NUM
1020/1040/1060T

PROGRAMMING MANUAL

VOLUME 1

0101938820/5

Table of Contents

Record of Revisions

DOCUMENT REVISIONS		
Date	**Revision**	**Reason for revisions**
04-92	0	Document creation (conforming to software index B)
11-93	1	Update to conform to software index D Manual revisions: - Classification of G preparatory functions and M miscellaneous functions. - Processing of blocks and programmed G and M functions (with G997 to G999). - Programming of error numbers and messages. - Counterboring, boring and tapping cycles. - The sections on structured programming and the use of table of variables are transferred from this manual to the supplementary programming manual. Taking into account of upgrades Software index C: - Special programming of PLC axes. - Creation of external parameter E41004. Software index D: - Spline curve interpolation. - Rigid tapping. - Creation of external parameters E42000 to E42127, E79003, E79004, E41005, E941xx, E960xx, E961xx, E962xx, E963xx.
09-94	2	Update to conform to software index F Manual revisions: - Circular interpolation defined by three points (G23) - Block sequencing without stopping movement, with sequence interruption and feed rate limiting after interrupt by EF (changes to G10) - Temporary suspension of next block preparation (G79+/-) - Automatic homing subroutine branch - Subroutine branch on reset - Message transmission by $0 to $6 (formerly in Chapter 3, moved to the end of Chapter 4) - Added a paragraph concerning access to the Profil function (see Sec. 5.2) - Unconditional call to a sequence by G77N..

		Added changes Software at index E: - Polar programming - Feed rate in fillets EB+ and chamfers EB- - Movements parallel to inclined axes (G05 and G07) - Extension of parameter E21000 - External parameters E49001 to E49128, E931xx, E932xx, E933xx, E7x100, E934xx, E951xx, E952xx, E41102, E33xyz, E43xyz, E34xxy, E44xxy, E20100 to E20111, E9030x, E9031x, E9032x, E9033x, E970xx, E971xx, E972xx, E11014, E11016 and E32001 - Acquisition of variables in the stack of another axis group by function VAR H.. N.. N.. - Adressing by function [.RG80] - Conversion of the internal unit to the programming unit by function U for linear axes.
02-95	3	Update to conform to software index G Manual revisions: - Spindle synchronisation - External parameters E11013, E41006, E935xx, E980xx
05-96	4	Update to conform to software index J Manual revisions: - transmission of a message from CNC to PC ($9) - call of a subroutine return block (G77 -i) - tool number T defined by 8 digits - inclined wheel p, grinding machine - external parameters E32002, E32003, E32004, E32005, E69002, E9034x, E9035x, E7x101, E913xx, E942xx, E973xx, E982xx and E983xx Inclusion of changes Software index H - external parameters E11008, E936xx

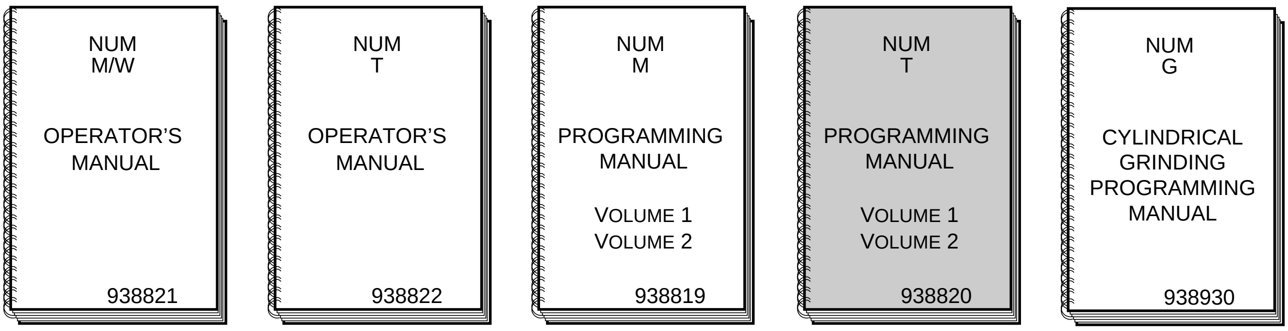
DOCUMENT REVISIONS		
Date	**Revision**	**Reason for revisions**
06-97	5	Update to conform to software index L Manual revisions: - ISO programme or block creation/deletion (G76+/-) - Conversion of the internal unit to the programming unit by function M for rotary axes Added changes: Software index J and K

Foreword

Structure of the NUM 1020/1040/1060 Documentation

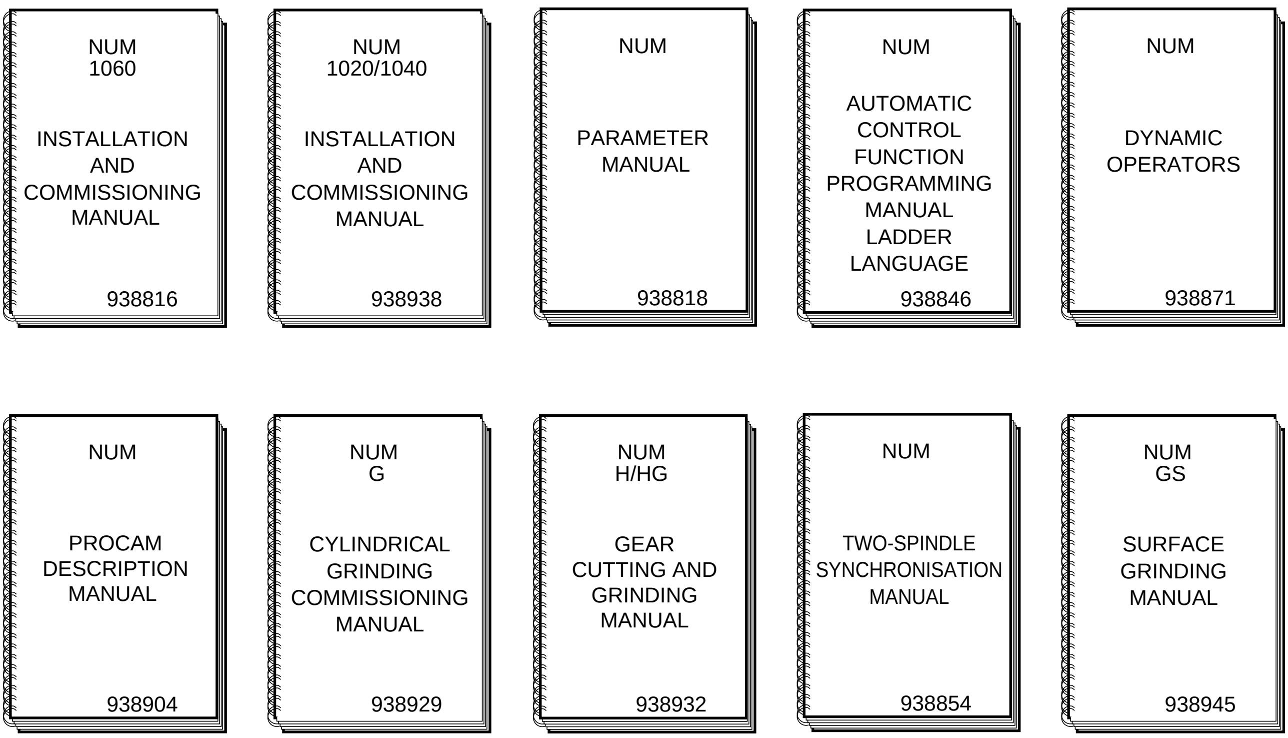
User Documents

These documents are designed for the operator of the numerical control.



OEM Documents

These documents are designed for the OEM integrating the numerical control on a machine.



OEM Documents (cont'd)

These documents are designed for the OEM integrating the numerical control on a machine.

NUM
SETTOOL PARAMETER INTEGRATION TOOL
938924

NUM
PLCTOOL LADDER LANGUAGE PROGRAMMING TOOL
938859

NUM
MMITOOL MAN/MACHINE INTERFACE CUSTOMISATION TOOL
938946

Special Programming Documents

These documents concern special numerical control programming applications.

NUM
SUPPLEMENTARY PROGRAMMING MANUAL
938872

NUM M
PROCAM MILL INTERACTIVE PROGRAMMING MANUAL
938873

NUM T
PROCAM TURN INTERACTIVE PROGRAMMING
938874

NUM
DUPLICATED AND SYNCHRONISED AXES
938875

NUM
PROFIL FUNCTION USER'S MANUAL
938937

NUM G
PROCAM GRIND INTERACTIVE PROGRAMMING
938931

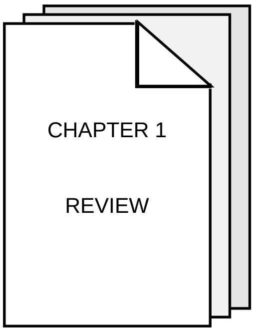
NUM
POLYGON CUTTING MANUAL
938952

NUM GS
PROCAM GRIND INTERACTIVE PROGRAMMING
938953

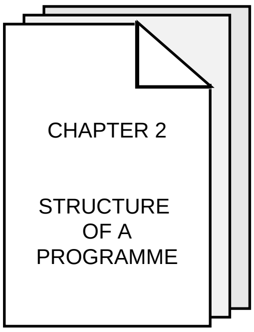
NUM T
PROCAM TURN TECHNOLOGICAL DATA
938959

NUM M
PROCAM MILL TECHNOLOGICAL DATA
938958

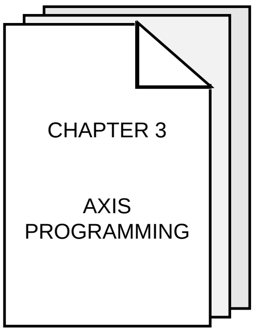
Programming Manual



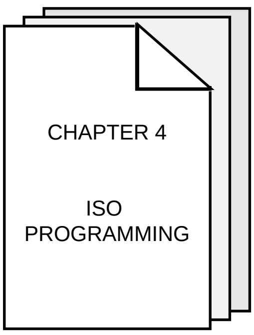
General description of the NC and its use with the machine tool.
Review of the rules and standards related to the NC/machine-tool combination.



Rules for writing a part programme by assembling characters into words, words into blocks and blocks into a complete programme.



Description of the features related to axis programming.



Detailed description of functions related to ISO programming.

CHAPTER 5

PROFILE GEOMETRY PROGRAMMING

Detailed description of profile geometry programming (PGP).

Description of access to the Profil function and the contour call created by Profil.

PGP and Profil are used to define contours as a sequence of geometric elements, with computation of intermediate points. PGP and Profil are extensions of ISO programming.

CHAPTER 6

PARAMETRIC PROGRAMMING

Gives the possibility of assigning variables to NC functions. The values of the variables can be obtained by computation or by reading machine data.

CHAPTER 7

PROGRAMME STACK- L VARIABLES AND SYMBOLIC VARIABLES

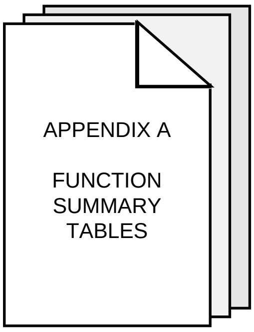
Possibility of saving or restoring a chain of L variables in a single instruction.

Possibility of naming the variables used in a part programme to make the programme easier to read.

CHAPTER 8

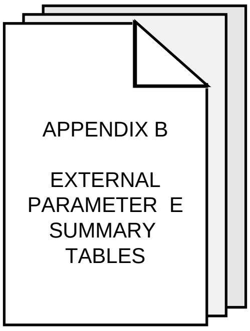
PROGRAMMING OF ERROR NUMBERS AND MESSAGES

Gives the possibility of programming and displaying error numbers and messages.



Tables given as lists of:

- G preparatory functions,
- M miscellaneous functions,
- other functions.



Tables given as lists of:

- exchange parameters with the PLC,
- parameters stored in the NC memory.

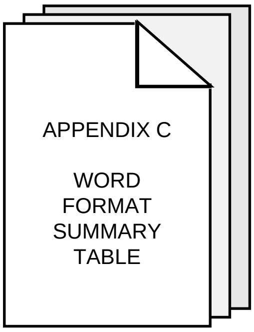
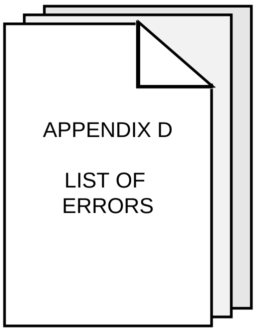


Table given as a list of words with their associated formats.



List of NC error numbers and definitions.

Use of this Programming Manual

Function Syntax Entry Conventions

The lines (blocks) of a part programme include several functions and arguments.

Special syntax rules apply to each of the functions described herein. These syntax rules specify how the programme blocks must be written.

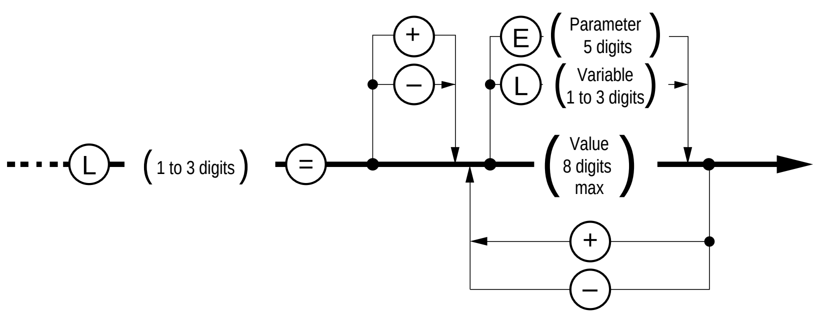
Certain syntax formats are given as a line. The following conventions simplify writing the line:
- the function to which the syntax format is related is highlighted by boldface type,
- terms between square brackets «[..]» are optional functions or arguments in the block (or functions activated earlier, with values unchanged, etc.) (except Sec. 6.6 and Chapter 7),
- «/» indicates a choice between several terms (equivalent to «or») (except Sec. 6.6 and Chapter 7),
- «..» after a letter replaces a numerical value,
- «...» replaces a character string (for instance a message).

Examples

Syntax of function G12

N.. [G01/G02/G03] **G12** X.. Z.. [F..] [$0…]

Syntax in the form of a Conway diagram



NC Operating Modes

Certain NC operating modes are mentioned herein when they are directly related to the use of ISO functions. For additional information on these modes, refer to the Operator Manual.

Optional Functionalities

The use of certain functionalities described herein requires validating the associated options. The «OPTIONS» system page is used to check for the presence of these functionalities (for access to the «OPTIONS» page and the list of functionalities, see Chapter 2 of the Operator Manual).

List of G, M and Other Functions

The lists at the beginning of the manual indicate the pages where the G, M and other functions are found (yellow pages).

Index

The index at the end of the manual facilitates access to information by keywords.

Agencies

The list of NUM agencies is given at the end of the manual.

Questionnaire

To help us improve the quality of our documentation, we kindly request you to return the questionnaire at the end of the manual.

Lists of G, M and Other Functions

G Functions

Code	Description	Page
G00	High-speed linear interpolation	4 - 29
G01	Linear interpolation at programmed feed rate	4 - 32
G02	Clockwise circular interpolation at programmed feed rate	4 - 36
G03	Counterclockwise circular interpolation at programmed feed rate	4 - 36
G04	Programmable dwell	4 - 238
G05	Movement on an inclined axis	4 - 269
G06	Spline curve execution command	4 - 216
G07	Initial tool positioning before machining on an inclined axis	4 - 268
G09	Accurate stop at end of block before going to next block	4 - 59
G10	Interruptible block	4 - 180
G12	Overspeed by handwheel	4 - 242
G16	Definition of tool axis orientation with addresses P, R	4 - 72
G20	Programming in polar coordinates (X, Z, C)	4 - 226
G21	Programming in cartesian coordinates (X, Y, Z)	4 - 229
G22	Programming in cylindrical coordinates (X, Y, Z)	4 - 234
G23	Circular interpolation defined by three points	4 - 44
G33	Constant lead thread cutting	4 - 92
G38	Sequenced thread cutting	4 - 99
G40	Tool radius offset (cutter compensation) cancel	4 - 80
G41	Left tool radius offset (cutter compensation)	4 - 79
G42	Right tool radius offset (cutter compensation)	4 - 79

Code	Description	Page
G48	Spline curve definition	4 - 216
G49	Spline curve deletion	4 - 216
G51	Mirroring	4 - 261
G52	Programming of movements in absoluted dimensions with reference to the measurement origin	4 - 203
G53	DAT1 and DAT2 offset cancel	4 - 206
G54	DAT1 and DAT2 offset enable	4 - 206
G59	Programme origin offset	4 - 209
G63	Roughing cycle with groove	4 - 151
G64	Turn/Face roughing cycle	4 - 128
G65	Groove roughing cycle	4 - 140
G66	Plunging cycle	4 - 146
G70	Inch data input	4 - 244
G71	Metric data input	4 - 244
G73	Scaling factor cancel	4 - 259
G74	Scaling factor enable	4 - 259
G75	Emergency retraction subroutine declaration	4 - 189
G76	Transfer of the current values of «L» and «E» parameters into the part programme	6 - 55
G76+/-	ISO programme or block creation/deletion	4 - 198
G77	Unconditional branch to a subroutine or block sequence with return	4 - 165
G77 -i	Call of a subroutine return block	4-196
G78	Axis group synchronisation	4 - 279
G79	Conditional or unconditional jump to a sequence without return	4 - 174
G79 +/-	Temporary suspension of next block preparation in a sequence with movements	4 - 187

Code	Description	Page
G80	Canned cycle cancel	4 - 91
G81	Centre drilling cycle	4 - 104
G82	Counterboring cycle	4 - 106
G83	Peck drilling cycle	4 - 108
G84 G84	Tapping cycle Rigid tapping cycle	4 - 113 4 - 111
G85	Boring cycle	4 - 117
G87	Drilling cycle with chip breaking	4 - 119
G89	Boring cycle with dwell at hole bottom	4 - 122
G90	Programming in absolute dimensions with respect to the programme origin	4 - 5
G91	Programming in incremental dimensions with respect to the start of the block	4 - 5
G92 G92 R.. G92 S..	Programme origin preset Programming of the tangential feed rate Spindle speed limiting	4 - 207 4 - 66 4 - 27
G94	Feed rate expressed in millimetres, inches or degrees per minute	4 - 61
G95	Feed rate expressed in millimetres or inches per revolution	4 - 64
G96	Constant surface speed expressed in metres per minute	4 - 15
G97	Spindle speed expressed in revolutions per minute	4 - 13
G98	Definition of the start X for interpolation on the C axis	4 - 228
G997	Enabling and execution of all the functions stored in state G999	4 - 264
G998	Enabling of execution of the blocks and part of the functions processed in state G999	4 - 264
G999	Suspension of execution and forcing of block concatenation	4 - 264

M Fonctions

Code	Description	Page
M00	Programme stop	4 - 248
M01	Optional stop	4 - 250
M02	End of programme	2 - 9
M03	Clockwise spindle rotation	4 - 11
M04	Counterclockwise spindle rotation	4 - 11
M05	Spindle stop	4 - 11
M06	Tool change	4 - 70
M07	Coolant 2 on	4 - 247
M08	Coolant 1 on	4 - 247
M09	Coolant off	4 - 247
M10	Clamp	4 - 246
M11	Unclamp	4 - 246
M12	Programmed feed stop	4 - 240
M19	Spindle index	4 - 21
M40 to M45	Spindle speed ranges	4 - 20
M48	Enable overrides	4 - 255
M49	Disable overrides	4 - 255
M61	Release of current spindle in the axis group	4 - 278
M62 to M65	Control of spindles 1 to 4	4 - 23
M66 to M69	Measurement of spindles 1 to 4	4 - 25
M997	Forced block sequencing	4 - 254
M998	Reactivation of edit (EDIT) and manual data input (MDI) modes and subroutine calls by the automatic control function	4 - 252
M999	Programmed cancellation of the edit (EDIT) and manual data input (MDI) modes and subroutine calls by the automatic control function	4 - 252

Other Functions

Code	Description	Page
$0	Message transmission to the display	4 - 288
$1 to $6	Message transmission to the PLC function or a remote server or a peripheral	4 - 290
/	Block skip	4 - 256
D..	Call to tool correction	4 - 74
ED..	Programmed angular offset	4 - 215
EG..	Programmed acceleration modulation	4 - 258
T	Tool number	4 - 70
M	Conversion of the internal unit of rotary axes	6-6 and 6-19
U	Conversion of the internal unit of linear axes	6-6 and 6-19

1 Review

1

The aim of this chapter is to introduce concepts that will be detailed in the rest of the manual, rather than to reflect the way an operator works on the machine.

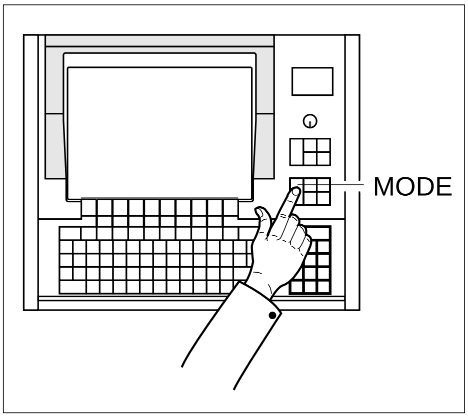
For instance, in Section 1.2.4 (Offset Definition), the aim is to define the offsets and corresponding origins or zero points rather than give a method for measuring the offsets.

1.1 System Overview

1.1.1 Overview of Modes

The operator uses the numerical control (NC) in various operating modes accessible from the operator panel.

Each mode corresponds to a particular use of the numerical control (continuous machining, programme loading, tool setting, etc.).



1.1.2 Defining a Programme

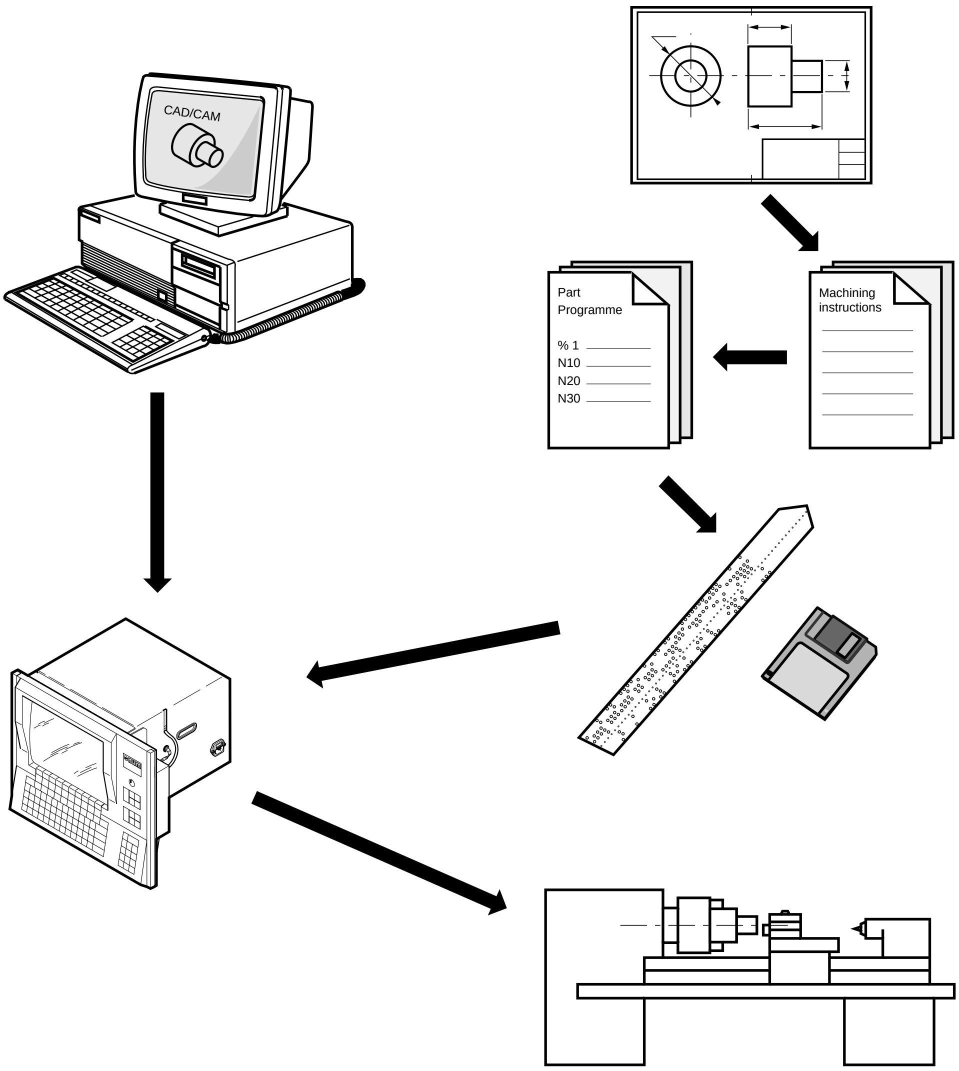
A programme is a sequence of instructions written in a programming language specific to the numerical control (the most widely used is ISO code: International Standards Organization).

The numerical control interprets the programme to control actions on a machine-tool.

The most widespread storage media for programmes are punched tape and diskettes.

1.1.3 Preparating a Programme

A part programme can be created by traditional programming or using a CAD/CAM system.

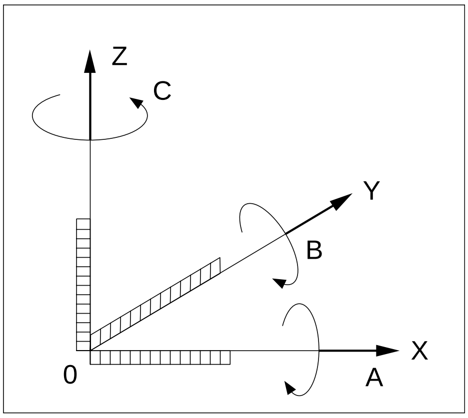


1.2 Machine Overview

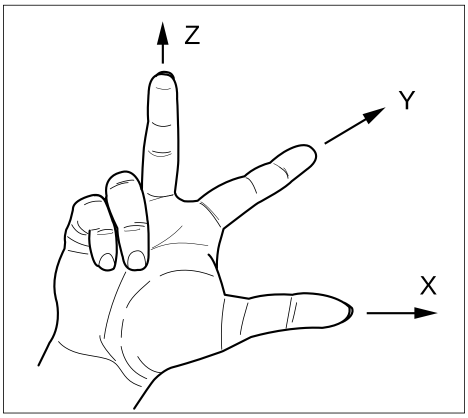
1.2.1 Review of Axis Definition and Direction

A coordinate system is used to identify the positions and movements of an object with respect to an origin or zero point.

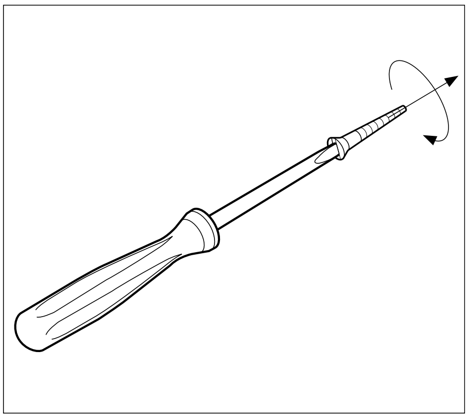
A rectangular cartesian coordinate system is a direct three-axis system of three linear axes, X, Y and Z, with which are associated three rotary axes, A, B and C.



The direction of axes X, Y and Z is easily remembered by the right-hand rule.



The positive direction of rotation of a rotary axis corresponds to the direction of screwing of a right-hand screw on the associated axis.



1.2.2 Machine Overview

The manufacturer defines the coordinate system associated with the machine in accordance with standard ISO 841 (or NF Z68-020).

The X, Y and Z axes, parallel to the machine slideways, form a right-handed rectangular cartesian coordinate system.

The coordinate system measures tool movements with respect to the part to be machined, assumed fixed.

REMARK *When it is the part that moves, it may be more convenient to identify its movements. In this case, axes X', Y' and Z', pointing in opposite directions from axes X, Y and Z, are used.*

The direction of the axis of a machine depends on the type of machine and the layout of its components.

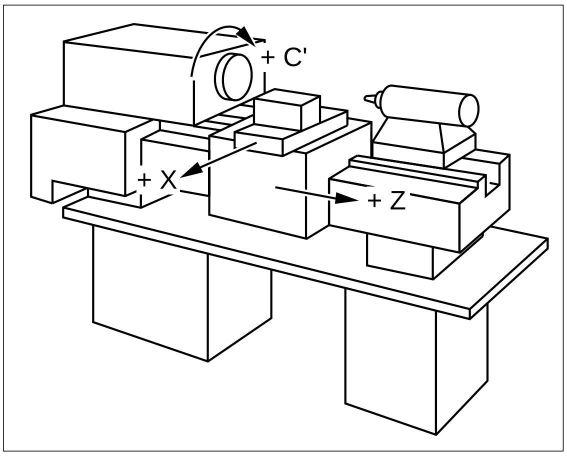
For a lathe:
- the Z axis is the same as the spindle axis,
- the X axis is perpendicular to the Z axis and corresponds to radial movement of the tool-holder turret,
- the Y axis (generally a dummy axis) forms a right-handed coordinate system with the X and Z axes.

Positive movement along the Z or X axis increases the distance between the part and the tool.

Rotary axes A, B and C define rotations around axes parallel to X, Y and Z.

Secondary linear axes U, V and W may or may not be parallel to primary axes X, Y and Z.

For more details, refer to the above-mentioned standard.



1.2.3 Definition of Travels and Origins

The NC processor computes all movements with respect to the measurement origin or zero point of the machine.

When the system is turned on, it does not know the measurement origin. The mechanical travel on each machine axis is limited by maximum and minimum limit switches.

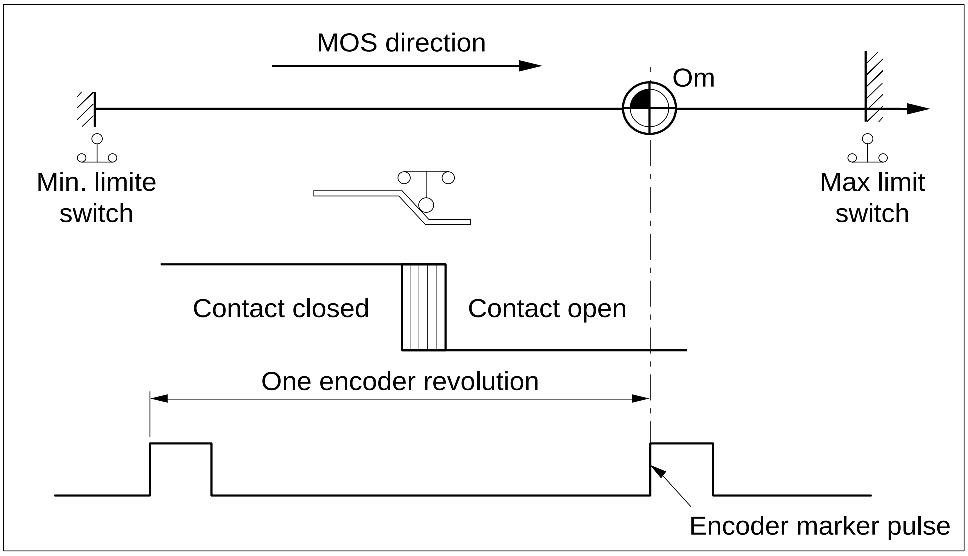
OM :

The system establishes the measurement origin (OM) via a homing procedure (MOS).

Om :

The home switch is set in a specific physical location: the machine zero point (Om) may or may not be the same as the measurement origin (OM).

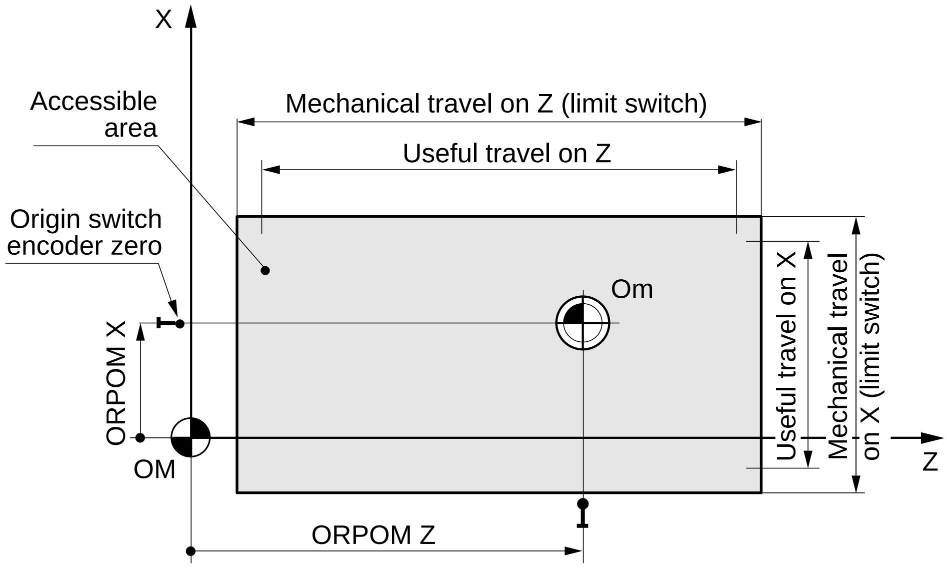
The homing procedure is completed for each of the axes when:
- the origin limit switch is actuated in the direction of movement specified by the m/c manufacturer (MOS direction),
- the encoder which measures axis movement outputs its marker pulse.



When homing (MOS) is completed, the system applies the offset defined by the manufacturer to each of the axes to establish the measurement origin (OM).

Measurement origin offset (OM/Om) = ORPOM

The useful travel on each of the axes is limited by software limits whose values are defined by the machine parameters.



1.2.4 Offset Definitions

To write a part programme, the programmer chooses a programme origin.

The programme origin is generally a starting point for dimensional measurements on the part drawing.

OP : 

The operator sets the programme origin (OP) as shown below:

Op : 

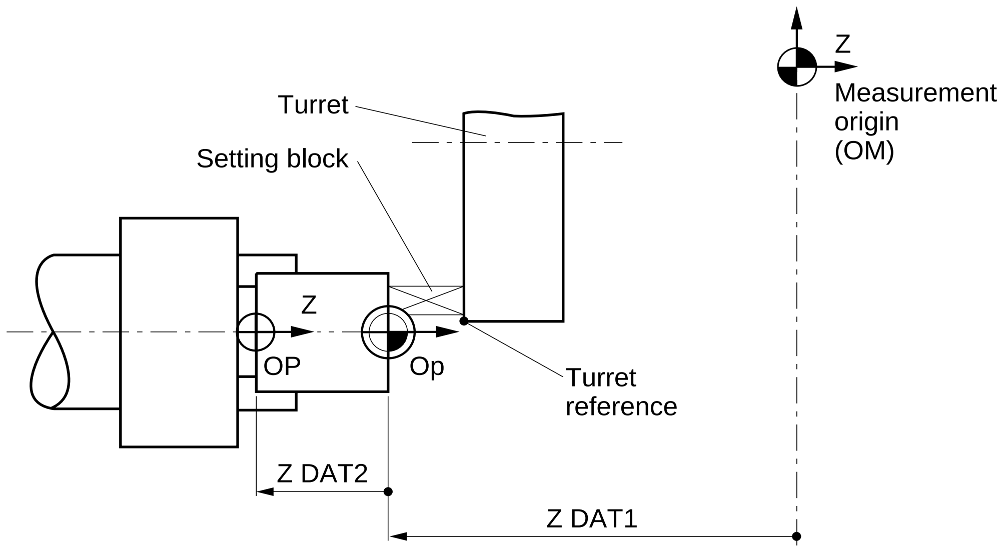
He sets (for each axis) a known, accessible point on the part, called the part origin, (Op). This may be the same point as the programme origin.

Part origin offset (Op/OM) = DAT1

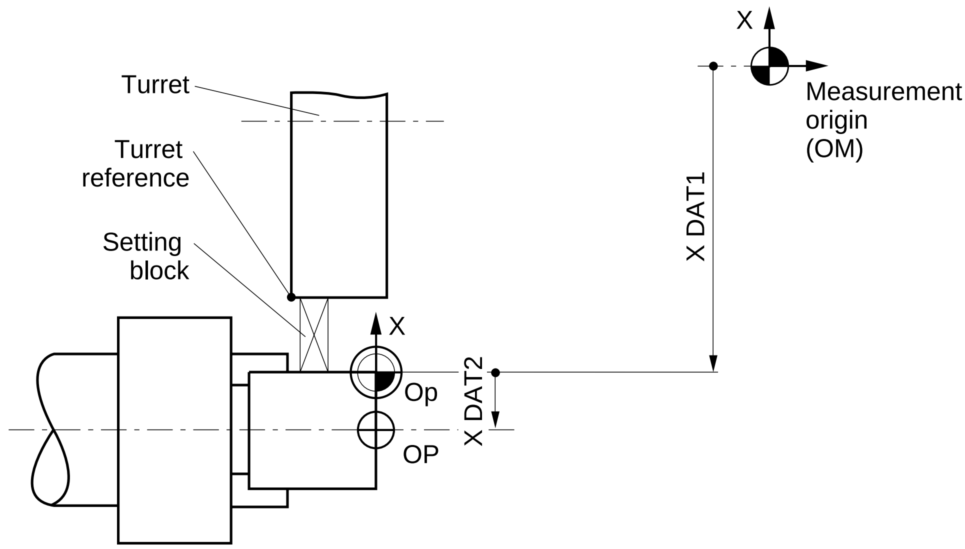
It is possible to set the DAT1, DAT2 values from the part programme.

Programme origin offset (OP/Op) = DAT2

Offsets on the Z axis

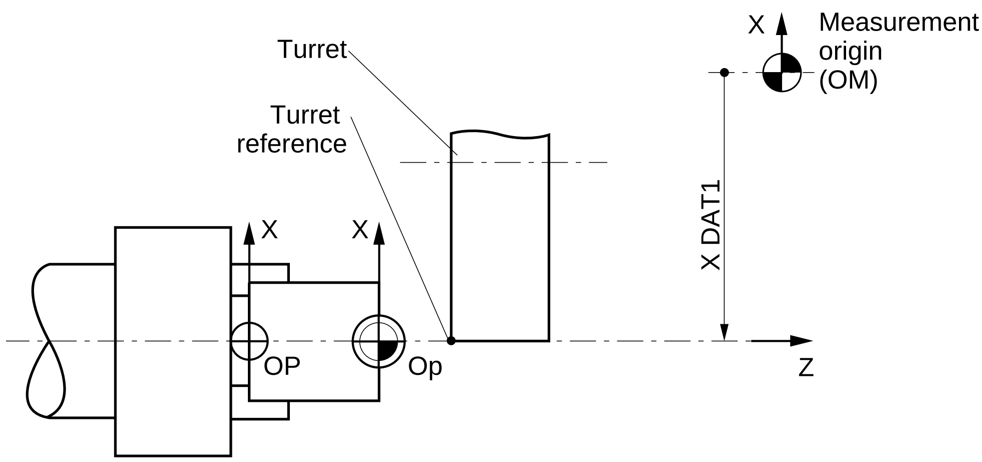


Offsets on the X axis (solution with DAT2)

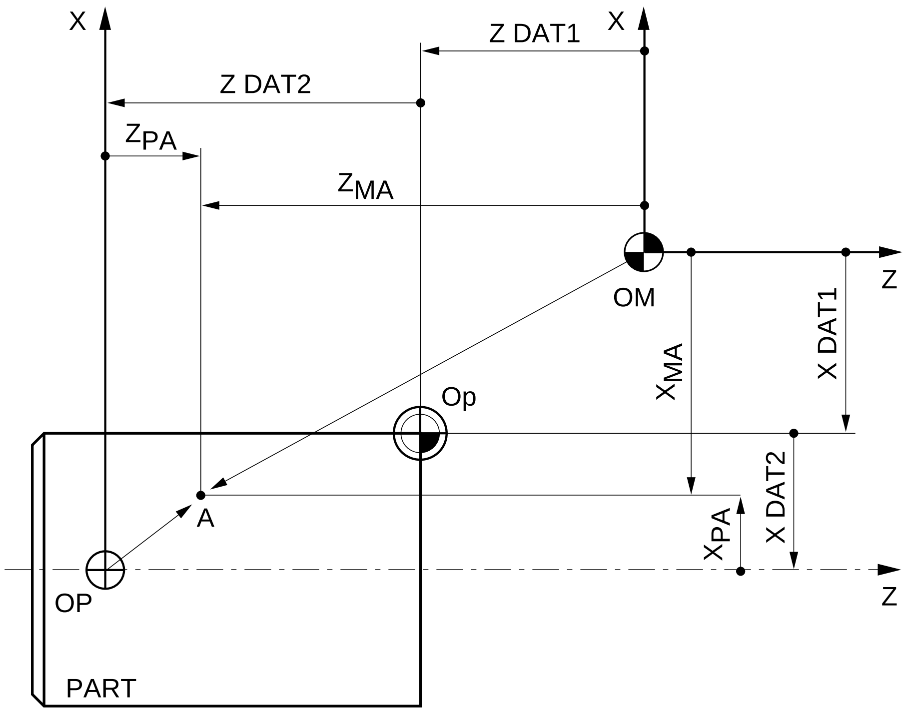


Offsets on the X axis (solution without DAT2)

X DAT1: Fixed value measured between OM and the spindle axis.



The coordinates of a point (A) defined with respect to the programme origin (OP) are converted by the NC to coordinates with respect to the measurement origin (OM):



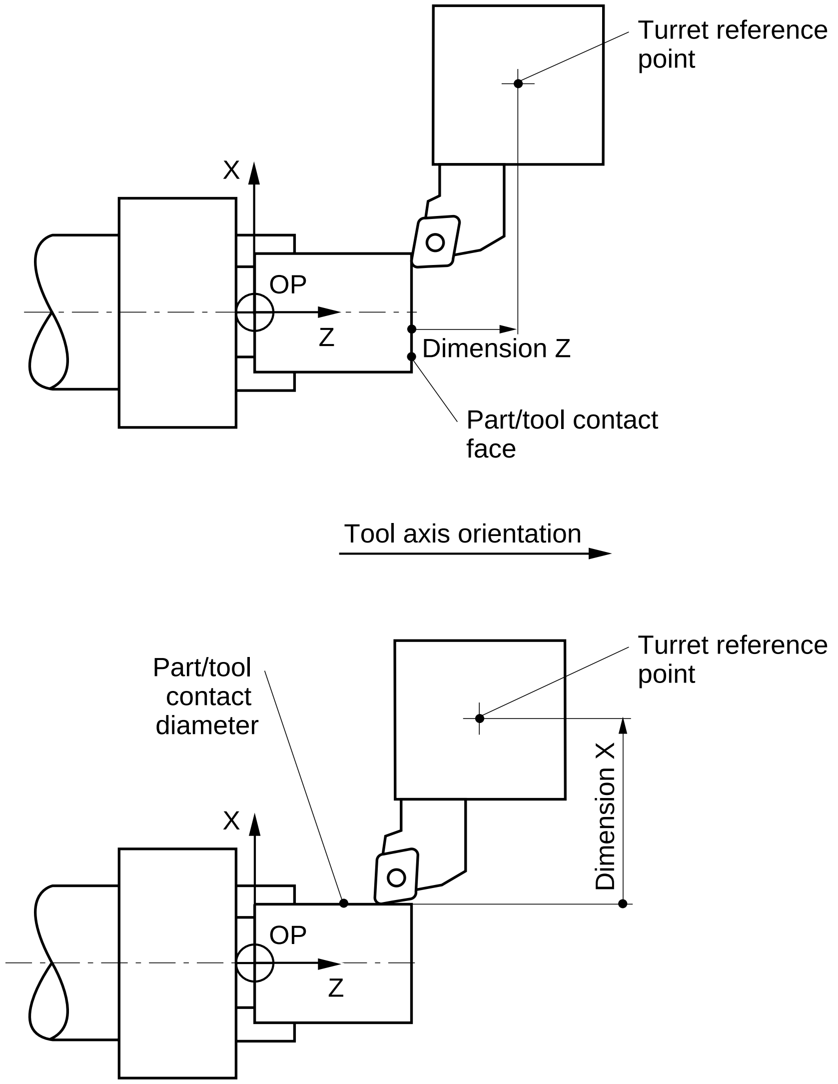
Programme dimensions (with respect to OP)	Measurement dimensions (with respect to OM)
X_{PA}	$X_{MA} = X_{PA}$ + X DAT1 + X DAT2
Z_{PA}	$Z_{MA} = Z_{PA}$ + Z DAT1 + Z DAT2

Programmed shifts can be added to the programme dimensions.

