1	**Stop axis motion without ramp**
Format	$1 X F = feed rate, X = arbitrary axis letter
Explanation	**Axis motion is interrupted when the PLC signal provided for this purpose is active.** **Assignment to PLC signal: *cncMem.axSect[n].FlgP2N.bAxStop* (n=axis number)**
Notes	**The motion is aborted immediately in the interpolation cycle, without velocity ramp.** **The following error will be corrected.** **If the axis interpolates with other axes interpolation will be aborted on all axes. The NC axis assigned to the input signal must be programmed directly after the path condition ($1 X...).** **The coordinate value of the NC axis indicates the maximum permissible travel. If the PLC signal is not active during the programmed path record change will occur when the programmed coordinate value is reached.** **An implicit G39 is executed in this function.** **For stopping an axis motion with an interrupt see $53 and $54.**

Example		
	N100 G0 Y20	
	N110 G1 $1 Y5 F500	
	N120 E1 = 0 BN ...	**(no part)**
	N130 SE ...	**(close gripper)**
	N140 G0 Y100	
	 **1 Gripper, 2 Range finder**	

4.2 $20 Handwheel enable for velocity superposition

$20	Handwheel enable for velocity superposition
Format	$20 X X = arbitrary axis letter
Explanation	**You can alter the velocity of the specified axes with a handwheel.** **The superposition is added to the programmed velocity.**
Notes	**Handwheel pulses can be saved in shared RAM variable *cncMem.axSect[n].wrdP2N.lValHdWhl*.** **The evaluation of the pulses can be saved in axSect[n].wrdP2N.fRateHdWhl.**

4.3 $21 Handwheel enable for path superposition

$21	Handwheel enable for path superposition
Format	$21 X X = arbitrary axis letter
Explanation	**You can alter the programmed end position of the specified axes with a handwheel.** **The superposition is added to the end coordinate and to the programmed velocity.**
Notes	**Handwheel pulses can be saved in shared RAM variable *cncMem.axSect[n].wrdP2N.lValHdWhl*.** **The evaluation of the pulses can be saved in cncMem.axSect[n].wrdP2N.fRateHdWhl.**

4.4 $23 Internal tracing operation on

$23	Internal tracing operation on
Format	$23 X X = arbitrary axis letter
Explanation	**The specified axes go into internal tracing operation.** **No set position for this axis is to be programmed as long as $23 is active.** **The actual position of the axis is traced. The given axes can then be externally moved (without control, e.g. with drive).**
Notes	**$23 is deselected with $25.** **As long as this function is active, the drive must not take any set points from the controller. Otherwise there will be feedback and the axis will drift away. Thus the user must ensure that the corresponding mode of operation is activated in the drive.**

4.5 $24 Tracing operation on

$24	Tracing operation on
Format	$24 X X = arbitrary axis letter
Explanation	**The specified axis goes into internal tracing operation.** **Tracing operation is used to temporarily interrupt position control, controlled by the program. This is always necessary when the axis is mechanically jammed or displaced by external factors, for example on an injection moulding machine by the discharger when parts are removed.**
Notes	**The position control circuit is opened, the "controller enable" relay drops, all increments in the actual-value system are recorded and transferred to the set position.** **A set position for this axis must not be programmed as long as $24 is active.**

Example	**See $25**

4.6 $25 Switch off tracing operation

$25	Switch off tracing operation
Format	$25 X X = arbitrary axis letter
Explanation	**Cancel programmed tracing operation for one axis.**
Notes	**An implicit G39 is executed in this function.**

Example	**X-axis is displaced by an external ejector and must therefore be taken out of position control.**	
	N100 G0 X100	**Gripper to part**
	N110 $24 X	**Tracing operation on**
	N120 SE1	**Move discharger forward (request to PLC)**
	N130 WN1	**Discharger retracted (acknowledgment from PLC)**
	N140 G0 $25 X50	**Tracing operation off, gripper is back with part**

4.7 $26 Exclude axes from interpolation context

$26	Exclude axes from interpolation context
Format	$26 X FX FX = feed rate, X = arbitrary axis letter
Explanation	**With this $ function you can exclude individual axes from the interpolation context and from the record change criterium ($29). They become "independent" axes.**
Notes	**The selected axes move at the axis-specific feed rate Fx in mm/min, regardless of the path feed rate.** **Independent rotary axes move at feed rate F"axis name" in °/min.** **Record change occurs when all axes in the NC subsystem meet the valid exact position condition.** **Independent axes do not normally reach their programmed end position at the same time.** **Record change without loss of velocity with G64 should not be used for independent axes.**

Example	**X and Y interpolate on a straight line, Z is independent.**
	N10 G1 X100 Y100 F500 $26 Z500 FZ1000
	See also $29.

Warning!

If $26 and $29 are used together and the G condition for feed rate changes in one of the following records, e.g. from G1 to G0, while the independent axis is still moving, G0 applies to all axes in the system. The feed rate of the independent axis will switch to the axis-specific G0 rapid-feed velocity (risk of collision).

Where necessary, limit the speed of the independent axis with OVR.

4.8 $27 Include independent axes in interpolation context

$27	Include independent axes in interpolation context
Format	$27 X X = arbitrary axis letter
Explanation	**The independent axis is reintegrated in interpolation and record change. $27 cancels function $26.**
Notes	**When $29 is active $28 must be set in the previous NC-Record.**

Example	**See $29**

4.9 $28 Include independent axis in record change

$28	Reintegrate independent axis in record change
Format	$28 X X = arbitrary axis letter
Explanation	**The independent axis is reintegrated in the record change, but not in the interpolation context.**
Notes	**Record change occurs when all interpolating axes and the independent axes meet their exact position condition. This word is the default setting, or cancels function of $29.**

Example	**See $29**

4.10 $29 Do not include independent axis in record change

Warning!

If $26 and $29 are used together and the G condition for feed rate changes in one of the following records, e.g. from G1 to G0, while the independent axis is still moving, G0 applies to all axes in the system. The feed rate of the independent axis will switch to the axis-specific G0 rapid-feed velocity (risk of collision).

Where necessary, limit the speed of the independent axis with OVR.

Note

Function $29 can only be called after $26. Cancelling with $28 must occur at least one record before $27.

$29	Do not include independent axis in record change
Format	$29 X X = arbitrary axis letter
Explanation	
Notes	**Record change occurs when the interpolating axes meet their exact position condition, regardless of the position of the independent axis.** **If the independent axis has not yet reached its end coordinates the motion of the independent axis will continue in the next record.** **If program ends with M17 or M30 the independent axis will stop too, regardless of its position.**

Example on the next page.

Example		
	N100 G1 X100 Y100 Z100 F1000	
	N110 G1 X120 Y80 F500 $26 Z200 FZ50 $29 Z	The Z-axis is excluded from interpolation ($26) and record change condition ($29).
	N120 X140 Y60	Z-axis moves independently of the axes programmed here.
	N130 X160 Y40	
	N140 $28 Z	Reintegrate Z-axis in record change ($28). Record change occurs when Z-axis has reached target position.
	N150 G1 X100 Y100 $27 Z100 F1000	Reintegrate Z-axis in interpolation from this record on ($27).
	N160 X50 Y120 Z90	Path interpolation restarts for the Z-axis.

4.11 $31 Switch on synchronous operation

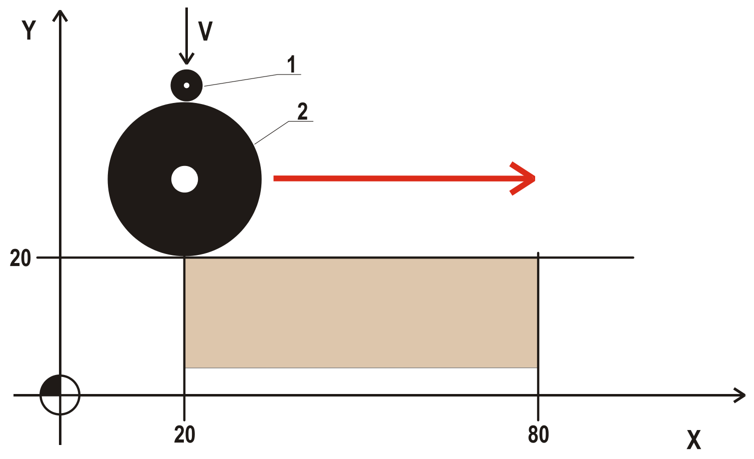
$31	Switch on synchronous operation
Format	$31 X Y X, Y = arbitrary axis letter
Explanation	**Synchronous operation allows synchronous operation of several axes to be programmed for a time.**
Notes	**The first axis named after $31 is the lead axis and all subsequently named axes are the following axes. The motion of the first named axis is also effective for the subsequent axes. If a motion is programmed for the following axes, the motion of the lead axis will be added to this.** **There is no check on whether the following axes can reach the required velocity.** **The distance moved by the lead axis does not appear in the actual value display of the following axes. Instead it is saved as an internal zero point offset. The displayed actual value of a following axis contains only the actual position of the distance programmed in this axis.** **The function is deselected with $32, M30 or Reset. M30 or Reset deletes the content of zero point offset, the absolute actual value is displayed.** **If bit 1=1 is set in Q37, $32 will delete the content of internal zero point offset (actual value display changes). In this case G39 is executed automatically.**

Example	See $32

4.12 $32 Switch off synchronous operation

$32	Switch off synchronous operation
Format	$32
Explanation	**This function cancels $31.**
Notes	**If bit 1=1 is set in Q37, $32 will delete the content of internal zero point offset (actual value display changes). In this case the controller executes an automatic G39.**

Example	**Continuous dressing of a grinding wheel**	
	N100 G0 X20 Y10	**Position grinding wheel**
	N110 VR1000	**Position dressing roll**
	N120 $31 VY	**V superposes Y**
	N130 $26 VR1001 FR102 $29 V	**Switch on dressing, V = independent axis**
	N140 X70 F100	**Grinding**
	...	
	N210 G91 $32 V1	**Lift off dressing roll**
	N220 G90 $28 V	**V-axis in position**
	N230 $27 V	**V-axis in interpolation**