1.2.5 Definition of the Tool Dimensions

1.2.5.1 Definition of the Tool Dimensions

Tool dimension = distance from tool cutting edge to turret reference point



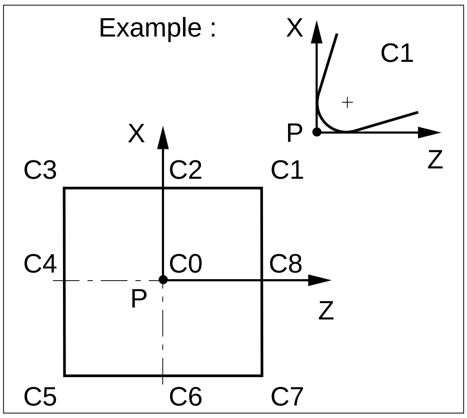
Tool X dimension = X Tool Z dimension = Z

1.2.5.2 Definition of Tool Tip Radius and Orientation

The description of a tool is shown below:

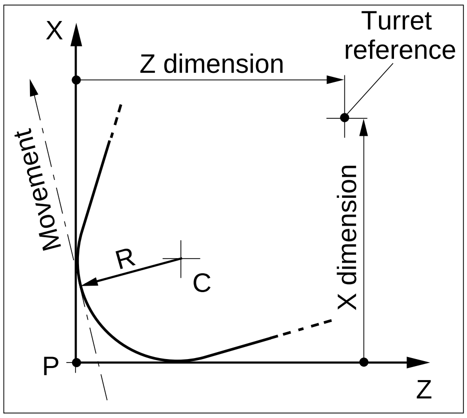
Tool tip orientation = codes C0-C8

The tool tip orientation code allows the system to locate the centre (C) of the tool tip from the theoretical cutting point (P).



Tool tip radius = R

The real cutting point of the tool is obtained by applying a vector of length R perpendicular to the direction of movement from C.

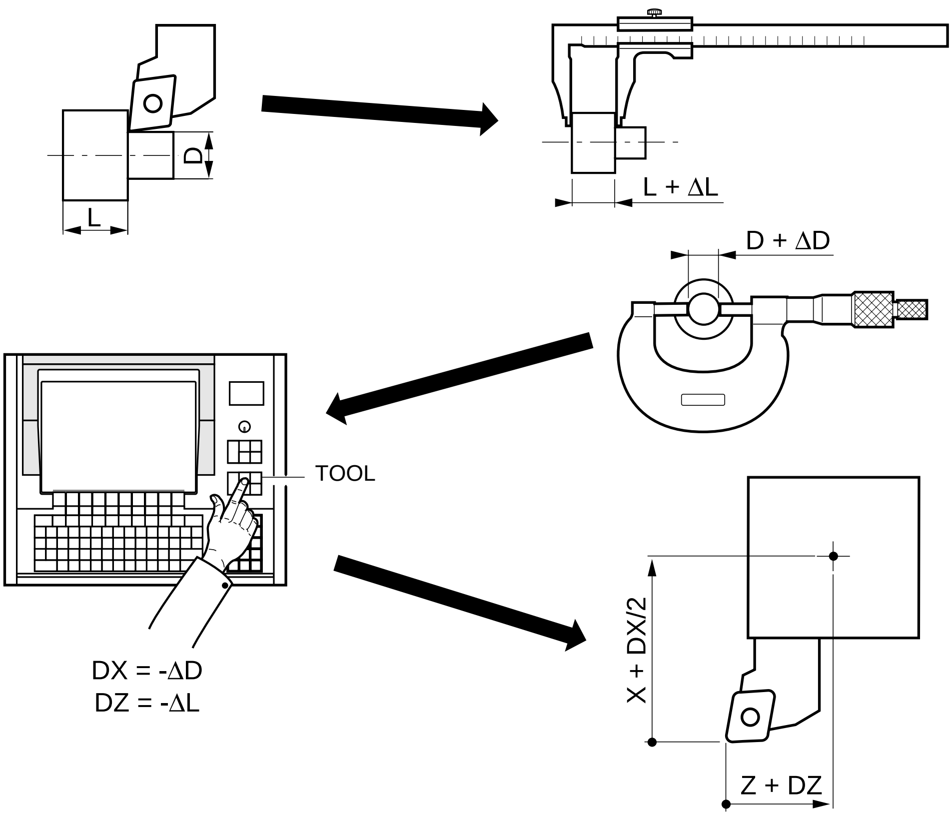


1.2.6 Definition of Dynamic Tool Corrections

At any time (even during machining), the operator can enter dynamic tool corrections when he observes a difference between the expected and the actual results on a part.

The corrections (positive or negative) compensate for slight dimensional variations of the tool or part (wear, expansion).

Dynamic tool correction on X = DX (diameter) Dynamic tool correction on Z = DZ



The system takes into account the corrected tool dimensions:

Corrected length on X = X dimension + DX/2 Corrected length on Z = Z dimension + DZ

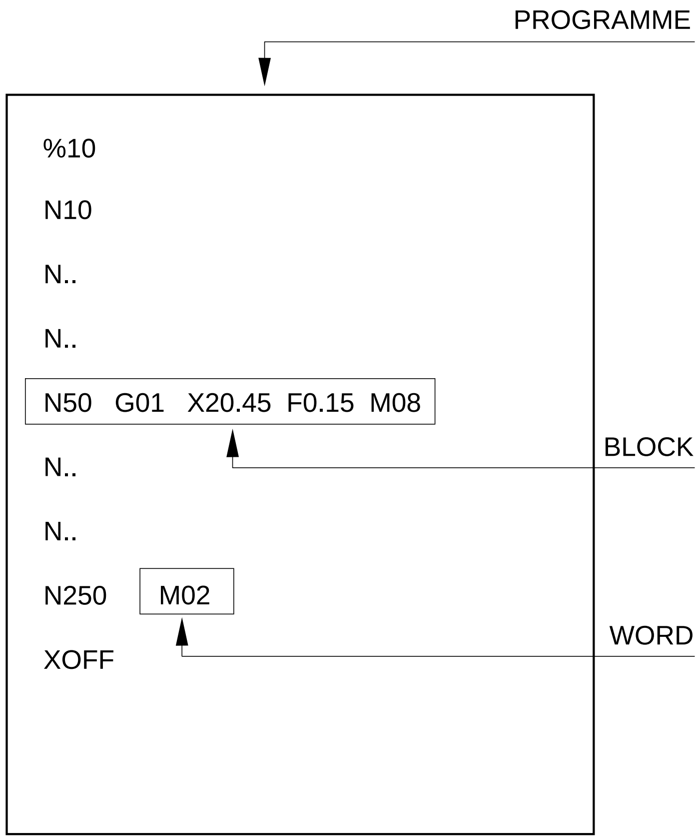
2 Structure of a Programme

A CNC part programme is a list of instructions and data to be transmitted to the control system.

The creation of a programme consisting of blocks and words must obey structure, syntax and format rules.

The programmes are variable in length with addresses as per the ISO and EIA codes and standards.

Programming is possible in both codes:
- ISO (International Standards Organization) 6983-1 (NF Z 68-035), 6983-2 (NF Z 68 036) and 6983-3 (NF Z 68-037).
- EIA (Electronic Industries Association) Standards RS 244 A and 273 A.



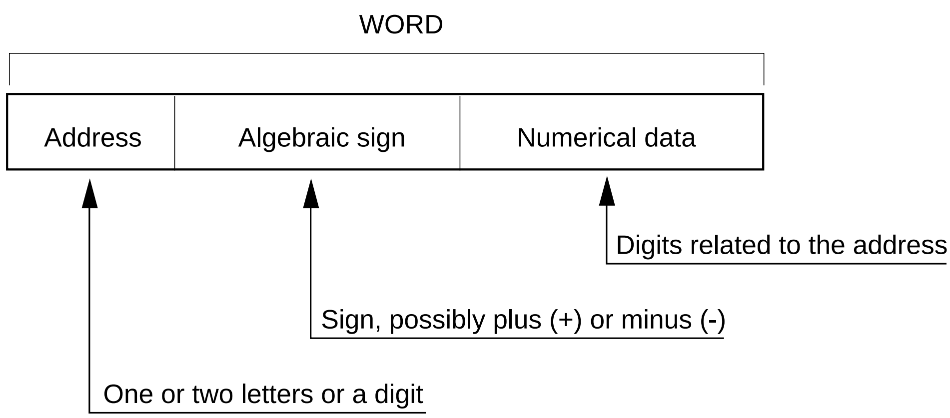
2.1 Word Format

A word contains an instruction or data to be transmitted to the control system.

Word types:
- words defining dimensions
- words defining functions.

The word format defines the specific characteristics of each code word used in programming (see table, Appendix C).

2.1.1 General Word Format



REMARK *For words defining a dimension, the decimal point is generally explicit. It separates the digits before and after the decimal point (it does not appear in the definition of the word format).*
The number of characters and spaces in a block must not exceed 118.

2.1.2 Special Features of the Dimension Word Format

The format of dimension words is determined by the choice of the internal system units specified by the OEM when integrating the CNC.

Internal system units are specified for:
- Linear axes
- Rotary axes.

The internal units directly affect the machine travels and the dimension acquisition and display formats for linear and rotary axes (modulo or not).

2.1.2.1 Internal System Unit for Linear Axes

The number of decimal digits available for programming the linear axes (where the basic unit is the mm) is declared in machine parameter P4, word N2 (see Parameter Manual).

Correspondence between the word format and internal unit for linear axes

Internal unit	Definition	Word format
0.1 mm	1 decimal digit	Format 071
0.01 mm	2 decimal digits	Format 062
µm	3 decimal digits	Format 053
0.1 µm	4 decimal digits	Format 044
0.01 µm	5 decimal digits	Format 035

2.1.2.2 Internal System Unit for Rotary Axes

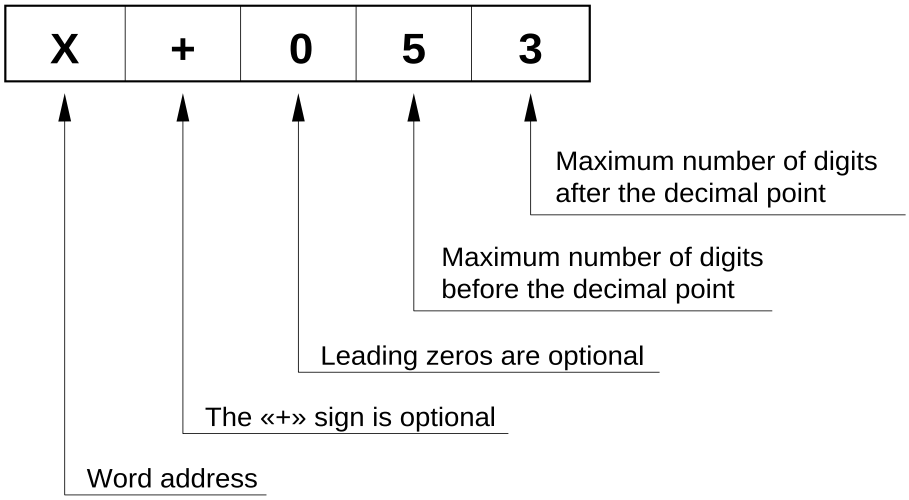
The number of decimal digits available for programming the rotary axes (for which the basic unit is the degree) is declared in machine parameter P4, word N4 (see Parameter Manual).

Correspondence between the word format and the internal system unit for rotary axes

Internal unit	Definition	Word format
0.1 degree	1 decimal digit	Format 031
0.01 degree	2 decimal digits	Format 032
0.001 degree	3 decimal digits	Format 033
0.0001 degree	4 decimal digits	Format 034

Examples of word formats:

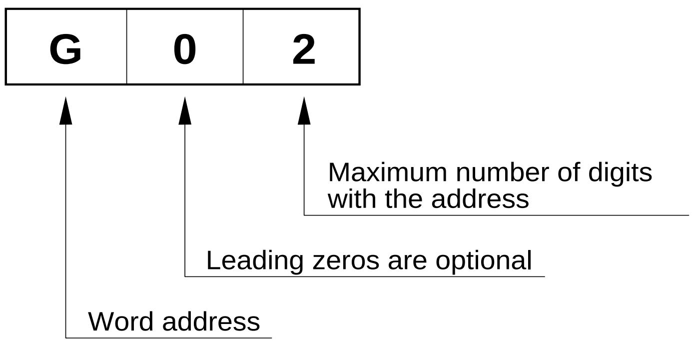
Word defining a dimension, address X (internal unit in µm)



The dimension 0.450 mm in X+053 format (variable word format), can be written:

X+0.450 or X.45

Word defining a function, address G



G function words in G02 format (variable word format).

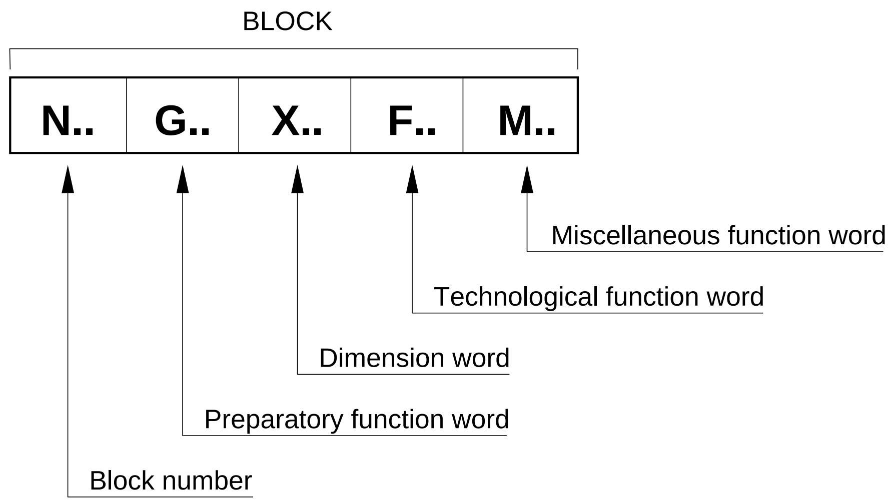
Word G01 can be written: G1

Word G04 can be written: G4

2.2 Block Format

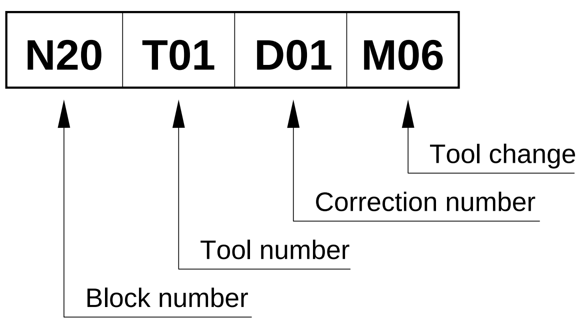
A block (or sequence) defines an instruction line of code words to be actioned by the control system.

The block format defines the syntax of the function and dimension words in each programming block.

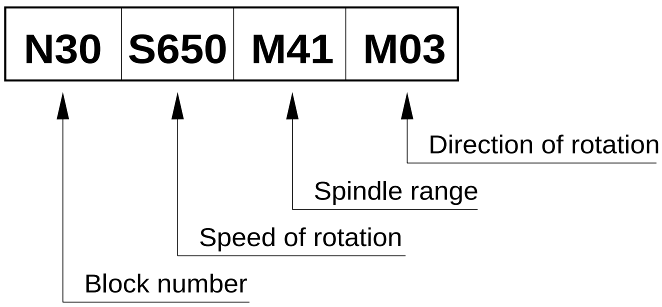


Examples of blocks

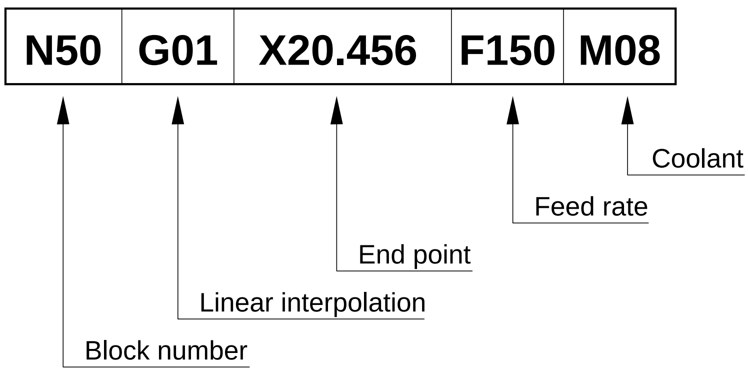
<u>A block defining a tool change and calling up the tool correction</u>



A block defining spindle rotation



A block defining a move



2.3 General Structure of a Programme

2.3.1 General

An NC programme must include start and end characters.

A programme is executed in the order in which the blocks are written between the programme start and end characters.

A programme is executed in the order in which the blocks are written, and not in the order of the block numbers. However, it is recommended to number the blocks in ascending order (in increments of ten, for instance).

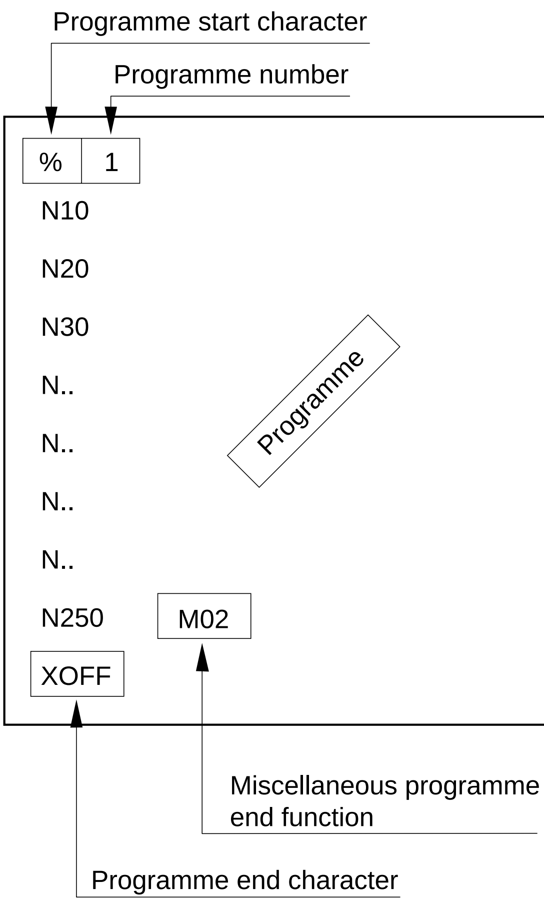
REMARK *A programme can be written in ISO code or EIA code. The ISO or EIA code is recognised by the system by reading the programme start character.*

Structure of an ISO Programme

Programme start: % character

Programme end: code M02

Programme end of load: XOFF character



Structure of an EIA programme

An EIA programme has the same structure as an ISO programme except for the programme start and end characters, which are different.

Programme start: EOR (End of Record) character

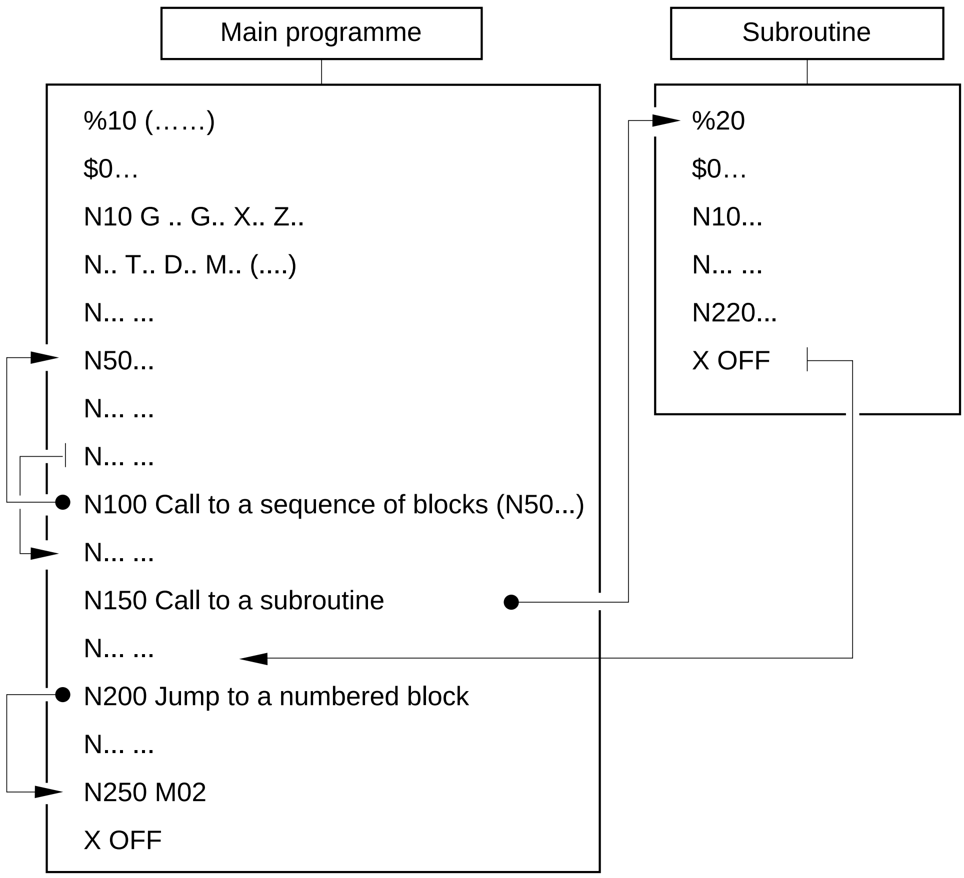
Programme end: BS (Back Space) character

REMARK *For an EIA programme, a programme end character other than BS can be declared by machine parameter P80 (see Parameter Manual).*

2.3.2 Branches and Subroutine Calls

Particular instructions (branches and subroutine calls) can modify the order in which a programme is executed.

A programme can be structured as follows:



2.3.3 Programme Numbering

Programme number: The permissible format is %051.

The % character is followed by a programme number and possibly by a comment in brackets.

Example:
```
%324 (PART No. 72 - PROG 3)
```

A programme number can be indexed (indices .1 to .8 with multiple axis group programming, see Sec. 4.15).

Example:
```
%425.2 (PROG FOR GROUP 2)
```

CAUTION

Programmes with numbers above %9000 are reserved for NUM and the OEM integrating the NC on the machine (check with NUM or the OEM for possible use of these numbers).

Programme Number and ISO Functions

When ISO functions are programmed after the programme (or subroutine) number on the same line, they are ignored.

Example:

%99 G1 X80	Movement G1 X80 is ignored

Programme Load from a Peripheral

When loading a programme from a peripheral, if the programme number does not comply with format %051, the excess digits are ignored.

Example:

%1234567.89 (comment)	Programme number received over the line
%12345 .8 (comment)	Number actually stored

Inhibiting display of subroutines being executed

Display on the programme page (PROG) of a subroutine and its internal subroutines during execution can be inhibited.

Placing the character «:» after the subroutine number (e.g. %110:) inhibits display. Only the subroutine call block is then displayed (for additional information, see Sec. 4.11.1).

2.3.4 Characteristics of the ISO and EIA Codes

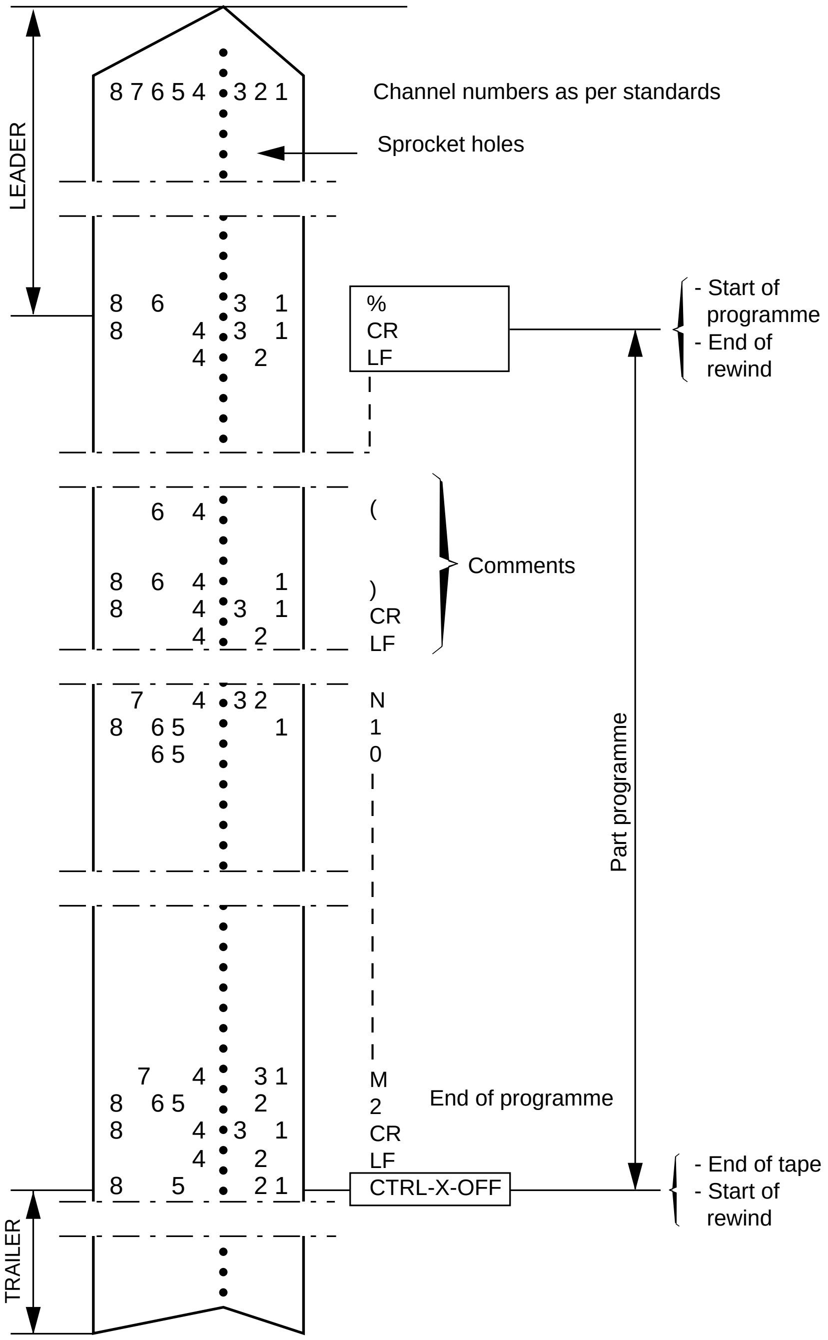
List of characters recognised by the system in ISO and EIA codes:

DESCRIPTION	ISO	EIA
10 digits	0-9	0-9
Letters of the alphabet	A-Z	A-Z
Programme start	%	EOR
Start of comment	(	,
End of comment	)	%
Plus sign	+	+
Minus sign	-	-
Decimal point	.	.
Greater than	>	
Less than	<	
Multiplied by	*	
Equal to	=	
Divided by	/	
At sign	@	
End of block	LF	CR
Skip block	/	/
Programme subdivision	:	letter O
Programme end	X OFF	BS

List of characters recognised by the system with no action on the machine:

DESCRIPTION	ISO	EIA
Tab	HT	TAB
Carriage return	CR	
Space	SP	SP
Error	DEL	DEL
	RUB OUT	RUB OUT

Structure of an ISO programme tape:



List of characters used in ISO code:

ISO CODE										
	Channel No.	8	7	6	5	4		3	2	1
Function	Charac-ter	Tape punch code								
Programme start, rewind stop	%	●		●			•	●		●
Plus sign	+			●		●	•		●	●
Minus sign	-			●		●	•	●		●
	0			●	●		•			
	1	●		●	●		•			●
	2	●		●	●		•		●	
	3			●	●		•		●	●
Digits	4	●		●	●		•	●		
	5			●	●		•	●		●
	6			●	●		•	●	●	
	7	●		●	●		•	●	●	●
	8	●		●	●	●	•			
	9			●	●	●	•			●
Angular direction about X axis	A		●				•			●
Angular direction about Y axis	B		●				•		●	
Angular direction about Z axis	C	●	●				•		●	●
Tool correction	D		●				•	●		
Peripheral parameter	E	●	●				•	●		●
Feed rate. Dwell	F	●	●				•	●	●	
Preparatory function	G		●				•	●	●	●
Subroutine No.	H		●			●	•			
Interpolation address	I	●	●			●	•			●
Interpolation address	J	●	●			●	•		●	
Interpolation address	K		●			●	•		●	●
Programmer parameter No.	L	●	●			●	•	●		
Miscellaneous function	M		●			●	•	●		●
Sequence number	N		●			●	•	●	●	
	O	●	●			●	•	●	●	●
	P		●		●		•			
Miscellaneous parameters	Q	●	●		●		•			●
	R	●	●		●		•		●	
Spindle speed function	S		●		●		•		●	●
Tool No.	T	●	●		●		•	●		
Secondary dimension parallel to X axis	U		●		●		•	●		●
Secondary dimension parallel to Y axis	V		●		●		•	●	●	
Secondary dimension parallel to Z axis	W	●	●		●		•	●	●	●
Primary X dimension	X	●	●		●	●	•			
Primary Y dimension	Y		●		●	●	•			●
Primary Z dimension	Z		●		●	●	•		●	
Programme subdivision	:			●	●	●	•		●	
Optional block skip	/	●		●		●	•	●	●	●
Carriage return	CR	●				●	•	●		●
End of block/line feed	LF					●	•		●	
Start of comment	(			●		●	•			
End of comment	)	●		●		●	•			●
Space	SP	●		●			•			
End of tape	X OFF	●			●		•		●	●
Horizontal tab	HT					●	•			●
Delete	DEL	●	●	●	●	●	•	●	●	●
No punch	NUL						•			

List of characters used in EIA code (RS.244.B):

EIA CODE										
Channel No.		8	7	6	5	4		3	2	1
Function	Charac-ter	Tape punch code								
Programme start, rewind stop	EOR					●	•		●	●
Plus sign	+		●	●	●		•			
Minus sign	-		●				•			
	0			●			•			
	1						•			●
	2						•		●	
	3				●		•		●	●
Digits	4						•	●		
	5				●		•	●		●
	6				●		•	●	●	
	7						•	●	●	●
	8					●	•			
	9				●	●	•			●
Angular direction about X axis	a		●	●			•			●
Angular direction about Y axis	b		●	●			•		●	
Angular direction about Z axis	c		●	●	●		•		●	●
Tool correction	d		●	●			•	●		
Peripheral parameter	e		●	●	●		•	●		●
Feed rate. Dwell	f		●	●	●		•	●	●	
Preparatory function	g		●	●			•	●	●	●
Subroutine No.	h		●	●		●	•			
Interpolation address	i		●	●	●	●	•			●
Interpolation address	j		●		●		•			●
Interpolation address	k		●		●		•		●	
Programmer parameter No.	l		●				•		●	●
Miscellaneous function	m		●		●		•	●		
Sequence number	n		●				•	●		●
	o		●				•	●	●	
	p		●		●		•	●	●	●
Miscellaneous parameters	q		●		●	●	•			
	r		●			●	•			●
Spindle speed function	s			●	●		•		●	
Tool No.	t			●			•		●	●
Secondary dimension parallel to X axis	u			●	●		•	●		
Secondary dimension parallel to Y axis	v			●			•	●		●
Secondary dimension parallel to Z axis	w			●			•	●	●	
Primary X dimension	x			●	●		•	●	●	●
Primary Y dimension	y			●	●	●	•			
Primary Z dimension	z			●		●	•			●
Programme subdivision	o		●				•	●	●	
Optional block skip	/			●	●		•			●
Carriage return							•			
End of block/line feed	EOB	●					•			
Start of comment	?			●	●	●	•		●	●
End of comment	%		●		●	●	•		●	●
Space	SP				●		•			
End of tape	BS			●		●	•		●	
Horizontal tab	TAB			●	●	●	•	●	●	
Delete	DEL		●	●	●	●	•	●	●	●
No punch	NUL						•			

Special ISO code characters:

Special characters

Channel numbers		8	7	6	5	4		3	2	1
Description	Charac-ter	Holes punched								
Less than	<			●	●	●	•	●		
Greater than	>	●		●	●	●	•	●	●	
Multiplied by	*	●		●		●	•		●	
Equal to	=	●		●	●	●	•	●		●
Divided by or block skip	/	●		●		●	•	●	●	●
At sign	@	●	●				•			
AND	&	●		●			•	●	●	
OR	!			●			•			●
Dollar sign	$			●			•	●		
Comma	,	●		●		●	•	●		
Period	.			●		●	•	●	●	
Single quote	'			●			•	●	●	●
Semicolon	;	●		●	●	●	•		●	●
Pound sign	#	●		●			•		●	●
Question mark	?			●	●	●	•	●	●	●
Double quote	"			●			•		●	

The «$» character is used in a programme to send messages (see Sec. 4.19).

Most of the other characters are mainly used for parametric programming (see Chapter 6).

Special characters of the EIA code:

As comments were not provided for by the EIA code, the characters «,» et «%» are used and have the same meaning as round brackets «( )» in ISO code.

As there is no equivalence in EIA code for ISO characters «>», «<«, «*», «=» and «@», parametric programming, tool data entry and tape punching are prohibited in this code.

The absence of a character on an EIA tape is reported as a parity error.

2.4 Classification of Preparatory G Functions and Miscellaneous M Functions

2.4.1 Classification of Preparatory G Functions

Types of G functions:
- Modal G functions,
- Nonmodal G functions.

Certain G functions must be programmed with the associated arguments.

Programming of certain G functions may be incompatible with the state of the current programme.

2.4.1.1 Modal G Functions

Functions belonging to a family of G functions that cancel one another.

Certain families of G functions include a default function that is initialised when power is applied (see A.1).

These functions remain enabled until cancelled by another function of the same family.

Example:

`N.. G00 X.. Z..`	High-speed linear interpolation
`N.. G01 Z..`	G00 cancelled by linear interpolation at machining feed rate

2.4.1.2 Nonmodal G Functions

Functions enabled only in the block where they are programmed (cancelled at the end of the block).

Example:

`N.. G09 X..`	Accurate stop at end of block cancelled at end of block.

2.4.1.3 G Functions Incompatible with the State of the Programme

Functions whose programming is enabled or not according to the state of the current programme.

Example:

`N.. G21 G42 X.. Y.. Z..`	Syntax correct, change of X Y Z coordinate system (G21), followed by radius offset (G42)
`N..`	
`N.. G42 G21 X.. Y.. Z..`	Syntax incorrect, change of coordinate system prohibited with radius offset

2.4.1.4 G Functions Associated with Arguments

Functions followed by one or more arguments that are specific to the G function announcing them.

The argument(s) must immediately follow the function.

The analysis of the arguments of a G function is ended by reading a word that does not belong to the list of arguments of this function.

Example:

`N.. G04 F2 T03 F0.2`	Syntax correct
`N.. G04 T03 F2 F0.2`	Syntax incorrect, argument F2 does not immediately follow G04

When a G functions has several arguments, they can be programmed in any order except for G functions that introduce breaks in the sequencing (G10, G76, G77 and G79, see Sec. 4.11).

The arguments associated with a function can be:
- compulsory,
- optional.

The argument of certain G functions can be programmed alone in a block.

Compulsory Arguments

The arguments are compulsory if:
- the G function serves only to announce arguments.

Example:

`N.. G16 P+`	G function and its argument P+

- the G function cancels a former modal state and characterises its argument differently.

Example:

`N.. G94 F100`	Feed in mm/min
`N..`	
`N.. G95 F0.5`	The change from feed in mm/min to mm/revolution requires redefining argument F

Optional Arguments

The arguments are optional if the G function allows them to be defined by default.

Example:

`N.. G96 [X..] S150`	Case where the X position (with respect to OP) was specified by an earlier block

Arguments Programmed Alone

The argument can be programmed alone in a block when the associated G function is still active.

Example:

`N.. G94 F150 X.. Z..`	Feed in mm/min
`N..`	
`N.. X.. Z.. F100`	Function G94 is not compulsory with its argument because the system is still in state G94.

2.4.2 Classification of Miscellaneous M Functions

Type of M functions:
- Modal M functions,
- Nonmodal M functions.

M functions can be:
- «pre» or «post» functions,
- encoded or decoded functions.

2.4.2.1 Modal M Functions

Functions belonging to a family of M functions that cancel one another.

Certain families of M functions include a default function that is initialised when power is applied (see A.2).

These functions remain enabled until they are cancelled by another function of the same family.

Example:

`N.. S500 M03`	Start of spindle rotation
`N.. M05`	Spindle stop, cancels M03

2.4.2.2 Nonmodal M Functions

Enabled only in the block where they are programmed.

Example:

`N.. M00`	Programme stop

2.4.2.3 «Pre» M Functions

Functions executed before axis movements programmed in the block.

Example:

`N.. X100 Z50 M08`	Coolant function M08 is executed before the movements on X and Z

2.4.2.4 «Post» M Functions

Functions executed after the axis movements programmed in the block.

Example:

`N.. X50 Z100 M09`	The coolant off function (M09) is executed after movements on X and Z

2.4.2.5 Encoded M Functions

The encoded functions are defined by the machine manufacturer and are specific to the machine (see manufacturer's technical data).

Encoded Functions M100 to M199

These functions with PLC handshake are generally nonmodal «post» functions, but these features can be redefined by the machine manufacturer.

Only one of these functions is allowed in a part programme block.

Encoded Functions M200 to M899

These so-called on-the-fly functions are modal «pre» functions. The programme continues without waiting for the execution report.

Only one of these functions is allowed in a part programme block.

REMARK *An encoded nonmodal function (M100 to M199) can be programmed in the same block with an encoded modal function (M200 to M899).*

2.4.2.6 Decoded M Functions

The decoded M functions are the basic system functions whose meaning is known.

REMARK *All these functions are acknowledged by a PLC handshake (CRM). The acknowledgement enables continuation of the part programme.*

Example:

`N.. T01 M06`	Tool change function M06

Several decoded M functions can be programmed in the same block.

Example:

```
N.. G97 S500 M03 M40 M08
```

3 Axis Programming

3.1 General

Programmable axes:
- Primary axes X, (Y), Z,
- Secondary axes U, (V), W,
- Rotary axes (A), (B), C.

Primary and secondary axes:
- they can be independent or form carrier/carried axis pairs (see machine parameter P64),
- they can be programmed in millimetres (basic unit) or inches.

Rotary axes:
- They can be modulo 360 degrees or have limited travel or be declared as non-rotary (see machine parameter P1),
- They are programmed in degrees (basic unit).

Reminder

Definition of the Internal System Measurement Units

The internal measurement unit is defined by the OEM when integrating the CNC. It directly affects the machine travels on the linear axes and rotary axes (modulo or not).

The number of decimal digits is declared in machine parameter P4 and determines the word formats (see Sec. 2.1 and Appendix C).

For linear axes, the internal unit can be 0.1 mm, 0.01 mm, µm, 0.1 µm or 0.01 µm.

For rotary axes, the internal unit can be 0.1 degree, 0.01 degree, 0.001 degree or 0.0001 degree.

REMARK *For ISO functions and programming arguments defining angular values (EA.., EC.., ED.., etc.), the unit is always 0.0001 degree.*

For additional information, refer to:
- The machine manufacturer's manual
- The Parameter Manual.

3.2 Programming the Independent Secondary Axes

Programming the independent secondary axes U, (V), W is unrelated to the programming of the primary axes X, (Y), Z.

For a primary axis, the machine dimension is expressed:

Mx (machine dimension) = Px (programmed dimension) + xDAT1 + xDAT2 + tool offset x

In the above example, x is the primary axis X (the equation is the same for the Y and Z axes).

For an independent secondary axis, the same machine dimension is expressed:

Mu (machine dimension) = Pu (programmed dimension) + uDAT1 + uDAT2

In the above example, u is the independent secondary axis U (the equation is the same for the (V) and W axes).

It should be noted that the tool length correction is not applied to the independent secondary axes.

3

3.3 Programming Carrier/Carried Parallel Axis Pairs

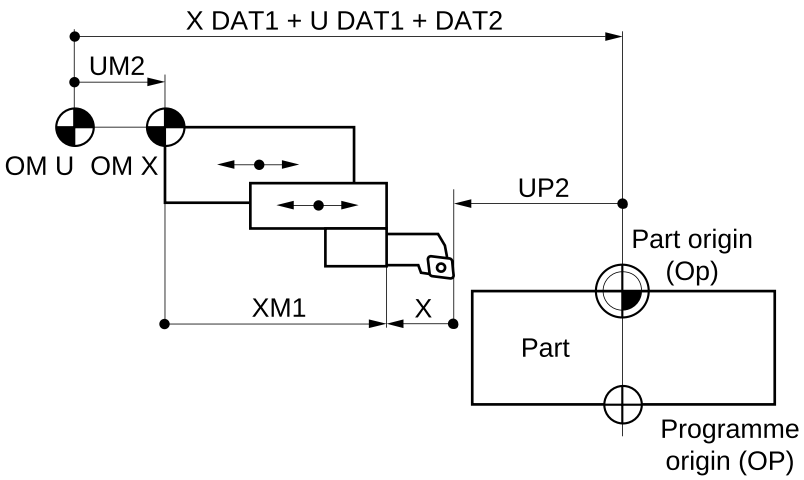
Movement of the axis pair with respect to the part.

The U axis is a large slide and the X axis is a small slide.

Representation of large slide approach programmed by UP2

Calculation of UM2, knowing that XP2 = UP2.

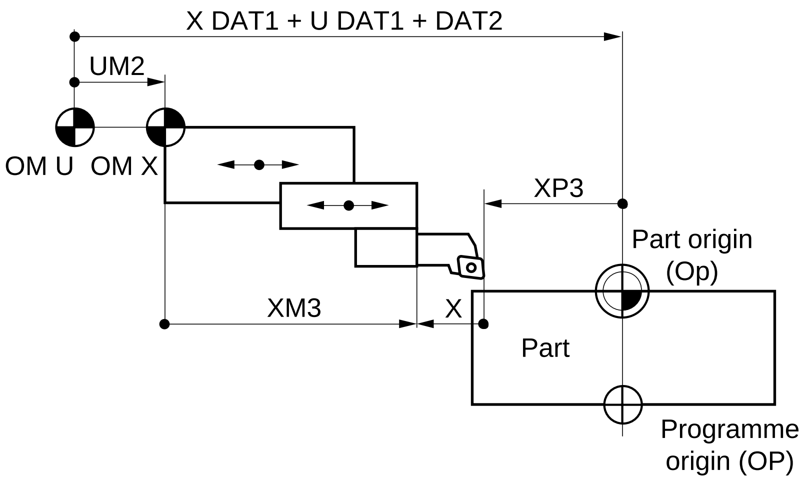
UM2 = UP2 = (DAT1 + DAT2 + X dimension) - XM1



Representation of the small slide approach programmed by XP3

Calculation of XM3, knowing that XP3 = UP3.

XM3 = XP3 = (DAT1 + DAT2 + X dimension) - UM2



3.4 Programming of Rotary Axes Modulo 360 Degrees

Rotary axis C programmed in absolute dimensions (G90)

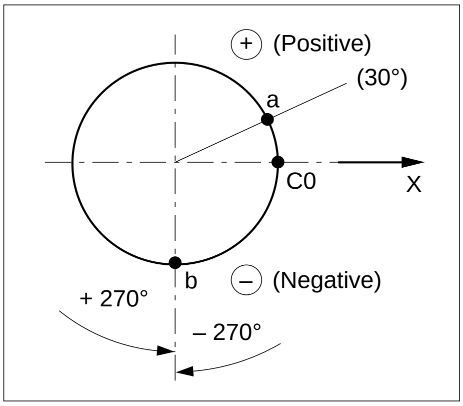
The angular value assigned to the axis is the position of the end point with reference to the programme origin, value between 0 and 360 degrees, maximum one revolution (see Sec. 4.1 for function G90).

The sign (+ or -) determines the direction of rotation to reach this point.

Example:

a: Start point

b: End point



Positive rotation

```
N.. ...
N.. G90 C+270
N..
```

Negative rotation

```
N..
N.. G90 C-270
N..
```

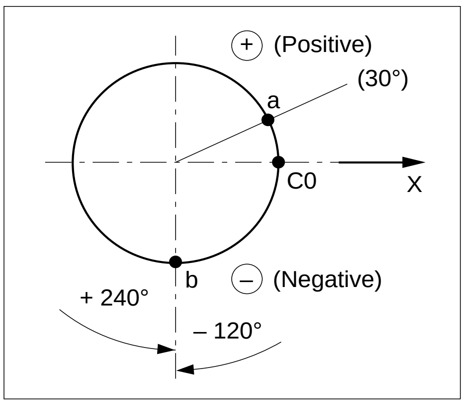
Rotary axis C programmed in incremental dimensions (G91)

The value assigned to the axis indicates the amplitude of rotation of the axis with reference to the previous position (see Sec. 4.1 for function G91).

Example:

a: Start point

b: End point



Positive rotation

```
N.. ...
N.. G91 C+240
N..
```

Negative rotation

```
N..
N.. G91 C-120
N..
```

REMARK *With incremental programming G91 (see Sec. 4.1 for function G91), a movement of more than one revolution is allowed on modulo rotary axes A, B or C. It should be noted that a maximum of 15 revolutions are allowed. If this value is exceeded, the system returns error message 1.*

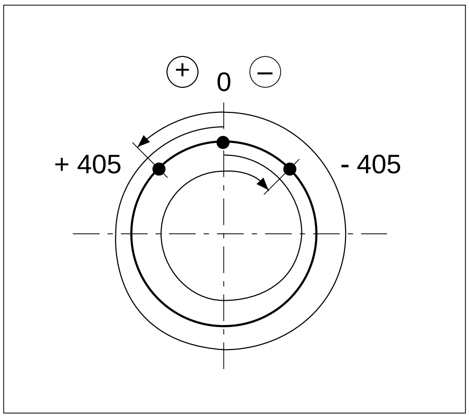
3.5 Programming of Slaved Rotary Axes with Limited Travel

Servoed rotary axes A, B or C with limited travel are defined by machine parameters like linear axes and therefore follow the same programming rules.