1 Dressing roll 2 Grinding wheel

4.13 $33 Select lead axis for thread cutting

$33	
Format	$33 Z Z = arbitrary axis letter
Explanation	**Specify lead axis for thread cutting / tapping with G33, G63.**
Notes	**$33 is self-maintaining and only has to be programmed again if the lead axis changes.**
Example	**See G33, G63.**

4.14 $34 Select radius axis

$34	Radius axis selection for G96
Format	$34 X X = arbitrary axis letter
Explanation	**The actual position of the selected axis enters the main spindle circumferential velocity calculation as the radius.**
Notes	**An additional offset can be entered in Q.019 of the axis selected with $34. The sign of Q.019 (OFFSET RADIUS) is taken into consideration in calculation of the constant cutting speed. The speed of the spindle is limited to the set maximum spindle speed.** **If no radius axis is selected with $34 Q.019 of the spindle axis will be interpreted as the radius. If Q.019 = 0 no speed will be output. Q.019 can be altered at any time by the PLC program or, if enabled, by the CNC program.** **The radius determined for calculation is formed from:** **Radius = actual position – tool compensation – Q.019.**

4.15 $35 Select tangential tracing axis

$35	Select tangential tracing axis
Format	$35 A X = arbitrary axis letter
Explanation	**Rotary axis, whose angular position is to be maintained constant. This function is effective together with G5, G6, G7 and G8 Tangential tracing for circle and straight line page 37.**
Notes	**The coordinate value is interpreted as the angle of attack. This angle must always be specified in degrees (0-359.999).**

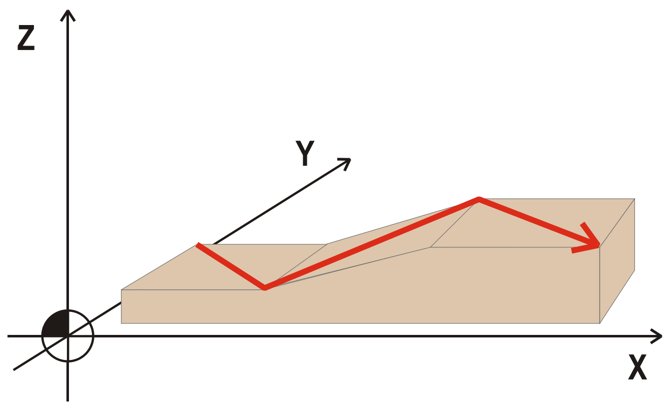
4.16 $37 Variant for path-length calculation

$37	
Format	$37
Explanation	**$37 (alternatively to $38, $39 and combinable) is used to calculate the path-length according to the lead-axis principle. The programmed feed rate refers to the axis with the largest path length (* 1000 / Q1079). By default ($39) the path length of the root mean square of the axes involved.**
Notes	**also see $38 and $39**

4.17 $38 and $39 Axis selection for path-feed rate calculation

$38	Exclude axes from path feed rate calculation
$39	Include axes in path feed rate calculation
Format	$38 X $39 X X = arbitrary axis letter
Explanation	**With these functions you can exclude individual axes from the path feed rate calculation.**
Notes	**The affected axes are carried along in the interpolation context. There is no check on whether the following axes can reach the required velocity.** **This function has no effect on axes which are involved in circular interpolation with G2/G3 or G12/G13.** **In the case of helix interpolation the third axis can be excluded from the path feed rate calculation with $38. Then the path feed rate is effective not on the spatial path but on the flat circular path (projection).**

Example		
	N100 G0 X10 Y40 Z10	**Start position**
	N110 G1 X40 Y10 F100	
	N120 X60 Y40 $38 Z25	**Exclude axis Z**
	N130 X100 Y10	
	N150 G0 $39 Z30 X0 Y0	**Include axis Z**



4.18 $40 Switch oscillation off

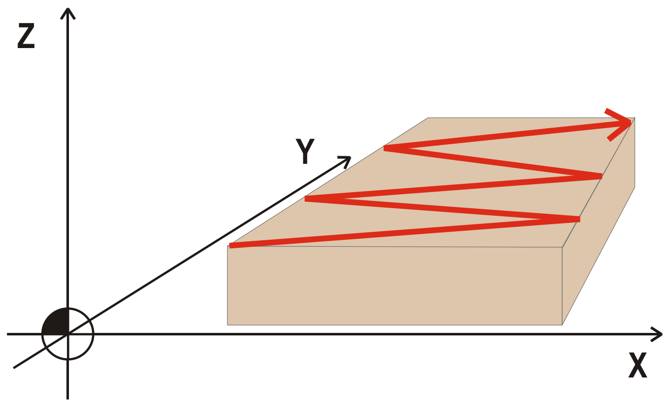
$40	**Switch oscillation off with spark-out passes**	
Format	$40 X $40 Xn **Xn = axis letter with specified number of spark-out passes**	
Explanation	**Oscillation switched off, the number of spark-out passes for an oscillating axis can be specified.**	
Notes	**When the end coordinate is reached the axis moves to the next reversal point. From this point the specified number of sparking out strokes is executed. One sparking out stroke is the distance between reversal points.** **With $40 the controller inserts an automatic G39.**	

Example		
	N10 $40 X5	**Axis X executes 5 sparking out strokes**

4.19 $41 Oscillation with continuous infeed

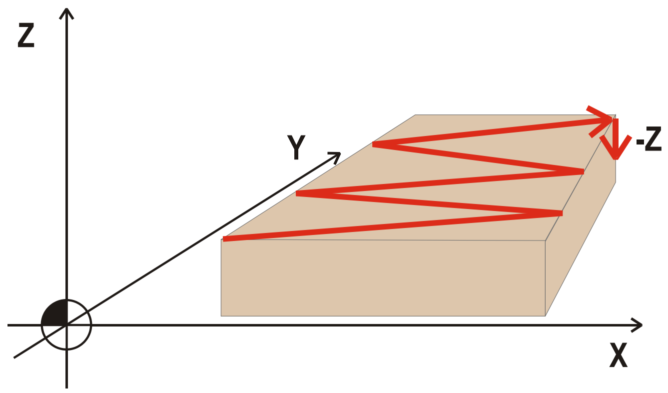
$41	Oscillation with continuous infeed on one axis
Format	$41 X Y X, Y = arbitrary axis letter
Explanation	**The oscillating axis oscillates between the start position (the axis position when the oscillation function is called) and the coordinate programmed in the oscillation record.** **Infeed is continuous, at path feed rate.**
Notes	**The oscillating axis is always the first axis programmed after the $-Word.** **The feed rate of the oscillating axis is programmed with F"axis name".** **The feed axis is the second programmed axis. The target coordinate is programmed.** **When the oscillate function is selected the oscillating axis automatically becomes an independent axis (corresponding to $26). When the oscillate function is deselected it returns to the interpolation context.** **The reversing behaviour of the oscillating axis can be controlled with G67.**

Example		
	N100 G0 X0 Y100	
	N110 G1 $41 X200 Y95 FX1000 FY5	**Oscillation on**
	N120 $40 X3	**Oscillation off with 3 sparking out strokes**
	N130 G0 Y150	



$41	Oscillation with continuous infeed on one axis and two feed axes
Format	$41 X Y Z R1001 R2 X, Y, Z = arbitrary axis letters R1001, R2 = arithmetic parameter addresses
Explanation	**A second feed axis is added to oscillating with one feed axis. This makes it possible to machine surfaces that are broader than the width of the grinding wheel.**
Notes	**The oscillating motion in X is as before. In Y there is continuous infeed until the programmed coordinate is reached. Now Z is fed by the content of R1001 in relation to R2. Y reverses and moves to the opposite reversal point (etc.).** **Record change occurs when Z has reached the final dimension and the valid exact position condition is met.**

Example	
	N100 G0 Y0 Z100 R1001:= 0.5 R2:= 1
	N110 G1 $41 X200 Y50 Z95 R1001 R2 FX1000 FY10 FZ500
	N120 $40 X3 Y0
	N130 G0 Z150

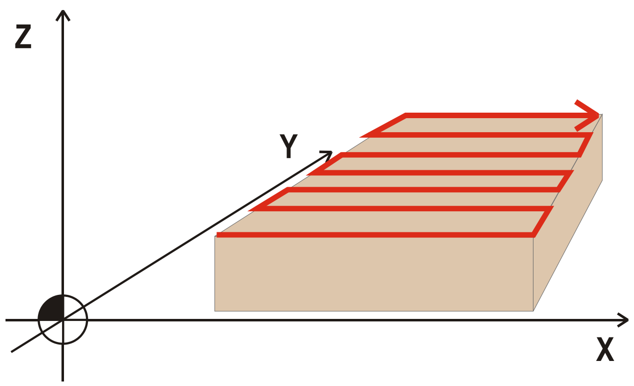


Record N110:	G1	Straight interpolation
	$41	Select oscillate function with continuous infeed
	X	Is the oscillating axis. The programmed position is the 1st reversal point. The 2nd reversal point is derived from the position of X at the start of oscillation.
	Y	is the 1st infeed axis. The programmed coordinate is the reversal point. Infeed in Y is continuous.
	Z	is the 2nd infeed axis. The programmed coordinate is the final dimension. Infeed in Z occurs at the reversal point of Y, dependant on R2.
	R1001	contains the infeed increment for Z
	R2	**controls infeed in Z:** **R2 =0 Infeed at front (smaller) reversal point of Y. R2 =1 Infeed at both reversal points of Y. R2 =2 Infeed at rear (greater) reversal point of Y.**
	FX	is the feed rate of the oscillating axis
	FY	is the feed rate of the 1st infeed axis.
	FZ	is the feed rate of the 2nd infeed axis.

4.20 $42 Oscillation with infeed at both reversal points

$42	Oscillating with infeed on one axis at both reversal points
Format	$42 X Y R X, Y = arbitrary axis letter R = arithmetic parameter
Explanation	**As $41 but with infeed at the corresponding reversal points. The respective feed increment is programmed in the arithmetic parameter.**
Notes	**A reversal dwell time can be programmed in the oscillation record with TI. The reversal dwell time starts as soon as infeed has occurred. The oscillating axis remains at the reversal point until the dwell time has expired.** **The reversing behaviour of the oscillating axis can be controlled with G67.**

Example	
	N100 G0 X0 Y100 R1001 := 0,5
	N110 G1 $42 X200 Y95 R1001 FX1000 FY500
	N120 $40 X3
	N130 G0 Y150

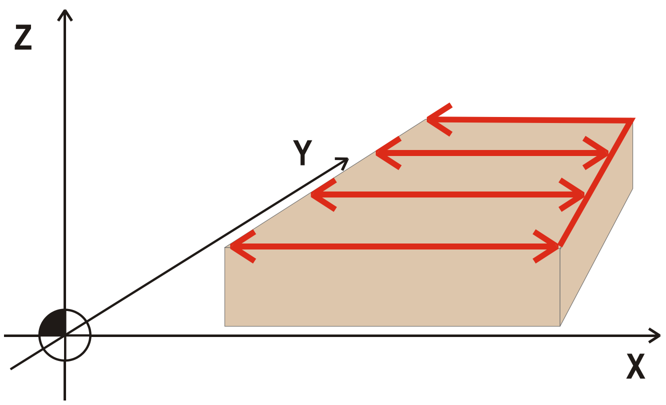


$42	Oscillating with infeed at both reversal points and a second feed axis	
Format	$42 X Y Z R1001, R1002, R3 X, Y, Z = arbitrary axis letters R1001, R1002, R3 = arithmetic parameter addresses	
Explanation	**A second feed axis is added to oscillating with one feed axis.**	

Example		
	N100 G0 X0 Y0 Z100 R1001:= 10 R1002:= 0.5 R3:= 1	
	N110 G1 $42 X200 Y50 Z95 R1001 R1002 R3 FX1000 FY500 FZ500	
	N120 $40 X3 Y0	
	N130 G0 Z150	
Record N110:	**G1**	**Straight interpolation**
	$41	**Select oscillate function with continuous infeed**
	X	**Is the oscillating axis. The programmed position is the 1st reversal point. The 2nd reversal point is derived from the position of X at the start of oscillation.**
	Y	**is the 1st infeed axis. The programmed coordinate is the reversal point. Infeed in Y is continuous.**
	Z	**is the 2nd infeed axis. The programmed coordinate is the final dimension. Infeed in Z occurs at the reversal point of Y, dependant on R2.**
	R1001	**Contains the infeed increment for Z**
	R1002	**Contains the infeed increment for Z**
	R3	**controls infeed in Z: R3 =0 Infeed at front (smaller) reversal point of Y. R3 =1 Infeed at both reversal points of Y. R3 =2 Infeed at rear (greater) reversal point of Y.**
	FX	**is the feed rate of the oscillating axis**
	FY	**is the feed rate of the 1st infeed axis.**
	FZ	**is the feed rate of the 2nd infeed axis.**

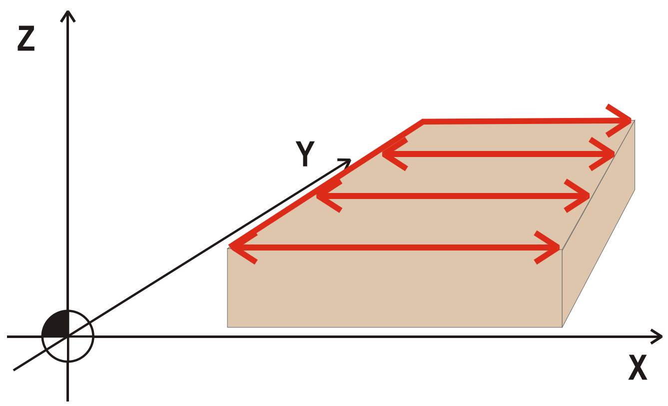
4.21 $43 Oscillation with infeed only at right reversal point

$43	Oscillation with infeed only at right reversal point
Format	$43 X Y R X, Y = arbitrary axis letter R = arithmetic parameter
Explanation	**Function and example as $42.**
Notes	**As in $42, a second feed axis can be used.** **A reversal dwell time can be programmed in the oscillation record with TI.** **The reversal dwell time starts as soon as infeed has occurred.** **The oscillating axis remains at the reversal point until the dwell time has expired.**

Example	
	

4.22 $44 Oscillating with infeed only at left reversal point

$44	Oscillation with infeed only at left reversal point
Format	$44 X Y R X, Y = arbitrary axis letter R = arithmetic parameter
Explanation	**Function and example as $42.**
Notes	**As in $42, a second feed axis can be used.** **A reversal dwell time can be programmed in the oscillation record with TI.** **The reversal dwell time starts as soon as infeed has occurred. The oscillating axis remains at the reversal point until the dwell time has expired.** **The reversing behaviour of the oscillating axis can be controlled with G67.**

Example	
	

4.23 $47 Define machining plane

$47	Define machining plane
Format	$47 U V U, V = arbitrary axis letters
Explanation	**The machining planes are defined by two axes each, which are specified in configuration parameter Q.054 as axes parallel to X, Y or Z.** **If several axes in a subsystem are defined parallel to the same spatial coordinate, the axes for the current machining plane are selected with $47.**
Notes	**The machining plane is activated with G17, G18 or G19.** **$47 selects the axes that define the plane, if this is not clear from parameter Q.054.**

4.24 $48 Enable axis for subsystem change

$48	**Enable axis**
Format	X $ 48 X = Arbitrary axis letter
Explanation	**If the controller is configured for controlling several subsystems, each axis must be assigned to one of the subsystems. Then an NC-Program can be started for each subsystem and these NC-Programs can run in parallel.** **It may therefore be necessary to program axes in several systems (example: multi-spindle lathes). In this case axes can be "borrowed" from a subsystem by assigning an arbitrary letter to the axis number.** **See: Lending NC Axes Between NC Subsystems Page 131.** **From SW version OS 08:05/0** **To lend axes to another subsystem, $48 "$_{\text{Axis letter}}$" must be used to release the axis that is to be lent ; only then can the axis be assigned to another subsystem with *`Axis letter:Axis .`*** **From SW version OS 10.43/0** **Q45 Bit2=1: For old NC-Programs without $48 monitoring.** **A release is not necessary for the axes; to ensure a safe program execution, the programs should be synchronized with words M90 - M98.** **With this function the borrowing system gives the "borrowed" axis back to the original system.**
Notes	**Axes are borrowed from other systems by programming X = axis number (X = arbitrary axis letter not used for another axis).** **To ensure a reliable working order, M-Functions M90 through M98 must be programmed in the subsystems (see page 108).**

In the example, 2 axes of subsystem 1 are released and assigned to subsystem 2 and released again.

Example		
System 1	%1	new syntax form
	N10 G1 X0 Y0 F500	
	N20 $48 XY	Sign off X and Y from system 1
	N30 M92	synchronize with 2nd system N10
	N40 – N60	Further processing
	N70 M92	synchronize with 2nd system N50
	N80 X:=1 Y:=2	Sign on X and Y in system 1
	N90 X-100 Y-100	
	N1000 M17	
System 2	%2	new syntax form
	N10 M91	synchronize with 1st system N30
	N20 X:=1 Y:=2	Sign on X and Y in system 2
	N30 G1 X20 Y20 F10	
	N40 $48 XY	Sign off X and Y from system 2
	N50 M91	synchronize with 1st system N70
	N1000 M17	

4.25 $53 - $54 Abort motion

$53	Abort motion with following error compensation
Format	$53 X X = arbitrary axis letter
Explanation	**Stop axis motion through interrupt signal**
Notes	**With active interrupt signal the axis motion is aborted immediately and record change is carried out.** **The current position of the axes during record change corresponds to that at the time of the interrupt.** **The function is effective record by record.** **See also $54.**

$54	Delete remaining distance through interrupt signal
Format	$54X I X = arbitrary axis letter, I = remaining distance data
Notes	**The axis position of the corresponding axis at the time of receipt of the interrupt signal is saved, the associated path position is determined. The distance programmed in "I" is travelled from this path position. The resulting difference between the programmed record end position and the actual record end position is saved in an internal zero point offset. This zero point offset remains until the next G39 (including implicit G39 e.g. $1 , E1 = 1). Record change can occur with G64. That also means at maximum velocity without decelerating. The internal zero point offset can be taken into consideration later in a G39 record.** **I = residual path of interrupt (measurement position) to record end on the path. I must be at least as great as the path that was travelled in three interpolation cycles. If the record change was programmed with G9 the braking distance must also be taken into account.** **The function is effective record by record.** **See also $53.**

Example	
	N100 G1 $54 X400 I100

Note

$54 is only permissible in connection with G1.