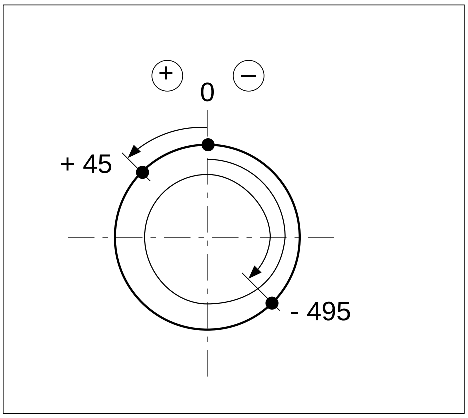
This definition of a rotary axis can be used for axes with more than 360 degrees of travel to be rotated by more than one revolution with respect to a preferential position.

Example:

Rotation greater than one revolution in absolute dimensions (G90).



Rotation greater than one revolution in incremental dimensions (G91).



3.6 Programming of Axes A, B or C Declared as Nonrotary

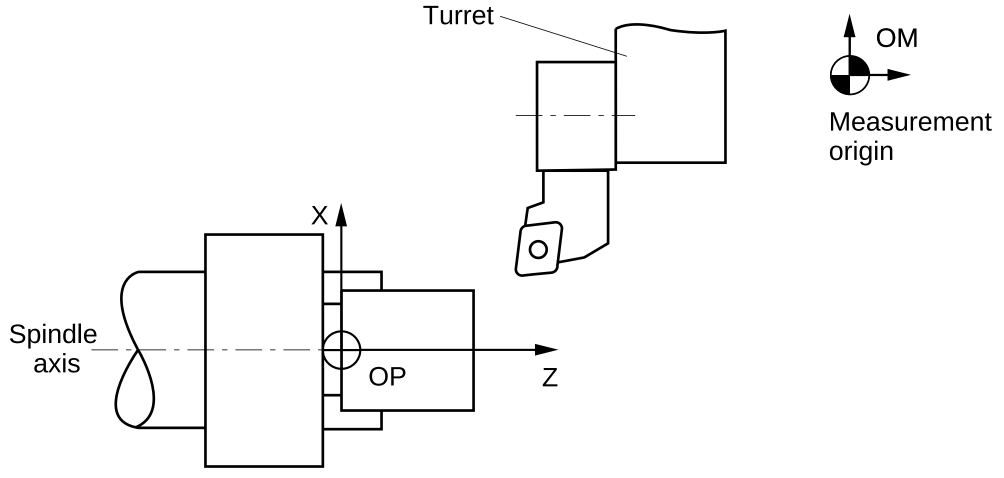
When axes A, (B) or C are declared as nonrotary (see machine parameter P1), they are considered as linear axes (in particular in keyboard Homing mode and Shift mode).

The speed of movement on axes A, B or C declared as nonrotary is expressed in mm/min. However, if they are programmed in a block together with primary and secondary axes X, (Y), Z, U, (V) or W, the programmed speed is assigned to the latter.

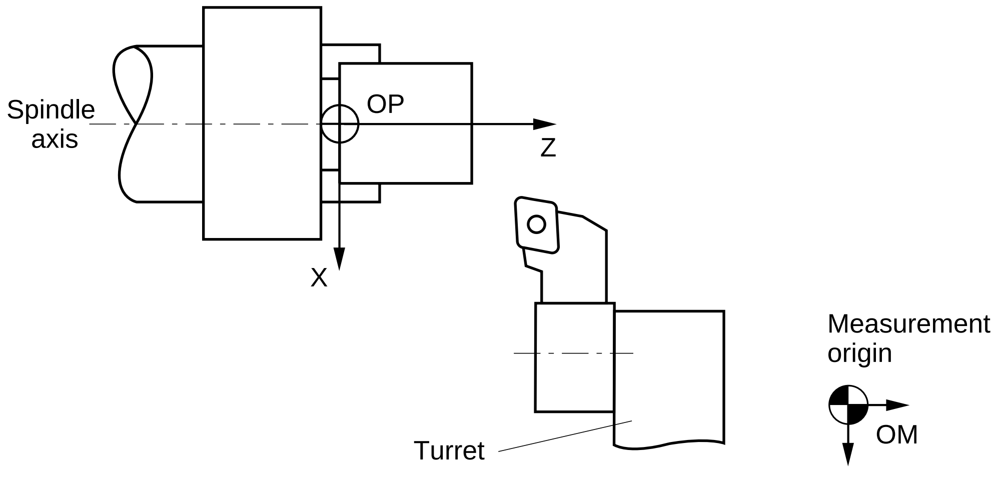
3.7 Features of Front Turret, Rear Turret

The front or rear position of the main turret defines the positive orientation of the X axis.

Lathe with rear turret



Lathe with front turret



4 ISO Programming

4

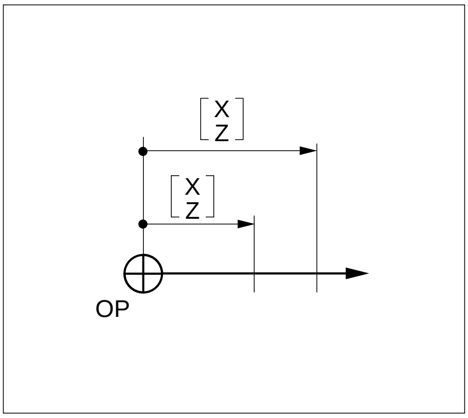
	4.15.13	Programmed Acceleration Reduction	4 - 258
	4.15.14	Scaling Factor	4 - 259
	4.15.15	Mirror Function	4 - 261
	4.15.16	Processing of Blocks and Programmed G and M Functions	4 - 264
4.16 «Inclined Axis» or «Inclined Wheel» State on a Grinder			4 - 267
	4.16.1	«Inclined Axis» State	4 - 267
	4.16.1.1	Initial Tool Positioning Before Machining on an Inclined Axis	4 - 268
	4.16.1.2	Movement Along the Inclined Axis	4 - 269
	4.16.2	«Inclined Wheel» State	4 - 271
4.17 Special Programming for Multiple Axis Groups			4 - 273
	4.17.1	Programme Declaration	4 - 273
	4.17.2	Programming Notes	4 - 273
	4.17.3	Subroutine branches for Multi-Axis Groups	4 - 275
	4.17.3.1	Branch to Automatic Homing Subroutine	4 - 275
	4.17.3.2	Subroutine Call by a Reset	4 - 275
	4.17.3.3	Subroutine Call by the Automatic Control Function	4 - 276
	4.17.3.4	Subroutine Call by M Function	4 - 276
	4.17.4	Spindle Programming	4 - 277
	4.17.5	Current Spindle Release by an Axis Group	4 - 278
	4.17.6	Axis Group Synchronisation	4 - 279
4.18 Special Programming of PLC Axes			4 - 283
	4.18.1	Programme Declaration and Storage	4 - 283
	4.18.2	Programming of the PLC Axes	4 - 285
	4.18.2.1	Emergency Retraction on a PLC Axis Group	4 - 285
	4.18.3	Editing the Programmes	4 - 286
	4.18.4	Exchanging Axes between Groups	4 - 286
	4.18.5	Exchanging Spindles Between Groups	4 - 287
4.19 Message Transmission			4 - 288
	4.19.1	Message Transmission to the Display	4 - 288
	4.19.2	Transmission to Automatic Control Function or Remote Server or Peripheral or PC	4 - 290
4.20 Spindle Synchronisation			4 - 293
	4.20.1	Spindle Acceleration Control	4 - 293
	4.20.2	Servo-Controlled Spindles and Synchronised Spindles	4 - 294
	4.20.2.1	Servo-Controlled Spindles	4 - 294
	4.20.2.2	Synchronised Spindles	4 - 295

4.1 Choice of the Programming System

4.1.1 Programming by Absolute or Incremental Dimensions

G90	Absolute dimensions with respect to the programme origin.

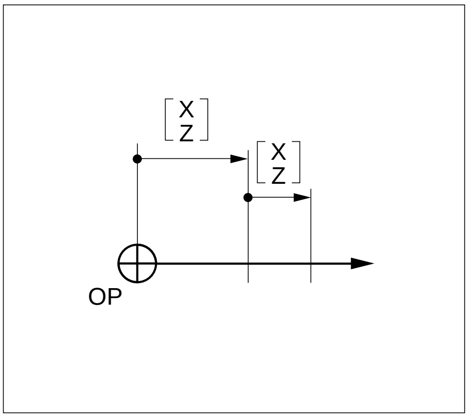
The value programmed on an axis is with reference to the programme origin (OP).



G91	Incremental dimensions.

The value programmed on an axis is with reference to the last programmed position.

The value is equal to the movement to be performed.



Syntax

N.. **G90/G91** X.. Z.. C..

G90	Absolute dimensions.
G91	Incremental dimensions.
X.. Z.. C..	End point.

Properties of the Functions

Functions G90 and G91 are modal.

G90 is the default function.

Cancellation

Functions G90 and G91 cancel one another.

Notes

The first movement must be programmed:
- in absolute dimensions (G90),
- by manual data entry (MDI) or in a programme with respect to the programme origin (OP) instead of with respect to the current position.

Incremental programming (G91) is prohibited for PGP (Profile Geometry Programming, see Chapter 5).

Combined programming

Both types of programming (G90/G91) can be included in a programme and even in a block. For instance:

```
N..
N.. G91 X.. Z..
N.. G90 X.. G91 Z..                X absolute, Z incremental
N.. G90 X.. Z..
N..
```

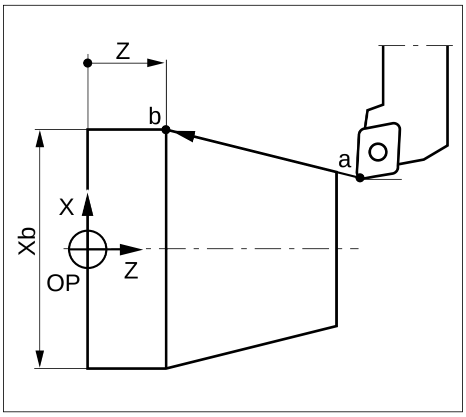
Examples

Absolute programming (G90), (System programmed on radius)

Tool located at point a (starting point)

Absolute programming of point b (coordinates of the end point).

```
N.. (G90)...
N.. Xa Ya
N.. Xb Yb
N..
```

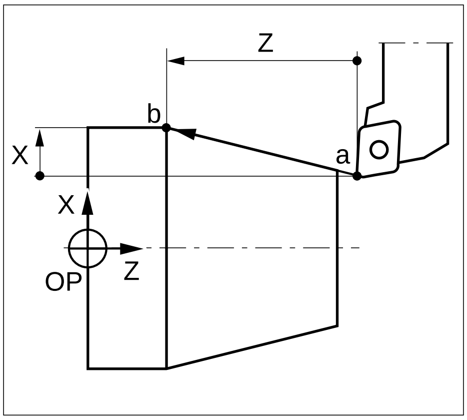


Incremental programming (G91), (System programmed on radius)

Tool located at point a (starting point)

Incremental programming of point b (amount of movement to reach end point b).

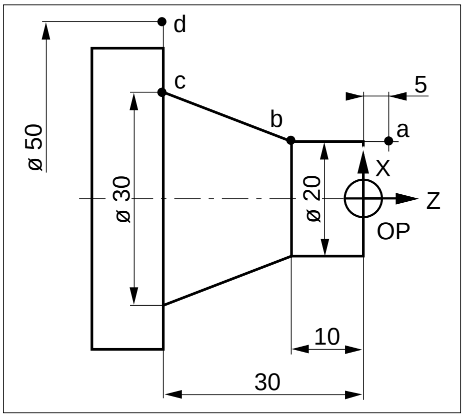
```
N.. (G90) ...
N.. Xa Za
N.. G91 Xb Zb
N..
```



Absolute programming (G90), (System programmed on diameter)

Coordinates of points a, b, c, d, with respect to the programme origin (OP).

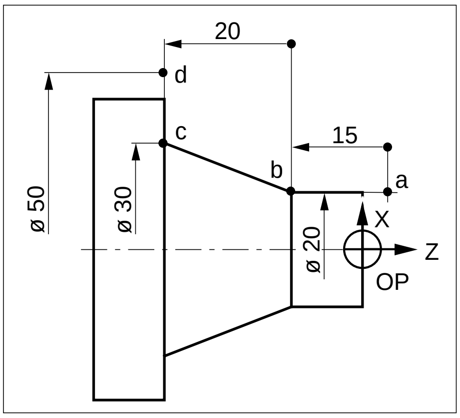
```
N.. (G90) ...
N.. X20 Z5
N.. Z-10
N.. X30 Z-30
N.. X50
N..
```



Incremental programming (G91), (System programmed on diameter)

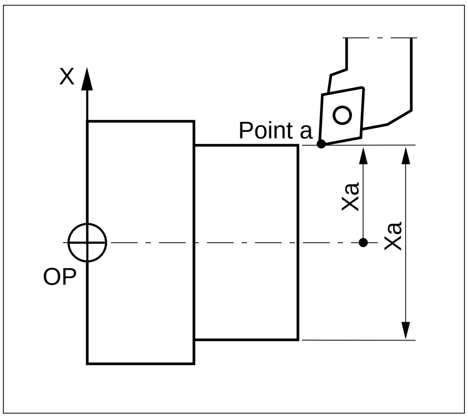
Absolute programming of point a, incremental movement to points b, c, d.

```
N.. (G90) ...
N.. X20 Z5
N.. G91 Z-15
N.. X5 Z-20
N.. X10
N.. G90
N..
```



4.2 Programming with Reference to Diameter or Radius

The part programme and certain data related to machining along the X (or U) axis are directly affected by whether programming is with reference to diameter or radius.



Programming of the system with reference to the diameter or radius is selected by machine parameter P4 (see parameter manual).

In both cases, certain functions are always expressed with respect to the diameter and others with respect to the radius.

System Programmed with Reference to Diameter

Programmed values expressed with reference to diameter:
- values programmed in absolute dimensions (G90): coordinates of a movement along X.. and position I.. of the centre of a circle,
- value of the starting diameter with constant surface speed (G96),
- value programmed with function G98.

Programmed values expressed with reference to radius:
- values programmed in incremental dimensions (G91): value of a movement along X.. and position I.. of the centre of a circle,
- circle radius with circular interpolation (R),
- fillet or chamfer (EB+, EB-),
- pass depth for roughing cycle (P or R),
- machining allowance for roughing cycles (I or K), (ER),
- positioning clearance for roughing cycle (Q),
- minimum depth of cut for roughing cycle (EQ),
- thread depth (P) and last pass for thread cutting (Q),
- pass depth for drilling cycle (P and Q),
- offsets programmed with functions G59 and G52.

Tool dimensions:
- Entry of values with respect to radius.

Dynamic tool corrections:
- Entry of values with respect to diameter, but display of the radius changes on the «DYNAMIC TOOL CORRECTIONS» page.

Value of offset DAT2:
- Entry of the value with reference to radius.

Movements related to the manual controls:
- Movements on the X axis with reference to radius, but display with reference to diameter on the «AXES» page.

System Programmed with Reference to Radius

Values expressed with reference to radius:
- All the programmed movements applied to the X axis and all the values entered related to machining along X.

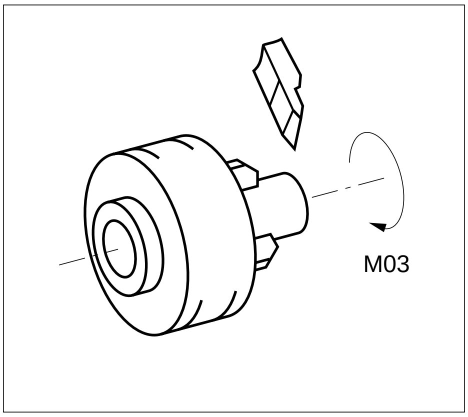
Values expressed with reference to diameter:
- Only the dynamic tool corrections on X are entered with reference to diameter.

4.3 Spindle Commands

4.3.1 Notes on Axis Programming

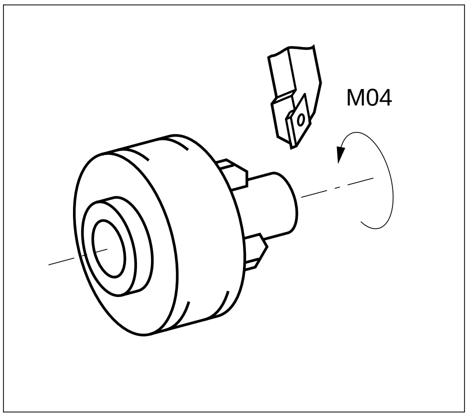
M03	Spindle clockwise rotation.

This command starts spindle rotation at the speed programmed.



M04	Spindle counterclockwise rotation.

This command starts spindle rotation at the speed programmed.



M05	Spindle off.

This command stops spindle rotation.

Syntax

N.. **M03/M04/M05**

M03	Spindle clockwise rotation.
M04	Spindle counterclockwise rotation.
M05	Spindle off.

Properties of the Functions

Functions M03 and M04 are decoded modal «pre» functions.

Function M05 is a decoded modal «post» function. It is the default function.

Cancellation

Functions M03, M04 and M05 cancel one another.

Functions M00, M19 and M01 (enabled) cancel functions M03 or M04.

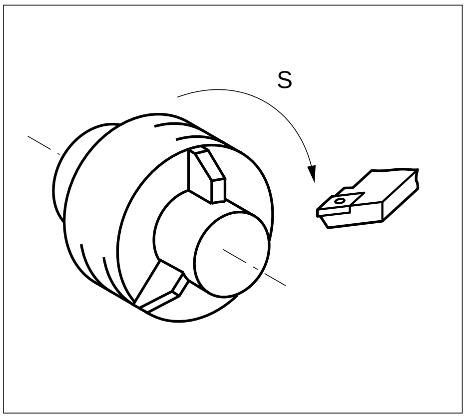
Example

```
N.. ...
N120 ...                    Tool call
N130 M03 ...                Spindle clockwise rotation
N..
N..
N220 M05 ...                Spindle off
N..
```

4.3.2 Spindle Speed Control

G97	RPM spindle speed.

This function defines a fixed spindle speed programmed with the S word.



4

Syntax

N.. **G97** S.. [M03/M04]

G97	Function setting the spindle speed in rpm.
S..	Mandatory argument associated with the function to define the speed.
M03/M04	Spindle direction of rotation.

Properties of the Functions

Function G97 is modal. It is the default function.

Cancellation

Function G97 is cancelled by function G96 S.. (constant surface speed).

The spindle speed programmed by G97 is cancelled by S0 or modified by programming a new value of S..

Notes

Spindle speed format

The format of rpm rotation may differ according to the type of machine:
- Format S05 (1 to 65000 rpm).
- Format S032 (0.01 to 650 rpm).

Example

```
N.. ...
N130 G97 S636 M04                    Spindle rotation
N..
```

Reminder

The spindle rotation speed N is determined from the required cutting speed (V).

The cutting speed V in meters per minute is mainly related to:
- the tool material,
- the part material.

Cutting speed V = 100 m/min.
Tool diameter D = 50 mm.

$$N\ (\text{rpm}) = \frac{1000 \text{ x } V}{3.14 \text{ x } D}$$

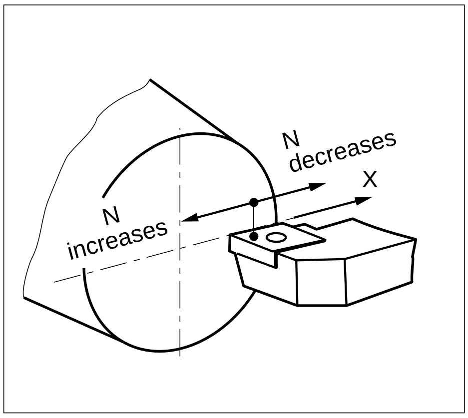
$$N = \frac{1000 \text{ X } 100}{3.14 \text{ X } 50}$$

N = 636.9 rpm, i.e. S636

4.3.2.1 Constant Surface Speed

G96	Constant surface speed is expressed in meters per minute.

This function varies the spindle rotation speed (N). The rotation speed varies according to the position of the tool centre with reference to the part diameter.



Syntax

N.. **G96** [X..] S..

G96	Function forcing a constant surface speed in m/min.
X..	Argument defining the current diameter.
S..	Mandatory argument associated with the function to define the programmed speed.

Property of the Function

Function G96 is modal.

Cancellation

Function G96 is cancelled by function G97 S..

Notes

This function can only be programmed when the machine is equipped with a variable speed spindle.

The machine spindle must be rotating when the function is called.

After a retraction with reference to the measurement origin (G52 X..), a new constant surface speed initialisation position in X (or U) must be reprogrammed.

When a constant surface speed is programmed:

- the X (or U) axis defining computation of the constant surface speed can be programmed in the same block or one of the previous blocks (between G52 and G96). If X is missing, the system returns error message 28.
- the X (or U) axis must be programmed with reference to the programme origin,
- the speed applies to the centre of the tool insert radius,
- it can be modified during the programme by redefining a new speed by G96 S..,
- enabling of offsets DAT1 and DAT2 does not affect the surface speed.

During machining with constant surface speed, it is recommended to:

- programme the feed rate in mm/revolution in order to perform machining with a constant chip thickness (see Sec. 4.7),
- cancel the constant surface speed by programming a rotation speed in rpm (G97 S..) before each tool change, then initialise the constant surface speed on the X position of the new tool (and D correction).

Review of surface speed

General equation for surface speed V:

V = 3.14 x D x N

Spindle rotation speed in rpm

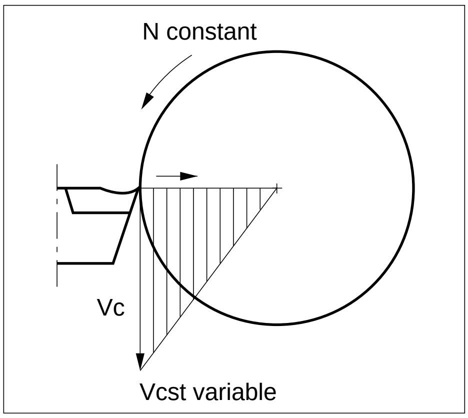
Part diameter at the tool contact point

Surface speed in meters per minute

Using a fixed spindle rotation speed (N)

The fixed spindle speed (N) is expressed in revolutions per minute.

The surface speed V decreases as the tool moves toward the centre of the part, (the surface speed is zero at the centre).

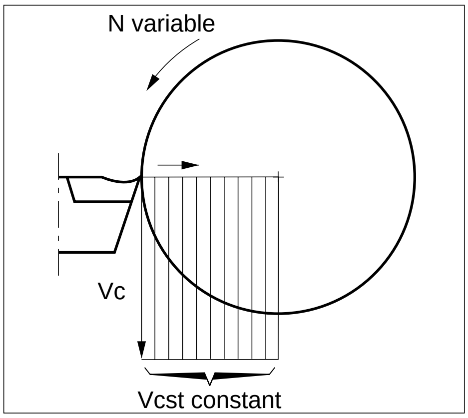


Using a constant surface speed (V)

The constant surface speed (Vcst) is expressed in meters per minute.

The surface speed is kept constant as the tool moves toward the centre of the part.

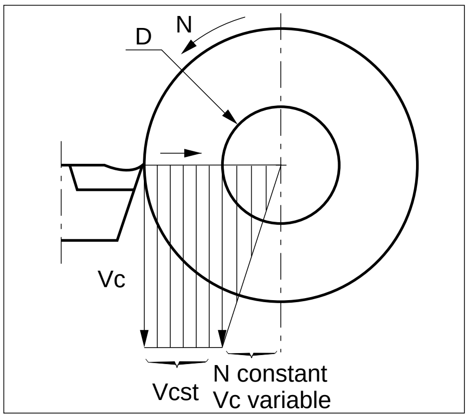
When the tool tip is at the centre, diameter D is equal to zero and speed N should be infinite (physically impossible because of the maximum physical rotation speed of the spindle).



Spindle speed limit (see Sec. 4.3.7)

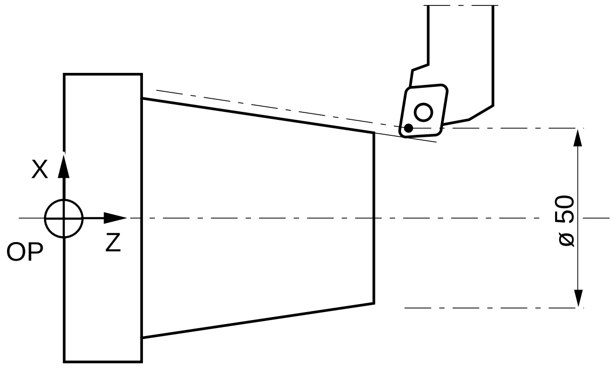
A safety limit can be defined by programming a maximum speed in revolutions per minute.

The limit defines a diameter D beyond which a constant surface speed no longer applies. The system returns to the case of a constant rotation speed and a variable surface speed.



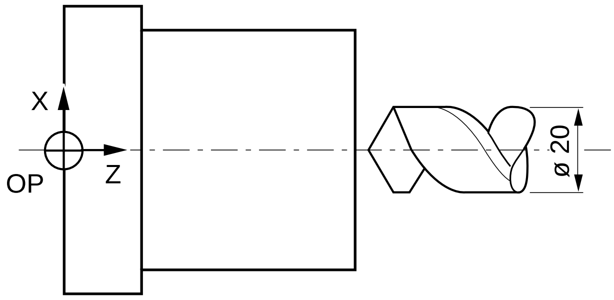
Examples

Programming a surface speed of 200 m/min for finishing a profile



`N.. ... (CARBIDE TOOL R=0.8)`	Tool change and correction
`N130 G97 S900 M40 M04`	Spindle rotation at 900 rpm
`N140 ... X50 Z70`	Tool tip positioned on dia. 50
`N150 G96 S200`	Constant surface speed initialised on X50
`N..`	
`N.. G97 S900`	Function G96 cancelled
`N..`	

Programming a surface speed of 30 m/min for drilling a hole with a diameter of 20 mm



`N.. ... (HSS DRILL DIAMETER=20)`	Tool change and correction
`N170 G97 S500 M40 M03`	Spindle rotation at 500 rpm
`N180 ... X0 Z60`	Drill axis positioned in the spindle axis
`N190 G96 X20 S30`	Constant surface speed initialised on the drill diameter (X20)
`N..`	
`N.. G97 S500`	Function G96 cancelled
`N..`	

4.3.3 Spindle Range

M40/M41/M42/M43/M44/M45	Spindle ranges.

The system allows the definition of six spindle ranges associated with address S.

Syntax

N.. [G97 S..] [M03/M04] **M40-M45**

G97 S..	RPM spindle speed.
M03/M04	Spindle direction of rotation.
M40 to M45	Spindle range selection.

Properties of the Functions

Functions M40 to M45 are decoded modal «pre» functions.

Cancellation

Functions M40 to M45 cancel one another.

Notes

The minimum and maximum speeds are defined for each range by the machine manufacturer. Example:

M40 = 50-500 rpm

M41 = 400-900 rpm

M42 = 800-4200 rpm

In a system with automatic range selection, the spindle range is determined simply by programming the S address and the rpm.

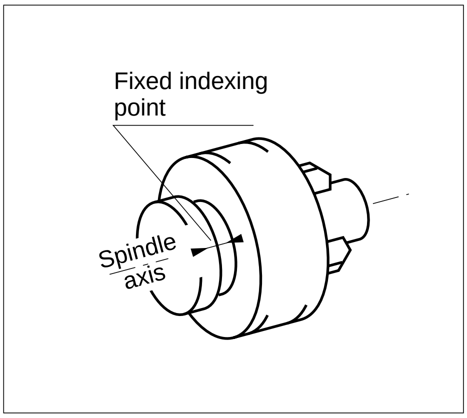
Example

```
N.. ...
N30 G97 S650 M41 M03                 Range M41
N..
```

4.3.4 Indexed Spindle Stop

M19	Indexed spindle stop.

This function stops the spindle at a position defined with reference to a fixed point.



Syntax

N.. [G97 S..] [M40-M45] [M03/M04] C±.. **M19**

G97 S..	RPM spindle speed (with G97).
M40-M45	Spindle speed range
M03/M04	Spindle direction of rotation.
C±..	Optional argument defining the indexing angle in degrees from the fixed point.
M19	Indexed spindle stop.

Properties of the Function

Function M19 is a decoded modal «pre» function.

Cancellation

Function M19 is cancelled by one of functions M03, M04 or M05.

Notes

The spindle may or may not be rotating when indexing is enabled. If the spindle is not rotating, indexing is carried out by positioning along the shortest path.

When the system includes a spindle probe, M19 can be programmed to index the spindle to any position with reference to the fixed position defined by the machine manufacturer (see manufacturer’s manual).

When the system includes bidirectional orientation capabilities, the stopped position is reached by the shortest path.

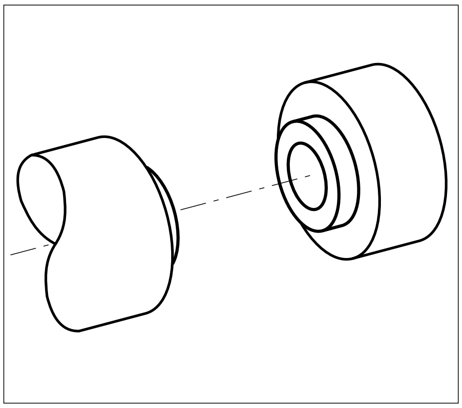
Example

Indexed spindle stop at + 90 degrees with reference to the origin.

`N.. ...`	Tool call
`N120 G97 S500 M42`	Spindle rotating
`N130 C90 M19`	Indexed spindle stop
`N..`	

4.3.5 Spindle Control Selection

M62/M63/M64/M65 Control of spindles 1-4.



When the machine is equipped with several spindles, these functions are used to direct control to the different spindle servo-drives.

The spindle characteristics are defined in machine parameter P6 (see parameter manual).

4

Syntax

N.. [G97 S..][M03/M04][M40-M45]**M62-M65**

G97 S..	RPM spindle speed.
M03/M04	Spindle direction of rotation.
M40-M45	Spindle ranges.
M62	Spindle 3 control.
M63	Spindle 4 control.
M64	Spindle 1 control.
M65	Spindle 2 control.

Properties of the Functions

Functions M62, M63, M64 and M65 are decoded modal «pre» functions.

Cancellation

Functions M62, M63, M64 and M65 cancel one another.

At power on, after a reset or at the end of the programme (M02), each spindle is assigned to the axis group with the same number (e.g. M64 is initialised for axis group one).

Notes

A spindle receives these functions for the axis group to which it is assigned:
- speed G96 S.. or G97 S..,
- spindle direction of rotation or stop (M03, M04, M05),
- spindle speed ranges (M40-M45),
- speed limitation (G92 S..),
- spindle indexing (M19 C..),
- speed override or override cancel (M48 or M49).

The spindle of one group is released by:
- selection of a new spindle (M62-M65),
- release function M61 (see Sec. 4.17.5).

After being released from a group, the spindle preserves all the characteristics it had at the time of release (see above) but any new functions in the group are not addressed to it but to the new spindle assigned to the group.

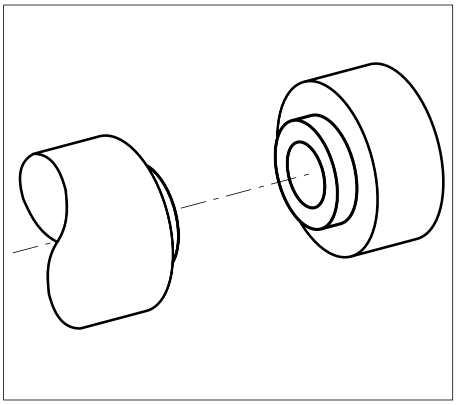
For notes on programming the spindles with multiple axis groups (see Sec. 4.17).

Example

```
N.. ...
N130 M65                  Assignment of spindle 2 to the group
N140 G97 S500 M03 M40     Control of spindle 2
N..
```

4.3.6 Spindle Measurement Selection

M66/M67/M68/M69
Measurement of spindles 1 to 4.



When the machine is equipped with several spindles, these functions are used to select the spindle measurement feedback.

The spindle characteristics are defined in machine parameter P6 (see parameter manual).

Syntax

N.. **M66/M67/M68/M69**

M66	Spindle 1 measurement
M67	Spindle 2 measurement
M68	Spindle 3 measurement
M69	Spindle 4 measurement

Properties of the Functions

Functions M66, M67, M68 and M69 are decoded modal «pre» functions.

Cancellation

Functions M66, M67, M68 and M69 cancel one another.

At power on, at the end of a programme M02 or after a reset, the spindle measurement is assigned to the axis group with the same number (e.g. M66 is assigned to axis group 1). If there is no spindle with the same number as the group, spindle 1 is assigned as default (M66).

Notes

Each axis group can use the measurement of any spindle.

Several groups can use the measurement of the same spindle.

When a spindle measurement is used by an axis group for thread cutting, the spindle speed override is inhibited for the entire duration of the thread cutting cycle (100 per cent override).

When a declared spindle does not have an axis encoder, measurement of this spindle is simulated by the NC.

For programming spindles with multiple axis groups (see Sec 4.17).

Example

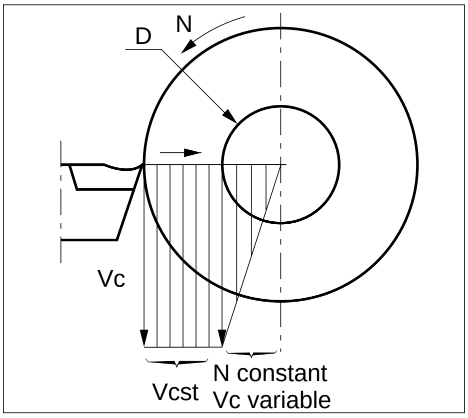
`N.. ...`	
`N180 M67`	Assignment of spindle measurement to group 2
`N190 G94 F50`	Feed in mm/rev related to spindle 2
`N..`	

4.3.7 Spindle Speed Limiting

G92 S..Spindle speed limiting.

This function specifies the maximum spindle speed not to be exceeded.

In case of a reduction in the machining diameter at constant surface speed (G96), the rotation speed can be limited to prevent problems due to centrifugal force, imbalance, etc.



Syntax

N.. **G92** S..

G92	RPM spindle speed limiting.
S..	Mandatory argument associated with the function to define the maximum spindle speed.

Property of the Function

Function G92 is modal.

Cancellation

Spindle speed limiting is cancelled by:
- cancellation function G92 S0,
- function G92 S.. with assignment of a different speed limit,
- the end of programme function (M02),
- a reset.

Notes

Spindle speed limiting:
- must be programmed before selecting constant surface speed (G96),
- must be programmed at a value below the maximum spindle speed,
- is independent of the maximum speed defined in one of the spindle speed ranges (M40-M45).

If the spindle speed programmed by G97 is higher than the speed defined by G92, the spindle rotates at the limited speed.

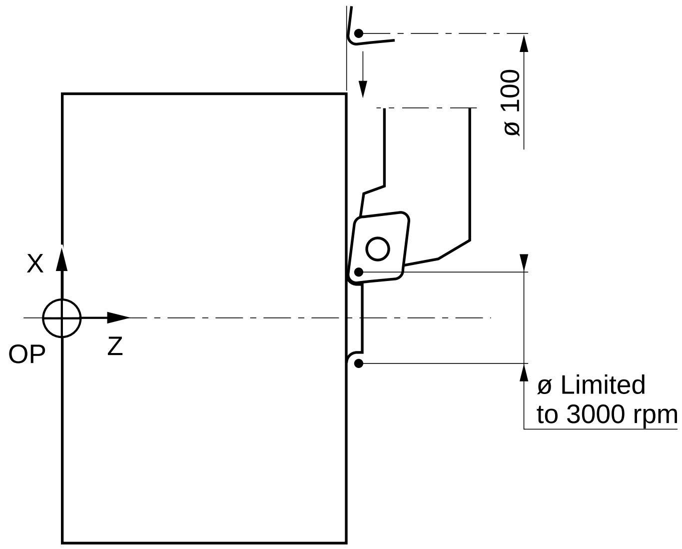
G92 must be immediately followed by its argument S... If they are separated by an axis (X.., Z.., etc.), the system interprets function G92 as preselection of the programme origin on the programmed axis (see Sec. 4.12.3).

Example

Programming speed limiting for facing at constant surface speed

On read of block N60, the spindle is initialised at 637 rpm on diameter 100 mm.

On execution of block N70, the rotation speed is gradually increased up to a maximum of 3000 rpm (approximately 21 mm diameter). The rest of facing up to X0 is carried out at a speed of 3000 rpm.



N.. ...	
N30 G97 S900 M40 M04	Spindle rotation at 900 rpm
N40 ... X100 Z60	Tool tip positioned on diameter 100
N50 G92 S3000	Spindle speed limited to 3000 rpm
N60 G96 S200	Constant surface speed initialised on X100
N70 ... X0	Facing
N..	
N..	
N200 G97 S900	Function G96 cancelled
N..	

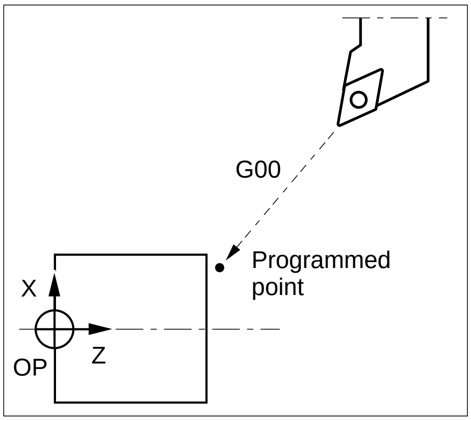
4.4 Rapid Positioning

G00	Linear interpolation at high speed.

The programmed point is reached at high speed by a linear path.

The path is a combination of all the axis movements programmed in the block.

Programmable axes:
- X, Z, (Y) primary axes,
- U, V, (W) secondary axes,
- (A), C, (B) rotary axes.



Syntax

N.. [G90/G91] **G00** [R-/R+] X.. Z..

G90/G91	Programming in absolute or incremental dimensions.
G00	Rapid positioning.
R-/R+	The position is before or after the programmed point. The distance is equal to the tool radius declared.
X.. Z..	End point: - Point coordinates with G90. - Value of the movement with G91.

Property of the Function

Function G00 is modal.

Cancellation

Function G00 is cancelled by one of functions G01, G02, G03 or G33.

Notes

The speed of movement on the path programmed in G00 is determined by the slowest axis (this axis moves at its maximum speed).

Optional arguments R+ or R-:
- are active only in the block where they are programmed,
- cannot be programmed in a block with PGP (see chapter 5).

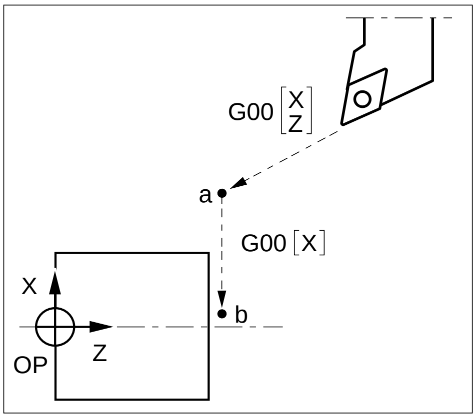
Programming of the additional axes and carrier/carried axis pairs

Two axes of carried/carrier axis pairs can be programmed in G00 using function G52 (programming with reference to the measurement origin, see Sec. 4.12.1).

Examples

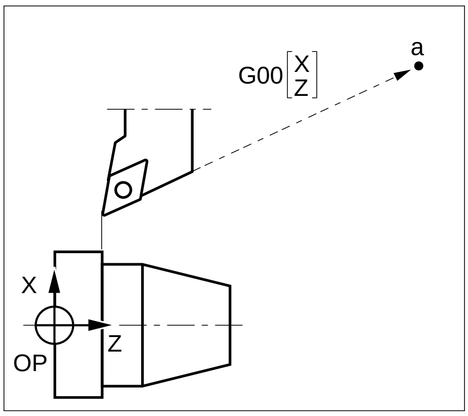
Rapid positioning before machining

```
N..
N.. ...                    Tool call
N30 G97 S600 M40 M04
N40 G00 Xa Za
N50 Xb
N..
```



Rapid retraction after machining

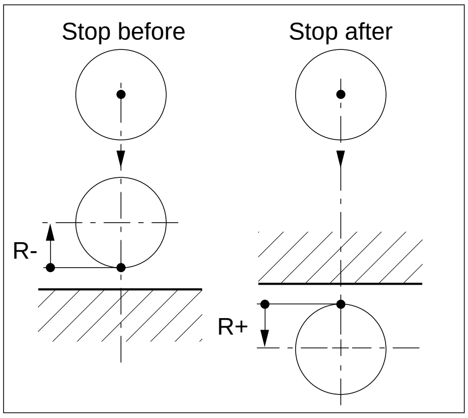
```
N.. ...
N130 G00 Xa Za
N..
```



Positioning with respect to the cutter dia. (R-/R+)

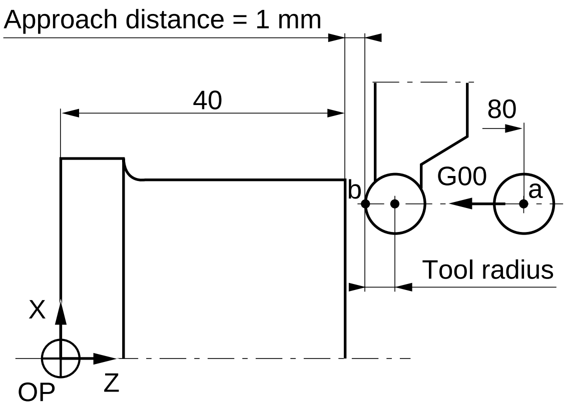
R+ or R- can be programmed to reach the specified approach point regardless of the real tool radius declared (see Sec. 4.8.3).

Positioning is only applied to the axes of the plane.



Example

Programming to stop at an approach distance of 1 mm.




```
N.. ...
N.. D..          Tool correction (see Sec. 4.8.3)
N140 G00 X40 Z80 Point a
N150 R- Z41      Point b, stop before the programmed point

N160 ...
N..
```

4.5 Programming of Movements

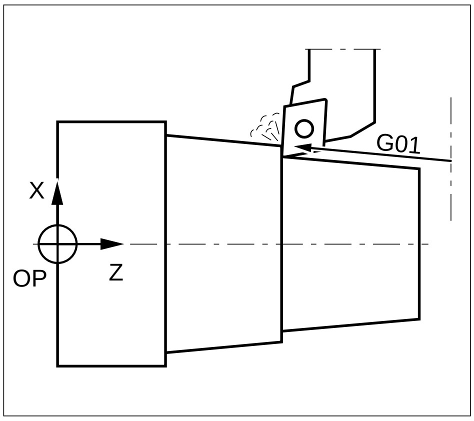
4.5.1 Linear Interpolation

G01	Linear interpolation at the programmed feed rate.

The programmed point is reached by a linear path at the programmed feed rate.

The path is a combination of all the axis movements programmed in the block.

Programmable axes:
- X, Z, (Y) primary axes,
- U, W, (V) secondary axes,
- (A), C, (B) rotary axes.



Syntax

N.. [G90/G91] **G01** [R+/R-] X.. Z.. [F..]

G90/G91	Programming in absolute or incremental dimensions.
G01	Linear interpolation at the programmed feed rate.
R-/R+	The position is before or after the programmed point. The distance is equal to the tool radius declared.
X.. Z..	End point: - Point coordinates with G90. - Value of the movement with G91.
F..	Feed rate (see Sec. 4.7).

Properties of the Functions

Function G01 is modal. It is the default function.

Cancellation

Function G01 is cancelled by one of functions G00, G02 or G03.

Notes

Automatic path smoothing may prevent exact passage through the programmed points unless specifically requested (see Sec. 4.6).

Optional arguments R+ or R-:
- are active only in the block where they are programmed
- cannot be programmed in a block with PGP (see Chapter 5).

Programming of additional axes and carrier/carried axis pairs

Linear interpolation can be programmed by combining movements on primary and additional axes.