4.26 $65, $66 Alternative joint configuration

$65	**Alternative joint configuration deselection**
$66	**Alternative joint configuration selection**
Format	$65 $66
Explanation	**$65 Cross-record spline interpolation deselection.** **$66 Cross-record spline interpolation selection.** **This function requires the coordinate transformation (page 47).**
Notes	**Software option "06 CNC Coordinate Systems" is required for this function. These $-words are described in the following manual: "Coordinate transformation, Article No. R4.322.1390.0 (322 140 05)".**

4.27 $70, $71 Cross-record spline interpolation

$70	**Spline interpolation deselection**
$71	**Spline interpolation selection**
Format	$70 X, $70 X100 $71 X, $71 X100 X = arbitrary axis letter
Explanation	**From SW version OS06.39/0** **$70 Cross-record spline interpolation deselection.** **$71 Cross-record spline interpolation selection.** **This function is only active with Q109 > 3.** **Functions $71 (select) and $70 (deselect) are only permissible in association with straight-line interpolation (G1). This applies to all axes programmed with $71 and $70 in the line (axis letter is sufficient) up to the next $ function.**
Notes	**$71 is used for selection and is self-maintaining. Deselection is made with $70.** **Both $ functions cancel each other.** **The default setting is $70** **The drive movements programmed with $71 are summed up and driven to the NC-Record similar to independent axes ($26 $29) using $70 The drive movement programmed with $70 travel is again driven as a straight-line interpolation. The transition to the straight line occurs without smoothing and without a speed jump.** ***Special case*: If the number of NC-Records programmed with $71 exceeds the value in Q109, half the path is traversed and then the axis is stopped. The rest of the path is traversed before the $70 record.**

4.28 $90, $91 Absolute/Incremental measurements, axis-specific

$90	**Absolute measurements**
91	**Incremental measurements**
Format	$90 X $91 X X = arbitrary axis letter
Explanation	**$90 Absolute measurements for this axis** **$91 Incremental measurements for this axis** **With these $ functions G-Functions G90 / G91 can be superposed for individual axes to mix absolute and incremental dimensions in one record.**
Notes	**$90 and $91 are self-maintaining until program end or until the dimensions are changed by programming G90 or G91 .** **The two $ functions cancel one another, programming G90 or G91 deletes all programmed $90 or 91 functions.** **The default setting is absolute.**

Example		
	N110 X100 Y35 F1050	**All axes with absolute position**
	N120 X120	
	N130 X125 $91 YR256	**X-axis moves on absolute coordinate, Y-axis moves incrementally with content of R256**
	N140 G91 X15 Y5	**All axes move incrementally**
	N150 X10 $90 Y75	**X-axis moves incrementally, Y-axis moves on absolute coordinate**

5 M-Functions

M-Functions can be used to program logic functions

M0	**Programmed stop**
M1	**Optional stop**
M3	**Spindle rotation clockwise (and special case M"axisname"**
M4	**Spindle rotation anticlockwise (and special case M"axisname"**
M5	**Spindle stop (and special case M"axis name")**
M17	**Subroutine end see Program ends with M17 and M30**
M30	**Program end / reset see Program ends with M17 and M30**
M90, M91 through M98	**Synchronization of NC subsystems**
From M1001	**M-Function with time stamp**

Up to max. 3 M-Functions can be programmed in each record. From Version OS10.28/0 onwards, 7 M-Functions can be programmed in each NC-Record. M-Functions which are not predefined can be evaluated at will in the PLC.

M-Functions in a record can lead to delays in record changing, because a PLC user program processes the signal and must enable the next record. This process takes at least two interpolation cycles.

Record change without loss of velocity (G64) is carried out without enable from the PLC user program.

5.1 M0 Programmed stop

M0	**Programmed stop**
Format	M0
Explanation	**Stop after record execution. Program can be continued with CNC start.**

Example	
	N120 G0 X100 M0

5.2 M1 Optional stop

M1	**Optional stop**
Format	M1
Explanation	**Stop after record execution if function activated from PLC program. Program can be continued with CNC start.**

Example	
	N120 G0 X100 M1

5.3 M3 and M4 Clockwise / Anticlockwise spindle rotation

M3	Clockwise spindle rotation
Format	M3
Explanation	**Starts an NC axis declared as main spindle or a PLC-controlled spindle**
Notes	M"axisname" 3 **If one or more axes in an NC subsystem are declared as rotary axes they can be operated as controlled spindles with M"axisname" 3. The speed is then programmed with S"axis name".**

M4	Anticlockwise spindle rotation
Format	M4
Explanation	**Starts an NC axis declared as main spindle or a PLC-controlled spindle**
Notes	M"axisname" 4 **If one or more axes in an NC subsystem are declared as rotary axes they can be operated as controlled spindles with M"axisname" 4. The speed is then programmed with S"axis name".**

5.4 M5 Spindle stop

M5	Spindle stop
Format	M5
Explanation	**Stops an NC axis declared as main spindle or a PLC-controlled spindle**
Notes	**Switching a controlled spindle from spindle mode to rotary axis mode is programmed with M5, and in the next record G39 for synchronizing up the actual position.** **Alternatively to M5 and G39 a controlled spindle can also be stopped at a target position by programming the target position after the axis letter of the spindle.** M"axisname" 5 **If one or more axes in an NC subsystem are declared as rotary axes they can be operated as controlled spindles with M"axisname" 5.**

Example		
	N10 M3 S500	**Spindle start**
	N20 X... Y...	
	N30 M5	**Spindle stop**
	N40 G39	**Synchronize up actual position**
	N50 C45 F300	**Position spindle as C-axis**

Example		
	N10 M3 S500	**Spindle start**
	N20 X... Y...	
	N30 C45	**Positioned spindle stop**

Note

On M3, M4 and M4

- **A main spindle is declared with Q.054 bit 3=1.**
- **Spindle operation is monitored by a PLC program. If there is no PLC program for operating the spindle, then the NC-Program stops.**

5.5 M17 Subroutine end

M17	Subroutine end
Format	M17
Explanation	**M17 causes a jump back to the calling NC-Program. If there is no calling program M17 has the same effect as M30.** **Not required if Q25 bit 5=1.**

5.6 M30 Program end

M30	Program end
Format	M30
Explanation	**Ends the NC-Program. Controller switches to RESET operating state.** **Not required if Q25 bit 5=1.**

5.7 M90 through M98 Synchronization of NC subsystems

M90	**Synchronization of all subsystems**
M91 through M98	**Synchronization with subsystem 1 through 8**
Format	M90 M95
Explanation	**These functions are required for controllers with several subsystems. In this case several NC-Programs (one per subsystem) can run at the same time.** **It is often necessary to divide these NC-Programs into parts that can be executed in parallel and parts which must be executed sequentially.** **Functions M90 through M98 are synchronization markers for controlling the execution of NC-Programs. Execution of the NC-Program stops at a synchronization label until the NC-Program of the corresponding subsystem has also reached a label. Then execution continues in all involved subsystems.**

Note

When using these functions to work in two subsystems, suitable measures must be taken in the PLC program to ensure both systems are always started in RECORD SEQUENCE. Thus, when a system is stopped, a suitable point must be found in which the other system is also stopped and then restarted in RECORD SEQUENCE. Operation in SINGLE RECORD or BLOCK RECORD is generally not allowed.

The following example shows synchronization of two subsystems with functions M91 and M92. If the controller is configured for exactly two subsystems all labels can be replaced with M90.

Example	**Subsystem 1**	**Subsystem 2**
	N10..... (#1 and #2 parallel)	
	N20.....	**N10.....**
	N30 M92 (sync with #2)	**N20 M91 (sync with #1)**
	N40.....	
	N50..... (only #1 active)	**(#2 waits at N30)**
	N60.....	
	N70 M92 (sync with #2)	**N30 M91 (sync with #1)**
	N80..... (parallel again)	**N40.....**
	N90.....	**N50.....**

Note

Inappropriate use of synchronization markers can lead to deadlock situations (jamming) in NC-Program processing.

5.8 M1001 M-Function with time stamp

From M1001	M-Function with time stamp
Format	M1001
Explanation	**From version OS10.03/1** **M words > 1000 = M-Function with a time stamp, a PLC program is required for executing (e.g. laser control).**

Example	
	N120 G0 X100 M1001

6 Interface CNC - PLC

	Bit variables
Format	Ennn, SEnnn, RSnnn, WAnnn, WNnnn nnn = number of bit variables, Global bit variable: 3-digit decimal number in range 0 through 255 From SW version OS05.49/0 System-specific bit variable: 3-digit decimal number in range 256 through 511
Note	**The global bit variables operate cross-system on all sub-systems. They can be used for controlling NC-Programs in several subsystems using central bit instructions.** **The system-specific bit variables operate in the subsystem in which the NC-Program is executed.** **In the PLC, the bit of a bit variable can be processed directly. Bit variables can be used in the NC-Program even without PLC processing.** **Access by the PLC to global bit variable is:** ***cncMem.comSect.abFlgPNRw[n]*** **(n=number of the bit variable 0-255)** **Access by the PLC to system-specific bit variable is:** ***cncMem.sysSect[n].abFlgPNRw[ii]*** **(n=number of the subsystem, ii=number of the bit variable 256-511)**

E	**Request a bit variable**
SE	**Set a bit variable at the start of record execution**
RS	**Reset a bit variable at the start of record execution**
WA	**Wait for bit variable = 1**
WN	**Wait for bit variable = 0**

6.1 E Request a bit variable

E	Request a bit variable
Format	Ennn = 1 Ennn = 0 nnn = number of the global bit variable, 3-digit decimal number in the range 0 – 255, the system-specific bit variable in the range 256 – 511.
Note	**Bit variables are executed at the time of record change from the preceding NC-Record. The controller executes an automatic G39.**

Example		
	N10 E0=1 B%9000	**(If bit variable 0=1, the system branches to subroutine %9000.)**
	N20 G90 G61 X100	**(Return from subroutine %9000 or bit variable 0 = 0 in N10)**

6.2 SE Setting a bit variable

SE	Setting a bit variable
Format	SEnnn nnn = number of the global bit variable, 3-digit decimal number in the range 0 – 255, the system-specific bit variable in the range 256 – 511.
Note	**The bit variable is set at the beginning of the record execution**

Example	
	N10 SE0

6.3 RS Resetting a bit variable

RS	Resetting a bit variable
Format	RSnnn nnn = number of the global bit variable, 3-digit decimal number in the range 0 – 255, the system-specific bit variable in the range 256 – 511.
Note	**The bit variable is reset at the beginning of the record execution**

Example	
	N10 RS0

6.4 WA and WN Wait for bit variable

WA	Wait for bit variable = 1
Format	WAnnn nnn = number of the global bit variable, 3-digit decimal number in the range 0 – 255, the system-specific bit variable in the range 256 – 511.
Explanation	**Record change to next record only if bit signal = 1.** **Bit variable checked at end of any axis motion.**

WN	Wait for bit variable = 0
Format	WNnnn nnn = number of the global bit variable, 3-digit decimal number in the range 0 – 255, the system-specific bit variable in the range 256 – 511.
Explanation	**Record change to next record only if bit signal = 0.**

Example		
	N10 G0 X100 WN0	**Motion executed regardless of instruction WN0. WN instruction evaluation and possible record change not until position X = 100**

7 Arithmetic parameters (R-Parameters)

General R-Parameters R2000 through R5999 (integer values)
General R-Parameters R6000 through R9999 (real values)
General R-Parameters (Retain) R31000 through R31499 (integer values)
General R-Parameters (Retain) R31500 through R31599 (real values)
System-specific R-Parameters R000 through R999 (integer values)
System-specific R-Parameters R1000 through R1999 (real values)
System-specific R-Parameters (Retain) R30000 through R30499 (int. values)
System-specific R-Parameters (Retain) R30500 through R30999 (real values)
Zero point offsets R10001 through R10564
Zero point overlays R10601 through R10664
R10701 through R10764 Reference point offset
Tool data memory R20000 through R29829

All parameters are in the shared RAM and can be read and written by the CNC and PLC.

Function assignment is defined by the NC-Program.

Real values can be programmed and entered in decimal form with up to 7 decimal places plus the sign.

Integer values are positive or negative whole numbers.

The active system of units (G70 and G71) is taken into consideration when substituting coordinate values and velocities.

With XCx: The content of the Retain-R-Parameters is stored in battery-backed RAM of the XCx.

With ProNumeric: The contents of the Retain-R-Parameters must be written into a file by the PLC program.

Important!

ProNumeric: The Retain-R-Parameters must be managed from the PLC. With the ProNumeric, the contents of the Retain-R-Parameters can only be saved in a file by a PLC program.

7.1 General R-Parameters R2000 through R5999 (integer values)

Number	Type
R2000 through R5999	Global R-Parameters, that are identical in all CNC subsystems

7.2 General R-Parameters R6000 through R9999 (real values)

Number	Type
R6000 through R9999	Global R-Parameters, that are identical in all CNC subsystems

7.3 General R-Parameters (Retain) R31000 through R31499 (integer values)

Parameter	Type
R31000 through R31499	Global R-Parameters, that are identical in all CNC subsystems

7.4 General R-Parameters (Retain) R31500 through R31599 (real values)

Parameter	Type
R31500 through R31599	Global R-Parameters, that are identical in all CNC subsystems

7.5 System-specific R-Parameters R000 through R999 (integer values)

Number	Type
R000 through R999	Local R-Parameters, which exist once per CNC subsystem

7.6 System-specific R-Parameters R1000 through R1999 (real values)

Parameter	Type
R1000 through R1999	Local R-Parameters, which exist once per CNC subsystem

7.7 System-specific R-Parameters (Retain) R30000 through R30499 (int. values)

Parameter	Type
R30000 through R30499	Local R-Parameters, which exist once per CNC subsystem

7.8 System-specific R-Parameters (Retain) R30500 through R30999 (real values)

Parameter	Type
R30500 through R30999	Local R-Parameters, which exist once per CNC subsystem

7.9 Zero point offsets R10001 through R10564

6 zero point offsets are available.
The zero point offsets are called with G54 through G59.
Each axis is assigned to a parameter number.

Parameter	
R10001	1st axis zero point offset 1 (G54)
through	
R10064	64th axis zero point offset 1 (G54)
R10101	1st axis zero point offset 2 (G55)
through	
R10164	64th axis zero point offset 2 (G55)
R10201	1st axis zero point offset 3 (G56)
through	
R10264	64th axis zero point offset 3 (G56)
R10301	1st axis zero point offset 4 (G57)
through	
R10364	64th axis zero point offset 4 (G57)
R10401	1st axis zero point offset 5 (G58)
through	
R10464	64th axis zero point offset 5 (G58)
R10501	1st axis zero point offset (G59)
through	
R10564	64th axis zero point offset (G59)

7.10 Zero point overlays R10601 through R10664

In these R-Parameters you can set a permanent zero overlay independent of the program.

A parameter is assigned each axis.

Parameter	
R10601	1st axis zero point overlay
through	
R10664	64th axis zero point overlay

The monitor display is altered according to this data. The internal controller actual value and software limit switch functions are unaffected.

If value = 0 no zero overlay occurs.

Important!

The content of this parameter is effective as zero overlay after homing.

For axes with absolute value encoder:
A value entered in these parameters must not be less than the reference point coordinate (Q.034).

7.11 R10701 through R10764 Reference point offset

The differences between actual value and reference point offset (G92) are entered in these parameters. This means they can be read by the CNC.

A parameter is assigned each axis.

Parameter	
R10701	1st axis zero point offset (G92)
through	
R10764	64th axis zero point offset (G92)

8 Overview Tables

8.1 Overview of G-Words

In this overview the G-Words are organized in groups.
Only one function from each group can be active.
Normally the functions remain active until they are deselected by another function from the same group.

Group	Properties D = Default setting S = Active for 1 record		Meaning
1		**G0**	**Contour control in rapid feed.**
	D	**G1**	**Straight interpolation**
		G2	**Clockwise circle-helix interpolation**
		G3	**Anticlockwise circle-helix interpolation**
		G10	**Point-to-point positioning in rapid feed**
		G11	**Home to reference point**
		G12	**Clockwise spiral interpolation**
		G13	**Anticlockwise spiral interpolation**
		G25	**Online curve interpolation OCI without tangential transition**
		G26	**Online curve interpolation OCI with tangential transition**
		G27	**Freeform interpolation of CNC programs created offline**
		G32	**Tapping with controlled spindle**
		G33	**Thread cutting**
		G63	**Tapping without compensating chuck**
		G76	**Thread cycle**
		G77	**Tapping cycle without compensating chuck**
2	**S**	**G4**	**Dwell time**
3	**D**	**G5**	Deselection of tangential tracing
		G6	Tangential tracing with the transition radius (inner circle)
		G7	Tangential tracing with the transition radius (outer circle)
		G8	Tangential tracing without transition radius
4	**D**	**G17**	**Plane selection X-Y**
		G18	**Plane selection X-Z**
		G19	**Plane selection Y-Z**
5	**D**	**G20**	**Deselection of coordinate transformation**
		G21	**Position specified in Cartesian coordinates**
		G22	**Position specified in Cartesian coordinates**
		G23	**Position specified by the axis positions**
		G24	**Position specified by the axis positions**
6	**D**	**G28**	**Update arithmetic parameters when record is executed**
		G29	**Update arithmetic parameters when record is executed**

Group	Properties D = Default setting S = Active for 1 record		Meaning
7	S	G39	Interrupt record preparation
8	D	G40	Switch off tool-radius compensation
		G41 G42	Tool radius compensation left/right
		G43 G44	Tool radius compensation positive/negative
	S	G50	Tool radius compensation without transition contour
9		G45 G46	Feed rate correction
10		G52	Coordinate rotation
11	D	G53 to G59	Zero point offset
12	S	G9	Exact positioning
	D	G60	Record change after exact stop boundary reached
		G61	Record change after elimination of set-actual deviation
		G62	Record change with acceleration monitoring
		G64	Record change without loss of velocity
		G66	Synchronization of the IPO interpolation points
13	S	G67	Special function for oscillating
14		G70	Units in inches; the last used function applies
		G71	Units in millimetres
15		G72	Coordinate systems: Selection of reference system
		G74	Coordinate systems: Selection of compensation system
16	D	G80 to G89	Machining cycles
17	D	G90	Absolute measurements
		G91	Incremental measurements
18		G92	Reference point offset
19		G93	Specification of feed rate in % of rapid feed
	D	G94	Feed rate in mm/min (in/min)
		G95	Feed rate in mm/rev. (in/rev.)
20		G96	Constant cutting speed
	D	G97	Spindle speed given in r.p.m.
21	D	G98	Accept self-maintaining preparatory functions
		G99	Do not accept self-maintaining preparatory functions

8.2 Overview of $-Words

In this overview the $-Words are organized in groups.
Only one function from each group can be active.
Normally the functions remain active until they are deselected by another function from the same group.

Group	Properties D = Default setting S = Active for 1 record		Meaning
1	**S**	**$1**	**Stop axis motion**
	S	**$53 - $54**	**Abort motion**
2	**S**	**$20**	**Handwheel enable for velocity superposition**
	S	**$21**	**Handwheel enable for path superposition**
3		**$23**	**Switch on internal tracing operation**
		$24	**Switch on tracing operation**
		$25	**Switch off tracing operation**
4		**$26**	**Independent. Switch on axis with individual feed rate**
		$27	**Independent. Switch off axis with individual feed rate**
		$28	**Independent. Incorporate axis in record change**
		$29	**Independent. Do not incorporate axis**
5		**$31**	**Switch on synchronous operation**
		$32	**Switch off synchronous operation**
6		**$33**	**Lead axis for thread cutting**
7		**$34**	**Radius axis for v = constant**
8		**$37**	**Path length calculation**
		$38	**Switch on contouring axis in IPO context**
		$39	**Switch off contouring axis in IPO context**
9		**$40**	**Switch oscillation off**
		$41	**Switch on oscillation with continuous infeed**
		$42	**Switch on oscillation with infeed on both sides**
		$43	**Switch on oscillation with infeed right**
		$44	**Switch on oscillation with infeed left**
10		**$47**	**Alternative machining plane**
11		**$48**	**Give back system axis**
12		**$65 / $66**	**Joint configuration for coordinate transformation**
13		**$70**	**Spline interpolation deselection**
		$71	**Spline interpolation selection**
14		**$90**	**Absolute measurements**
		91	**Incremental measurements**