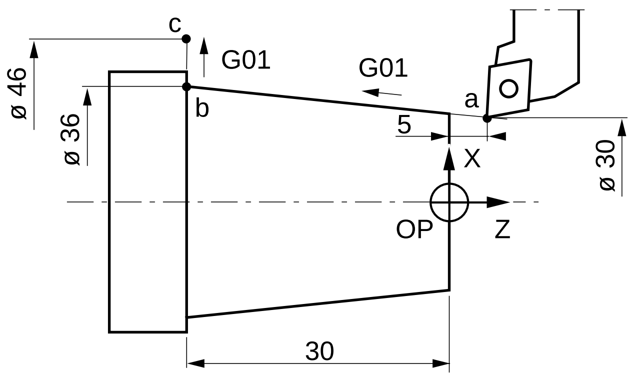
Example:

Linear interpolation on a primary axis and a secondary axis.
```
N.. G01 Y.. W.. F..
```

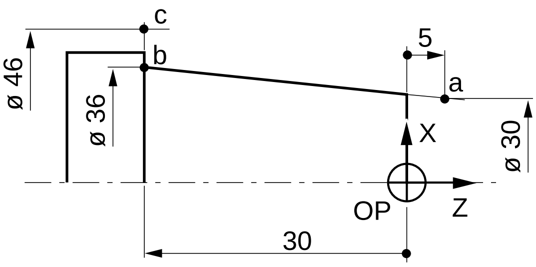
Linear interpolation on a secondary axis and a rotary axis.
```
N.. G01 U.. C.. F..
```

Examples

Linear interpolation along the X and Z axes (machining path a, b, c)

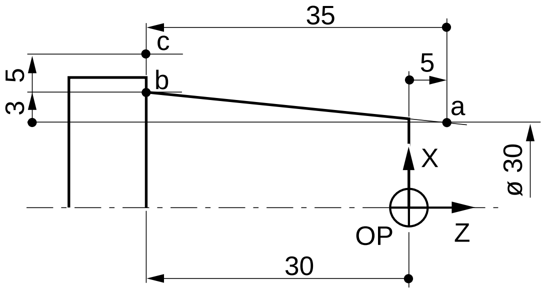


Turning and facing in absolute programming (G90)




```
%20
N10 G00 G52 X.. Z..          Tool change position
N20 ...                      Tool call
N30 G97 S600 M40 M04
N40 X30 Z5                   Point a, approach
N50 G96 S200
N60 G95 F0.2                 Feed rate in mm/rev.
N70 G01 X36 Z-30             Point b, turning
N80 X46 F0.1                 Point c, facing
N..
```

Turning and facing in incremental programming (G91)

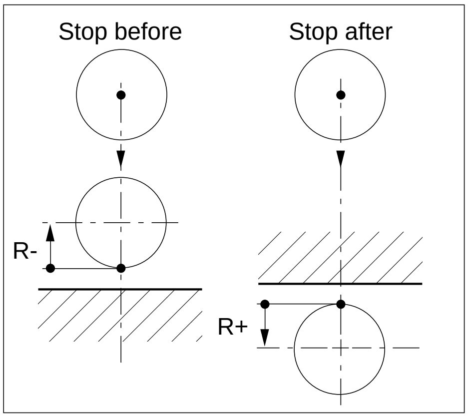



```
%25
N10 G00 G52 X.. Z..          Tool change position
N20 ...                      Tool call
N30 G97 S600 M40 M04
N40 X30 Z5                   Point a, approach
N50 G96 S200
N60 G95 F0.15                Feed rate in mm/rev.
N70 G91 G01 X3 Z-35          Point b, turning
N80 X5                       Point c, facing
N90 G90 ...
```

Positioning with respect to the cutter dia. (R+/R-)

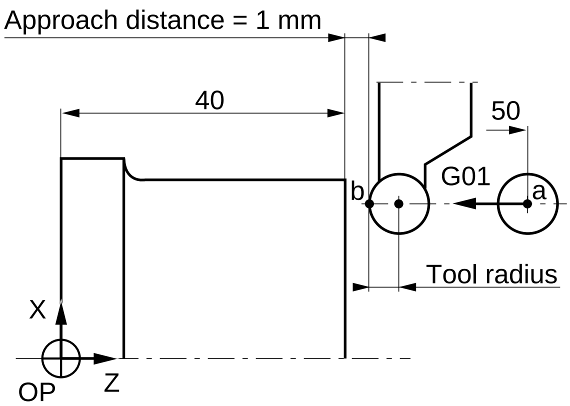
R+ or R- can be programmed to reach the specified approach distance regardless of the read tool radius (see Sec. 4.8.3).

Positioning only applies to the axes of the interpolation plane.



Example

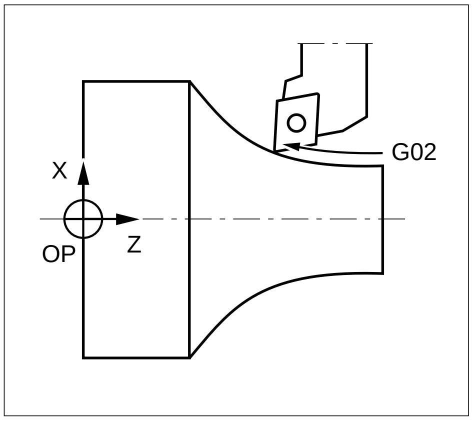
Programming to stop at an approach distance of 1 mm.



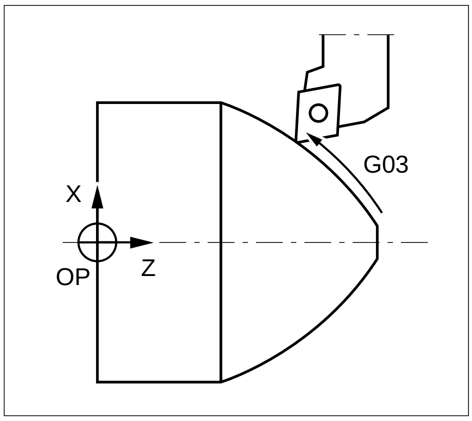
N.. ...	
N.. D..	Tool correction (see Sec. 4.8.3)
N140 G00 X40 Z50	Point a
N150 G01 R- Z41 G95 F0.2	Point b, stop before the programmed point
N..	

4.5.2 Circular Interpolation

G02	Clockwise circular interpolation at the programmed feed rate.



G03	Counterclockwise circular interpolation at the programmed feed rate.



The programmed point is reached by a circular path.

The centre of the tip radius moves at the programmed feed rate.

The two linear axes controlled depend on the interpolation plane:
- Axes Z (or W) and X (or U) in G20 (see Sec. 4.14).

Syntax

N.. [G90/G91]**G02/G03** X.. Z.. I.. K../R..[F..]

G90/G91	Programming in absolute or incremental dimensions.
G02	Clockwise circular interpolation.
G03	Counterclockwise circular interpolation.
X.. Z..	End point Coordinates of the end point with G90. Movement value with G91.
I.. K..	Location of the circle centre in the XZ plane (I along X, K along Z). - With reference to the programme origin for G90. - With reference to the interpolation start point for G91.
R..	Radius of the circle.
F..	Feed rate (see Sec. 4.7).

Properties of the Functions

Functions G02 and G03 are modal.

Cancellation

Function G02 is cancelled by functions G00, G01, G03 or G33.

Function G03 is cancelled by functions G00, G01, G02 or G33.

Notes

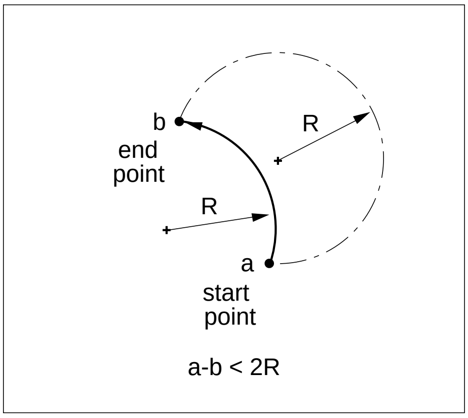
The programmed point may not be reached if the next block is sequenced with path smoothing (see Sec. 4.6).

In a block programmed in G02 or G03, all the addresses used for the interpolation must be specified, even if they are zero (I0, K0) or unchanged with reference to the previous block (for X and Z).

Programming a circle by its radius (R..)

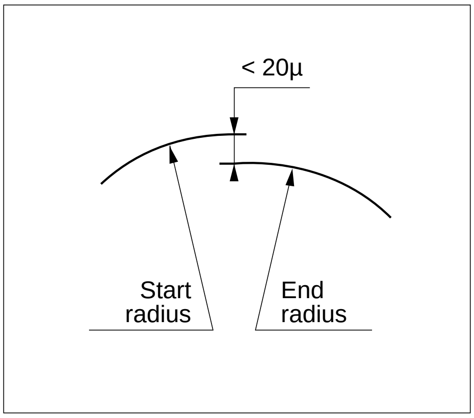
The system chooses the path whose angle is less than 180 degrees (a path with an angle greater than 180 degrees can only be obtained by programming the circle by the coordinates of its centre or in PGP (see Chapter 5).

If the distance between the start point and end point is greater than twice the programmed radius, the system generates an error message.



Programming a circle by the coordinates of its centre (I and K)

When the difference in radii between the start point and end point is greater than 20 µ, the system generates an error message.



Programming the additional axes and carrier/carried axis pairs

Circular interpolation can be carried out on the following primary and secondary axis pairs:

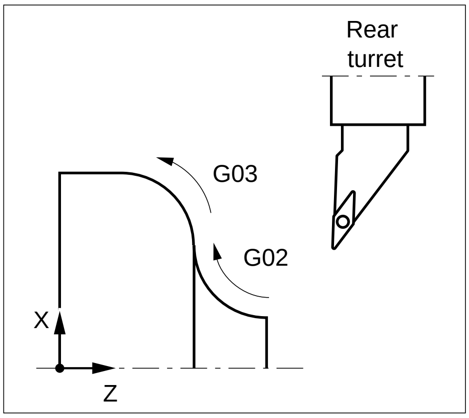
Main interpolation plane	Function	Axis Pairs
ZX	G20	ZU, WX, WU

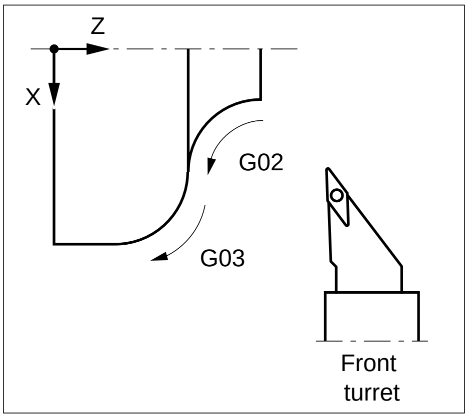
Other axes such as rotary and/or linear axes can be associated in a circular interpolation block, but these axes are interpolated linearly.

A movement on a carrier axis causes an identical movement on the carried axis.

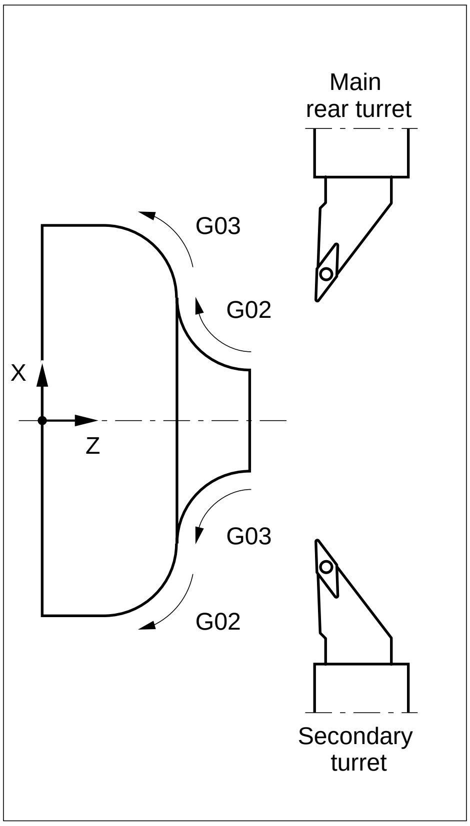
Direction of movement according to the turret position

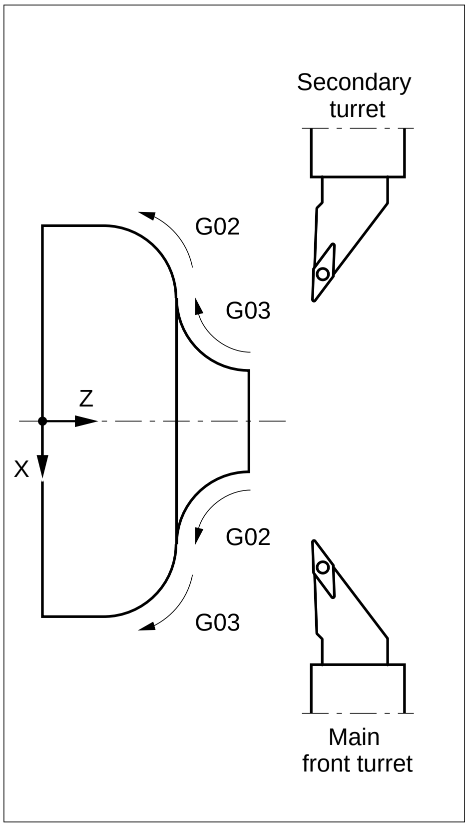
Lathe with one turret





Lathe with two mechanically coupled turrets

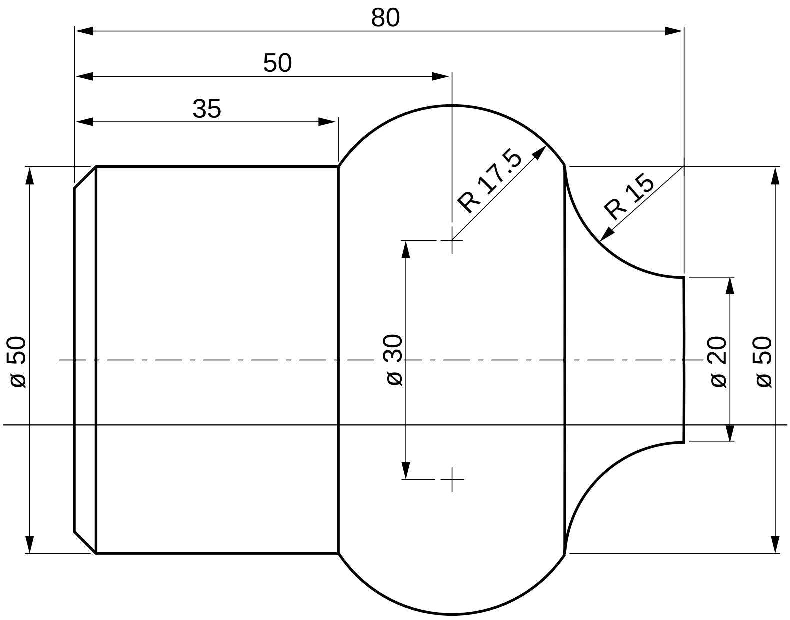




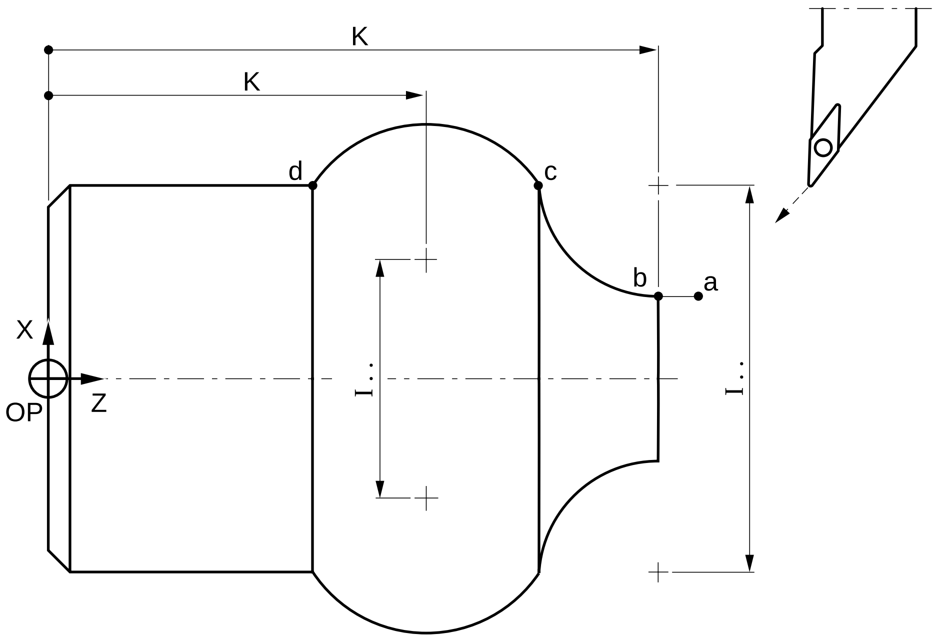
Examples

<u>Circular interpolation by programming in absolute dimensions (G90)</u>

Circular interpolation G02 is carried out by programming the radius (R) and circular interpolation G03 is carried out by programming the circle centre (with I and K).



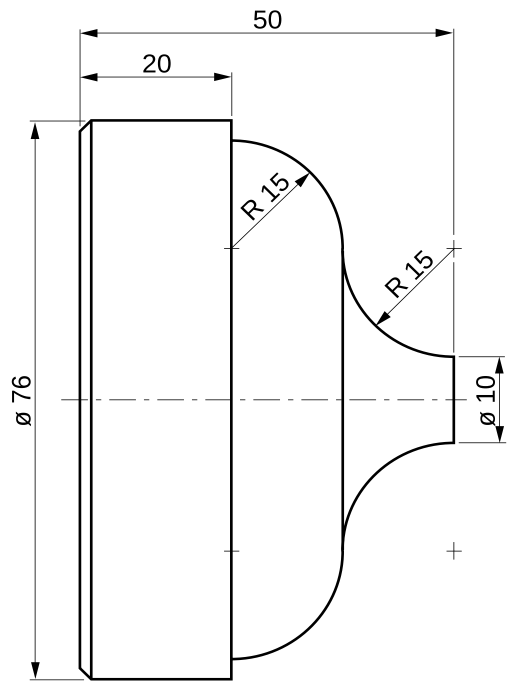
Machining paths (finishing)



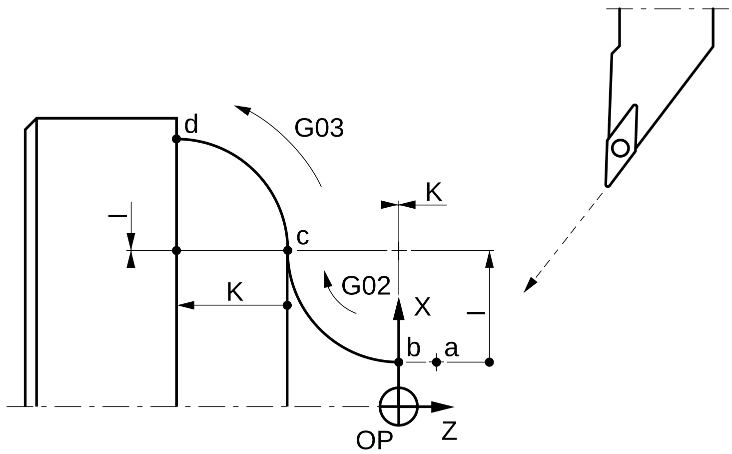

```
%28
N10 G00 G52 X.. Z..                 Tool change position
N20                                 Tool call
N30 S900 M40 M04
N40 G95 F0.2                        Feed rate in mm/rev.
N50 X20 Z85                         Point a, approach
N60 G96 S200
N70 G01 Z80                         Point b
N80 G02 X50 Z65 R15                 Point c
N90 G03 X50 Z35 130 K50             Point d
N100 G00 X80
N110 G52 X.. Z.. G97 S900 M05
N120 M02
```

Circular interpolation by programming in incremental dimensions (G91)

Both interpolations G02 and G03 are carried out by programming the circle centre (with I and K).



Machining paths (finishing)

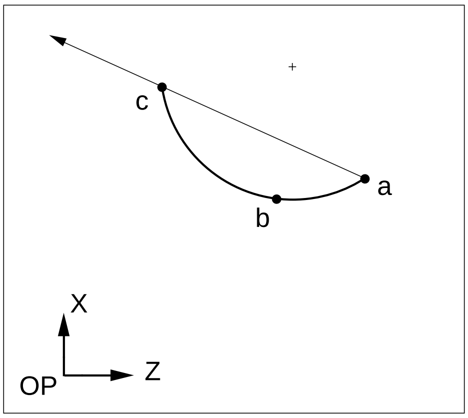


%30	
N10 G00 G52 X.. Z..	Tool change position
N20 ...	Tool call
N30 S900 M40 M04	
N40 G95 F0.15	Feed rate in mm/rev.
N50 X10 Z5	Point a, approach
N60 G96 S200	
N70 G01 Z0	Point b
N80 G91 G02 X15 Z-15 I15 K0	Point c
N90 G03 X15 Z-15 I0 K-15	Point d
N100 G90 G01 X80	
N110 G52 X.. Z.. G97 S900 M05	
N120 M02	

4.5.3 Circular Interpolation Defined by Three Points

G23	Circular interpolation defined by three points.

Circular interpolation can be carried out by programming:
- the start point (defined in the block preceding function G23),
- the end point and the intermediate point (defined in the same block as function G23).



The direction of circular interpolation is defined by the position of the intermediate point (b) with respect to the start point (a) and the end point (c), i.e.:
- to the left of line ac: clockwise,
- to the right of line ac: counterclockwise.

Syntax (XY plane)

N.. [G90/G91] **G23** X.. Z.. I.. K.. [F..]

G90/G91	Absolute or incremental programming.
G23	Clockwise or counterclockwise circular interpolation.
X.. Z..	End point.
I.. K..	Intermediate point.
F..	Feed rate (see Sec. 4.7).

Properties of the Function

Function G23 is nonmodal. However, function G02 or G03 created by the system for a clockwise or counterclockwise arc is modal.

Cancellation

Function G23 is cancelled at the end of the block.

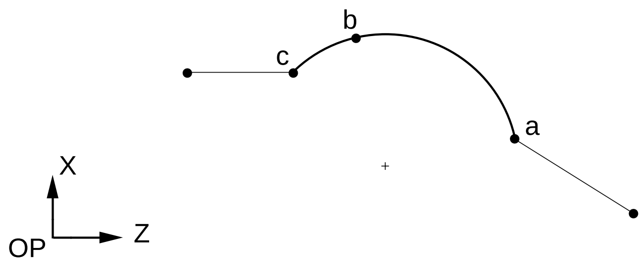
Notes

The arguments of function G23 must not be separated by any other address. Otherwise, the system returns error message 101. Example:
`N.. G23 X.. Z.. F.. I.. K..` Programming incorrect

Circular interpolation defined by G23 can be in absolute dimensions (G90) or incremental dimensions (G91).

Example

Circular interpolation from a through b to c




```
N..
N90 G01 Xa Za G95 F0.2          Point a, approach
N100 G23 Xc Zc Ib Kb F0.15      Circular interpolation
N110 G01 X.. Z.. F0.2
N..
```

4.5.4 Polar Programming

Polar programming is used to define paths or positions when one of the part dimensions is angular.

General

Polar programming is used for line or circle geometric elements defined in the same plane.

Polar programming:
- can coexist with ISO programming and Profil Geometry Programming (PGP) (see Chapter 5)
- can be in absolute dimensions (G90), incremental dimensions (G91) or a combination thereof (G90 and G91, possibly in the same block)
- can be combined with cartesian programming.

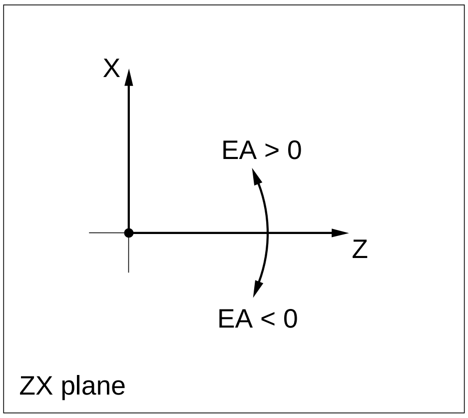
Polar programming can be carried out in one of the interpolation planes selected: ZX or XY (in all cases, the line or circle start point must be defined in the same plane as the one used in programming).

Reference Axis of the Polar Angle

For polar programming of a line or circle, the polar angle is always programmed by argument EA (whatever the interpolation plane selected).

For turning, the polar angle is defined with respect to the reference axis of the interpolation plane:
- Z in the ZX plane (G20)
- X in the XY plane (G21).

The positive or negative direction of the angle is defined in the trigonometric circle.

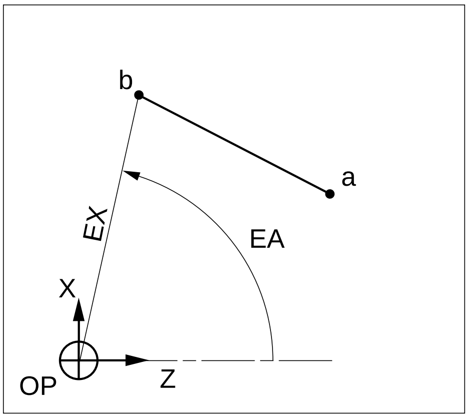


4.5.4.1 Polar Programming of a Line

A line can be programmed:
- in absolute dimensions with function G90
- in incremental dimensions with function G91.

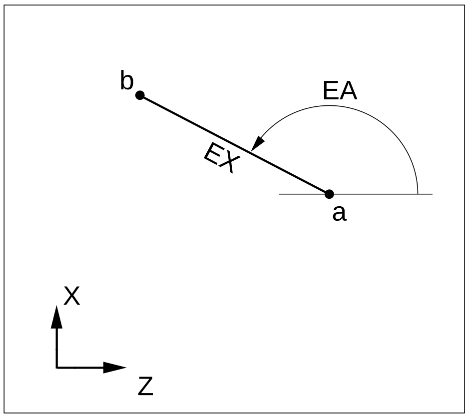
Polar Programming of a Line in Absolute Dimensions (G90)

The line is defined in absolute dimensions by:
- its start point (a) contained in the block preceding the polar programming block
- the polar coordinates of its end point (b) defined with respect to the programme origin (OP).



Polar Programming of a Line in Incremental Dimensions (G91)

The line is defined in incremental dimensions by:
- its start point (a) contained in the block preceding the polar programming block
- the polar coordinates of its end point (b) defined incrementally with respect to its start point (or the last programmed point).



Syntax

N.. [G20] [G90/G91] G00/G01 EA.. EX.. [F..]

G20	ZX plane selected.
G90/G91	Programming in absolute or incremental dimensions.
G00/G01	Linear interpolation.
EA..	Angle of line EX.
EX..	Length of the line. In G90: EX = distance from programme origin to end point. In G91: EX = distance from start point to end point.
F..	Feed rate (see Sec. 4.7).

Notes

Arguments EA and EX in the same block must both be programmed either in incremental dimensions or in absolute dimensions. It is not acceptable to programme EA in absolute dimensions (G90) and EX in incremental dimensions (G91).

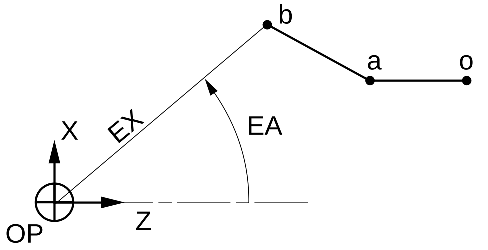
In the block, EA must be programmed first, before EX.

Argument EX:
- is always addressed by the same letters, whatever the interpolation plane
- must always be programmed as a positive value.

Examples

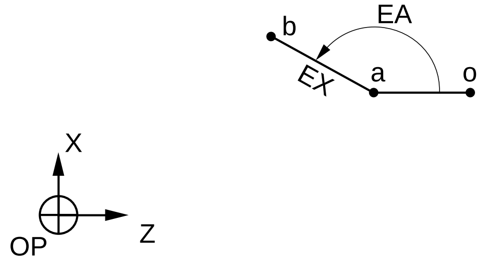
Line programmed in absolute dimensions

```
N..
N.. G90 G20
N200 X20 Z60          Point o
N210 G01 X20 Z40
N220 EA30 EX35
N..
```



Line programmed in incremental dimensions

```
N..
N.. G90 G20
N120 X20 Z60          Point o
N130 G01 X20 Z40
N130 G91 EA120 EX15
N..
```

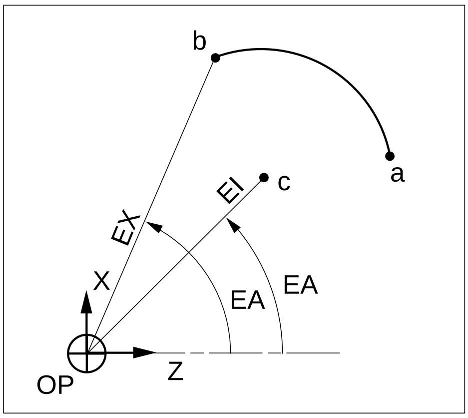


4.5.4.2 Polar Programming of a Circle

A circle can be programmed:
- in absolute dimensions with function G90 or incremental dimensions with function G91
- as a combination of cartesian and polar coordinates
- as a combination of incremental and absolute dimensions (G90/G91) and cartesian and polar coordinates
- by its arc angle and its centre defined in cartesian or polar coordinates.

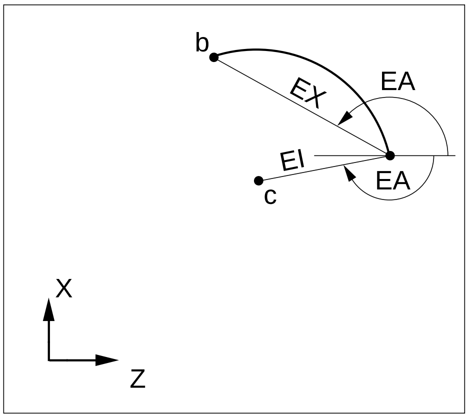
Polar Programming of a Circle in Absolute Dimensions (G90)

The circle is defined in absolute dimensions by:
- its start point (a) contained in the block preceding the polar programming block,
- the polar coordinates of the end point (b) and the centre (c) defined in absolute dimensions with respect to the programme origin (OP).



Polar Programming of a Circle in Incremental Dimensions (G91)

The circle is defined in incremental dimensions by:
- its start point (a) contained in the block preceding the polar programming block,
- the polar coordinates of the end point (b) and the centre (c) defined in incremental dimensions with respect to the circle start point (or last point programmed).



Syntax

N.. [G20] [G90] G02/G03 EA.. EX.. EA.. EI.. [F..]

G20	ZX plane selected.
G90/G91	Programming in absolute or incremental dimensions.
G02/G03	Circular interpolation.
EA..	Angle of line EX.
EX..	Length of the line. In G90 : EX = distance from programme origin to end point. In G91: EX = distance from start point to end point.
EA..	Angle of line EI.
EI..	Length of the line. In G90 : EI = distance from programme origin to circle centre. In G91 : EI = distance from start point to circle center.
F..	Feed rate (see Sec. 4.7).

Notes

The following notes apply to all cases of circles programmed in absolute or incremental dimensions.

In a block, the programming order must be complied with:
- End point EA then EX
- Centre point EA then EI.

Arguments EX and EI must always be programmed in the positive direction.

Arguments EX and EI are addressed by the same letters whatever the interpolation plane chosen.

Examples

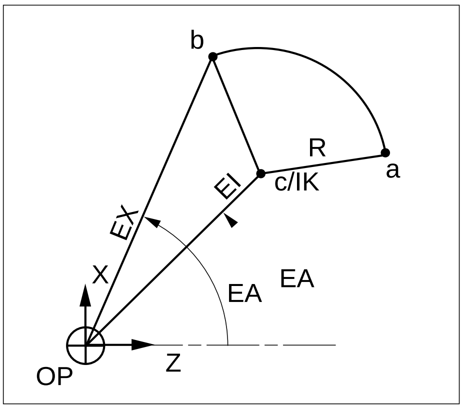
Definition of a circle in absolute dimensions (G90) using cartesian and polar programming.

Cartesian and polar programming can be combined in a block, making it possible to use other circle programming syntaxes.

Example:

Cartesian and polar programming in absolute dimensions in the YZ plane (G19).

```
N.. G90 G20 G01 Xa Za
N.. G02 EAb EXb EAc EIc
or
N.. G90 G20 G01 Xa Za
N.. G02 EAb EXb Ic Kc
or
N.. G90 G20 G01 Xa Za
N.. G02 Xb Zb EAc EIc
or
N.. G90 G20 G01 Xa Za
N.. G02 EAb EXb R..
```



Cartesian and polar programming can also be applied to a circle defined in incremental dimensions.

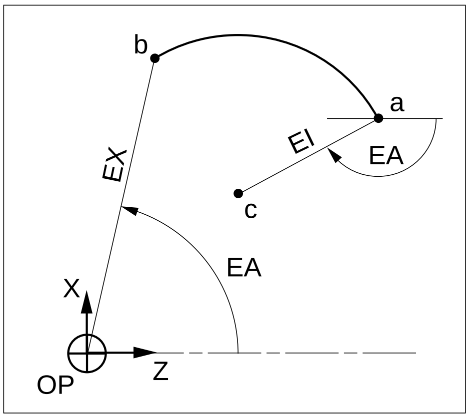
Definition of a circle by programming in absolute/incremental dimensions (G90/G91) and cartesian/polar coordinates

Absolute and incremental dimensions and cartesian and polar coordinates can be programmed in the same block, making it possible to use other circle programming syntaxes.

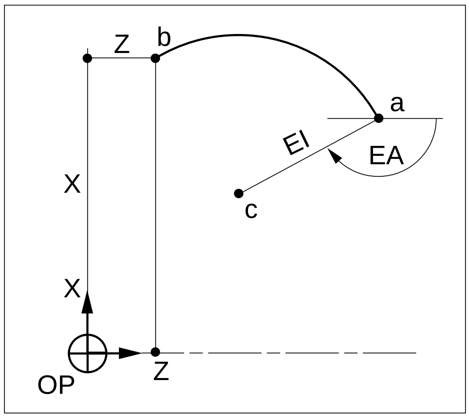
Example:

Absolute/incremental programming (G90 and G91) combined with cartesian and polar programming in the ZX plane (G20).

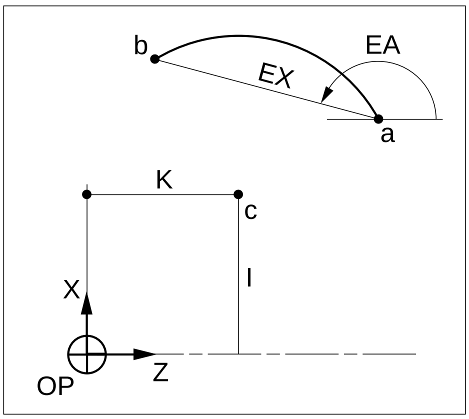
```
N.. G90 G20 G01 Xa Za
N.. G03 EAb EXb G91 EAc EIc
```



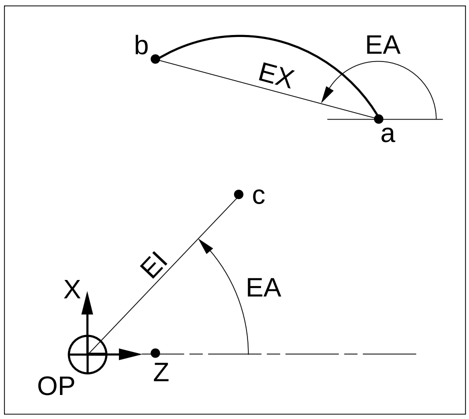

```
N.. G90 G20 G01 Xa Za
N.. G03 Xb Zb G91 EAc EIc
```




```
N.. G90 G20 G01 Xa Za
N.. G91 G03 EAb EXb G90 Ic Kc
```




```
N.. G90 G20 G01 Xa Za
N.. G91 G03 EAb EXb G90 EAc EIc
```



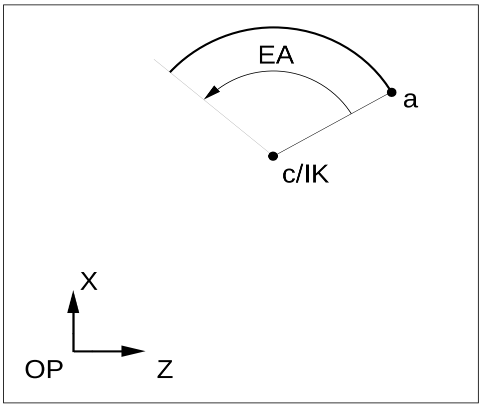
4.5.4.3 Defining a Circle by the Arc Angle

Defining a circle by the arc angle and cartesian programming of its centre defined in absolute or incremental dimensions

The circle is defined by:
- its start point (a) contained in the block preceding the arc angle programming block,
- the cartesian coordinates of its centre (c) and its arc angle.

The centre can be programmed in:
- absolute dimensions with G90,
- incremental dimensions with G91.

The arc angle EA is defined in absolute dimensions.



Syntax

N.. [G20] [G90/G91] G02/G03 I.. K.. EA.. [F..]

G20	ZX plane selection.
G90/G91	Programming of the circle centre in absolute or incremental dimensions.
G02/G03	Circular interpolation.
I.. K..	Coordinates of circle centre in the ZX plane (I along X and K along Z): - in G90 with reference to the programme origin - in G91 with reference to the circle start point.
EA..	Arc angle. Angle of the end point with respect to the start point.
F..	Feed rate (see Sec. 4.7).

Notes

The notes below concern only circles defined by the arc angle (with the centre defined in cartesian coordinates).

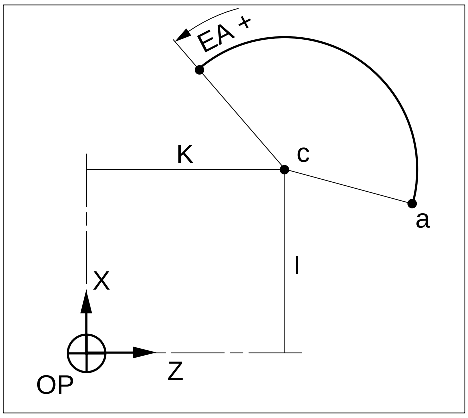
When a value of zero is assigned to EA, the system describes a complete circle.

When the circle is defined from Z to X, EA is positive; in the opposite direction, EA is negative.

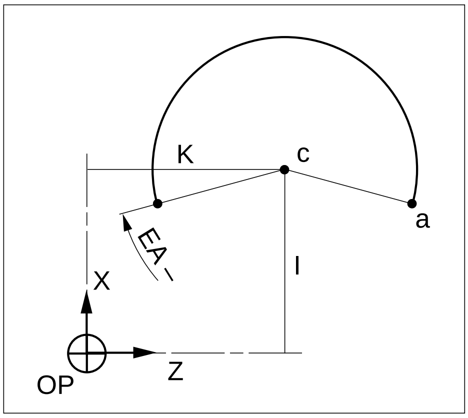
Four types of circles are possible in absolute programming, depending on the direction of the programmed circular interpolation (G02 or G03) and the sign (positive or negative) of arc angle EA.

Example

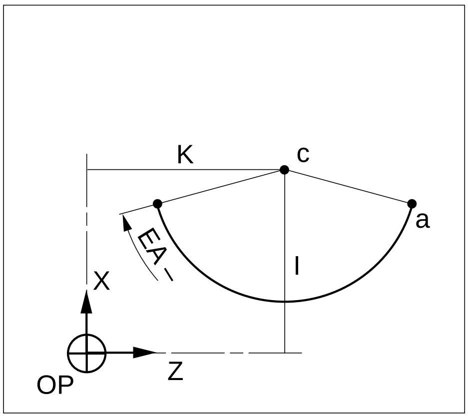
```
N.. G90 G20 Xa Za
N.. G03 I.. K.. EA+160
```



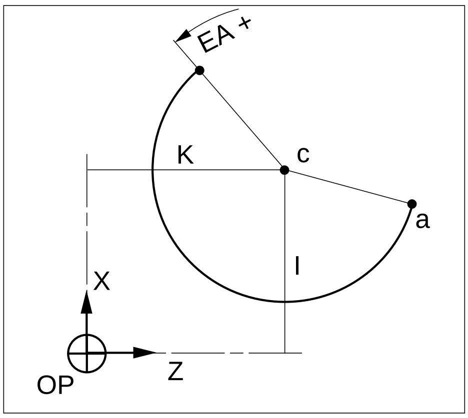

```
N.. G90 G20 Xa Za
N.. G03 I.. K.. EA-160 (ou EA+200)
```




```
N.. G90 G20 Xa Za
N.. G02 I.. K.. EA-160
```



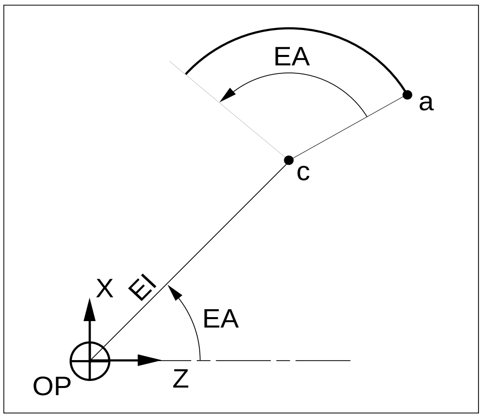

```
N.. G90 G20 Xa Za
N.. G02 I.. K.. EA+160 (ou EA-200)
```



Defining a circle by its arc angle and polar programming of its centre in absolute coordinates (G90)

The circle is defined by:
- its start point (a) contained in the block preceding the arc angle programming block
- the polar coordinates of its centre (c) and its arc angle.

Arc angle EA is defined in absolute dimensions.



4

Syntax

N.. [G20] [G90] G02/G03 EA.. EI.. EA.. [F..]

G20	ZX plane selection.
G90	Circle centre programmed in absolute dimensions.
G02/G03	Circular interpolation.
EA..	Angle of line EI. programme origin/circle centre
EI..	Length of the line. EI = distance from programme origin to circle centre.
EA..	Arc angle. Angle of the end point with respect to the start point.
F..	Feed rate (see Sec. 4.7).

Notes

The notes below concern only circles defined by the arc angle (with the centre defined in polar coordinates).

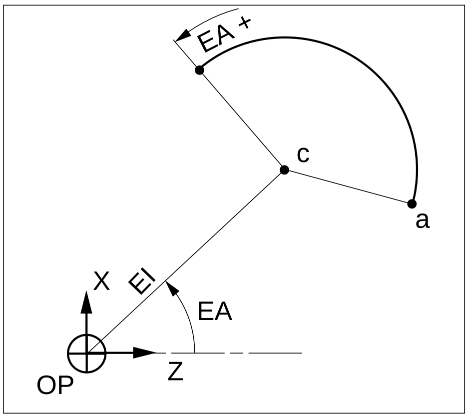
When a value of zero is assigned to EA, the system describes a complete circle.

When the circle is defined from Z to X, EA is positive; in the opposite direction, EA is negative.

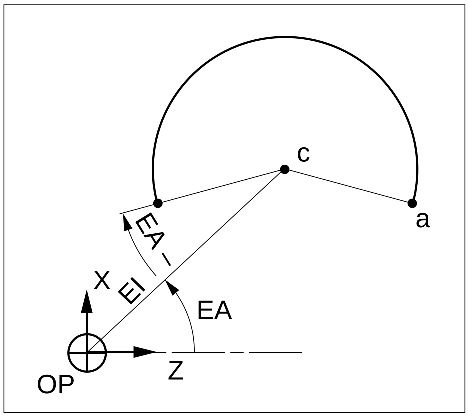
Four types of circles are possible in absolute programming, depending on the direction of the programmed circular interpolation (G02 or G03) and the sign (positive or negative) of arc angle EA.

Example:

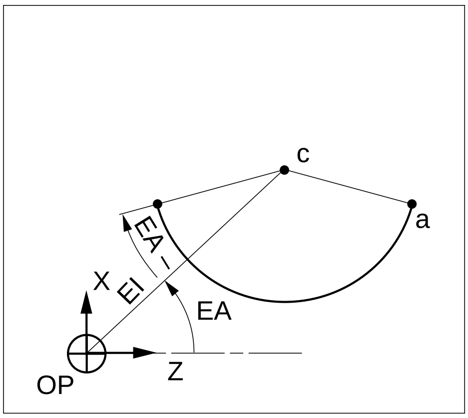
```
N.. G90 G20 Xa Za
N.. G03 EA.. EI.. EA+160
```



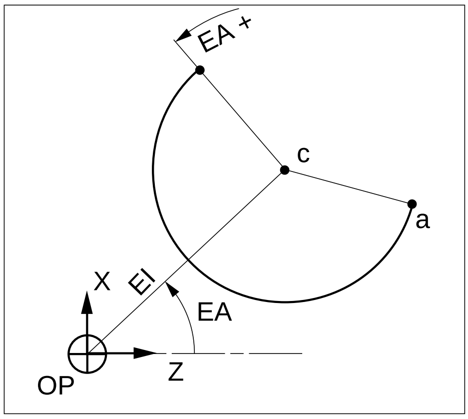

```
N.. G90 G20 Xa Za
N.. G03 EA.. EI.. EA-160 (or EA+200)
```




```
N.. G90 G20 Xa Za
N.. G02 EA.. EI.. EA-160
```




```
N.. G90 G20 Xa Za
N.. G02 EA.. EI.. EA+160 (or EA-200)
```

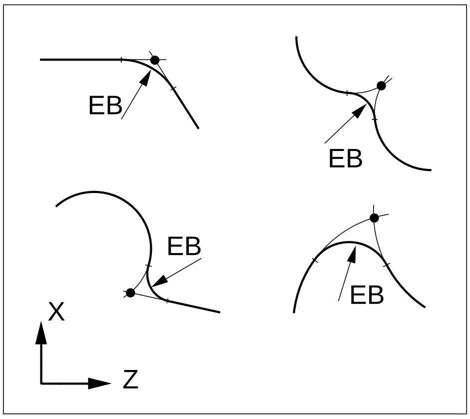


4.5.5 Programming Fillets and Chamfers

4.5.5.1 Fillet Between Two Interpolations

EB+	Fillet between two interpolations.