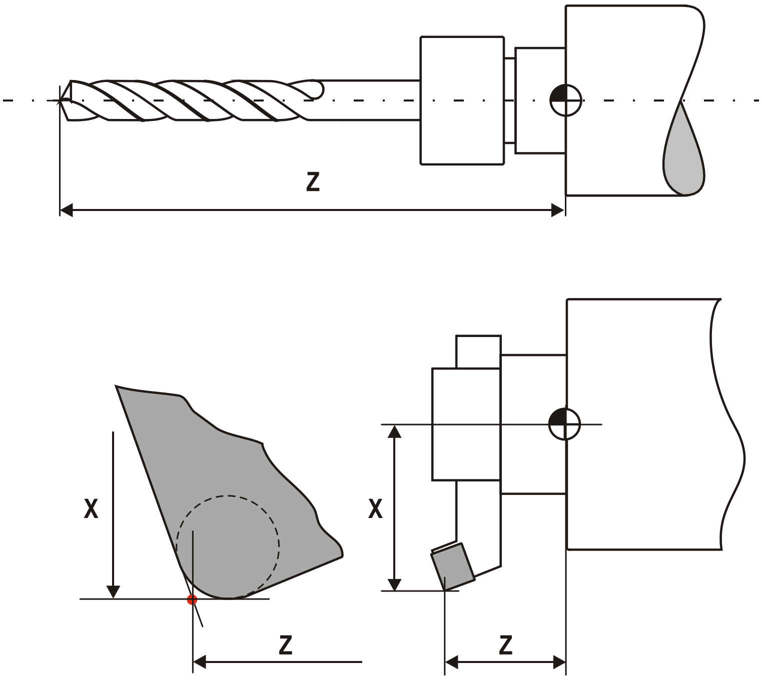
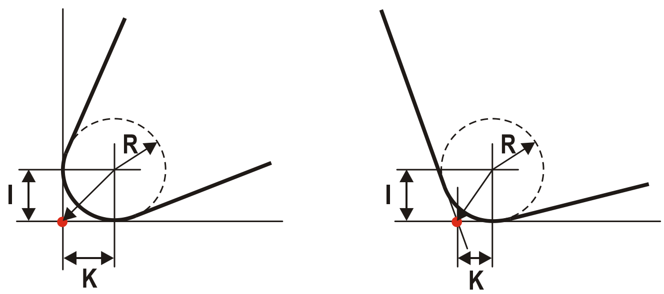
8.3 M-Functions

M0	**Programmed stop**
M1	**Optional stop**
M3	**Clockwise spindle rotation**
M4	**Anticlockwise spindle rotation**
M5	**Spindle stop**
M17	**Subroutine end**
M30	**Program end/reset**
M90, M91 through M98	**Synchronization of NC subsystems**
M1001	**Fast M-Function > M1000**

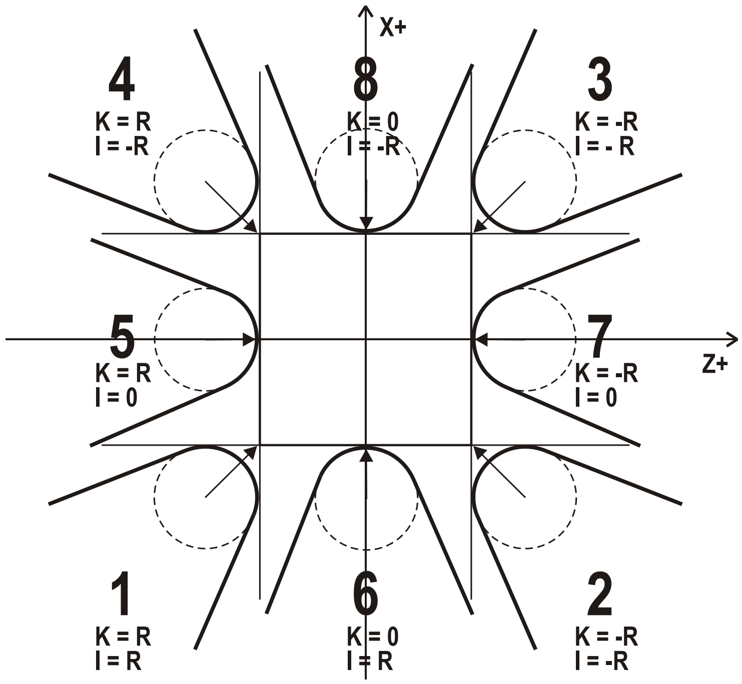
8.4 CNC – PLC interface

E	**Request a bit variable**
SE	**Set a bit variable at the start of record execution**
RS	**Reset a bit variable at the start of record execution**
WA	**Wait for bit variable = 1**
WN	**Wait for bit variable = 0**

9 Annex
9.1 Tool compensations
9.1.1 Measuring tools

Determining tool length The tool length is the distance between the tool reference point and the theoretical cutting point	
Determining tool length compensation values I and K	 R = tool nose radius I, K = compensation values

9.1.2 Quadrant assignment for cutting edge radius compensation

Quadrant assignment for tool nose radius compensation behind and in front of turning centre. Using example of XZ plane (XZ plane selection with G18).	 R = tool nose radius I, K = compensation values

Important!

The quadrant number must be entered in the Tool data memory if tool nose radius compensation is to be activated.

For cutter radius compensation, quadrant number 0 or 9 must be entered.

9.2 Tool data memory

The 99 tool data memories are mapped to different arithmetic parameters, and each begin from:

R20000	**1st tool data memory (selected with T01)**
R20100	**2nd Tool data memory (selected with T02)**
..	**..**
R29800	**99th tool data memory (selected with T99)**

The tool data memories are system-specific.

Structure of tool data memory

R-Parameters	Format	Designation	Explanation
R2xx00	0000000000	IZ	Actual time - wear monitoring in min
R2xx01	0.000	X	Tool length in X-direction in mm
R2xx02	0.000	Y	Tool length in Y-direction in mm
R2xx03	0.000	Z	Tool length in Z-direction in mm
R2xx04	0.000	I	Tool length compensation value for X-direction in mm
R2xx05	0.000	J	Tool length compensation value for Y-direction in mm
R2xx06	0.000	K	Tool length compensation value for Z-direction in mm
R2xx07	0.000	R	Tool radius in mm
R2xx08	0.000	Q	Quadrant
R2xx09	0000000000	SZ	Tool life in min
R2xx10	0000000000	VS	Tool worn, if value = 1
R2xx11	0000000000	IH	Tool call frequency, actual number
R2xx12	0000000000	SH	Tool call frequency, target number
R2xx13 through R2xx14	0000000000	---	Reserved
R2xx15 through R2xx19	0.000	---	Reserved
R2xx20 through R2xx24	0.000	User data 01 through User data 05	User data
R2xx25 through R2xx29	0000000000	User data 06 through User data 10	User data

The tool number selected with the T function (nn-1)

9.2.1 Tool monitoring

Tool monitoring for the CNC includes monitoring tool life and tool-call frequency.

Tool life monitoring records the effective operating time of the tool (not with G0, G4 and TI) and compares it with the specified limit value.
The actual time is recorded in IZ (R2xx00), the limit time (tool life) is recorded in SZ (R2xx09) in minutes.
Tool life monitoring occurs only if the life in SZ (R2xx09) is greater than zero.

Tool call frequency monitoring records the frequency of tool calls and is incremented when the T function is called. The actual frequency is entered in IH (R2xx11), the maximum permissible call frequency is entered in SH (R2xx12).
Tool call frequency monitoring is performed only when the max. call frequency in SH (R2xx12) is greater than zero.

The error message (0x02100008) 'Tool worn (System n)' is output when one of three conditions is met:

The actual time is equal to or greater than the tool life

The actual frequency is equal to or greater than the max. call frequency

By a PLC signal (coupling memory variable *cncMem.sysSect[n].flgP2N.bToolWornExt* TRUE)

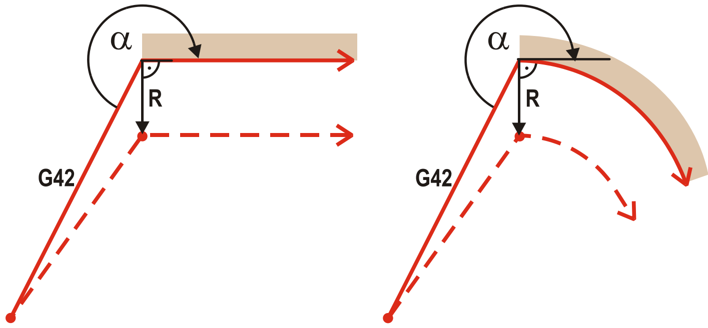
Also, a '1' will be entered in VS (R2xx10) and the coupling memory *variable cncMem.sysSect[n].flgN2P.bToolWorn* is set to TRUE.

9.3 Approach and departure strategies

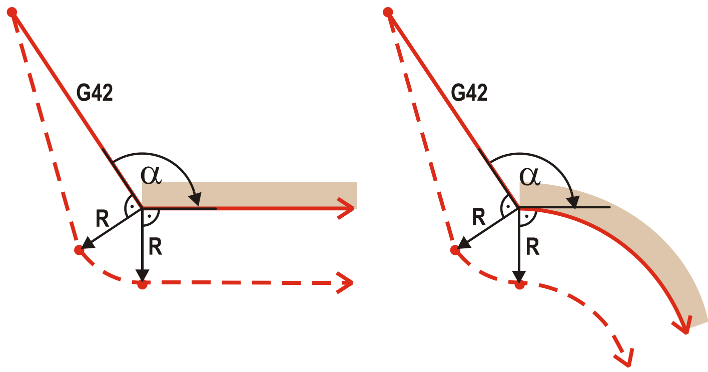
Approach

at various angles
When an SRK command is called, the start position for contour machining is approached at an angle of 90° to the contour. Depending on the approach angle, the approach is made with or without transition radii.

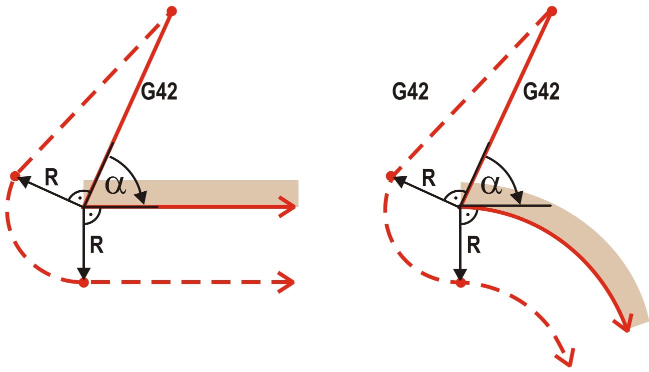
$\alpha > 180°$



$90° \leq \alpha \leq 180°$



$\alpha < 90°$

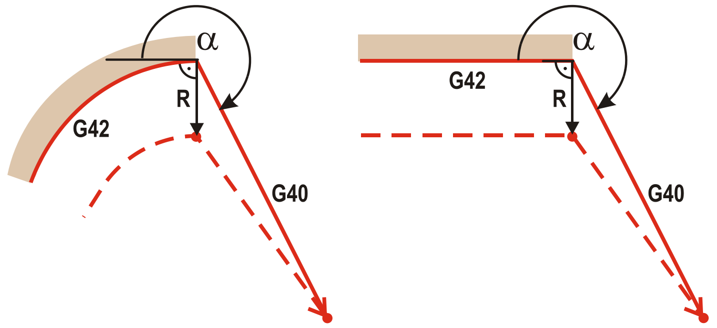


Departure

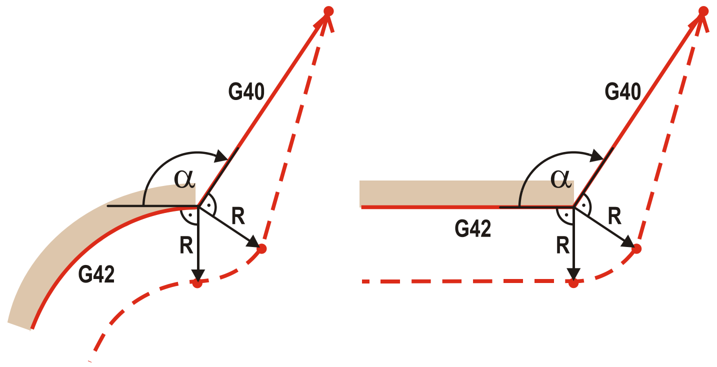
at various angles

When SRK processing is deselected, the system departs from the end position of the contour at an angle of 90° to the contour. Depending on the coordinates programmed for subsequent movement, the departure is made with or without transition radii.

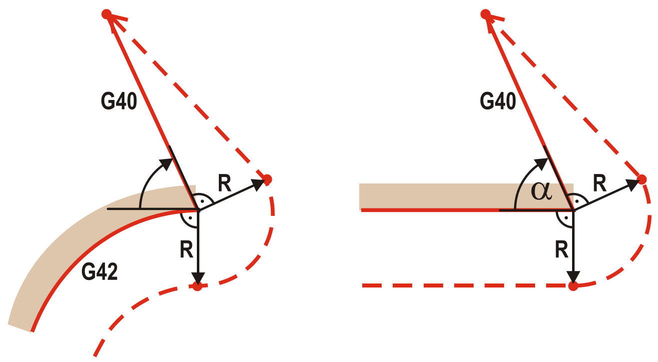
$\alpha > 180°$



$90° \leq \alpha \leq 180°$



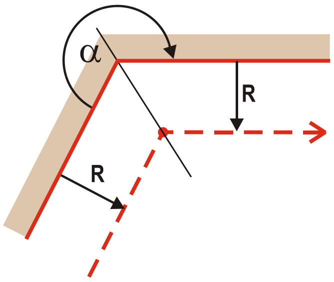
$\alpha < 90°$

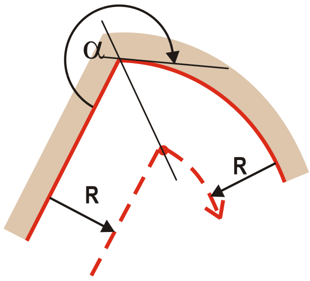


9.4 Contour transitions

Straight - straight
Straight - arc

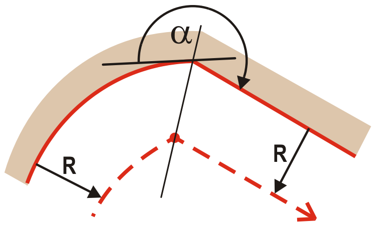
$\alpha > 180°$

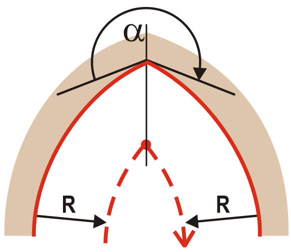




Arc - straight
Arc - arc

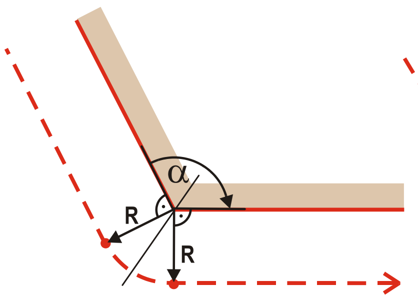
$\alpha > 180°$

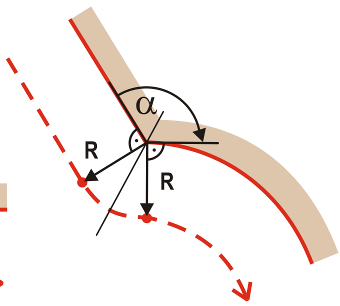


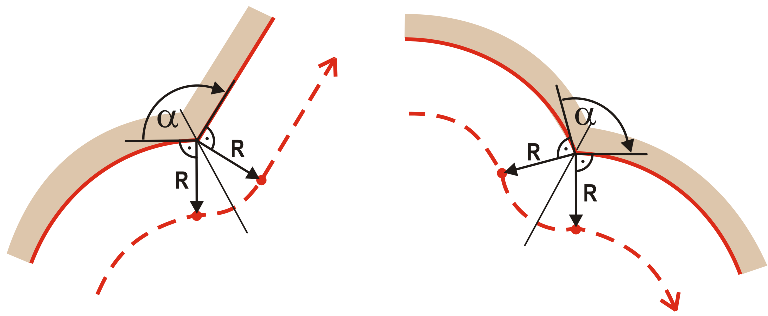
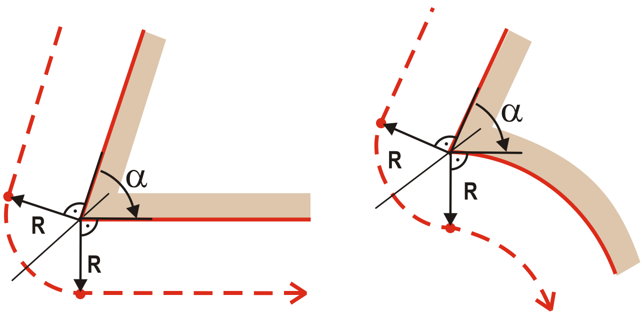
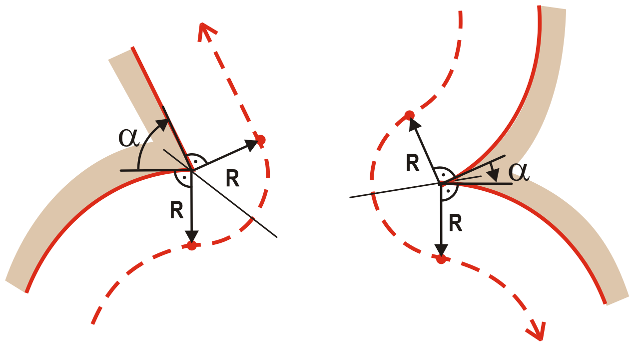


Straight - straight
Straight - arc

$90° \leq \alpha \leq 180°$



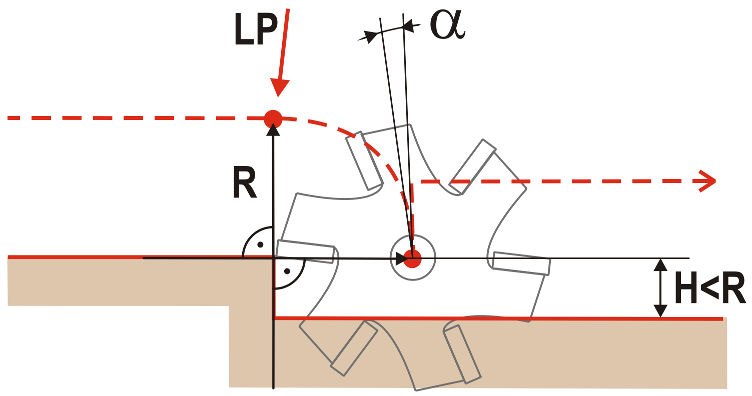


Arc - straight Arc - arc	$90° \leq \alpha \leq 180°$ 
Straight - straight Straight - arc	$\alpha < 90°$ 
Arc - straight Arc - arc	$\alpha < 90°$ 

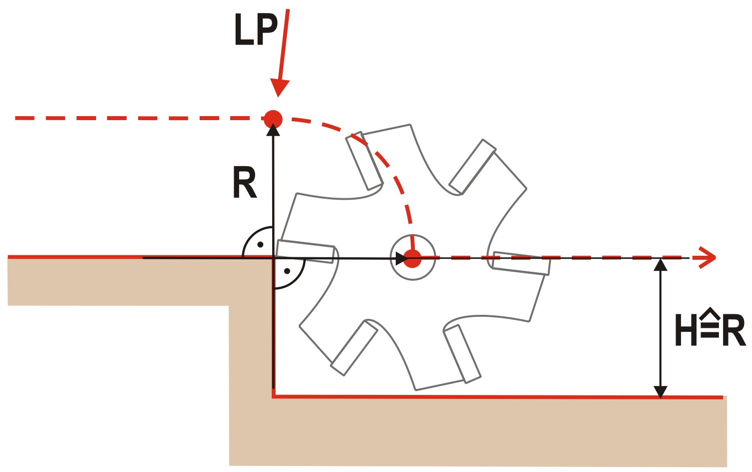
Contour transitions with error messages and STOP

Stop is activated as soon as the contour transition has been interpreted. Interpretation is predictive so the position where STOP occurs may be far before the contour transition. The last position (LP) can be approached by repeating START.