This functions is used to make a fillet between two linear and/or circular interpolations.



Syntax (XY plane)

N.. G01/G02/G03 X.. Z.. I.. K.. / R.. [F..] **EB+..** [EF..]

G01 / G02 / G03	Linear or circular interpolations.
X.. Z..	Programmed intersection point.
I.. K.. / R..	Circle centre or radius in G02 or G03.
F..	Feed rate (see Sec. 4.7).
EB+..	Fillet dimension.
EF..	Feed rate specific to the fillet (see Sec. 4.7).

Property of the Function

Function EB+.. is nonmodal.

Cancellation

Function EB+ is cancelled at the end of the block.

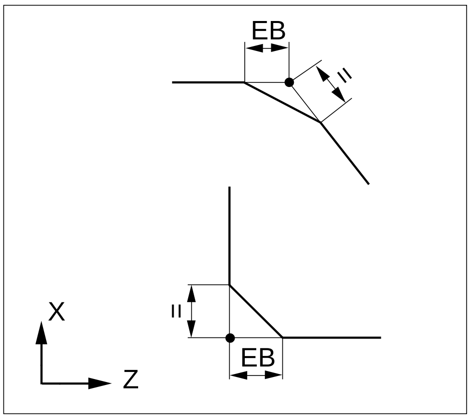
Example

Refer to the example in Sec. 4.7.4 (feed rate specific to fillets and chamfers).

4.5.5.2 Chamfer Between Two Linear Interpolations

EB-	Chamfer between two linear interpolations.

This functions is used to make a chamfer between two linear interpolations.



Syntax (XY plane)

N.. G01 X.. Z.. [F..] **EB-..** [EF..]

G01	Linear interpolation.
X.. Z..	Programmed intersection point.
F..	Feed rate (see Sec. 4.7).
EB-..	Chamfer dimension.
EF..	Feed rate specific to the chamfer (see Sec. 4.7).

Property of the Function

Function EB-.. is nonmodal.

Cancellation

Function EB- is cancelled at the end of the block.

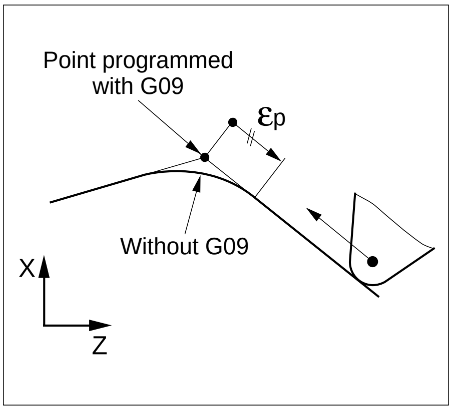
Example

Refer to the example in Sec. 4.7.4 (feed rate specific to fillets and chamfers).

4.6 Path Sequencing Conditions

G09	Accurate stop at end of block before continuation to next block.

The programmed point is reached when the function is programmed in the block.



4

Syntax

N.. **G09** [G00/G01/G02/G03] X.. Z.. [F..]

G09	Accurate stop at the end of a block before continuation to the next block.
G00/G01/G02/G03	Linear or circular interpolation.
X.. Z..	Coordinates of the end point.
F..	Feed rate (See Sec. 4.7).

Property of the Function

Function G09 is nonmodal.

Cancellation

Function G09 is cancelled at the end of the block.

Notes

The tracking error εp is directly proportional to the feed rate.

The more acute the angle between two paths, the greater the smoothing effect at a given feed rate and therefore a given εp.

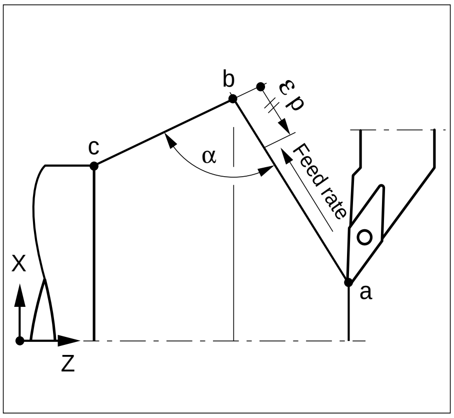
When G09 is programmed:
- the tracking error εp is closed down at the end of the path,
- the feed rate is zero at the end of the block.

Examples

Programming with G09

The axes are decelerated along path ab at a distance εp from point b and go through point b.

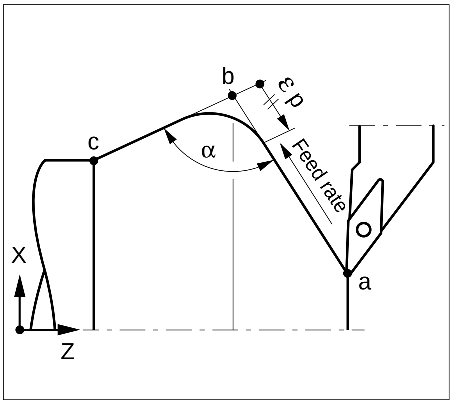
```
N.. ...
N100 G01 Xa Za G95 F0.3
N110 G09 Xb Zb
N120 G09 Xc Zc
N..
```



Programming without G09

The axes are not decelerated and do not go through point b.

The resulting curve between the paths result from the feed rates along ab and bc and the angular value of the vectors.

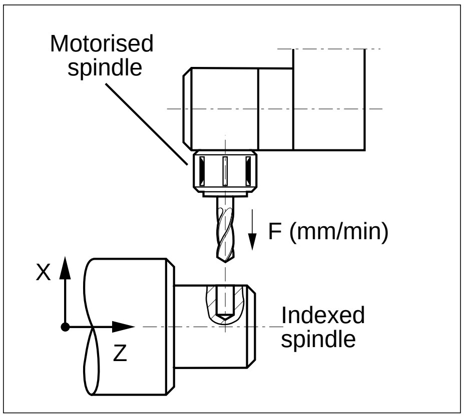


4.7 Feed Rate

4.7.1 Feed Rate Expressed in Millimetres, Inches or Degrees per Minute

G94	Feed rate expressed in millimetres, inches or degrees per minute.

The feed rate is expressed in millimetres or inches per minute on linear axes and in degrees per minute on rotary axes programmed alone.



4

Syntax

N.. **G94** F.. G01/G02/G03 X.. Z.. C..

G94	Function setting the feed rate: - in millimetres/min, - in inches/min - in degrees/min.
F..	Mandatory argument associated with the function and defining the feed rate.
G01/G02/G03	Linear or circular interpolation.
X.. Z..	End point on linear axes.
C..	Angular end point on rotary axes.

Properties of the Functions

Function G94 is modal. It is the default function.

Reminder

Address F is assigned a default value of 1000 mm/min (F1000).

If the system is initialised with G95 by the OEM, address F is assigned a default value of 1 mm/rev (F1) (for additional information, refer to machine parameter P7 in the Parameter Manual).

Cancellation

Function G94 is cancelled by function G95.

Notes

The feed rate limits are defined by the machine manufacturer (see manufacturer's technical data). When the programmed feed rate exceeds the permissible minimum or maximum rate, the system automatically limits it to the permissible feed rate.

The feed rate cannot be programmed in inches per minute unless the system is in state G70 (see Sec. 4.15.4, programming in inches).

When the programme units are changed, a G function defining the new feed rate type and its programming format must be followed by F.. (if the system was already in a G94 state, address F.. can be programmed alone in a block).

Example

`N..`	
`N140 G00 X.. Z..`	
`N150 G95 F0.3`	Feed rate in mm per revolution
`N160 G01 Z..`	
`N170 X.. Z.. F0.2`	
`N..`	
`N240 G00 X.. Z..`	
`N250 G94 F200 G01 X.. W..`	Feed rate in mm/min on a primary axis and a secondary axis
`N260 W.. F100`	
`N..`	

Programming additional axes

The feed rate on rotary axes or independent secondary axes results from the combined movement vector in the basic reference system.

Rotary axes programmed alone in a block are assigned a feed rate calculated from the orthogonal resultant of their relative dimensions.

Equation for determining the feed rate in this case:

$$F.. = \sqrt{\Delta A^2 + \Delta B^2 + \Delta C^2} / \Delta t.$$

Programming a modulo rotary axis programmed alone

Example:

```
N.. G91 G94 F40 G01 C30
```

Feed rate F40 is expressed in degrees per minute (execution time = 45 seconds).

Programming a modulo rotary axis and a linear axis

Example:

```
N.. G91 G94 F100 G01 X10 C30
```

The feed rate on the X axis is expressed in millimetres per minute. The feed rate on the C axis depends on the time required to execute the linear path on the X axis.

$t = \Delta X / F = 10 / 100 = 0.1$ minute, i.e. 6 seconds.

The feed rate on the C axis is equal to 30 deg/6 seconds, i.e. 5 deg/s.

Reminder

Determination of the feed rate (Fr) in mm/min.

Feed rate Fr = N x fz x Z

Number of teeth of the tool

Feed per tooth

Rotation speed

Example:

N = 750 revolutions per minute,

fz = 0.1 mm,

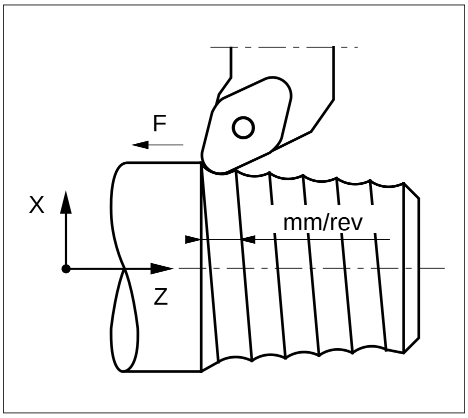
Z = 2 teeth,

Fr = 750 x 0.1 x 2 = 150 mm/min, i.e. F150.

4.7.2 Feed Rate Expressed in Millimetres or Inches per Revolution

G95	Feed rate expressed in millimetres or inches per revolution.

The feed rate is expressed in millimetres or inches per spindle revolution.



Syntax

N.. **G95** F.. G01/G02/G03 X.. Z..

G95	Function setting the feed rate: - in mm/rev, - in in./rev.
F..	Mandatory argument associated with the function defining the feed rate.
G01/G02/G03	Linear or circular interpolation at the programmed feed rate.
X.. Z..	End point on the linear axes.

Property of the Function

Function G95 is modal.

Reminder:

- G94 (mm/min) is the default function.
- Address F is assigned a default value of 1000 mm/min (F1000) at power on (for additional information, refer to machine parameter P7 in the Parameter Manual).

Cancellation

Modal function G95 is cancelled by function G94.

REMARK *If the system is initialised in G95 mode by the machine manufacturer, address F is assigned a default value of 1 mm/rev (F1).*

Notes

The feed rate limits are defined by the machine manufacturer (see manufacturer's technical data). When the programmed feed rate exceeds the permissible values, the system automatically limits it (maximum limit 30 mm/revolution). If a higher value is programmed, the system does not return an error message but simply limits the feed rate to 30 mm/revolution).

When the feed rate unit is changed, the G function defining the new feed rate unit and its programming format must be followed by the argument F.. (if the system is already in state G95, address F.. can be programmed alone in a block).

The feed rate cannot be programmed in inches per minute unless the system is in G70 mode (programming in inches).

Example

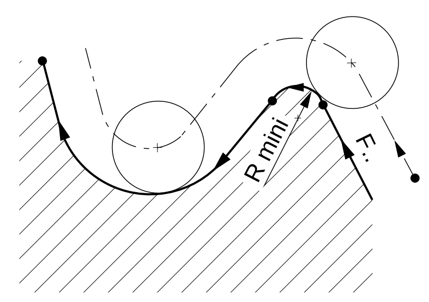
```
N..
N.. G00 X.. Z..
N140 G94 F200                      Feed rate in mm/min
N150 G01 Z..
N160 X.. Z.. F100
N..
N240 G00 X..  Z..
N250 G95 F0.3 G01 X.. W..          Feed rate in mm/rev on a primary axis
                                   and a secondary axis
N260 W.. F0.2
N..
```

4.7.3 Tangential Feed Rate

G92 R..	Programming of tangential feed rate.

This function applies a tangential feed rate when machining curves with tool radius correction (see Sec. 4.8.4).

Feed rate F.. is no longer applied to the tool centre as it may be to large.



Syntax

N.. **G92** R..

G92	Tangential feed rate applied to tool radius correction.
R..	Mandatory argument defining the minimum value of curve radius below which the tangential feed rate is to be ignored.

Properties of the Function

Function G92 followed by argument R is modal.

Cancellation

Tangential feed rate G92 R.. is cancelled by:
- cancellation function G92 R0,
- function G92 R.. with a different radius,
- the end of programme function (M02),
- a reset.

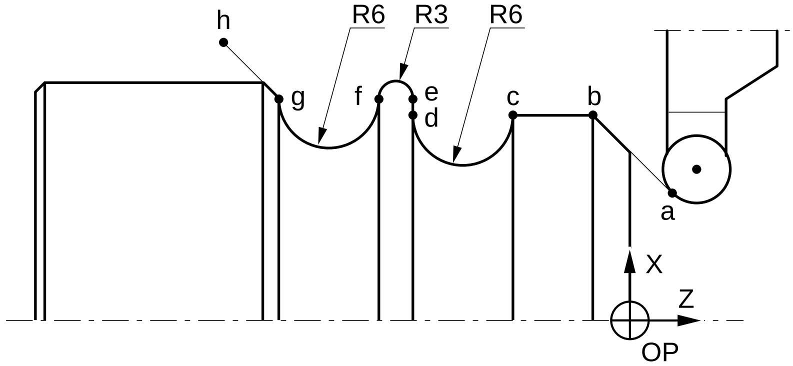
Notes

Function G92 is ignored during automatic creation of a connecting circle between two secant elements (lines or circles) with radius correction. The feed rate then is the same as the feed rate programmed in the previous block.

Function G92 programmed in a block must not be accompanied by axis commands.

Example

In this example, the tangential feed rate is applied to curves whose radius is greater than 4 mm.

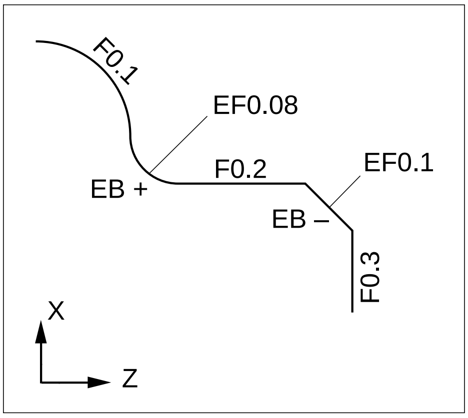


%23	
N10 G00 G52 X.. Z..	Tool change position
N20_	Tool call
N30 S900 M40 M04	
N40 G95 F0.2	
N50 G00 G42 Xa Za	Right radius offset
N60 G92 S3000	
N70 G96 S100	
N80 G92 R4	Tangential feed rate limit
N90 G01 Xb Zb F0.2	
N100 Xc Zc	
N110 G02 Xd Zd R6	Feed rate applied to the point of tangency
N120 G01 Xe, Ze	
N130 G03 Xf Zf R3	Feed rate applied to the tool centre
N140 G02 Xg Zg R6	
N150 G01 Xh Zh	
N160 G92 R0	Tangential feed rate cancelled
N170 G00 G40 G52 X.. Z.. G97 S900 M05	
N180 M02	

4.7.4 Feed Rate Specific to Fillets EB+ and Chamfers EB-

EF	Feed rate specific to fillets EB+ and chamfers EB-.

A feed rate different from the modal machining feed rate F can be programmed for fillets and chamfers programmed by EB+ and EB-.



Syntax

N.. Interpolation EB+.. / EB-.. **EF..**

Interpolation	Linear interpolation (G01) or circular interpolation (G02 or G03).
EB+	Fillet dimension.
EB-	Chamfer dimension.
EF..	Feed rate.

Property of the Function

Function EF.. is modal.

Cancellation

Function EF followed by a value is cancelled by programming:
- function EF followed by a new value, or
- the end of the programme (M02).

Notes

Feed rate EF is substituted for the modal feed rate F if its value is nonzero and is less than feed rate F.

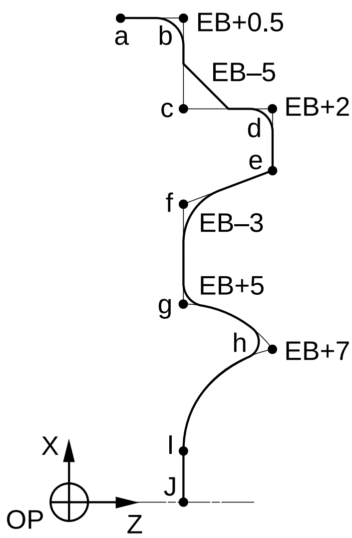
Feed rate EF is in the unit specified by G94 (mm/min) or G95 (mm/rev).

Feed rate F.. in mm/min (G94) or mm/rev (G95) remains modal when executing fillets and/or chamfers.

4

Example

Finishing a contour with feed rate EF in the chamfers and fillet.



When executing the contour, the linear and circular interpolations are carried out at modal feed rate G95 F0.4.

%39	
N.. ...	
N110 G00 G52 X.. Z.. G97 S900 M04	Tool spindel positioning
N120 ...	Tool call
N130 X160 Z30	Point a
N140 G96 S400	
N150 G92 S4000	
N160 G95 F0.4	Feed rate for G1, G2 or G3
N170 G01 Z50 EB+0.5 EF0.3	Point b (feed rate 0.3)
N180 X140 EB-5 EF0.2	Point c (feed rate 0.2)
N190 Z60 EB+2	Point d (feed rate 0.2)
N200 X110	Point e (feed rate 0.4)
N210 Z60 EB-4 EF0.15	Point f (feed rate 0.15)
N220 X90 Z50 EB+5	Point g (feed rate 0.15)
N230 X70 EB+5 EF0.2	Point h (feed rate 0.2)
N240 G02 X40 Z70 R30 EB+7	Point i (feed rate 0.2)
N250 G03 X10 Z60 R20	Point j (feed rate 0.4)
N260 G1 X0	Point k (feed rate 0.4)
N270 ...	

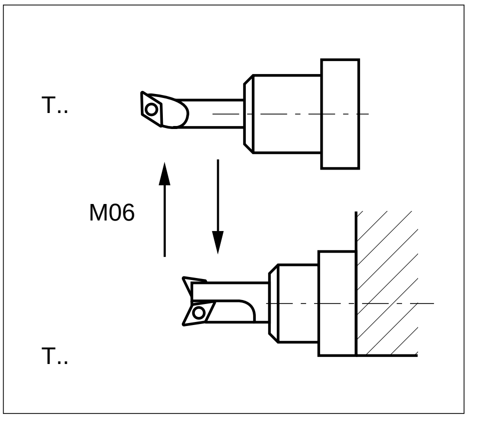
4.8 Programming of Tools

4.8.1 Tool Change

M06	Tool change

This function calls for a tool to be placed in the machining station.

The tool is loaded automatically or manually.



Syntax

N.. T.. **M06** [$0.. or (...)]

T..	The tool is selected by function T followed by a number. The number corresponds to the tool station on the machine turret.
M06	Tool change.
$0 or (...)	Possible message or comment concerning the tool characteristics (see Sec. 4.19).

Properties of the Function

Function M06 is a decoded nonmodal «post» function.

Cancellation

Function M06 is reset when the NC detects the M function report (CRM).

Notes

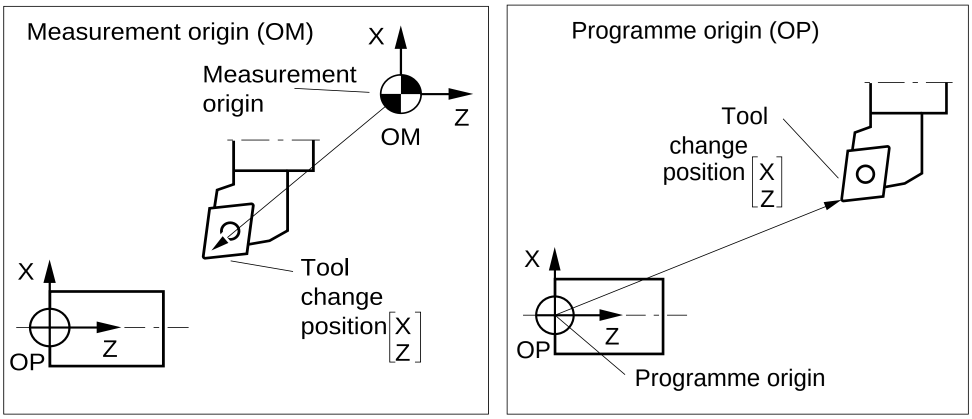
Function T defining the tool number must not be assigned a value greater than 99999999. Above, the system returns error message 1.

Before a tool change, it is recommended to programme a safe position for indexing the turret:
- with reference to the programme origin (OP), or
- with reference to the measurement origin programmed with function G52 (see Sec. 4.12.1).

Example

Automatic tool call

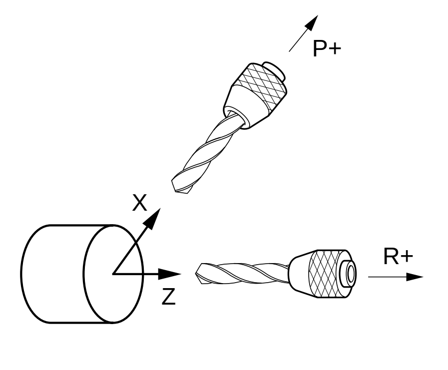
```
N.. ...
N90 G00 X.. Z.. or G00 G52 X.. Z..
N100 T05 M06 (TURN/FACING TOOL R=0.8)
N..
```



4.8.2 Tool Axis Orientation

G16	Definition of the tool axis orientation with addresses P, R.

The tool axis orientation is defined by this function with one of the arguments P or R followed by a plus or minus sign.



Syntax

N.. **G16** P±/R±

G16	Definition of the tool axis orientation.
P+	Points along X+.
P-	Points along X-.
R+	Points along Z+.
R-	Points along Z-.

Properties of the Function

Function G16 followed by one of arguments P or R is modal.

Function G16 followed by R+ is the default function.

Cancellation

Function G16 is cancelled by programming G16 with a different P or R argument.

Notes

By convention, the tool vector points from the tool tip (cutting part) to the tool reference (spindle mounting).

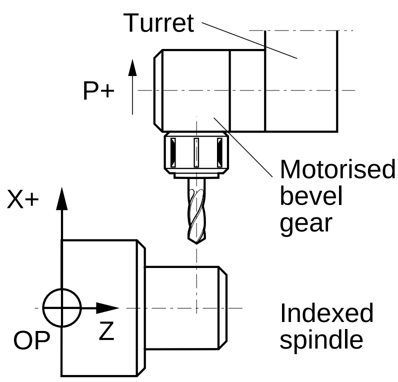
The tool axis cannot be an independant secondary axis.

When defining the tool axis orientation:
- it is recommended to cancel tool radius offset (G40) and canned cycles (G80),
- the block containing G16.. may include movements, miscellaneous M functions and technological S and T functions.

Example

```
%44
N10 G00 G52 X.. Z.. (G16 R+)            Orientation initialised along Z+
N20 T08 ... M06 (TOOL)
N30 G97 S900 M40 M04
N..
N..
N170 G00 G52 X.. Z..
N180 T11 ... M06 (DRILL)
N190 G97 S200 M03 M40
N200 C0 M19
N210 G16 P+                             Axis orientation along X+
N220 ...
N..
```

Machine equipped with a live tool.

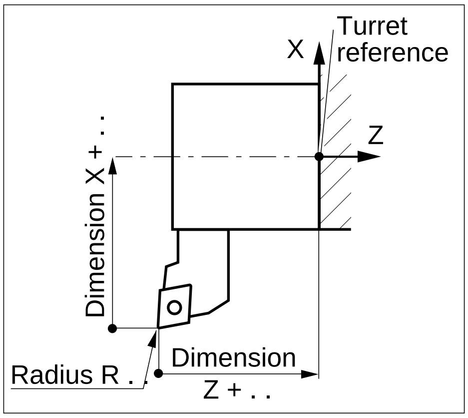


4.8.3 Tool Correction Call

D..	Tool correction call.

Address D followed by a number selects the tool correction.

The stored tool dimensions are validated for the programmed axes



The tool dimensions are displayed on the «TOOL CORRECTIONS» page:
- X = Tool dimension on the X axis,
- Z = Tool dimension on the Z axis,
- R = Tool radius,
- C = Tool tip orientation.

The dimensions can be entered:
- manually or via a peripheral (see operator’s manual),
- by parametric programming (see Sec. 6.2).

Syntax

N.. [G16 R+] **D..** [G40/G41/G42] X.. Z..

G16 R+	Tool axis orientation along Z.
D..	Correction number (1 to 255 corrections).
G40	Tool radius offset cancel.
G41/G42	Tool radius offset.
X.. Z..	End point.

Properties of the Function

Function D.. is modal. Correction D0 is the default correction.

Cancellation

Function D.. is cancelled by programming a new correction or D0.

Notes

The correction number may be different from the tool number.

Several correction numbers can be assigned to the same tool.

D0 always contains zero.

The system includes 255 tool corrections (X,Z,R,C). If the number assigned to the correction is higher than 255, the system returns error message 8.

<u>Tool corrections (dimensions) in X and Z</u>

The tool corrections are assigned to the tool axis orientation defined by G16.. (see Sec. 4.8.2).

The X and Z axes (U and W if carried) are affected by the tool dimensions stored in corrections X and Z.

The declared tool dimensions are taken into account when programming:
- a correction number D..,
- a movement of the axis parallel to the tool axis orientation.

During machining, the tool dimensions are re-applied during:
- correction number changes,
- the use of wear corrections,
- a tool axis orientation change.

The corrections are suspended by programming G52 (see Sec. 4.12.1, programming in absolute dimensions referenced to the measurement origin).

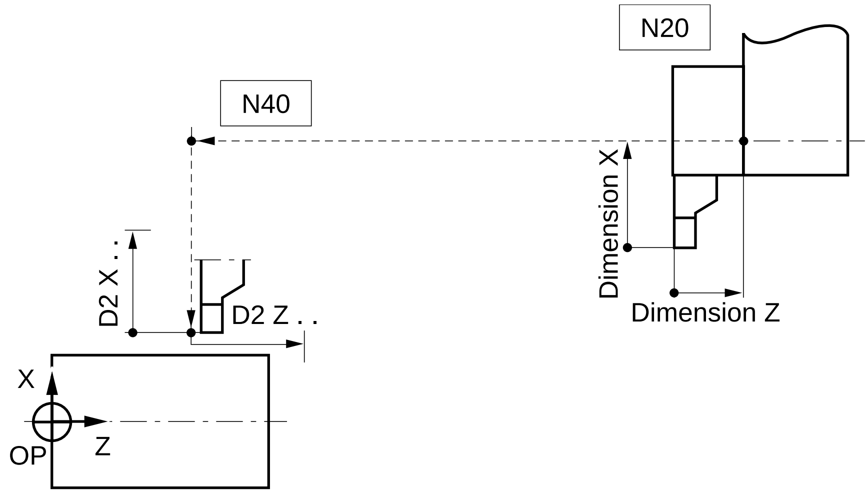
The maximum dimension of the X and Z corrections is 9999.999 mm.

REMARK *The tool axis can be a primary axis or a carried secondary axis (but not an independent secondary axis).*

Example

Machining with tool T02 to which are assigned two corrections, D02 and D12.
The X and Z corrections of tool T02 are taken into account during the first movement on the X and Z axes programmed after D02 and D12.

```
%55
N10 G00 G52 X.. Z.. G16 R+
N20 T02 D02 M06                Call of tool T02 and correction D02
N30 G97 S900 M40 M04
N40 G00 Z100                   Taking into account dimension Z.. of D02
N50 X60                        Taking into account dimension X.. of D02
N..
N100 D12 G00 X80 Z100          Taking into account dimensions X.. and
                               Z.. of D12
N..
```



Tool radius correction (R)

The tool radius correction is assigned to programming according to the two axes of the interpolation plane (X, U, Z, W, etc.).

The tool radius declared is taken into account when programming:
- the correction number D..,
- one of functions G41 or G42,
- one of the axes of the interpolation plane.

During machining, a change in the tool radius is made by canceling the radius offset by G40 then reprogramming the radius offset by G41 or G42 after:
- changing the correction number,
- changing the tool wear compensation.

The maximum dimension of R correction is 999.999 mm.

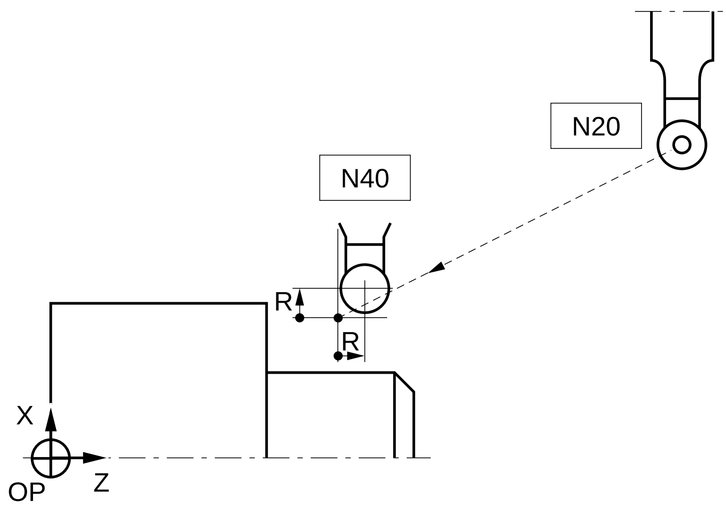
REMARK *The two axes of the interpolation plane can be primary, secondary, carried or independent axes.*

Example

Machining with tool T05 with two corrections, D05 and D15.

Radius correction R.. of tool T05 is taken into account by reading functions G41 or G42 and a movement on one of the axes of the plane programmed after D...

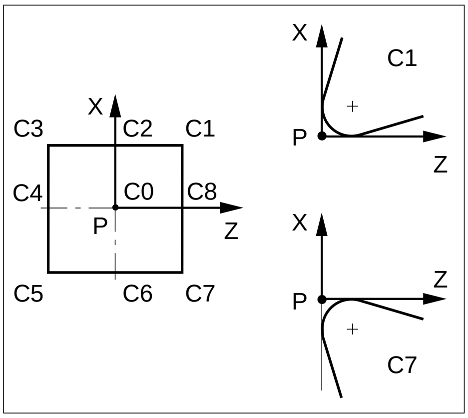
```
%65
N10 G00 G52 X.. Z..                     Tool change position
N20 T05 D05 M06                         Call of tool T05 and correction D05
N30 S900 M40 M04
N40 G00 G42 (or G41) X60 Z120           Activation of radius offset R in D05.
N70
N..
N150 G00 G40 Z60                        D05 radius offset cancel
N160 G41 (or G42) X100 Z50 D15          Activation of radius offset R in D15
N..
N300 G00 G40 X200 Z100                  D15 radius offset cancel
N..
```



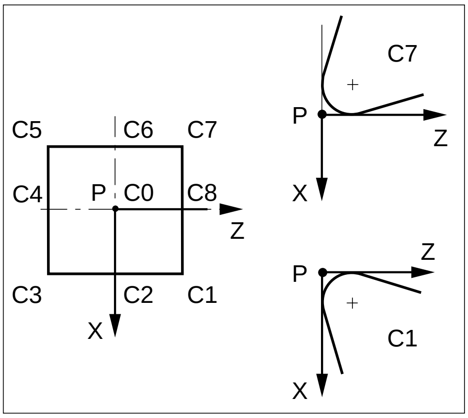
Tool tip orientation (C) according to the turret position

The tool tip orientation is defined by codes C0 to C8 (see Sec. 1.2.5.2).

Rear turret



Front turret



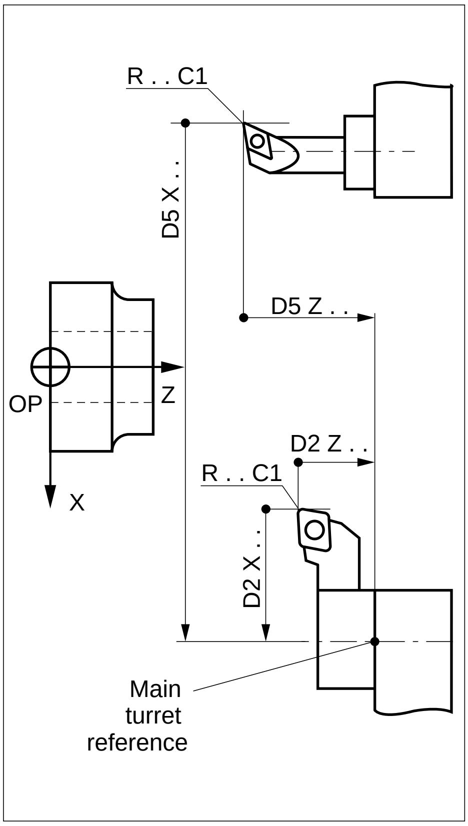
Use of tool corrections (X, Z, R, C) on a lathe with two mechanically coupled turrets

Case of a main front turret.

The dimensions of the tools in the secondary turret are defined with respect to the reference point of the main turret.

In this case, the tool tip orientation is defined with reference to the front position of the main turret, i.e. C1.

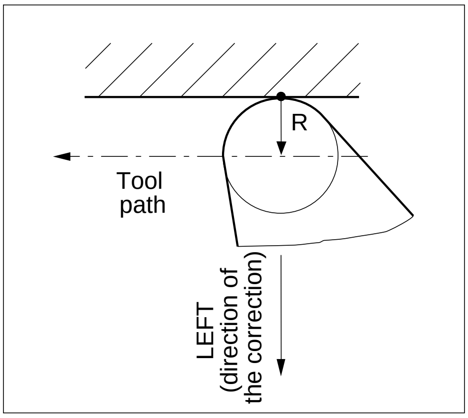
Definition of the tool tip orientation is only important when radius offset G41 and G42 are programmed. With G40 (radius offset cancelled), C0 is defined.



4.8.4 Positioning the Tool with Respect to the Part

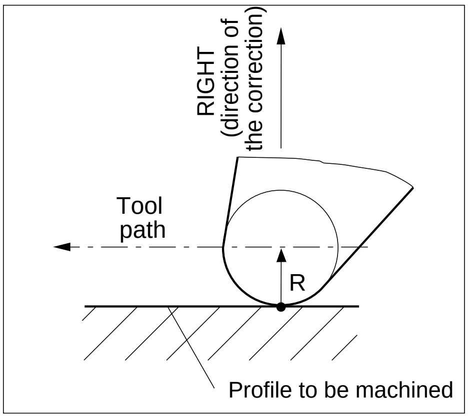
G41	Left radius offset.

The tool paths programmed are corrected (offset to the left) by a value equal to the tool radius (R) declared by corrector D...



G42	Right radius offset.

The tool paths programmed are corrected (offset to the right) by a value equal to the tool radius (R) declared by corrector D...



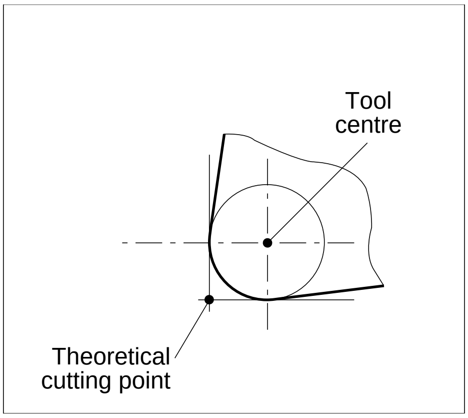
Syntax

N.. [D..] [G00/G01/G02/G03] **G41/G42** X.. Z..

D..	Corrector number containing the tool radius offset.
G00/G01/G02/G03	Linear or circular interpolation.
G41	Radius offset to the left of the profile.
G42	Radius offset to the right of the profile.
X.. Z..	End point.

G40	Radius offset cancel.

Control of the theoretical tool cutting point. Radius offset is no longer applied to the tool.



Syntax

N.. [G00/G01] **G40** X.. Z..

G00/G01	Linear interpolation.
G40	Tool radius offset cancel.
X.. Z..	End point.

Properties of the Functions

Functions G40, G41 and G42 are modal.

Function G40 is the default function.

Cancellation

Modal functions G41/G42 cancel one another.

Modal function G40 cancels function G41 and G42.

Notes

Functions G41 or G42 allow programming of a part profile in real profile dimensions without taking the tool radius into account.

With radius correction:
- the paths defining the part profile are followed even if the tool radius used and stored is smaller or larger than the theoretical tool radius programmed,
- the tool is positioned to the left or right of the profile to be machined with respect to the direction of movement along the path.

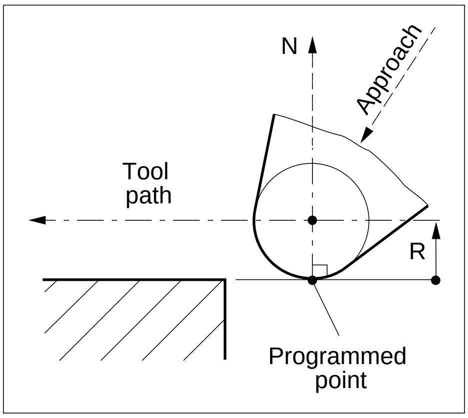
Radius correction is carried out along a vector perpendicular to the profile with radius R.. declared in correction D as the vector length.

The following functions must be programmed without radius offset (system in state G40), or the system returns error message 140.
- M00 (programme stop),
- M01 (optional programme stop),
- M02 (end of programme),
- G52 (programming with respect to the measurement origin).
- $0 (message transmission),
- L100 to L199 and L900 to L959 (programme variables, see Sec. 6.1),
- E800xx and E8x000 to E8x999 (external parameters, see Sec. 6.2).

When changing the direction of correction (change from G41 to G42 or vice versa), it is not necessary to cancel the radius offset by G40.

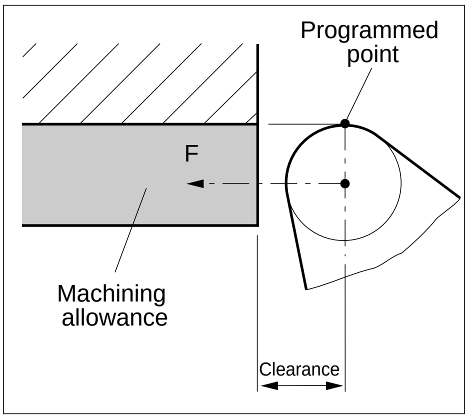
Tool positioning

At the end of the first block programmed with radius correction (which must be linear), the tool centre is positioned:
- on the normal (N) to the next path,
- offset from the programmed point by the value of the radius correction (R).



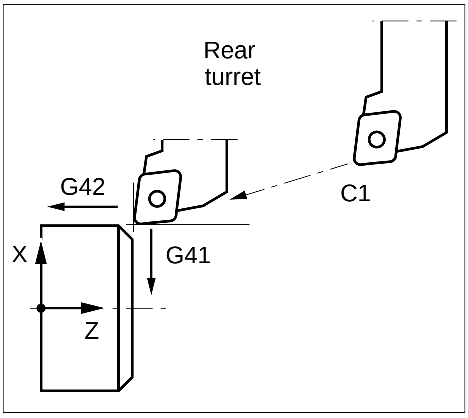
Precaution for positioning the tool

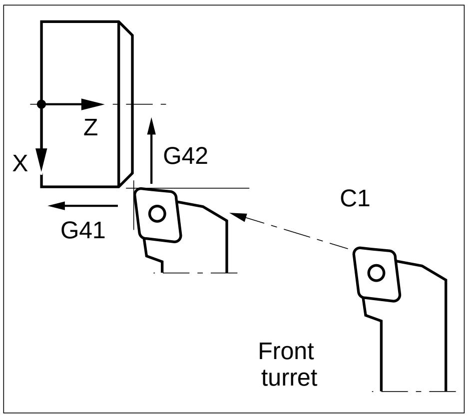
When rapid positioning, provide a clearance with a value higher than the declared tool radius.



Radius correction according to the turret position

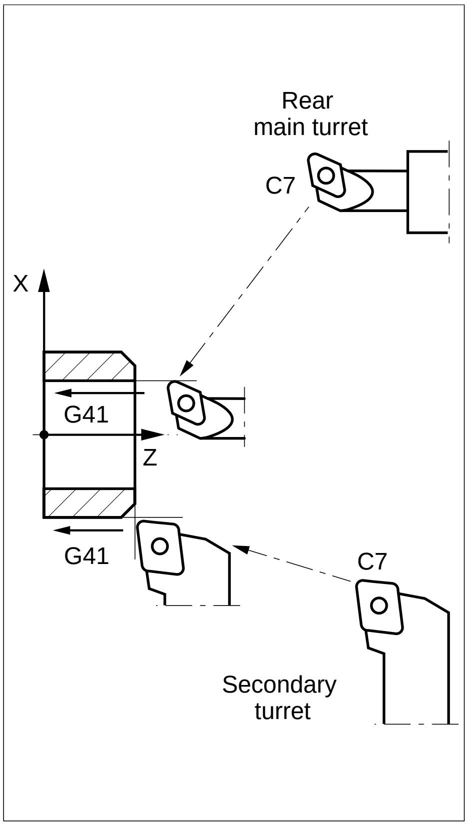
Lathe with one turret

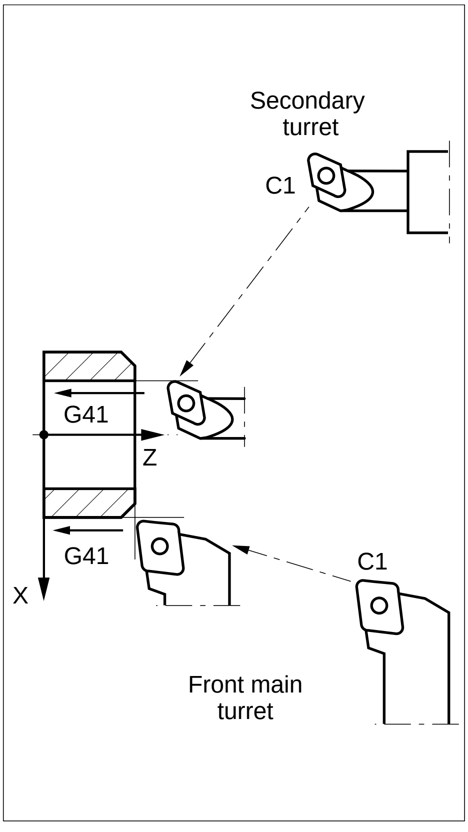




Lathe with two mechanically coupled turrets

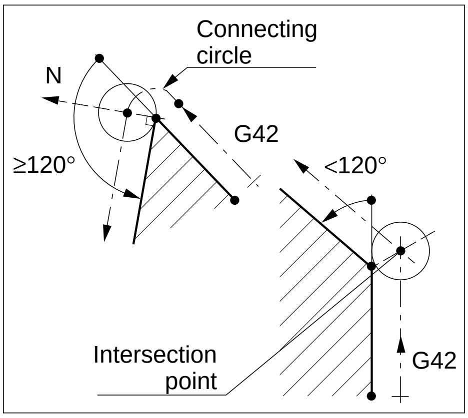
The tool tip orientation (C) is defined with reference to the main turret.

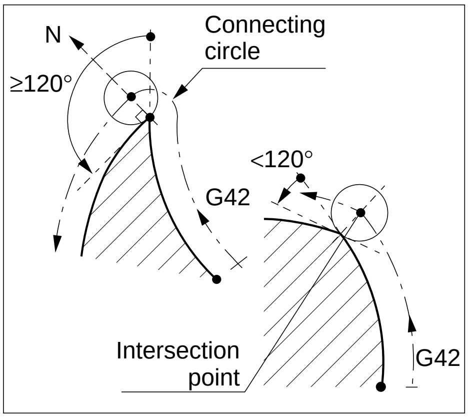




Tool outside the profile (line/line or circle/circle)

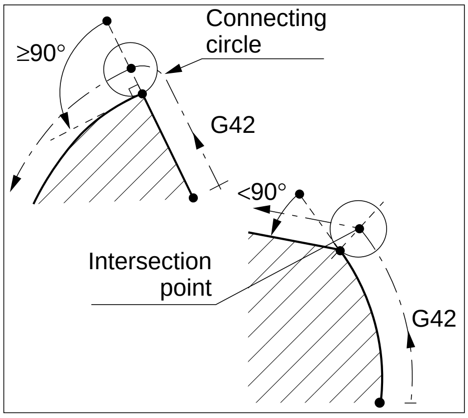
At the end of the block, the tool centre is positioned:
- offset from the programmed point, on the normal to the next path (angle ≥ 120 degrees) after describing a connecting arc,
- on the point of intersection between the current and next path (angle < 120 degrees).