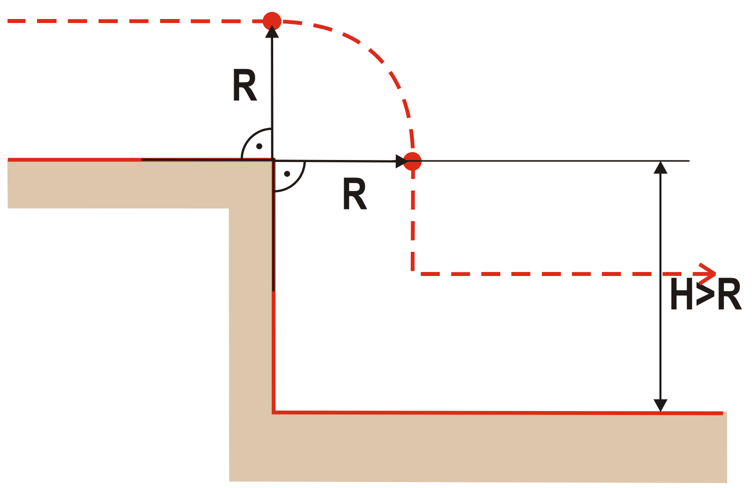
Error 0x21300003
"Angle too sharp"



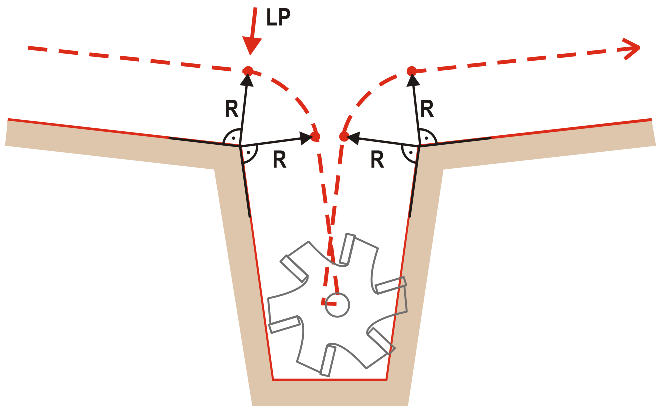
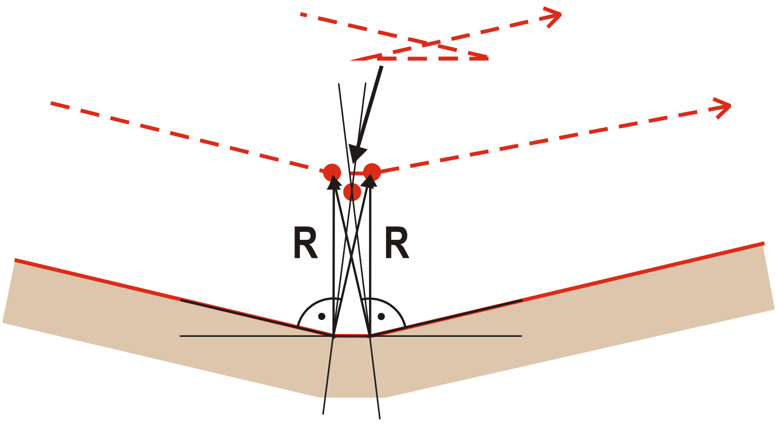
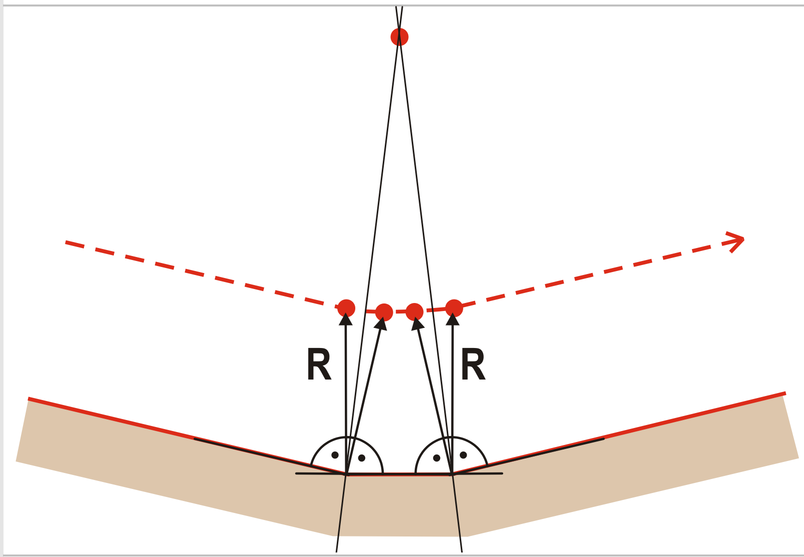
Error 0x21300005
"Start point = End point"

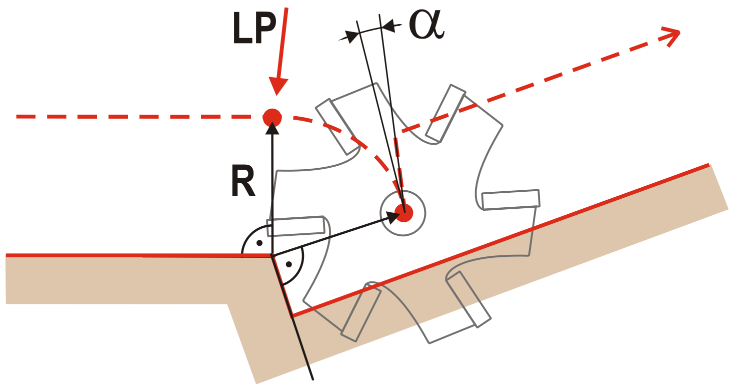
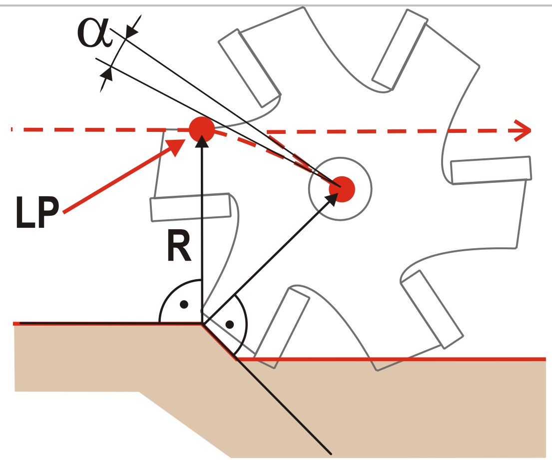


No error message, contour and tool radius are OK



LP = Last position that can be approached
H = Height
R = Tool radius

Error 0x21300004 "Tool radius too large"	 LP = Last position that can be approached R = Tool radius
Error 0x21300004 "Tool radius too large"	
No error message, contour and tool radius are OK	 R = Tool radius

Error 0x21300003 "Angle too sharp"	
Error 0x21300003 "Angle too sharp"	
	LP = Last position that can be approached R = Tool radius

schleicher control systems

9.5 Lending NC Axes Between NC Subsystems

When the axes of one NC subsystem are to interpolate with the axes of another NC subsystem, it is first necessary to sign off the axes to be lent from their currently assigned NC subsystem; this is done with $48. The axes to be lent are assigned axis letters from the number of the axis (see $48 Enable axis for subsystem change Page 100).

Example:
Axis X is the 3rd axis of the controller and usually belongs to NC subsystem 1. If this axis is to move to NC subsystem 2 you have to assign a number to the axis. (See also M90, page 108)

Example		
System 1	%1	new syntax form
	N10 G1 X0 Y0 F500	
	N20 $48 XY	Sign off X and Y from system 1 using $48
	N30 M90	synchronize with 2nd system N10
	N40 – N60	Further processing
	N70 M90	synchronize with 2nd system N50
	N80 X:=1 Y:=2	Sign on X and Y in system 1
	N90 X-100 Y-100 M17	
System 2	%2	new syntax form
	N10 M90	synchronize with 1st system N30
	N20 X:=1 Y:=2	Sign on X and Y in system 2
	N30 G1 X20 Y20 F10	
	N40 $48 XY	Sign off X and Y from system 2 using $48
	N50 M90 M17	synchronize with 1st system N70

System 1	%1	old syntax form
	N10 G0 X1000	
	N20 M90	(Wait for NC subsystem 2)
	N30 (X must not be used)	(Axis X currently moving in NC subsystem 2)
	N40 M90	(Wait for NC subsystem 2)
	N50 G0 X0 M17	
System 2	%2	old syntax form
	N10 M90	(Wait for NC subsystem 1)
	N20 X:=3	(The 3rd axis of NC subsystem 1 is moved in subsystem 2 as the X axis)
	N30 G1 X500 F2000	
	N40 $48 X	(X released again with $48)
	N40 M90	(Wait for NC subsystem 1)
	N50 M17	

10 Index

Schleicher Electronic Berlin GmbH
Wilhelm-Kabus-Straße 21-35
10829 Berlin
Germany

Tel.: +49 30 33005 - 0
E-Mail: info@schleicher.berlin
Internet: http://www.schleicher.berlin