Tool outside the profile (line/circle)

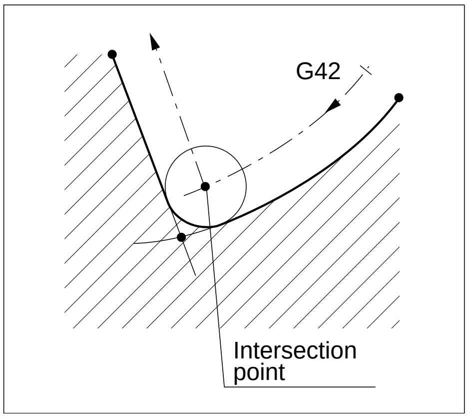
At the end of the block, the tool centre is positioned:
- offset from the programmed point on the normal to the next path (angle ≥ 90 degrees) after describing an intersecting arc,
- on the point of intersection between the current and next path with offset (angle < 90 degrees).



Tool inside the profile

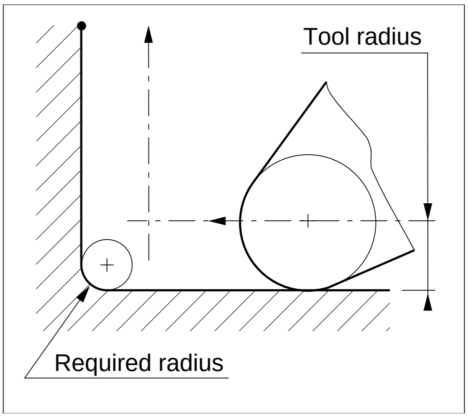
The tool follows the path until its centre reaches the intersection point between the current and next path with offset.

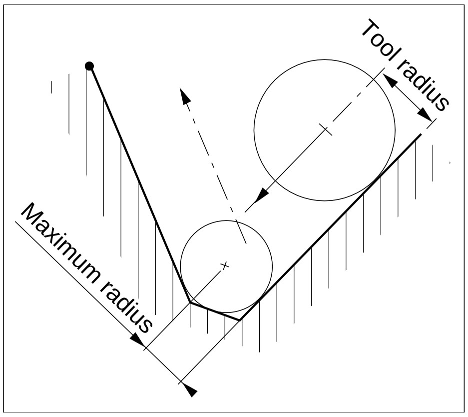
The programmed point is not reached: the shape of the tool generates a fillet between the two consecutive paths.



Tool inside the profile, special cases

When the tool size is too large to be tangent to one of the programmed paths (path radius less than the tool radius or path inaccessible), the system returns error message 149.

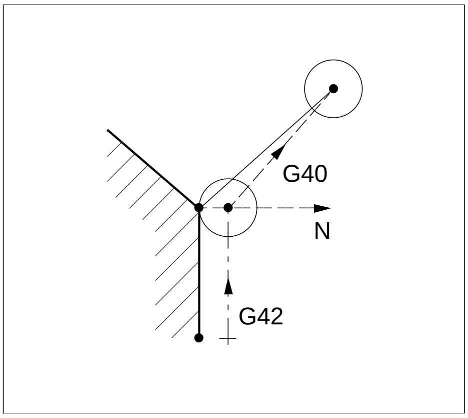




Tool retraction

At the start of the first block programmed with tool radius offset cancelled (must be a line), the tool centre starts:
- from the normal to the previous path,
- offset from the programmed point by the value of the radius correction.

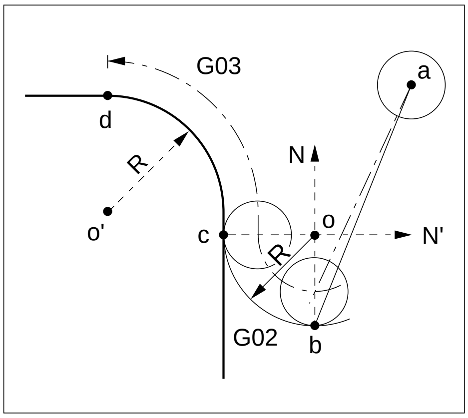
At the end of the move, the tool centre coincides with the programmed point end point.



Examples

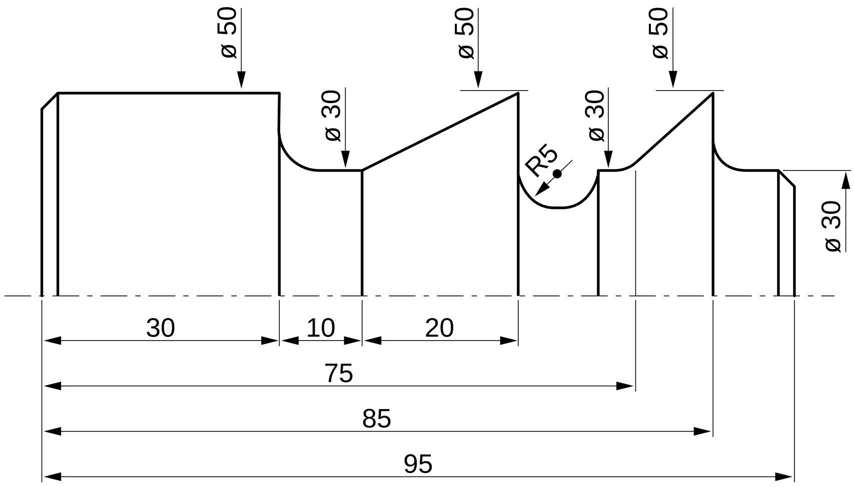
Engagement on an outside circle in G03

```
N.. ...
N.. D04
N40 Xa Za
N50 G01 G42 Xb Zb F..
N60 G02 Xc Zc Io Ko F..
N70 G03 Xd Zd Io' Ko'
N..
```

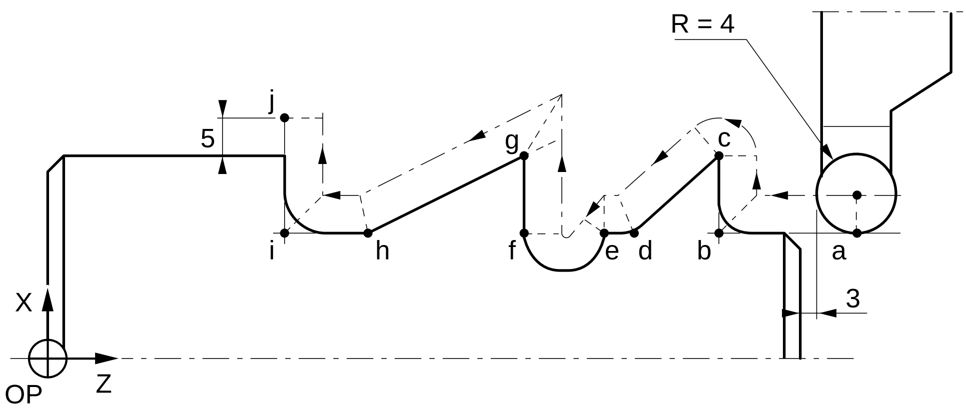


Contouring an external profile with right radius correction

Unspecified radii = Tool radius

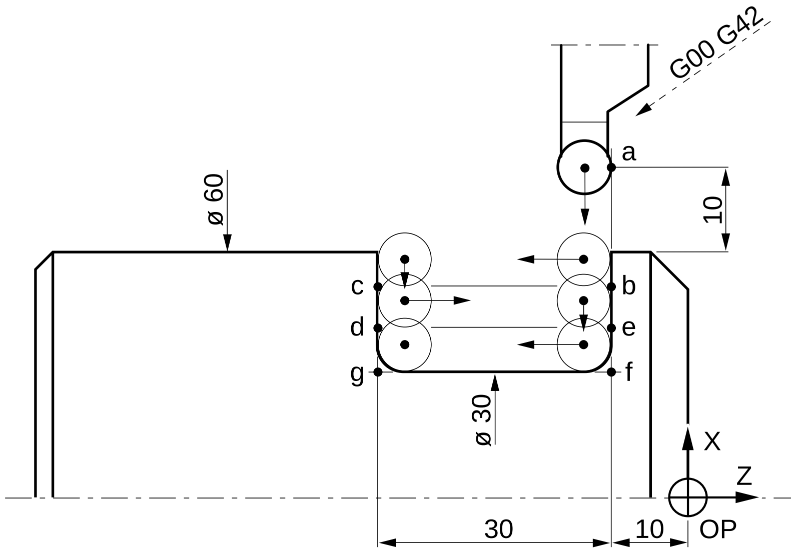


Machining paths (finishing)



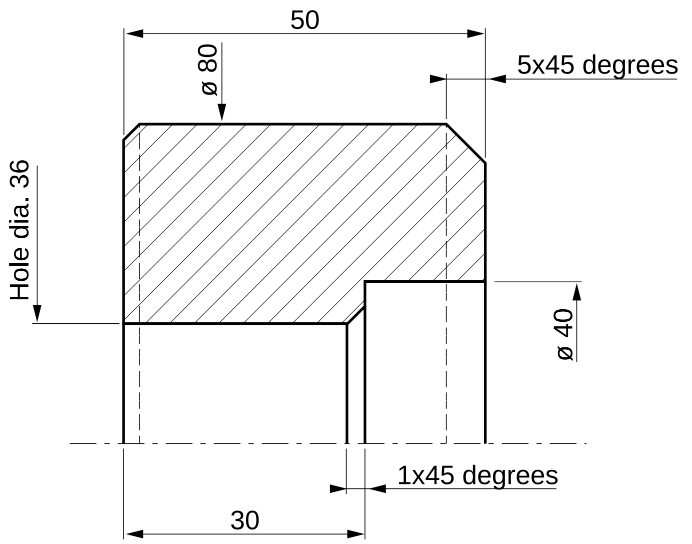
%37	
N10 G00 G52 X.. Z..	Tool change position
N20 T03 D03 M06 (COPYING TOOL R=4)	
N30 S900 M40 M04	
N40 G95 F0.2	
N50 G00 X30 Z102	Point a, right radius offset
N60 G92 S3500	
N70 G96 S200	
N80 G01 Z85	Point b
N90 X50	Point c
N100 X30 Z75	Point d
N110 Z70	Point e
N120 G02 X30 Z60 R5 F0.05	Point f
N130 G01 X50 F0.2	Point g
N140 X30 Z40	Point h
N150 Z30	Point i
N160 X60	Point j
N170 G97 S900	
N180 G00 G40 G52 X.. Z.. M05	Radius offset cancel
N190 M02	

Machining a groove by reciprocating cuts with alternating radius correction (G41-G42)

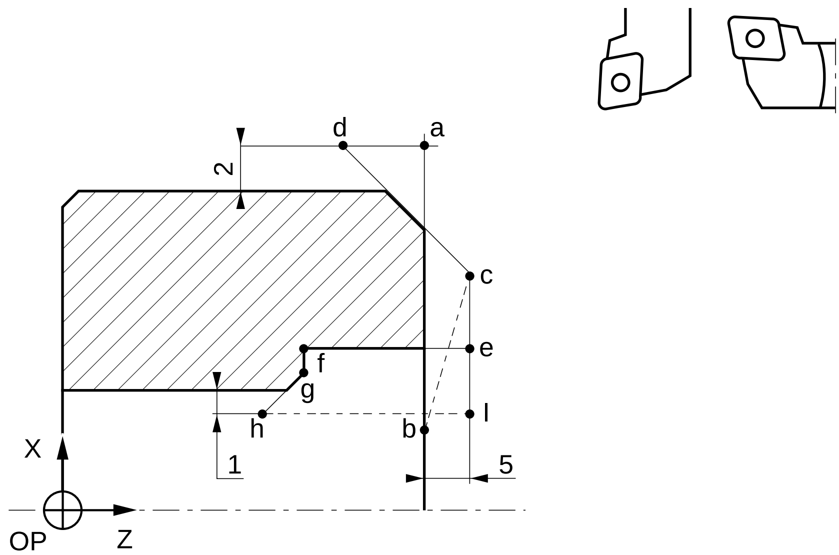


%85	
N10 G00 G52 X.. Z..	Tool change position
N20 T01 D01 M06 (COPYING TOOL R=4)	
N30 S900 M40 M04	
N40 G95 F0.1	
N50 G00 G42 X80 Z-10	Point a, right correction
N60 G92 S3000	
N70 G96 S100	
N80 G01 X50	Point b
N90 Z-40 F0.2	Point c
N100 G41 X40 F0.1	Point d, left correction
N110 Z-10	Point e
N120 G42 X30	Point f, right correction
N130 Z-40 F.2	Point g
N140 X80	
N150 G97 S900	
N160 G00 G40 G52 X.. Z.. M05	Radius offset cancel
N170 M02	

External machining and internal contouring after tool retraction and positioning



Machining paths



```
%72
N10 G00 G52 X.. Z..                      Tool change position
N20 T01 D01 M06 (OUTSIDE TOOL R=0.8)
N30 S900 M40 M04
N40 G95 F0.12
N50 G00 G41 X84 Z50                      Point a, left correction
N60 G92 S3500
N70 G96 S180
N80 G01 X30                              Point b
N90 G00 G42 X60 Z55                      Point c, right correction
N100 G01 X84 Z43 F0.08                   Point d
N110 G97 S900
N120 G00 G40 G52 X.. Z..                 Radius offset cancel
N130 T07 D07 M06 (BORING TOOL R=0.4)
N140 G41 X40 Z55                         Point a, left correction
N150 G96 S100
N160 G01 Z30 F0.1                        Point f
N170 X38                                 Point g
N180 X34 Z28                             Point h
N190 G00 Z55                             Point i
N200 G97 S900
N210 G40 G52 X.. Z.. M05                 Radius offset cancel
N220 M02
```

4.9 Basic Cycles

4.9.1 Cancellation of a Canned Cycle

G80	Canned cycle cancel.

This function cancels canned cycles.

Syntax

N.. **G80**

G80	Canned cycle cancel.

Properties of the Function

Function G80 is modal. It is the default function.

Cancellation

Modal function G80 is cancelled by one of functions G64, G81-G85, G87 or G89.

Example

```
N..
N110 G00 X.. Z..                 Tool positioning
N120 G94 (or G95) F..
N130 G83 Z-10 P8                 Drilling cycle
N140 G80 G00 G52 X.. Z..         Cycle cancel
N..
```

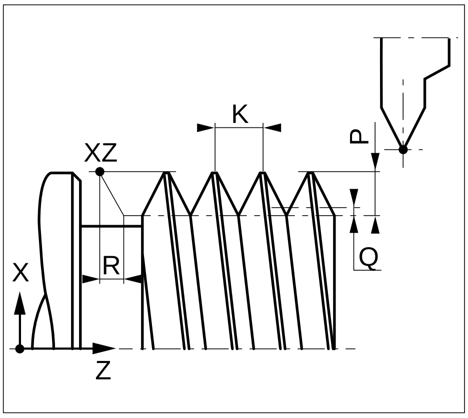
4.9.2 Constant Pitch Thread Chasing Cycle

G33	Constant pitch thread chasing cycle.

This function is used for cutting parallel, tapered or scroll threads.

The threads can be single or multiple start and can be cut by straight or angular penetration.

The consecutive passes are carried out with reducing depths of cut.



Syntax

N.. **G33** X.. Z.. K..[EA..][EB..]P..[Q..][R..][F..][S..] / [ES..]

G33	Constant pitch thread chasing cycle.
X.. Z..	Tool position at the end of thread cutting with respect to the programme origin.
K..	Pitch in the thread cutting axis (X or Z) in mm.
EA..	Angle for tapered thread (see Fig. 5). - The default value of EA = 0: parallel thread cutting - EA90: scroll thread cutting.
EB..	Tool penetration angle (see Figs. 3 and 4) - The default value of EB = 0: straight penetration. - EB > 0: penetration down the thread flank in the direction of machining. - EB < 0: penetration down the thread flank opposite the machining direction.
P..	Total thread depth (including Q).
Q..	Depth of the final sizing pass (included in P) - Default no sizing pass (omit Q..). - Q = 0: spring cut required.

R..	Length of the tool retraction slope at the end of a pass (in the thread cutting axis) (see Figs. 1 and 2). - The default value of R = 0: retraction perpendicular to the thread cutting axis.
F..	Number of thread starts (format F01) (see Fig. 6). - Default single start.
S../ES..	Number of passes (format S02 or ES02) (not including pass Q), default 1 pass: - S..: Passes with reduced depth of cut, - ES..: Constant value passes.

Property of the Function

Function G33 is nonmodal.

Cancellation

Function G33 is cancelled at the end of the block.

Notes

The modal functions present before the G33 cycle are restored after execution of the cycle.

The thread pitch is applied to the thread cutting axis (axis on which the movement is largest).

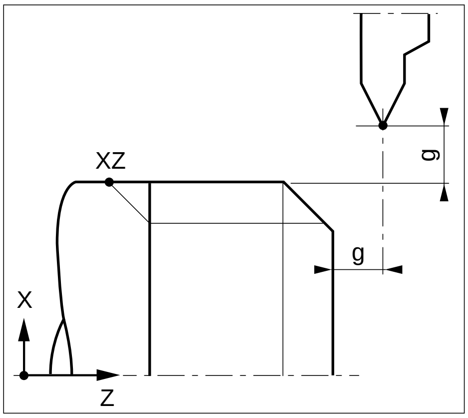
During the cycle, spindle speed override is inhibited (value forced to 100 per cent).

REMARK *The maximum possible pitch is limited by the feed rate in mm/min acceptable by the machine (feed rate = pitch x rotation speed). Refer to the machine manufacturer's documentation.*

Thread cutting start point

The tool must be backed off from the material by a clearance (g).

The clearance in the axis of penetration is mandatory. If the tool positioning and end of thread cutting values in the axis of penetration are the same, the system will generate an error message.



Cycle Steps

The steps below are given for information. The thread cutting cycle contains only the main parameters.

```
N.. ...
N220 G00 Xa Za                    Rapid to start of thread cutting
N230 G33 Xb Zb K.. P.. S..
N..
```

Step 1: Tool positioning in X and Z to the thread cutting start point (point a).

Step 2: Rapid advance in the axis of penetration (first pass).

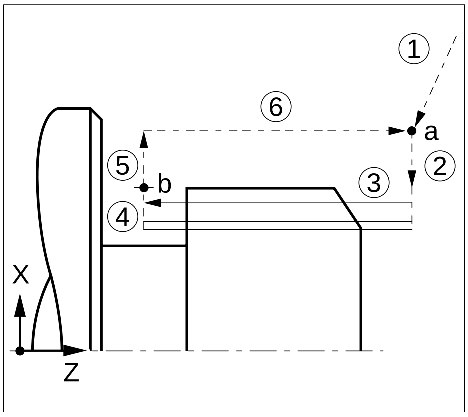
Step 3: Execution of the first pass in the thread cutting axis Z (point Zb).

Step 4: Retraction along X (point Xb).

Step 5: Rapid retraction of the penetration axis.

Step 6: Rapid return to the thread cutting start point.

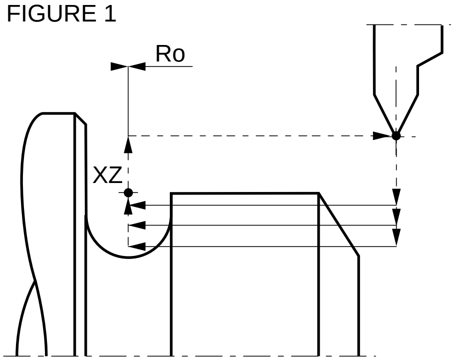
Successive passes are executed in the same way as steps 2 to 6 and may be followed by a spring pass.



Tool retraction slope at the end the pass defined by R

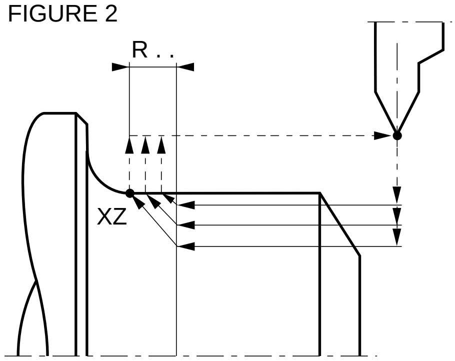
Thread cutting without retraction slope

FIGURE 1



Thread cutting with retraction slope

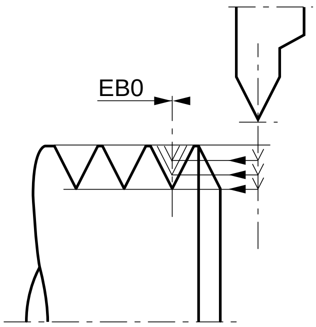
FIGURE 2



Penetration defined by EB

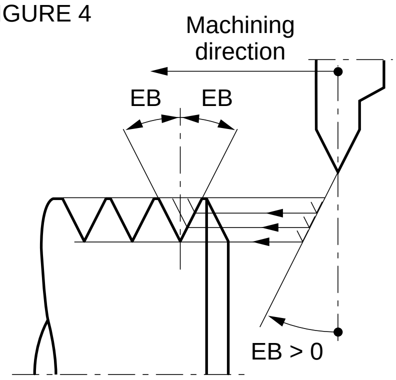
Thread cutting by straight penetration

FIGURE 3



Thread cutting by penetration down the thread flank

FIGURE 4



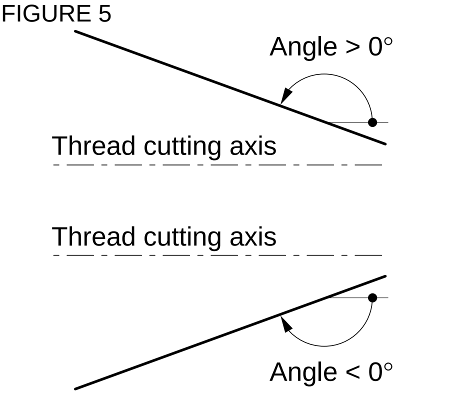
Tapered thread cutting, angle defined by EA

The value of EA is positive in the counterclockwise direction, modulo 180 degrees.

i.e. -45 deg ≤ EA ≤ 45 deg for:
- Z thread cutting axis
- X penetration axis.

I.e. EA > 45 deg or EA < -45 deg for:
- Z axis of penetration
- X thread cutting axis.

FIGURE 5

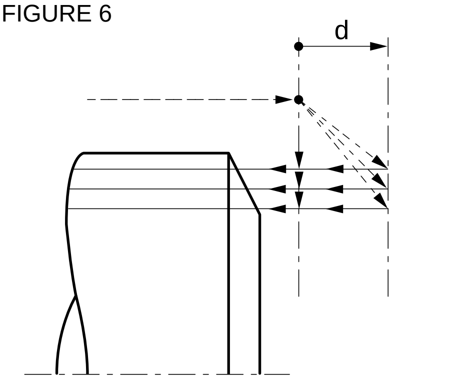


Multiple start thread cutting

The offset «d» (pitch) corresponding to each start is in the direction opposite the thread cutting direction.

Each start will be completed at one depth before continuing to the next pass depth.

FIGURE 6



Reminders

Determination of the pass depth (P) for ISO threads.

External thread: 0.613 x pitch

Internal thread: 0.577 x pitch

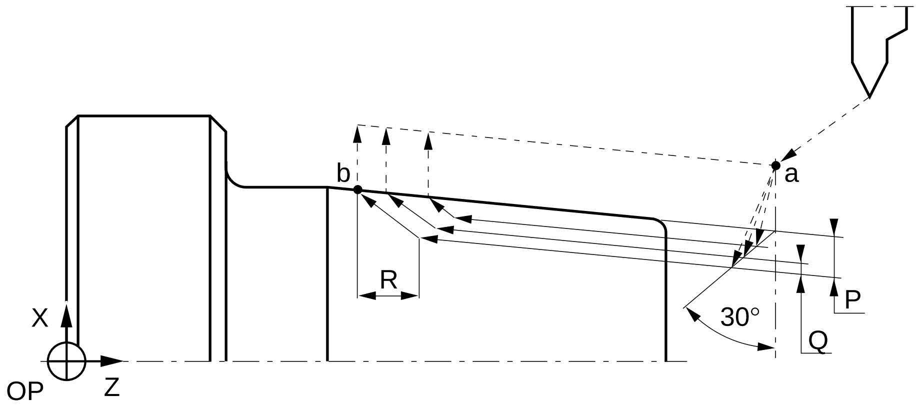
Determination of the approximate number of passes (S)

S = P x 7

- Coefficient: 7
- Pass depth: P
- Approximate number of passes (integer): S

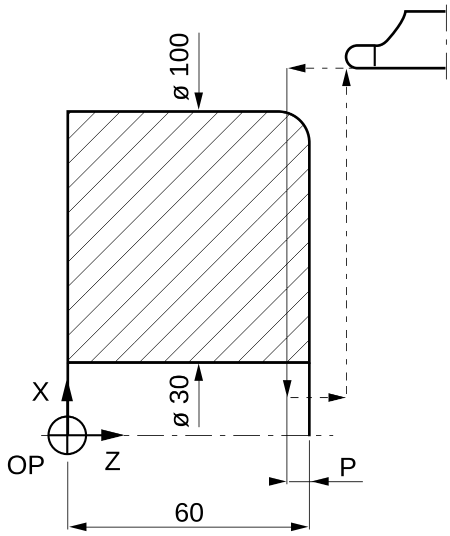
Examples

Cutting of external tapered thread (5 deg)




```
N.. ...
N140 T09 D09 M06 (THREAD CUTTING TOOL, PITCH=1 RH THREAD)
N150 G97 S1000 M40 M03
N160 G00 Xa Za                        Start of thread cutting
N170 G33 Xb Zb K1 EA175 EB30 P0.61 Q0.02 R4 S5
N.. ...
```

Cutting of front threads




```
N.. ...
$ FRONT THREAD ALONG X
N140 T05 D05 M06 (TOOL R=2)
N150 G97 S200 M40 M03
N160 G00 X110 Z64
N170 G33 X20 Z60 K4 EA90 P2 Q0 S14
N..
```

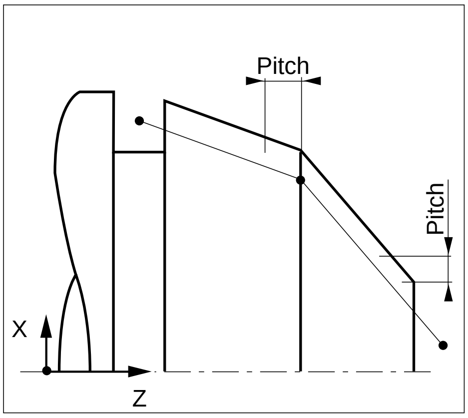
Point a, start of thread cutting

4.9.3 Sequenced Thread Cutting

G38	Sequenced thread cutting.

This functions allows the execution of several consecutive thread cutting blocks.

The threads can be cylindrical or tapered.



Syntax

N.. **G38** X.. Z.. K..

G38	Sequenced thread cutting.
X.. Z..	Tool position at the end of thread cutting with respect to the programme origin.
K..	Pitch in the thread cutting axis (X or Z) in mm.

Property of the Function

Function G38 is modal.

Cancellation

Function G38 is cancelled by one of functions G00, G01, G02 or G03.

Notes

Different pitches (K) can be programmed in consecutive thread cutting blocks with G38.

The thread pitch is applied to the thread cutting axis (axis on which the movement is largest).

The consecutive passes, to obtain the required depth, can be programmed by subroutine calls (see Sec. 4.11.1).

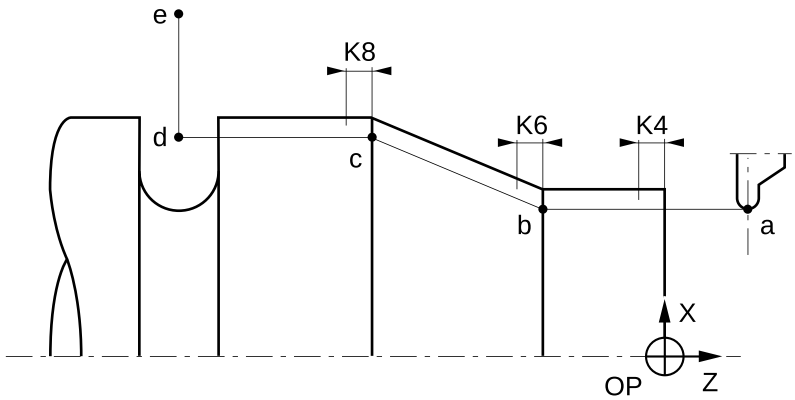
During execution of the cycle, spindle speed override by potentiometer is inhibited (value forced to 100 percent).

Function G38 must be programmed:
- without miscellaneous functions (M) in the cycle blocks,
- without use of constant surface speed (nor retract in step for successive passes).

REMARK *The maximum possible pitch is limited by the feed rate in mm/min acceptable by the machine (feed rate = pitch x rotation speed). Refer to the machine manufacturer's documentation.*

Example

Sequenced external thread cutting



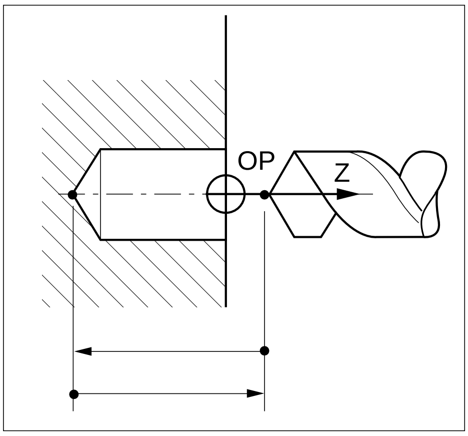

```
N.. ...
N100 T04 D04 M06 (TOOL R=2)
N110 G97 S400 M40 M03
N120 G00 Xa Za              Approach point
N130 G38 Xb Zb K4           Cycle with 4 mm pitch
N140 Xc Zc K6               Cycle with 6 mm pitch
N150 Xd Zd K8               Cycle with 8 mm pitch
N160 G00 Xe Ze              Cycle cancelled
N..
```

4.9.4 Drilling, Boring and Tapping Cycles

4.9.4.1 General

G8x	Machining cycles in the tool axis.

Axes programmable with the basic cycles:
- primary axes X, Z,
- secondary axes U, W,
- rotary axes A or C are used only for positioning when the machine is equipped with them.



Syntax

N.. **G8**x X.. / Z.. [ER..] [EH..]

G8x	Machining cycle.
X.. / Z..	End point on the machining axis.
ER..	Dimension of the approach (or retraction) plane in the machining axis.
EH..	Dimension of the impact plane in the machining axis.

Properties of the Functions

Functions G8x are modal.

Cancellation

Functions G8x are cancelled by another function G8x or by functions G64, G65, G66.

Notes

When a cycle (G8x) is programmed, the system must be in state G40 (G41 or G42 tool radius offset cancelled).

⚠ CAUTION

The use of programme variables L900 to L959 (see Chapter 6) is not recommended in a programme including canned cycles, since some of these variables could be overwritten when a cycle is called.

Dimensions ER and EH

Dimension ER of the approach (or retraction) plane on the machining axis is assigned to the primary axis (Z) or the secondary axis (W) programmed last.

Dimension EH of the impact plane on the machining axis is assigned to the primary axis (Z or X) or the secondary axis (W or U) programmed last.

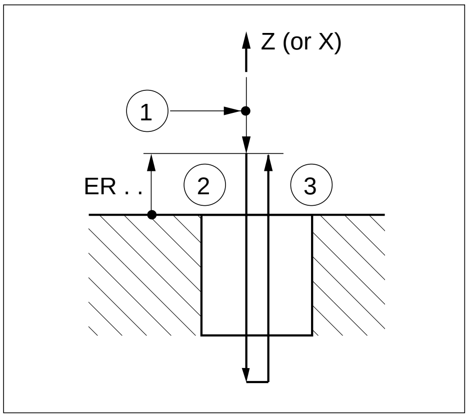
EH must always be programmed in the same block of the cycle as ER.

Cycle steps with ER (without EH)

Step 1: Rapid positioning (linear or circular) in the plane then on ER..

Step 2: Tool penetration to the value programmed on the tool axis (Z).

Step 3: Retraction along the tool axis (Z) up to ER..



ER.. not programmed:
The previous value programmed on the Z axis is used for approach.

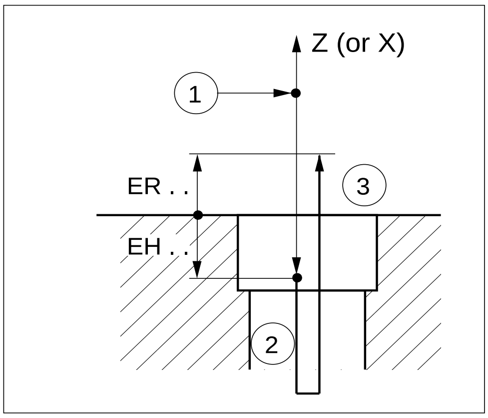
ER.. programmed alone:
The tool is positioned in the Z axis.

Cycle steps with EH and ER

Step 1: Rapid linear positioning in the axis then on EH..

Step 2: Tool penetration down to the value programmed on the tool axis (Z or X).

Step 3: Retraction along the tool axis (Z or X) up to ER..



4

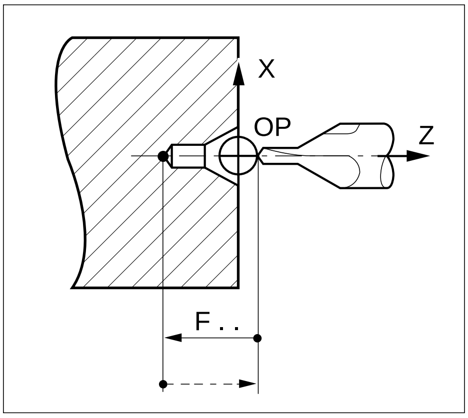
EH and ER.. programmed:
EH differentiates between the impact plane and the backoff plane.

EH.. not programmed and ER.. programmed:
The value of ER is used (ER = EH).

REMARK *The above descriptions of the cycle steps with ER alone and with ER and EH take into account only the start and end planes. For the detail of these steps, refer to the cycle concerned.*

4.9.4.2 Centre Drilling Cycle

G81	Centre drilling cycle.



Syntax

N.. **G81** X.. / Z.. [ER..] [EH..]

G81	Centre drilling cycle.
X.. / Z..	End point on the machining axis.
ER..	Dimension of the retraction plane on the machining axis.
EH..	Impact plane in the machining axis.

Property of the Function

Function G81 is modal.

Cancellation

Function G81 is cancelled by one of functions G80, G82-G85, G87, G89 or functions G64, G65, G66.

Cycle Steps

Step 1: Fast positioning on the machining axis.

Step 2: Infeed at feed rate F..

Step 3: Outfeed at high speed in the tool axis.

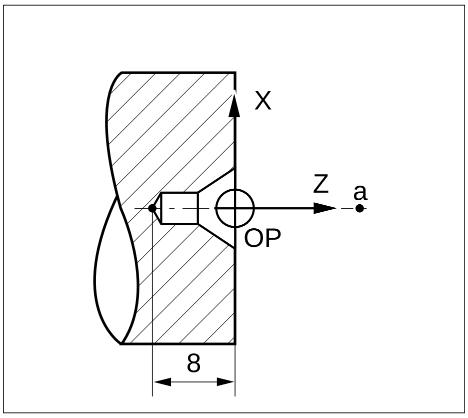
Example

Execution of centre drilling.

```
N.. ...
N50 G00 Xa Za
N60 G94 (or G95) F..
N70 G81 Z-8
N80 G80 G00 X.. Z..
N..
```

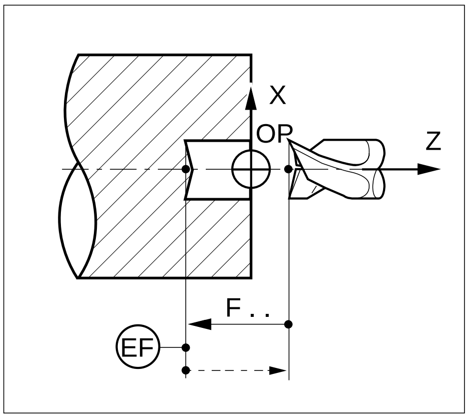
or

```
N..
N50 G94 (or G95) F..
N60 G81 Xa ERa Z-8
N70 G80 G00 X.. Z..
N..
```



4.9.4.3 Counterboring Cycle

G82	Counterboring cycle.



Syntax

N.. **G82** X.. / Z.. [ER..] [EH.] EF..

G82	Counterboring cycle.
X.. / Z..	End point on the machining axis.
ER..	Dimension of the retraction plane on the machining axis.
EH..	Impact plane in the machining axis.
EF..	Mandatory dwell expressed in seconds (maximum 99.99 s, format EF022).

Property of the Function

Function G82 is modal.

Cancellation

Function G82 is cancelled by one of functions G80, G81, G83-G85, G87, G89 or functions G64, G65, G66.

Cycle Steps

Step 1: Fast positioning in the machining axis.

Step 2: Infeed at feed rate F..

Step 3: Dwell at the end of drilling (or boring).

Step 4: Outfeed at high speed on the tool axis.

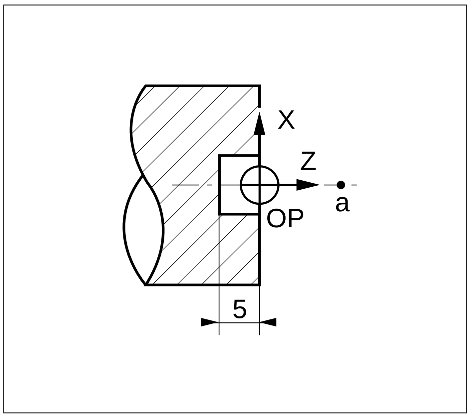
Example

Execution of counterboring

```
N.. ...
N50 G00 Xa Za
N60 G94 (or G95) F..
N70 G82 Z-5 EF2
N80 G80 G00 X.. Z..
N..
```

or

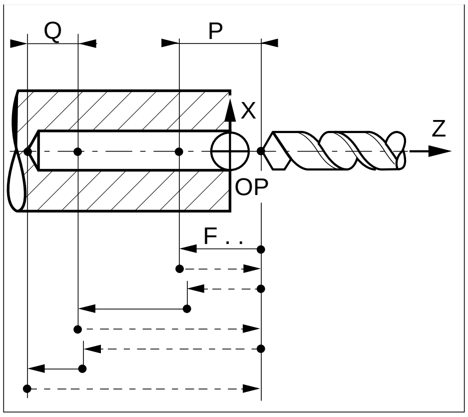
```
N..
N50 G94 (or G95) F..
N60 G82 Xa ERa Z-5 EF2
N70 G80 G00 X.. Z..
N..
```



4.9.4.4 Peck Drilling Cycle

G83	Peck drilling cycle.

This cycle can be used for drilling in the X or Z axis



Syntax

N.. **G83** X.. / Z.. [ER..] [EH..] [P..] / [ES..] [Q..] [EP..] [EF..]

G83	Peck drilling cycle.
X.. / Z..	End point on the machining axis.
ER..	Retraction plane on the machining axis.
EH..	Impact plane in the machining axis.
P..	Value of first peck.
ES..	Number of infeeds (pecks) at constant value (see Fig. 1).
Q..	Value of last peck.
EP..	Backoff after each peck (default 1 mm).
EF..	Dwell at the end of each peck.

Property of the Function

Function G83 is modal.

Cancellation

Function G83 is cancelled by one of functions G80-G82, G84, G85, G87 and G89 or functions G64, G65, G66.

Notes

If addresses P and Q are programmed, the consecutive pecks between P and Q are degressive values.

At least one of arguments P and ES must be programmed or the system returns error message 889.

If the value of P is greater than delta Z, the system returns error message 881.

Notes related to ES (number of constant infeeds)

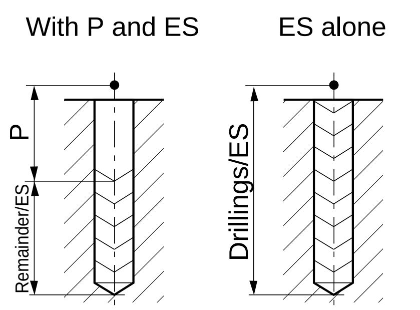
If P and ES are programmed:

The first infeed is equal to P and drilling is continued as a number ES of infeeds.

If ES is programmed alone (without P):

All the drilling is executed as a number ES of infeeds.

Figure 1



Cycle Steps

The steps below are given as an illustration. The number of steps depends on the values programmed in the cycle.

Step 1: Fast positioning in the machining axis.

Step 2: First peck to depth P.. in the tool axis at the machining feed rate.

Possible dwell at the end of the peck.

Rapid retraction to the start point.

Rapid infeed to 1 mm (or EP..) above depth P..

Step 3: Second peck at feed rate.

Possible dwell at the end of the peck.

Rapid retraction to the start point.

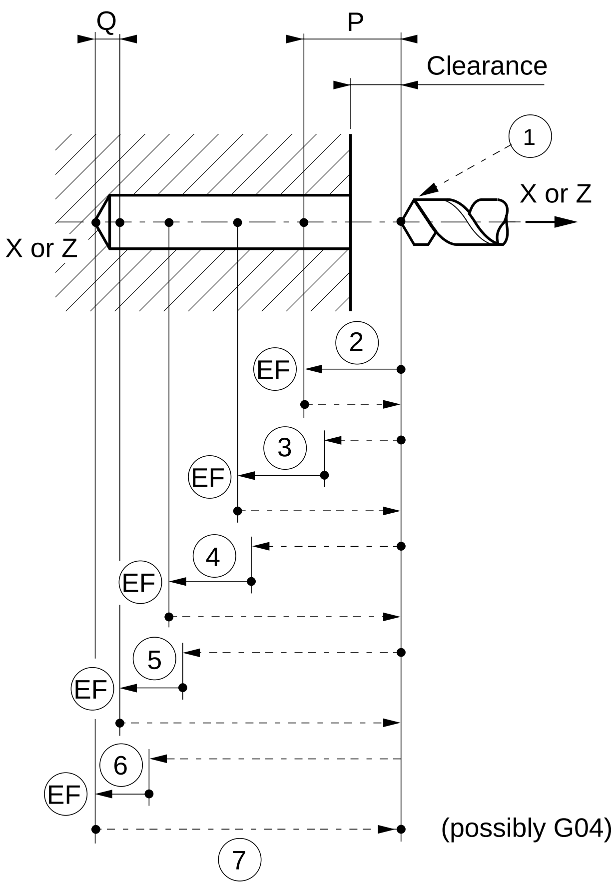
Rapid infeed to 1 mm (or EP..) above the previous peck.

Steps 4 and 5: Pecks and retractions same as step 3.

Step 6: Peck to depth Q.. at feed rate.

Step 7: Rapid retraction to the start point.

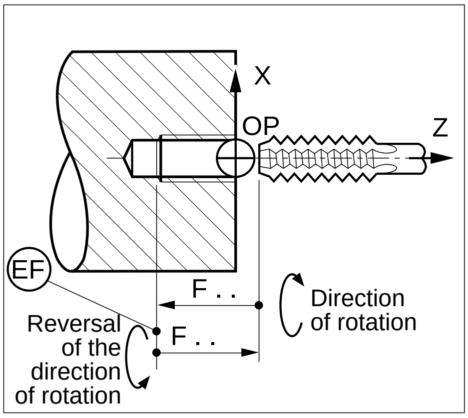
Possible dwell G04 F.. in the start point.


Q
P
Clearance
1
X or Z
X or Z
2
EF
3
EF
4
EF
5
EF
6
EF
(possibly G04)
7

4.9.4.5 Tapping Cycle

G84	Tapping cycle.

This cycle is used to tap a hole with a floating tap-holder.



Syntax

N.. **G84** X.. / Z.. [ER..] [EH..] EF..

G84	Tapping cycle.
X.. / Z..	End point on the machining axis.
ER..	Dimension of the retraction plane on the machining axis.
EH..	Impact plane in the machining axis.
EF..	Dwell expressed in seconds (maximum 99.99 s, format EF022, default 1 second).

Property of the Function

Function G84 is modal.

Cancellation

Function G84 is cancelled by one of functions G80-G83, G85, G87, G89 or functions G64, G65, G66.

Notes

In this tapping cycle, since feed is not slaved to spindle rotation, the tap must be floating to compensate for position errors.

During execution of the cycle, feed rate override by potentiometer is inhibited (value forced to 100 percent).

Determination of the feed rate in mm/min

F.. = Tap pitch (in mm) x spindle speed rotation (revolutions/minute).

Cycle Steps

Step 1: Fast positioning in the machining axis.

Step 2: Infeed at the specified feed rate F..

Step 3: Reversal of the direction of rotation at the end of tapping.

Step 4: Dwell at the end of tapping.

Step 5: Outfeed at the specified feed rate F.. along the tool axis.

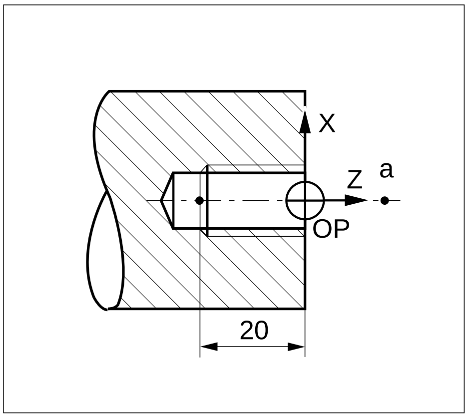
Example

Execution of one M8 tapping, pitch 1.25.

```
N.. ...
N40 S300 M41 M03
N50 G00 Xa Za
N60 G94 (or G95) F..
N70 G84 Z-20 EF1
N80 G80 G00 X.. Z..
N..
```

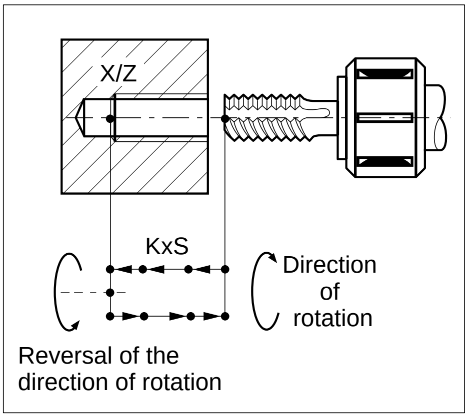
or

```
N..
N50 G94 F375
N60 G84 Xa ERa Z-20 EF1
N70 G80 G00 X.. Z..
N..
```



4.9.4.6 Rigid Tapping Cycle

G84	Rigid tapping cycle.



This cycle slaves the tap to the spindle rotation. The feed rate is automatically calculated from the actual spindle speed and pitch.

This cycle can be used to programme tapping in the X or Z axes.

Syntax

N.. **G84** X.. / Z.. K.. [ER..] [EH..] [EK..]

G84	Tapping cycle.
X.. / Z..	End point on the machining axis.
K..	Tap pitch in mm (K specifies that tapping is rigid).
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
EK..	Spindle outfeed/infeed ratio (the default value of EK=1).

Property of the Function

Function G84 is modal.

Cancellation

Function G84 is cancelled by one of functions G80-G83, G85, G87 and G89 or functions G64, G65, G66.

Notes

When the cycle is called, the tool axis (X or Z) is coupled with spindle rotation.

When the tapping axis is not located on the centre line of rotation of the part, the machine must be equipped with:
- a C axis,
- a measured auxiliary spindle.

During execution of the cycle:
- the following error on the tapping axis is cancelled during feed at constant speed,
- feed rate and spindle speed override by potentiometer is inhibited (value forced to 100 percent).

In the end-of-tapping region, the spindle is decelerated then rotation is reversed.

At the end of the cycle, the spindle is returned to its initial state.

Rigid tapping can be carried out over several penetrations, but this requires programming several consecutive blocks.

Rigid tapping can be carried out over several penetrations, but this requires programming several consecutive blocks.

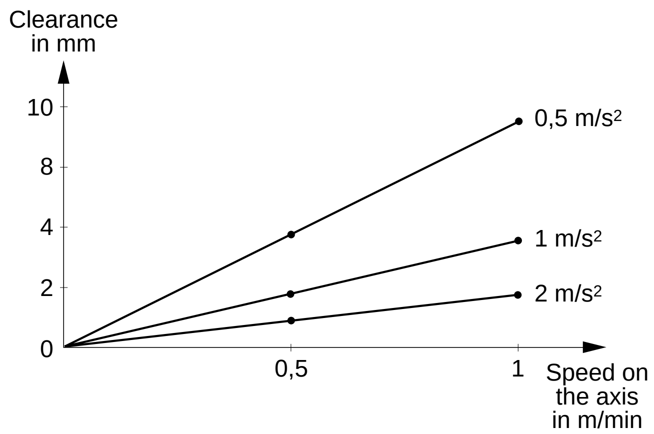
With rigid tapping, the system returns error message 899 in the following cases:
- use with an axis group number above 5,
- use with a spindle number above 2,
- the axis group does not control the spindle used or does not provide its measurement.

Tapping Clearance

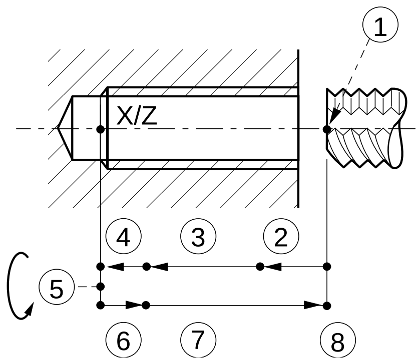
Before starting the cycle, sufficient clearance must be provided to allow the tapping axis to reach correct speed before impacting the material. This clearance depends on the required tapping speed and the acceleration allowed on the axis. The following chart can be used to approximate the clearance required.

Use of the chart for tapping an M10 hole (pitch = 1.5 mm), for instance:
- rotation speed = 320 rpm,
- axis feed rate = 480 mm/min or 0.48 m/min,
- acceleration = 0.5 m/s^2.

Required clearance read on the chart: approximately 4 mm



Cycle Steps



Step 1: Rapid to start of hole (provide clearance).

Step 2: Infeed with acceleration of the spindle speed and feed rate.

Step 3: Feed at constant speed.

Step 4: Deceleration before reaching depth.

Step 5: Reversal of the direction of rotation.

Step 6: Retraction with acceleration over a distance equal to the deceleration.

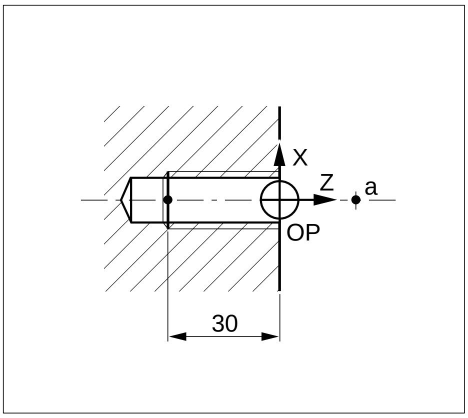
Step 7: Feed at constant speed.

Step 8: Return of the spindle to the initial state.

Examples

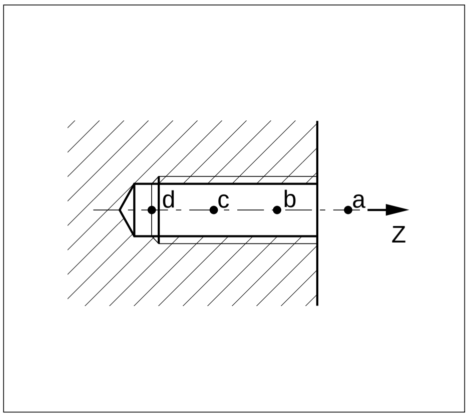
Execution of one M10 hole, pitch 1.50

```
N.. ...
N130 G97 S200 M41 M03
N140 G00 Xa Za
N150 G84 Z-30 K1.5 EK2
N160 G80 G00 G52 X.. Z..
```



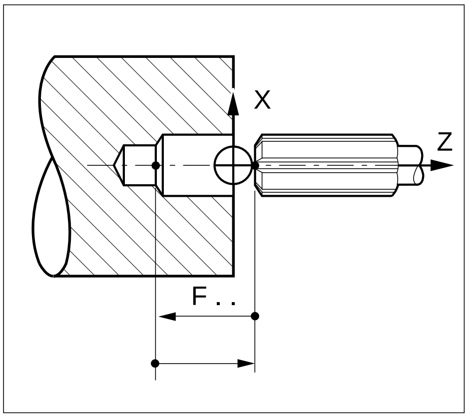
Execution of one tapped hole with several consecutive penetrations

```
N.. ...
N240 G97 S400 M41 M03
N250 G00 Xa Za
N260 G84 Zb K.. EK..
N270 Zc
N280 Zd
N290 G80 G00 G52 X.. Z..
N..
```



4.9.4.7 Boring Cycle

G85	Boring cycle.



Syntax

N.. **G85** X.. / Z.. [ER..] [EH..] [EF..]

G85	Boring cycle.
X.. / Z..	End point on the machining axis.
ER..	Dimension of the retraction plane on the machining axis.
EH..	Impact plane in the machining axis.
EF..	Retraction rate (default = feed rate F..).

Property of the Function

Function G85 is modal.

Cancellation

Function G85 is cancelled by one of functions G80-G84, G87 and G89 or functions G64, G65, G66.

Cycle Steps

Step 1: Fast positioning in the machining axis.

Step 2: Infeed at feed rate F..

Step 3: Outfeed at feed rate F.. along the tool axis.

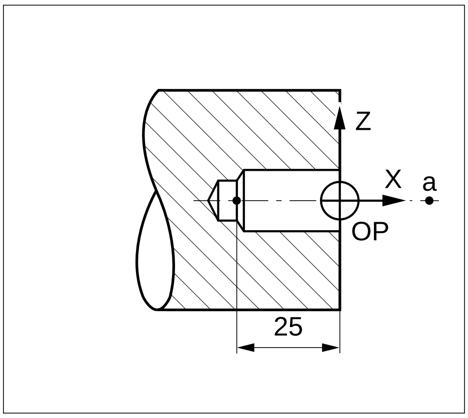
Example

Execution of a bore.

```
N.. ...
N50 G00 Xa Za
N60 G94 (or G95) F..
N70 G85 Z-25
N80 G80 G00 X.. Z..
N..
```

or

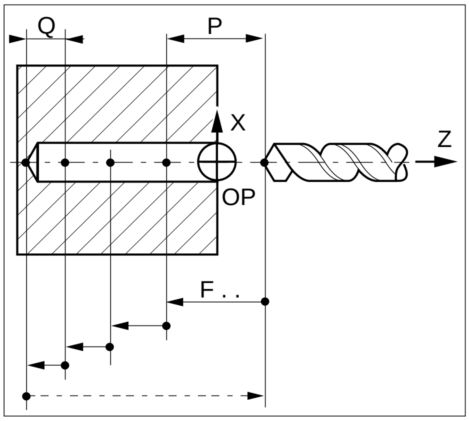
```
N..
N50 G94 (or G95) F..
N60 G85 Xa ERa Z-25
N70 G80 G00 X.. Z..
```



4.9.4.8 Drilling Cycle with Chip Breaking

G87	Drilling cycle with chip breaking.

This cycle is used to programme machining in the X or Z axes.



Syntax

N.. **G87** X.. / Z.. [ER..] [EH..] [P..] / [ES..] [Q..] [EP..] [EF..]

G87	Drilling cycle with chip breaking.
X.. / Z..	End point on the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis
P..	Value of first infeed.
ES..	Number of infeeds at constant value (see Fig. 1)
Q..	Value of last infeed.
EP..	Backoff between two infeeds (default no backoff, EP = 0).
EF..	Dwell at the end of each infeed.

Property of the Function

Function G87 is modal.

Cancellation

Function G87 is cancelled by one of functions G80-G85 and G89 or functions G64, G65, G66.

4

Notes

If addresses P and Q are programmed, the consecutive infeeds between P and Q are degressive values.

At least one of arguments P and ES must be programmed or the system returns error message 889.

If the value of P is greater than delta Z, the system returns error message 881.

<u>Notes related to ES (number of constant infeeds)</u>

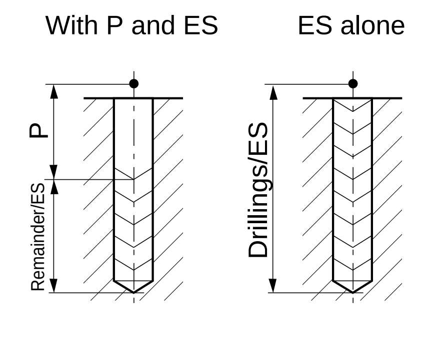
If P and ES are programmed:

The first infeed is equal to P and drilling is continued as a number ES of infeeds.

If ES is programmed alone (without P):

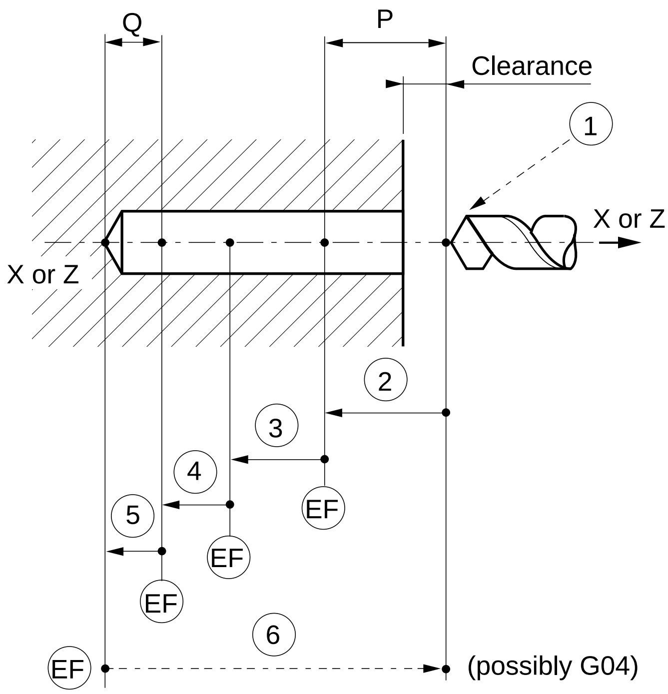
All the drilling is executed as a number ES of infeeds.

Figure 1



Cycle Steps

The steps below are given as an illustration. The number of steps depends on the values programmed with the cycle.



Step 1: Fast positioning in the machining axis.

Step 2: First infeed of depth P.. at programmed feed rate.

Possible dwell at the end of the infeed.

Steps 3 and 4: Consecutive infeeds and dwells as step 2.

Step 5: Infeed depth Q.. at the feed rate.

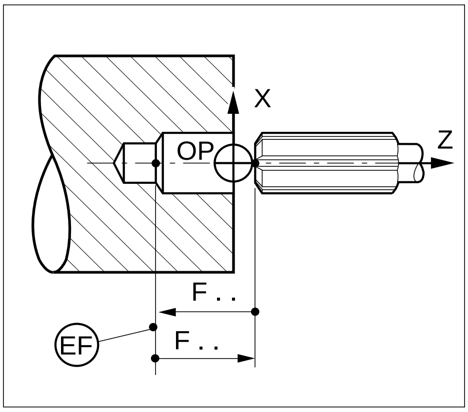
Possible dwell at the end of the peck.

Step 6: Rapid retraction in the tool axis.

Possible dwell G04 F.. at the start point.

4.9.4.9 Boring Cycle with Dwell at the Bottom of the Hole

G89	Boring cycle with dwell at the bottom of the hole.



Syntax

N.. **G89** X.. / Z.. [ER..] [EH..] EF..

G89	Boring cycle with dwell at the bottom of the hole.
X.. / Z..	End point on the machining axis.
ER..	Dimension of the retraction plane on the machining axis.
EH..	Impact plane in the machining axis.
EF..	Mandatory dwell expressed in seconds (maximum 99.99 s, format EF022, the default value of EF = 1 second).

Property of the Function

Function G89 is modal.

Cancellation

Function G89 is cancelled by one of functions G80-G85, G87 or functions G64, G65, G66.

Cycle Steps

Step 1: Fast positioning in the machining axis.

Step 2: Infeed at the feed rate F..

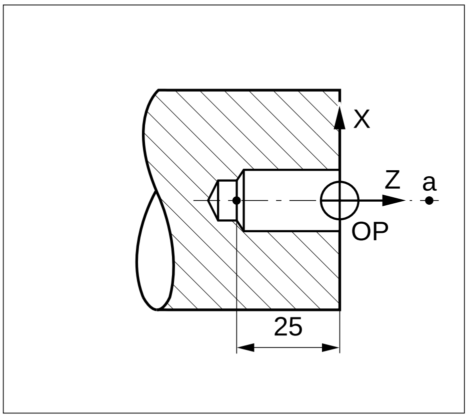
Step 3: Dwell at the end of boring

Step 4: Outfeed at the feed rate F.. along the tool axis.

Example

Execution of a bore.

```
N.. ...
N50 G00 Xa Za
N60 G94 (or G95) F..
N70 G89 Z-25 EF1
N80 G80 G00 X.. Z..
N..
```

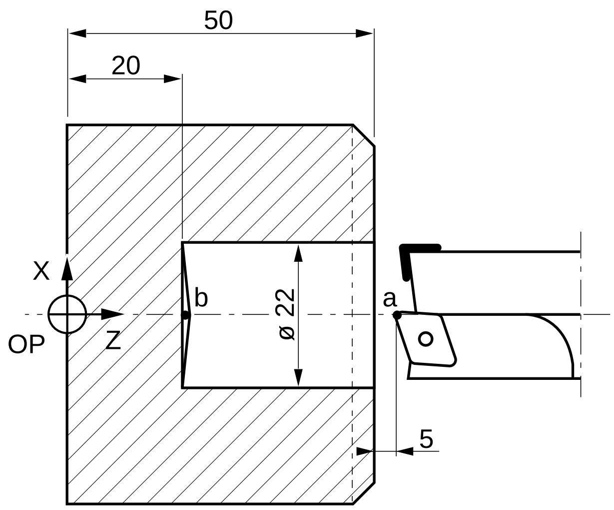


or

```
N..
N50 G94 (or G95) F..
N60 G89 Xa ERa Z-25 EF1
N70 G80 G00 X.. Z..
N..
```

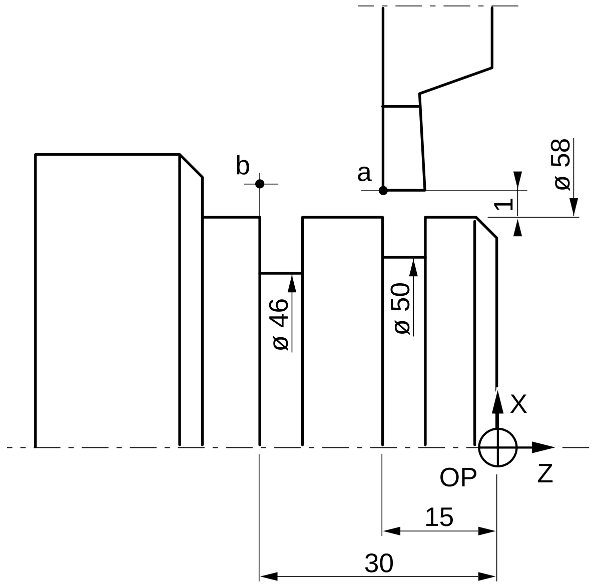
4.9.4.10 Examples of Programming Cycles

Drilling in the Z axis (cycle G83 or G87)




```
%62
N10...
N.. G92 S3500
N.. ...
N120 T06 D06 M06 (CARBIDE DRILL D=22)
N130 G97 S900 M40 M03
N140 G00 X0 Z55                          Point a
N150 G96 X22 S100
N160 G95 F0.15
N170 G83 (or G87) Z20 P10 Q5             Point b, cycle
N180 G97 S900
N190 G80 G00 G52 X.. Z..                 Cycle cancellation
N..
```

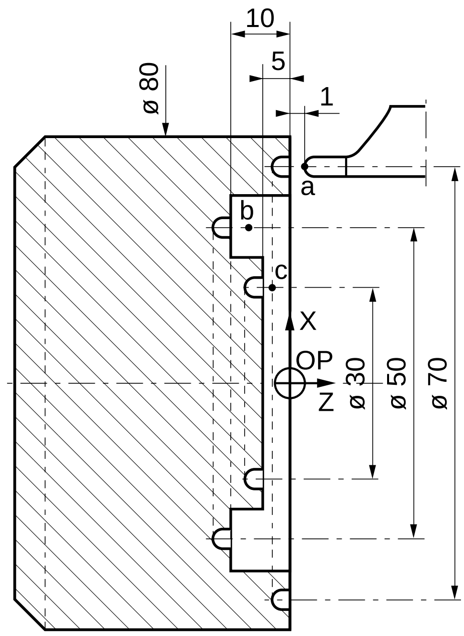
Execution of two grooves in the X axis (cycle G83 or G87)




```
%56
N10...
N.. G92 S3000
N..
N.. ...
N220 T08 D08 M06 (GROOVING TOOL L=4)
N230 G97 S900 M40 M04
N240 G16 P+                           Tool axis orientation on X+
N250 G00 X60 Z-15                     Point a
N260 G96 S50
N270 G95 F0.1
N280 G83 (or G87) X50 P2 Q1 G04 F1    Cycle
N290 X46 Z-30                         Point b, cycle repeat
N300 G97 S900
N310 G80 G16 R+ G00 G52 X.. Z..       Cycle cancellation
N..
```

<u>Execution of three front grooves along the Z axis (cycle G82) with positioning on different levels using ER</u>

Grooves with radius 3, depth 3.




```
%72
N10...
N.. G92 S200
N..
N..
N190 T06 D06 M06 (GROOVE CUTTER R3)
N200 G97 S900 M04 M40
N210 G00 Z10
N220 G96 X80 S40
N230 G95 F0.08
N240 G82 X70 Z-3 ER1 EF2              Point a, cycle
N250 X50 Z-13 ER-9                    Point b, cycle
N260 ER-4
N270 X30 Z-8 EF1                      Point c, cycle
N280 G97 S900
N290 G80 G00 G52 X.. Z..              Cycle cancellation
N..
```

4.9.4.11 Table Summarising Cycles G81 to G89

CYCLES / SEQUENCE OF MOVEMENTS	G81	G82	G83	G84	G85	G87	G89
	Centre drilling	Counterboring	Peck drilling	Tapping	Reaming	Drilling with chip breaking	Boring with dwell in bottom of hole
Downward	Work	Work	Rapid then work with n consecutive penetrations (P, Q)	Work	Work	Work with n consecutive penetrations (P, Q)	Work
Number of constant infeeds			Programmed by ES			Programmed by ES	
Retraction after infeed			Rapid				
Clearance after pecking			Programmed by EP				
Backoff between 2 infeeds						Programmed by EP	
Dwell on each infeed			Programmed by EF			Programmed by EF	
HOLE BOTTOM – Dwell		Programmed by EF	Programmed by EF on a combination machine used for turning	Programmed by EF			Programmed by EF
HOLE BOTTOM – Spindle				Reversal of rotation			
HOLE BOTTOM – Retraction in the plane							
Outfeed	Rapid	Rapid	Rapid	Work	Work rate of other feed rate if EF present	Rapid at the end of penetration	Work

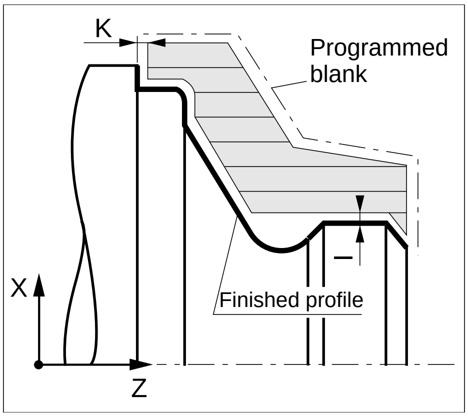
4.10 Other Machining Cycles

4.10.1 Rough Turning/Facing Cycle

G64	Turning/Facing cycle.

This function is used to rough a volume of material located between the specified dimensions of a given blank profile and a finished profile.

This turning or facing cycle can be used for external or internal working.



Syntax

N.. **G64** [N.. N..] / [EP..] [I.. K..] P.. / R..

N.. BLANK DEFINITION

G64	Turning/Facing cycle.
N.. N..	Numbers of the first and last block defining the finished profile (minimum two blocks, maximum 50 blocks).
EP..	Number of the contour created with the PROFIL function (see PROFIL User's Manual)
I..	Finishing allowance in X. - The default value of I = 0.
K..	Finishing allowance in Z. - The default value of K = 0.
P.. / R..	Cut depth (unsigned). P: depth of cut in X (turning) R: width of cut in Z (facing)
N.. BLANK DEFINITION	Sequence of blocks defining the blank dimensions (these blocks are located between G64 and the cycle cancellation code G80).

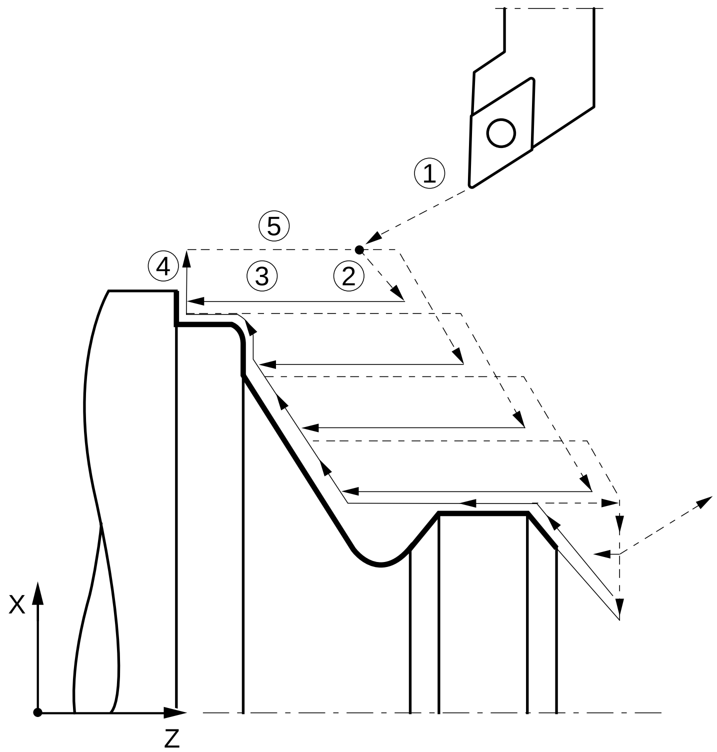
Property of the Function

Function G64 is modal.

Cancellation

Function G64 is cancelled by function G80.

Cycle Steps



Step 1: Tool positioning in XZ.

Step 2: Rapid to pass depth (to depth «P» for turning).

Step 3: Execution of the first pass at the feed rate.

Step 4: Lift along the profile at the feed rate.

Step 5: Rapid return to the cycle start point.

Execution of the following passes in the same way as steps 2 to 5, then retraction.

Notes

When the cycle is programmed, the system must be in state G40 (G41 or G42 tool radius offset cancelled).

The feed rate and its argument can be programmed in the cycle block, e.g.:
`N.. G64 N.. N.. I.. K.. P.. G95 F0.25`

Notes on definition of the finished profile

Both first and last blocks N.. to N.. defining the finished profile dimensions must include both coordinates X and Z as they may be analysed in reverse order.

Addresses I and K must be signed according the direction of the machining allowance (e.g. «I-..» for a bore).

The cycle will not execute any grooves (in the face or on diameter) included in the definition of the finished profile.

At the end of the cycle, the system is left in G0 mode.

The finished profile definition can be placed before the cycle call, but in this case it is necessary to use branch function G79 (see Sec. 4.11.3).

The following programme variables cannot be used in the finished profile definition:

The following programme variables cannot be used in the finished profile definition. Otherwise, the system returns error message 96:
- L100-L199
- L900-L959 (see Sec. 6.1).

(The only case where these variables can be used is when M999 is programmed, see Sec. 4.15.9.)

Notes on the blank definition

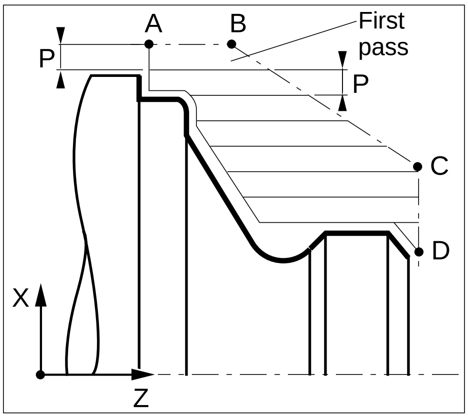
The blank definition cannot contain:
- PGP (Profile geometry Programming) blocks,
- a block defining a curve.

The blank definition blocks can include miscellaneous M functions.

The roughing cycle uses P or R to determine turning or facing, plus the directional sense of increasing/decreasing dia. from the blank definition, to determine internal/ external working.

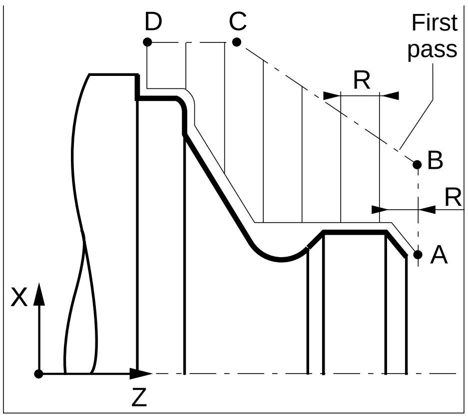
Rough turning

```
N200 ...
N210 G64 N.. N.. I.. K.. P4
N220 X.. Z..                Point A
N230 Z..                    Point B
N240 X.. Z..                Point C
N250 X..                    Point D
N260 G80 G52 X.. Z..
N..
```



Rough facing

```
N200 ...
N210 G64 N.. N.. I.. K.. R4
N220 X.. Z..                Point A
N230 X..                    Point B
N240 X.. Z..                Point C
N250 Z..                    Point D
N260 G80 G52 X.. Z..
N..
```

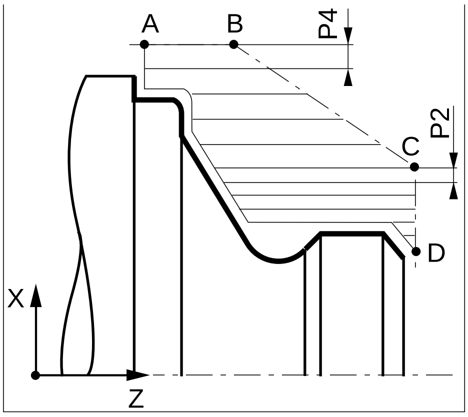


4

The pass depth defined in the cycle block (P or R) can be modified during the blank definition blocks.

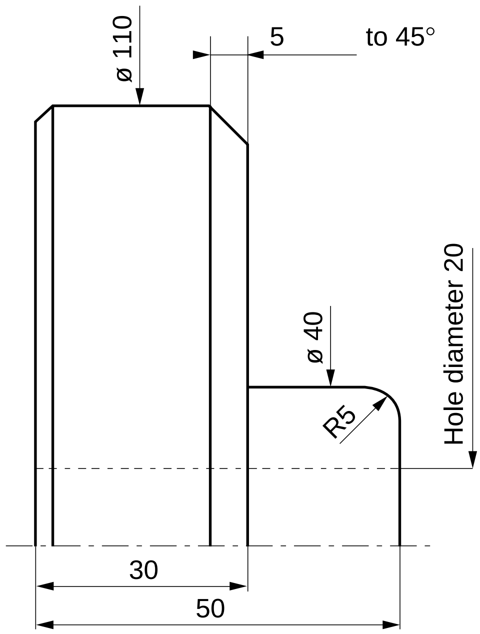
Example

```
N200 ...
N210 G64 N.. N.. I.. K.. P4
N220 X.. Z..              Point A
N230 Z..                  Point B
N240 X.. Z..              Point C
N250 X.. P2               Point D,
                          new P
N260 G80 G52 X.. Z..
N..
```

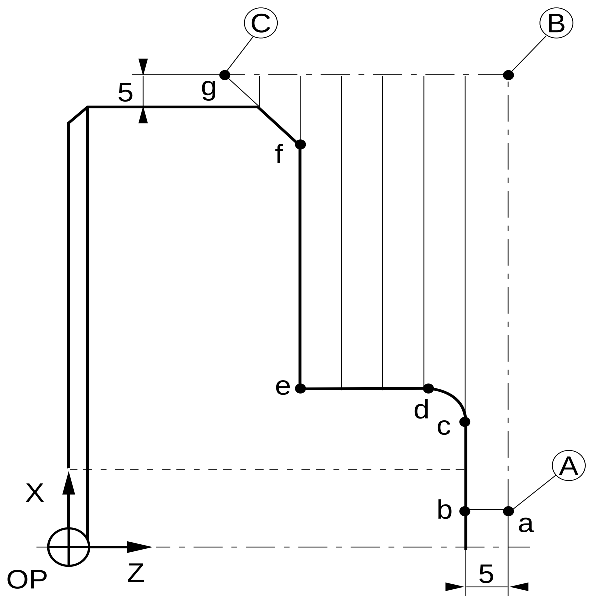


Examples

Roughing an external profile by facing



Machining paths

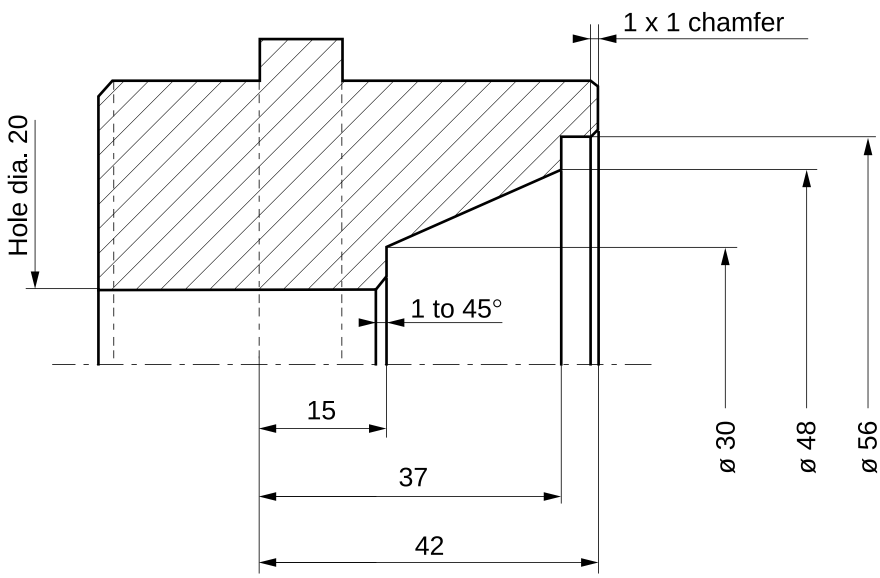


```
%46
N10 G00 G52 X.. Z..                          Tool change position
N20 T01 D01 M06 (FACING TOOL R=0.8)
N30 S900 M40 M04
N40 X120 Z55                                 Point B, tool at the start of roughing
N50 G92 S3500
N60 G96 S200
N70 G95 F0.2
N80 G64 N140 N210 I0.4 K0.2 R3               Cycle
N90 X16 Z55                                  Point (A) }
N100 X120                                    Point (B) } Blank definition
N110 Z20                                     Point (C) }
N120 G80 G52 X.. Z.. G97 S900                Cycle cancellation
N130 T03 D03 M06 (COPYING TOOL R=0.4)
N140 G42 X16 Z55                             Point a, tool at the start of finishing
N150 G96 S250
N160 G01 Z50 F0.1                            Point b
N170 X30                                     Point c
N180 G03 X40 Z45 R5                          Point d
N190 G01 Z30                                 Point e
N200 X100                                    Point f
N210 X120 Z20                                Point g
N220 G00 G40 G52 X.. Z.. G97 S900 M05
N230 M02
```

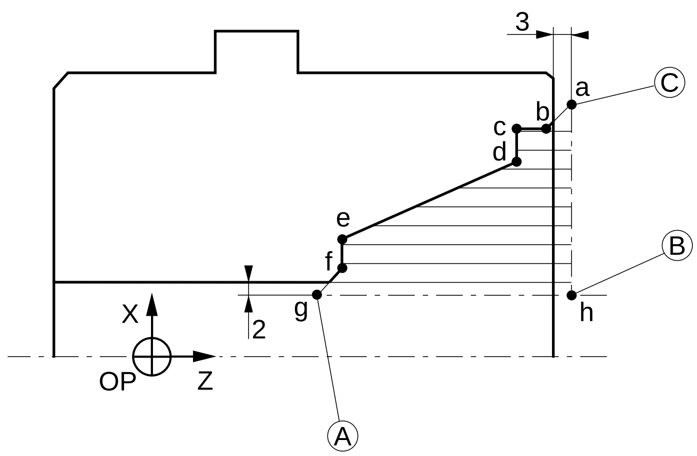
Changes required to programme %46 for turning.

```
N80 G64 N140 N210 I0.4 K0.2 P3               Address R replaced by P
N90 X120 Z20                                 }
N100 Z55                                     } Blank definition reversed
N110 X16                                     }
```

Roughing of an internal profile by turning

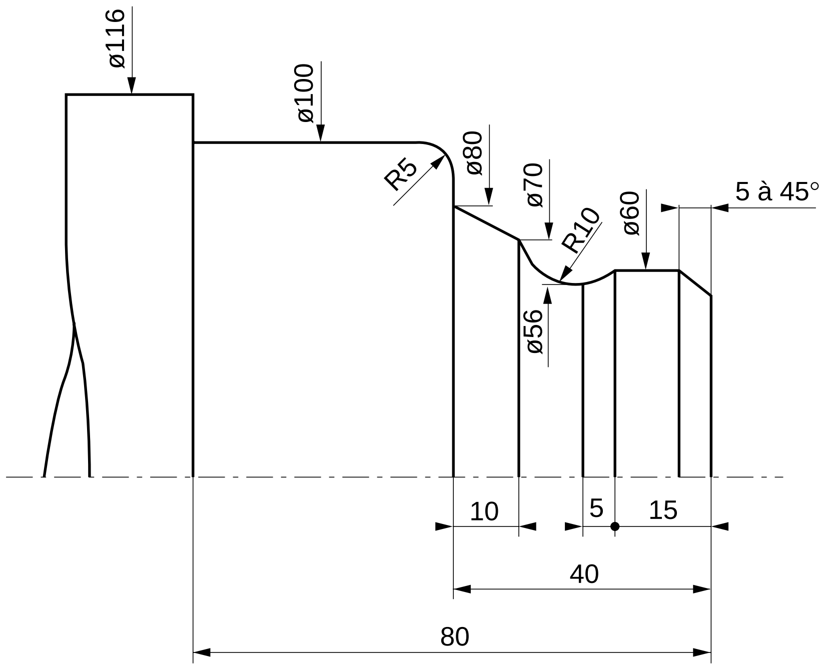


Machining paths

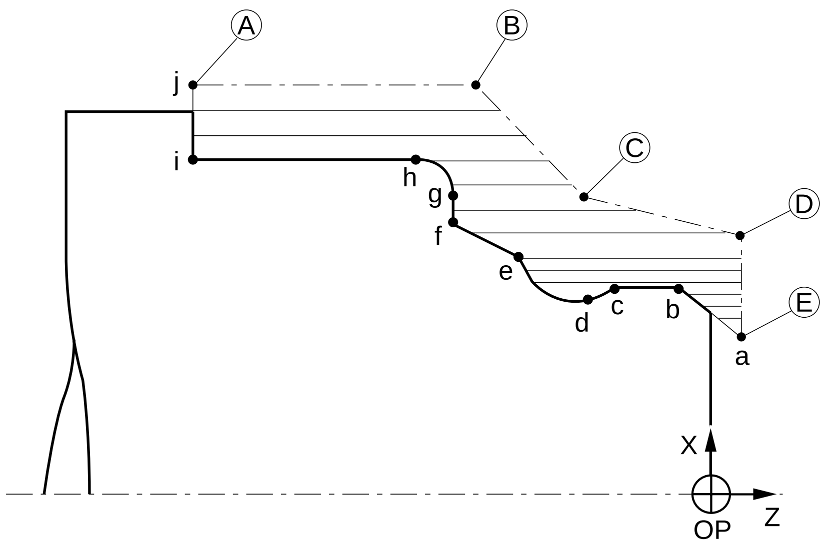


```
%48
N10 G00 G52 X.. Z..                        Tool change position
N20 T07 D07 M06 (BORING TOOL R=0.4)
N30 S900 M40 M04
N40 X18 Z45                                Point B, tool at the start of roughing
N50 G92 S3500
N60 G96 S100
N70 G95 F0.15
N80 G64 N140 N210 I-0.2 K0.1 P2            Cycle
N90 X16 Z12                                Point (A)
N100 Z45                                   Point (B)   Blank definition
N110 X64                                   Point (C)
N120 G80 G52 X.. Z.. G97 S900              Cycle cancellation
N130 T09 D09 M06 (BORING TOOL R=0.2)
N140 G41 X64 Z45                           Point a, tool at the start of finishing
N150 G96 S120
N160 G01 X56 Z41 F0.08                     Point b
N170 Z37                                   Point c
N180 X48                                   Point d
N190 X30 Z15                               Point e
N200 X22                                   Point f
N210 X16 Z12                               Point g
N220 G00 Z45 G97 S900                      Point h
N230 G40 G52 X.. Z.. M05
N240 M02
```

Roughing of an external profile by turning (preformed blank profile)

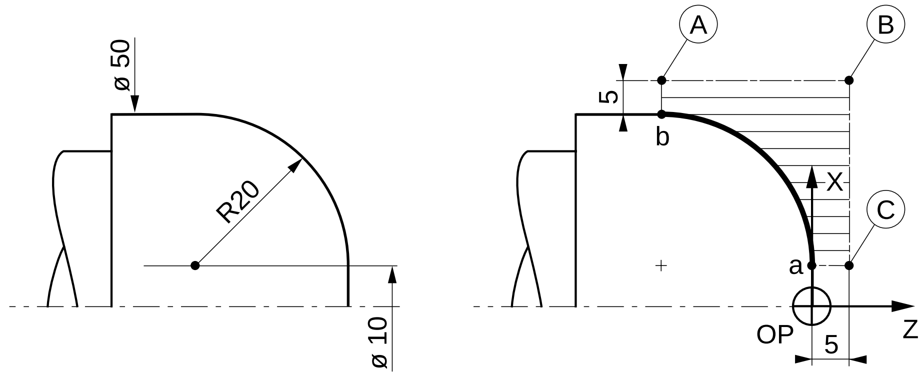


Machining paths



```
%62
N10 G00 G52 X.. Z..                            Tool change position
N20 T01 D01 M06 (BIDIRECTIONAL TOOL R=0.8)
N30 S900 M40 M04
N40 X120 Z-30                                  Point B, tool at the start of roughing
N50 G92 S3500
N60 G96 S180
N70 G95 F0.25
N80 G64 N170 N270 I0.3 K0.2 P4                 Cycle
N90 X120 Z-80                                  Point (A)  }
N100 Z-40                                      Point (B)  }
N110 X80 Z-20                                  Point (C)  } Blank definition
N120 X70 Z5                                    Point (D)  }
N130 X40 P2                                    Point (E)  }
N140 G97 S900
N150 G80 G52 X.. Z..                           Cycle cancellation
N160 T03 D03 M06 (COPYING TOOL R=0.4)
N170 G42 X40 Z5                                Point a, tool at the start of finishing
N180 G96 S250
N190 G01 X60 Z-5 F0.1                          Point b
N200 Z-15                                      Point c
N210 X56 Z-20                                  Point d
N220 G02 X70 Z-30 R10                          Point e
N230 G01 X80 Z-40                              Point f
N240 X90                                       Point g
N250 G03 X100 Z-45 R5                          Point H
N260 G01 Z-80                                  Point i
N270 X120                                      Point j
N280 G00 G40 G52 X.. Z.. G97 S900 M05
N230 M02
```

Roughing then finishing by turning of a profile defined by two points (a and b).

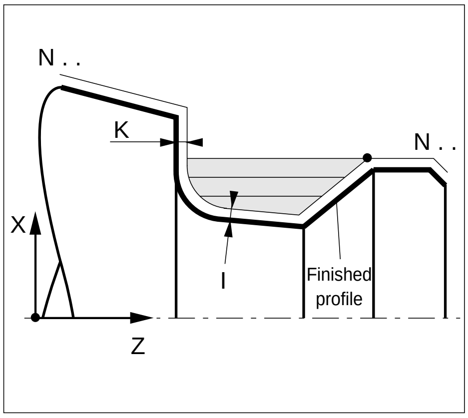


%64	
N10 G00 G52 X.. Z..	Tool change position
N20 T01 D01 M06 (TOOL R=0.8)	
N30 S900 M40 M04	
N40 X60 Z5	Point B, tool at the start of roughing
N50 G96 S300	
N60 G92 S3000	
N70 G95 F0.3	
N80 G64 N150 N160 P3 I0.2 K0.2	Cycle
N90 X60 Z-20	Point A } Blank definition
N100 Z5	Point B } Blank definition
N110 X10	Point C } Blank definition
N120 G80 G52 X.. Z.. G97 S900	Cycle cancellation
N130 T03 D03 M06 (OUTIL R=0.4)	
N140 X10 Z2	Approach point
N150 G42 G01 X10 Z0	Point a
N160 G03 X50 Z-20 I10 K-20	Point b
N170 G01 G40 X54 G97 S900	Retraction point
N180 G00 G52 X.. Z.. M05	
N190 M02	

4.10.2 Groove Roughing Cycle

G65	Groove roughing cycle.

This function is used to rough a groove whose profile is programmed during definition of a finished profile.
The cycle executes axial grooves by turning and front grooves by facing.



Syntax

N.. **G65** [N.. N..] / [EP..] X.. / Z.. [I.. K..] [EA..] P.. / R.. [Q..] [EF..]

G65	Groove roughing cycle.
N.. N..	First and last finished profile block numbers. The groove profile must be located between these blocks (maximum 50 blocks).
EP..	Number of the contour created with the PROFIL function (see PROFIL User's Manual).
X.. / Z..	End of cut position in the appropriate axis. - X for front rough facing, - Z for axial rough turning.
I..	Finishing allowance in X. - The default value of I = 0.
K..	Finishing allowance in Z. - The default value of K = 0.
EA..	Cut introduction angle in the groove (see Fig. 1).
P.. / R..	Cut depth (unsigned). P: depth of cut in X (turning). R: width of cut in Z (facing).
Q..	Positioning clearance (see Fig. 2). - The default value of Q = 0 (approach distance without clearance at the machining feed rate).
EF..	Material penetration feed rate. - The default value of EF is the same as the modal feed rate F.

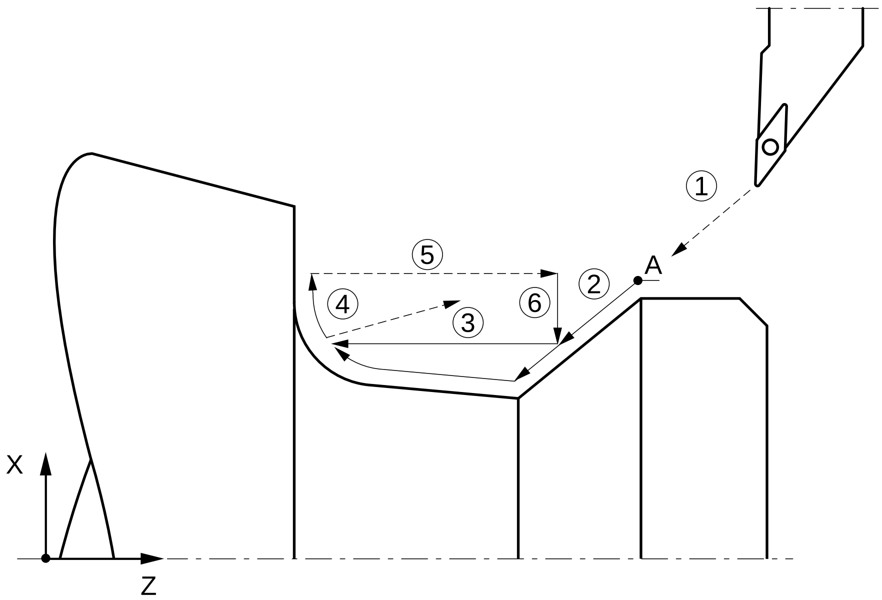
Property of the Function

Function G65 is nonmodal.

Cancellation

Function G65 is cancelled at the end of the block.

Cycle Steps



Step 1: Tool positioning (point A, cycle start).

Before the cycle start, the system takes into account the declared tool radius.

Step 2: Infeed at the introduction angle at the machining feed rate (to depth P for turning).

Step 3: Execution of the pass at the machining feed rate.

Step 4: Retraction at the machining feed rate along the groove profile.

Step 5: Rapid return to the perpendicular over the start point for the next pass.

Step 6: Positioning at the machining feed rate (if Q with a value not programmed).

The following passes are executed in the same way as steps 2 to 6.

After execution of the last pass, the tool retracts at rapid to the cycle start point.

Notes

Both the finished profile definition blocks N.. to N.. must include both the coordinates X and Z.

Tool positioning to the cycle start must be programmed in G40 (G41 or G42 radius offset cancelled).

The feed rate function and argument can be programmed in the cycle block, e.g.:
`N.. G65 N.. N.. I.. K.. P.. G95 F0.1.`

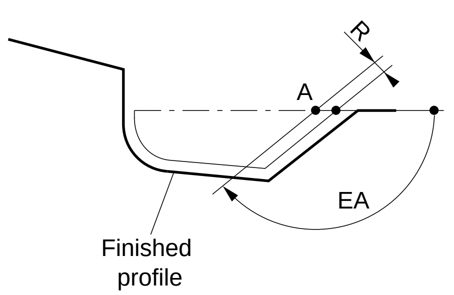
At the end of the cycle, the system is left in G0 mode.

Notes on argument EA

- EA: cut introduction angle defines a line from the cycle start point (point A) to the intersection with the finished profile.

After positioning to the start point, the system reads the declared tool radius (R) before executing the first pass.

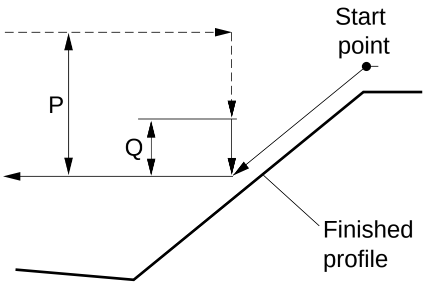
Figure 1



Notes on argument Q

After the first pass, when argument Q is programmed with its value, the approach distance before each new cut is carried out in two steps:
- At high speed up to value Q,
- At the machining feed rate until the start of the next pass.

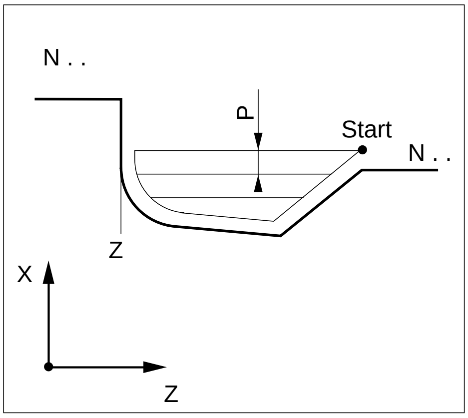
Figure 2



Cycle used for dia. or face machining

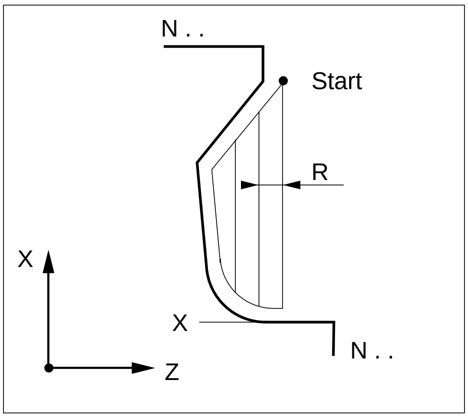
Dia. roughing

```
N70 ...
N80 G65 N.. N.. Z.. I.. K.. EA.. P..
N90 G52 X.. Z..
N..
```



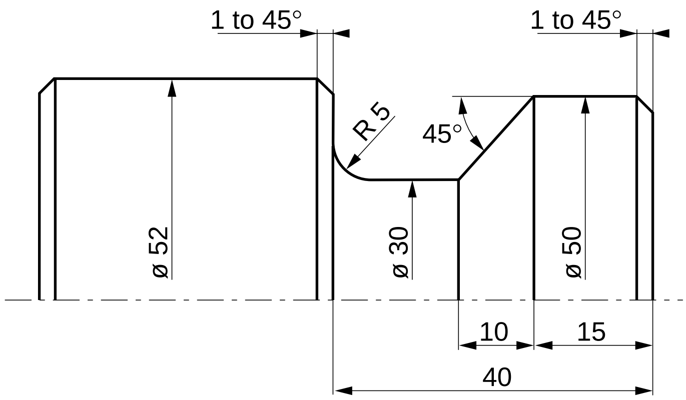
Face roughing

```
N70 ...
N80 G65 N.. N.. X.. I.. K.. EA.. R..
N90 G52 X.. Z..
N..
```

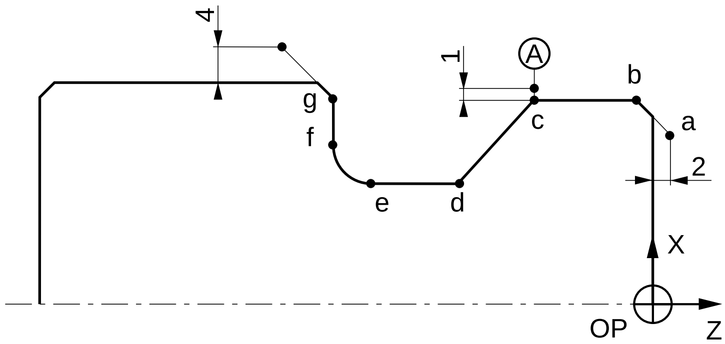


Example

<u>Roughing an axial groove</u>



Machining paths

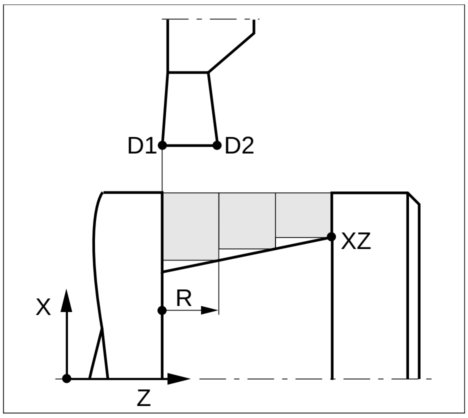


```
%34
N10 G00 G52 X.. Z..                               Tool change position
N20 T03 D03 M06 (GROOVE ROUGHING TOOL R=0.4)
N30 S900 M03 M40
N40 X52 Z-15                                      Point A, tool at the start of the groove
N45 G92 S3000
N50 G96 S100
N60 G95 F0.2
N70 G65 N100 N180 Z-40 I0.2 K0.1 EA-135 P2
N80 G52 X..Z.. G97 S900
N90 T05 D05 M06 (SLIDING TOOL R=0.8)
N100 G42 X44 Z2                                   Point a, tool at the start of the profile
N110 G96 S250
N120 G01 X50 Z-1 F0.1                             Point b
N130 Z-15                                         Point c
N140 X30 Z-25                                     Point d
N150 Z-35                                         Point e
N160 G02 X40 Z-40 R5                              Point f
N170 G01 X50                                      Point g
N180 X60 Z-45                                     Point h
N190 G00 G40 G52 X.. Z.. G97 S900 M05
N.. ...
```

4.10.3 Plunging Cycle

G66	Plunging cycle.

This function is used to rough a dia. or frontal groove by consecutive penetrations.



Syntax

N.. **G66** D.. X.. Z.. [EA..] P../R.. [EP..] [EF..]

G66	Plunging cycle.
D..	Number of the second grooving tool correction (the first correction must be programmed in one of the previous blocks, see Fig. 1).
X.. Z..	Groove machining end point (see Fig. 2).
EA..	Angle defining the slope across the groove bottom (see Fig. 2). - The default value of EA = 0 (no slope).
P../R..	Step between each penetration. P..: value along X (frontal groove). R..: value along Z (dia. groove).
EP..	Retraction (at 45 degrees) at the end of the pass (see example of special use of the cycle).
EF..	Dwell at the end of each penetration in seconds (Format F022). - The default value of EF = 0.

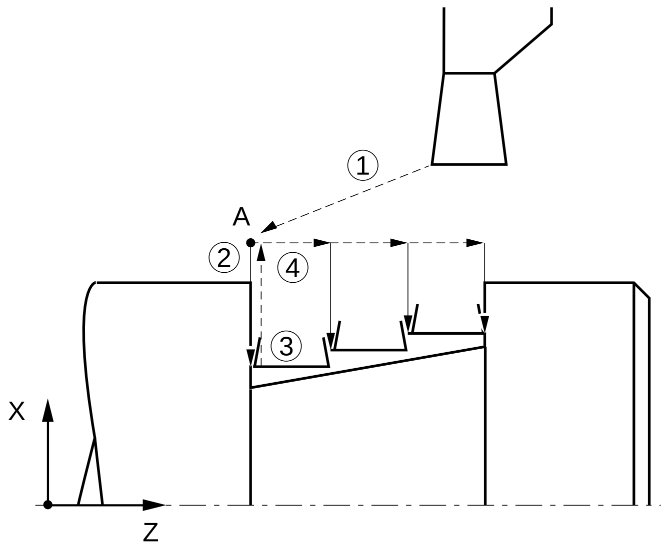
Property of the Function

Function G66 is nonmodal.

Cancellation

Function G66 is cancelled at the end of the block.

Cycle Steps



Step 1: Tool positioning above the material (point A, cycle start).

The first tool correction corresponding to the start side of the groove must be programmed in this block or in one of the previous blocks.

Step 2: Penetration at the machining feed rate (to the groove bottom point possibly modified by angle EA).

Step 3: Retraction at the machining feed rate.

Step 4: Rapid movement by the value of the step for execution of the next penetration (taking into account the correction corresponding to the other side of the groove).

Execution of the following passes in the same way as steps 2 to 4 until reaching the end of the groove.

After execution of the last pass, retraction clear of the material.

Notes

When the cycle is programmed, the system must be in state G40 (G41 or G42 tool radius offset cancelled).

The tool penetrations are uniformly distributed over the width of the groove. In certain cases, the system recomputes the programmed step.

The feed rate function and argument can be programmed in the cycle block, e.g.:
`N.. G66 D.. X.. Z.. R.. EA.. EF.. G95 F0.1.`

At the end of the cycle, the system is left in G0 mode.

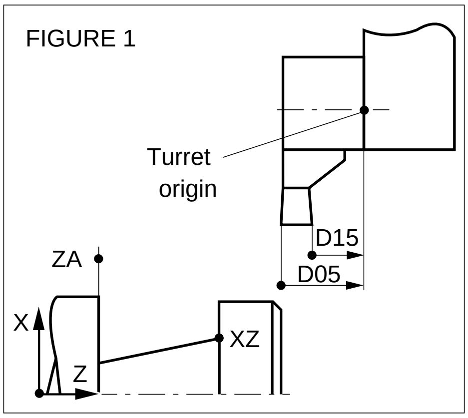
<u>Declaration of the two tool corrections</u>

Example:

Correction D05: Groove start side along Z (point A).

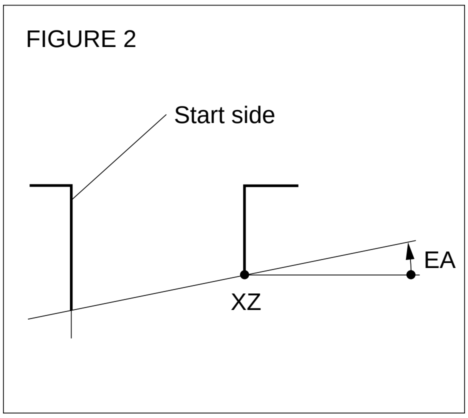
Correction D15: Groove end side along Z (point XZ).

The difference in value between the two corrections must be equal to the tool width.



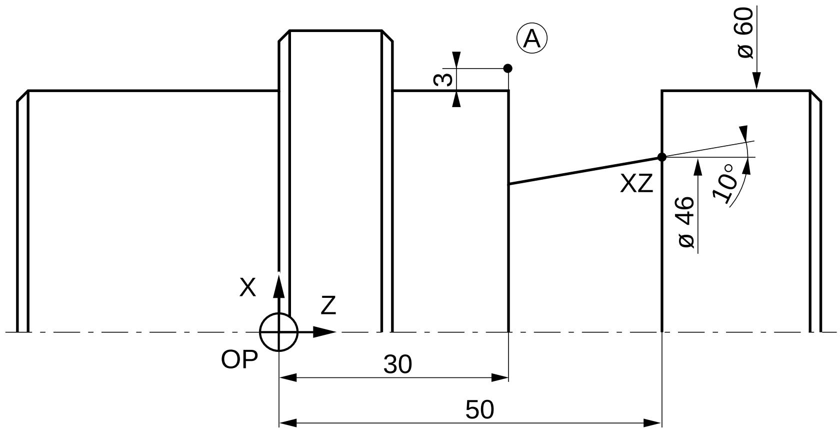
<u>Notes on arguments XZ and EA</u>

The groove bottom slope is defined from the groove end point (point XZ) to the intersection with the groove start side.



Examples

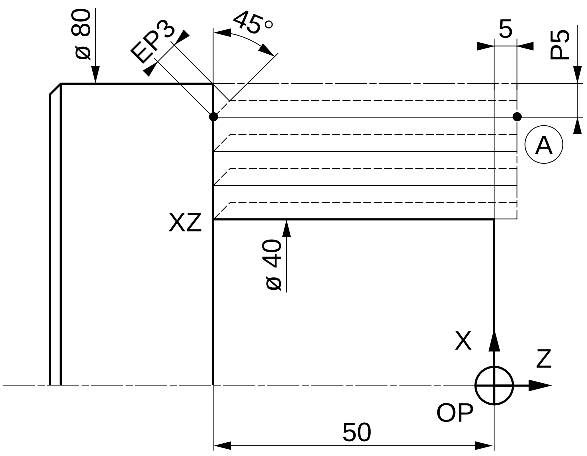
Plunging an axial groove




```
N.. ...
N210 G00 G52 X.. Z..                    Tool change position
N220 T05 D05 M06 (GROOVING TOOL L=6)
N230 G97 S900 M40 M03
N240 X66 Z30                            Point A
N250 G96 S80
N260 G95 F0.1
N270 G66 D15 X46 Z50 EA10 R5 EF0.5      Cycle
N280 G52 X.. Z.. G97 S900 M05
N..
```

Special use of cycle G40 with EP

EP can be added to the cycle syntax to rough a diameter.



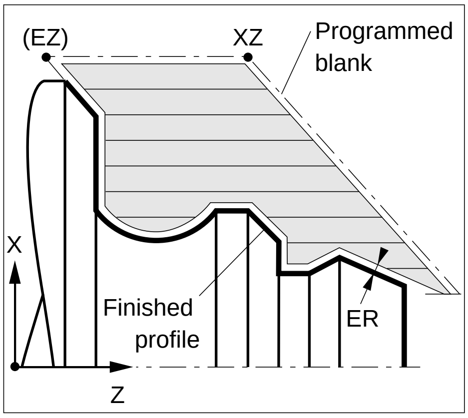
N.. ...	
N100 G00 G52 X.. Z..	Tool change position
$0 TURNING, DIAMETER 40	
N110 T01 D01 M06 (BIDIRECTIONAL TOOL R=08)	
N120 G97 S900 M40 M03	
N130 X70 Z5	Point A
N140 G96 S200	
N150 G95 F0.3	
N160 G66 X40 Z-50 P5 EP3	Cycle
N170 ...	

4.10.4 Roughing Cycle with Groove

G63	Roughing cycle with groove.

This function is used for roughing a volume of material located between the definition of a blank and a finished profile.

The cycle will machine a groove whose profile is compatible with the geometry and radius of the tool used. It can be carried out by facing or turning and for external or internal machining.



4

Syntax

N.. **G63** [N.. N..] / [EP..] X.. Z.. EX.. / EZ.. P.. / R.. EA.. / EU.. / EW.. [EB..] [EC..] [ER..] [Q..] [EQ..] [EF..]

G63	Roughing cycle with groove.
N.. N..	First and last block numbers defining the finished profile (maximum 95 blocks).
EP..	Number of the contour created with the PROFIL function (see PROFIL User's Manual).
X.. Z.. (or U.. W..)	Cycle start point.
EZ.. / EX..	Pass end point on the roughing axis: - EZ for rough turning along Z. - EX for rough facing along X. Direction of roughing pass execution: - Z+ if EZ > Z Z- if EZ < Z - X+ if EX > X X- if EX > Z
P.. / R..	Pass depth: - P: value along X (rough turning) - R: value along Z (rough facing).

EA../EU../EW..	Roughing pass start positions (see Figs. 1 and 2). These arguments are programmed to define a start angle for cut application and can be combined: EA EU, EA EW or EU EW.
EB..	Maximum groove penetration angle. Angle used to take into account the groove profile geometry and the tool rake angle (α) (see Fig. 3). - By default, EB is parallel to the roughing axis.
EC..	Maximum cut exit angle. Angle used to take into account the groove profile geometry and the tool angle of attack (K) (see Fig. 4). - By default, EC is perpendicular to the roughing axis.
ER..	Finishing allowance. Value applied normal to the profile. - The default value of ER = 0.
Q..	Positioning clearance. Approach distance at feed rate before contact with the material (for rapid to next cut depth). - The default value of Q = 0 (no clearance).
EQ..	Minimum chip value. - Below the programmed value, the pass is not executed.
EF..	Penetration feed rate in the material. - The default value of EF is the same as the modal feed rate F.

Property of the Function

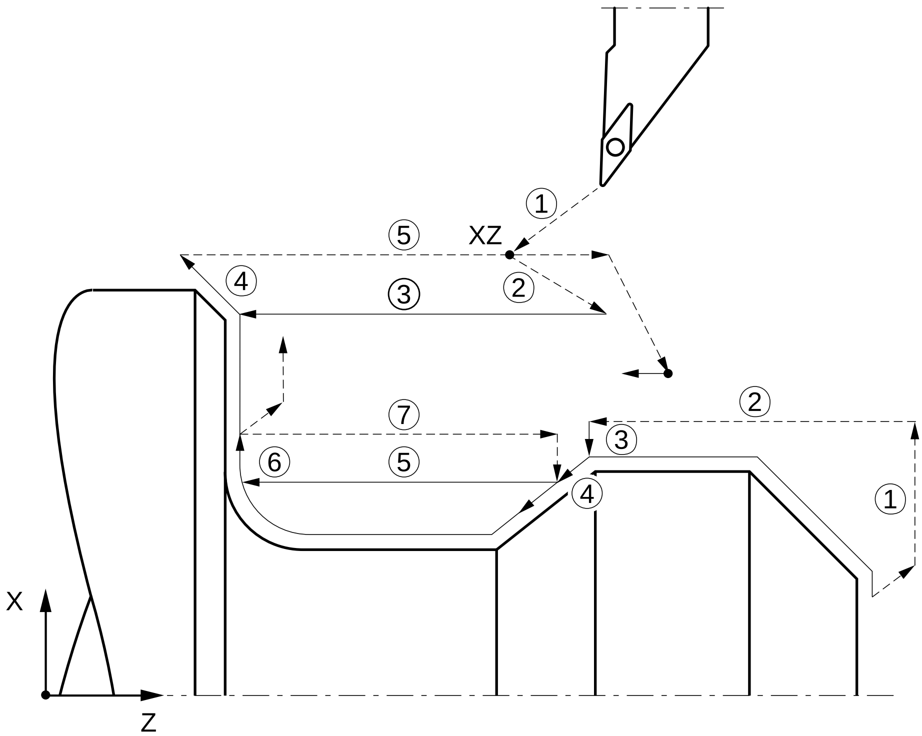
Function G63 is nonmodal.

Cancellation

Function G63 is cancelled at the end of the block.

Cycle Steps

Description of the first rough turning and grooving passes



Rough turning pass

Step 1: Tool positioning to X Z.

Before the cycle begins, the system reads the declared tool radius .

Step 2: Rapid to first cut depth P (for turning).

Step 3: Execution of the first pass at the machining feed rate.

Step 4: Lift along the profile at the machining feed rate.

Step 5: Rapid return to the cycle start point.

The following passes are executed in the same way as steps 2 to 5.

After execution of the last roughing pass, the tool is retracted from the surface and positioned in X.

Grooving pass

Step 1: Rapid positioning in X.

Step 2: Rapid positioning in Z.

Step 3: Positioning in X at the machining feed rate.

Step 4: Cut application at the machining feed rate along the groove profile.

Step 5: Execution of the pass at the machining feed rate.

Step 6: Lift at the machining feed rate along the groove profile.

Step 7: Rapid return to the cycle start point.

The following passes are executed in the same way as steps 3 to 6.

After execution of the last pass, the tool is retracted from the surface and positioned in X.

Notes

The number of blocks from N.. to N.. must not exceed 95. Otherwise, the system returns error message 92.

When the cycle is programmed, the system must be in state G40 (G41 or G42 tool radius offset cancelled).

At the end of the cycle, the system is left in G0 mode.

The finished profile definition can be programmed before the cycle call, but in this case it is necessary to use the branch function G79 (see Sec. 4.11.3).

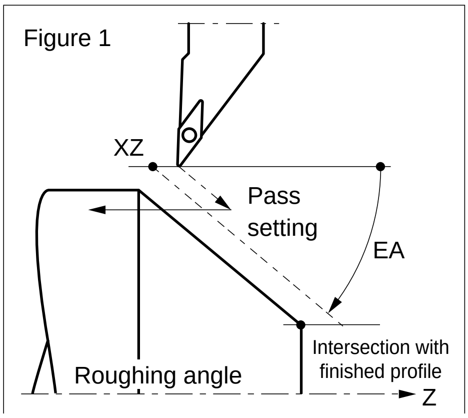
The cycle arguments defining positive angles are specified counterclockwise.

The feed rate function and argument can be programmed in the cycle block, e.g.:
```
N.. G63 N.. N.. X.. Z.. EZ.. P.. EA.. EB.. ER.. Q.. EQ.. EF.. G95 F0.3
```

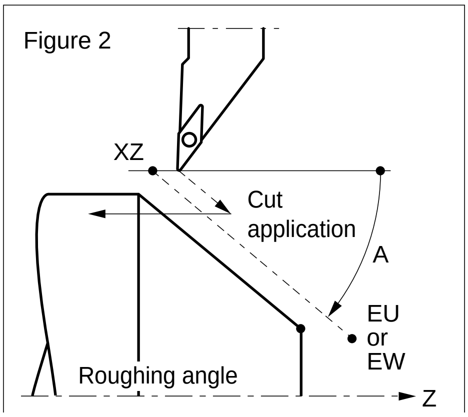
Notes on arguments EA, EU and EW

- EA: Angle limiting the start of cut application between point XZ (cycle start) and intersection with the finished profile. (If there is no intersection, the last pass is limited to the first point of the finished profile).

After positioning to point XZ, the system reads the declared tool radius before carrying out the first cut application.



- EU or EW: End point of the last roughing pass. If argument EA is not programmed, the values declared with EU and/or EW and XZ (cycle start) are used to define the cut application start angle (A).



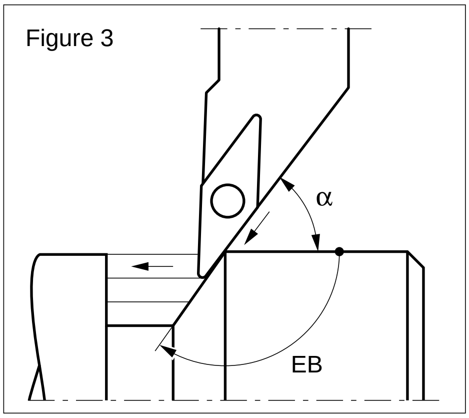
The angle is also used to differentiate between applying the cut at feedrate or rapid for a groove or not, respectively:

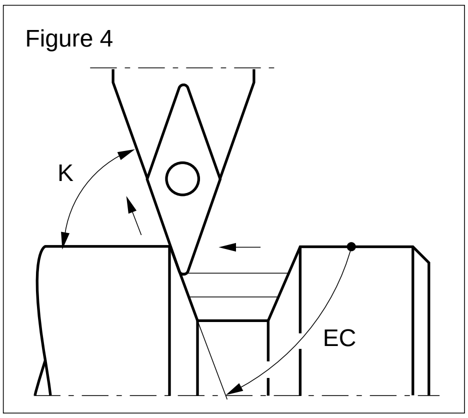
- Machining feed rate (in the material) if the angle between the cut application line and the roughing axis is $\geq$ 90 degrees.
- Traversing rate (off the material) if the angle between the cut application line and the roughing axis is < 90 degrees.

Definition of EB and EC

Angle defined by EB

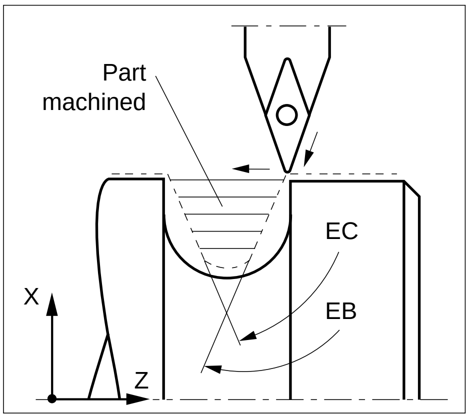
Angle defined by EC

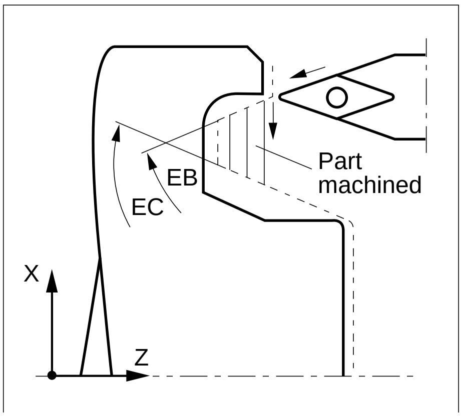




Machining of a groove limited by declaration of angles EB and EC.

When the groove profile and tool geometry are completely incompatible, the groove is not executed.

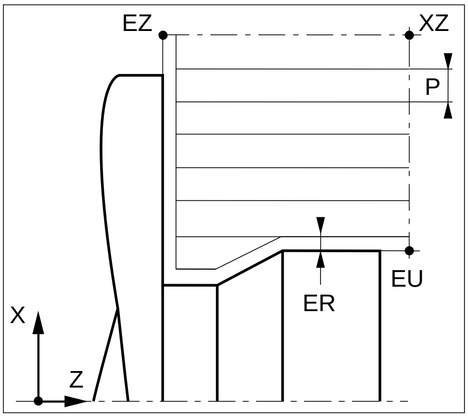




Cycle differences between turning and facing

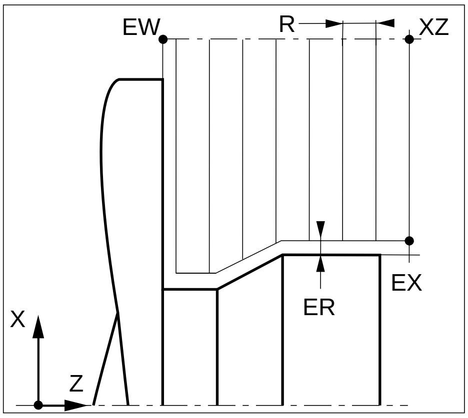
Rough turning

```
N70 ...
N80 G63 N.. N.. X.. Z.. EZ.. P.. EU.. EB.. ER..
N90 G52 X.. Z..
N..
```



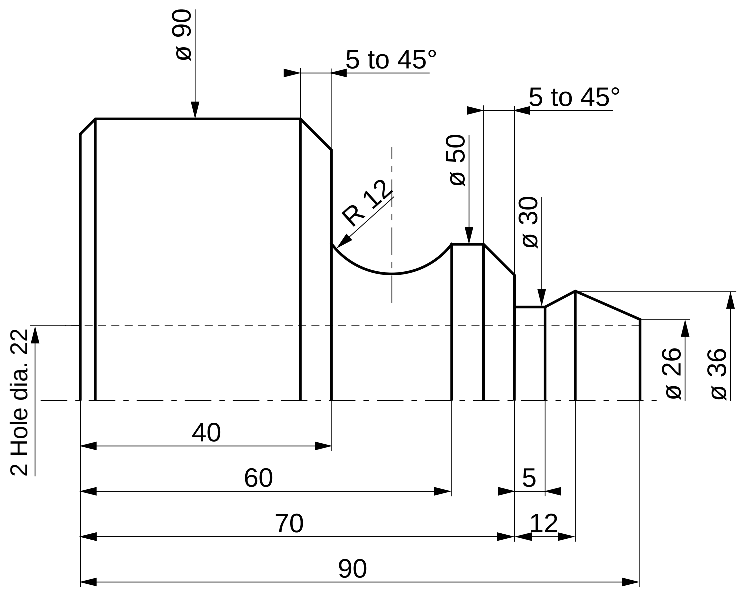
Rough facing

```
N70 ...
N80 G63 N.. N.. X.. Z.. EX.. R.. EW.. EC.. ER..
N90 G52 X.. Z..
N..
```

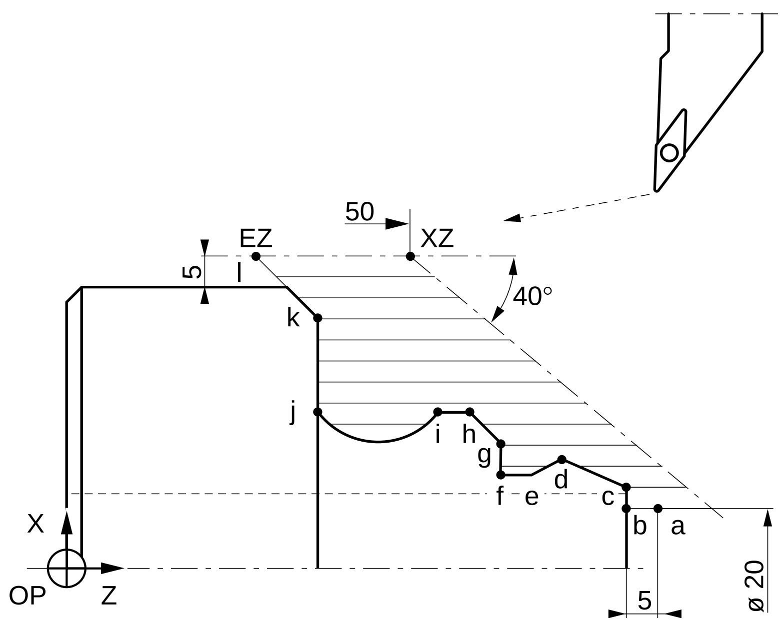


Examples

Rough and finish turning of an external profile



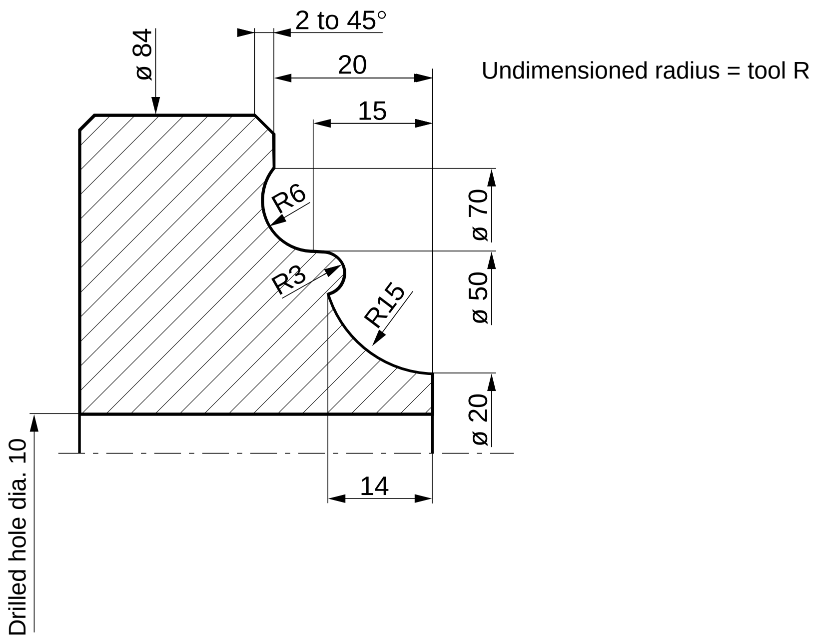
Machining paths



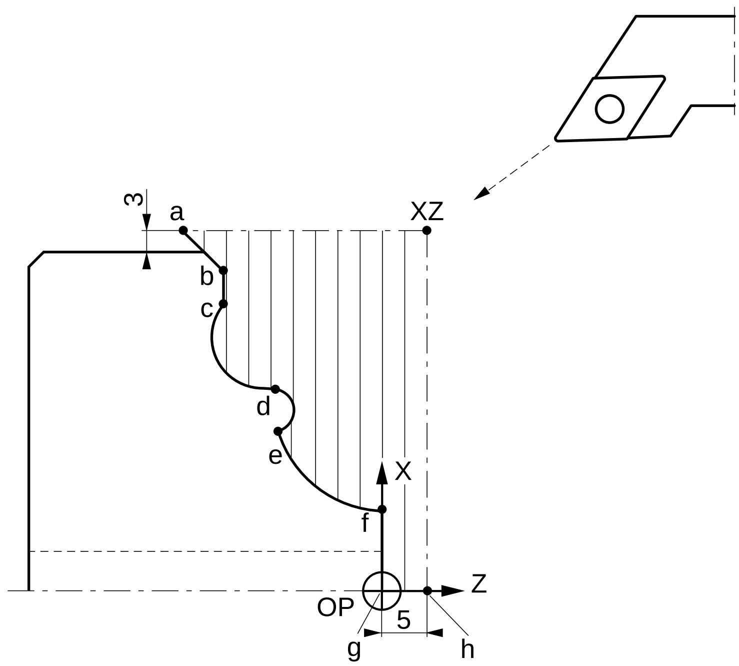
```
%32
N10 G00 G52 X.. Z..                                        Tool change position
N20 T01 D01 M06 (SLIDING TOOL R=0.8)
N30 S900 M40 M04
N40 X100 Z50                                               Tool position at the start of roughing
N50 G92 S3500
N60 G96 S200
N70 G95 F0.25
N80 G63 N110 N230 X100 Z50 EZ30 P2 EA-40 EB-145 ER0.4 Q2 EQ1 EF0.1
N90 G52 X.. Z..
N100 T03 D03 M06 (COPYING TOOL R=0.4)
N110 G42 X20 Z95                                           Point a, tool at the start of finishing
N120 G96 S250
N130 G01 Z90 F0.4                                          Point b
N140 X26 F0.1                                              Point c
N150 X36 Z82                                               Point d
N160 X30 Z75                                               Point e
N170 Z70                                                   Point f
N180 X40                                                   Point g
N190 X50 Z65                                               Point h
N200 Z60                                                   Point i
N210 G02 X50 Z40 R12                                       Point j
N220 G01 X80                                               Point k
N230 X100 Z30                                              Point l
N240 G40 G52 X.. Z.. G97 S900 M05
N250 M02
```

4

Rough and finish facing of an external profile



Machining paths

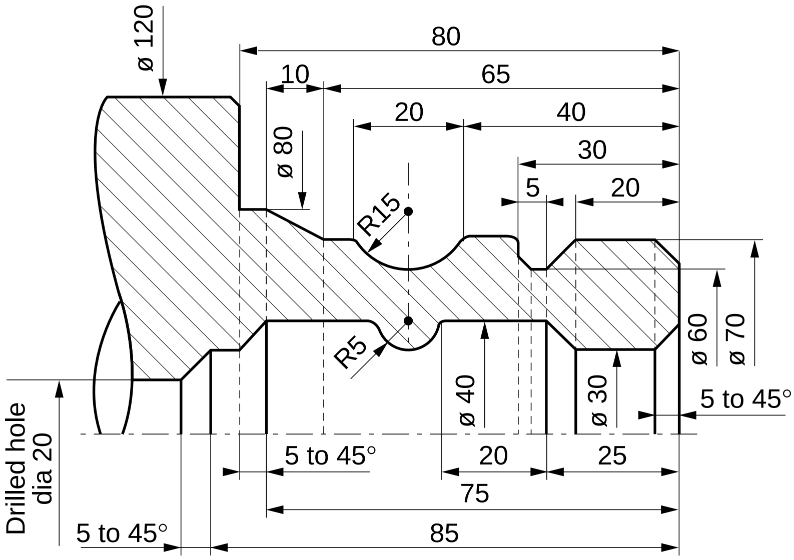


```
%39
N10 G00 G52 X.. Z..                               Tool change position
N20 T07 D07 M06 (FACING TOOL =0.8)
N30 S900 M40 M04
N40 X100 Z10                                      Tool position at the start of roughing
N50 G92 S3500
N60 G96 S200
N70 G95 F0.2
N80 G63 N110 N190 X90 Z5 EX0 R2 EA-180 EB-150 EC-170 ER0.2 Q1 EF0.1
N90 G52 X.. Z..
N100 T09 D09 M06 (COPYING TOOL R=0.4)
N110 G41 X90 Z-25                                 Point a, tool at the start of finishing
N120 G96 S250
N130 G01 X80 Z-20 F0.1                            Point b
N140 X70                                          Point c
N150 G03 X50 Z-15 R6                              Point d
N160 G02 X40 Z-14 R3                              Point e
N170 G03 X20 Z0 R15                               Point f
N180 G01 X0                                       Point g
N190 G00 Z5                                       Point h
N200 G40 G52 X.. Z.. G97 S900 M05
N210 ...
N.. M02
```

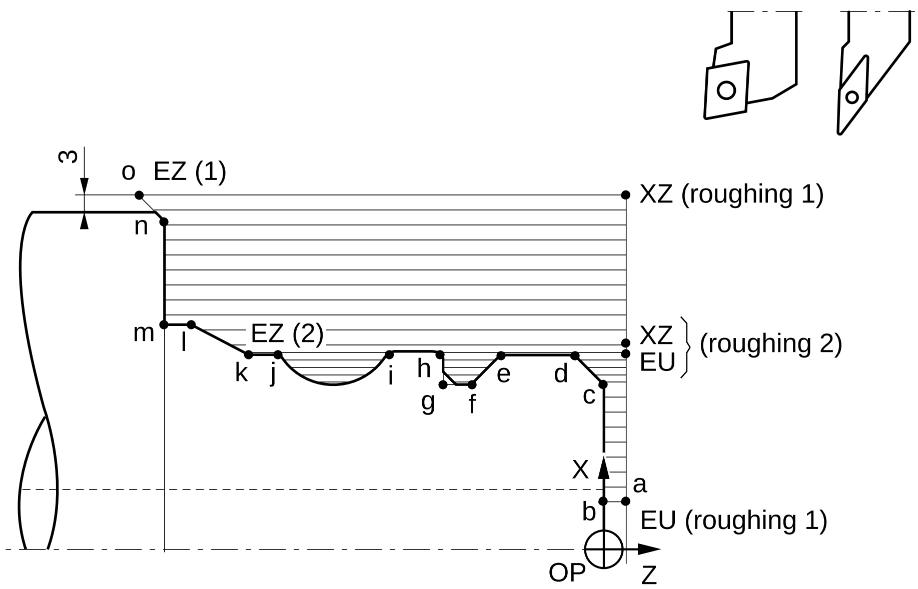
4

Execution of a profile by two outside rough turnings then finishing

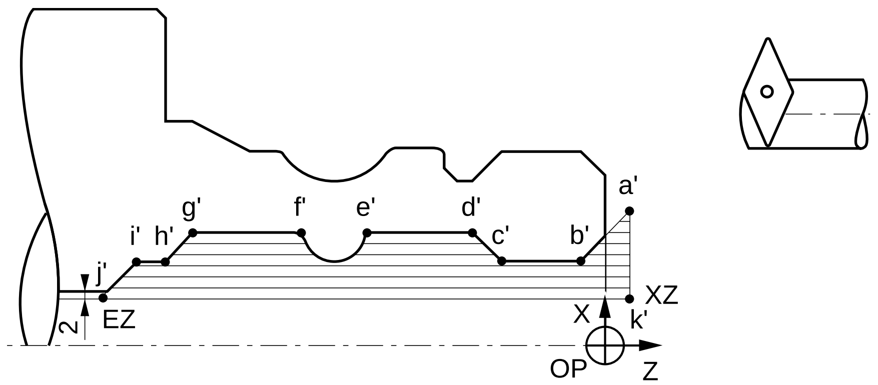
Rough and finish turning an internal profile



External machining paths



Internal machining paths



`%163`	
`N10 G0 G52 X.. Z..`	Tool change position
`$0 EXTERNAL ROUGHING 1`	
`N20 T01 D01 M06 (TOOL R0.8)`	
`N30 S900 M40 M04`	
`N40 X126 Z5`	Tool position at the start of roughing 1
`N50 G92 S4000`	
`N60 G96 S260`	
`N70 G95 F0.3`	
`N80 G63 N150 N300 X126 Z5 P3 EZ-85 EU16 EB-170 EC-93 ER0.2 EQ1 Q1 EF0.2`	
`N90 G0 G52 X.. Z.. G97 S900`	
`$0 EXTERNAL ROUGHING 2`	
`N100 T03 D03 M06 (TOOL R0.8)`	
`N110 X80 Z5`	Tool position at the start of roughing 2
`N120 G63 N150 N300 X72 Z5 P1.5 EZ-65 EU70 EB-125 EC-93 ER0.2 Q1 EF0.15`	
`N130 G0 G52 X.. Z.. G97 S900`	
`$0 EXTERNAL FINISHING`	
`N140 T11 D11 M06 (COPYING TOOL R0.8)`	
`N150 G42 G0 X16 Z5`	Point a
`N160 G96 S300`	
`N170 G01 Z0 F0.08`	Point b
`N180 X60`	Point c
`N190 X70 Z-5`	Point d
`N200 Z-20`	Point e
`N210 X60 Z-25`	Point f
`N220 Z-30 EB-2`	Point g

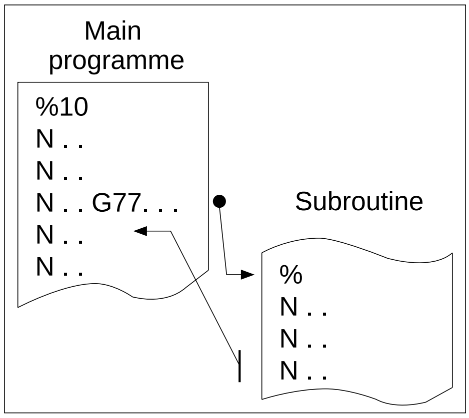
N230 X70 EB2	Point h
N240 Z-40 EB2	Point i
N250 G02 X70 Z-60 R15 EB2	Point j
N260 G01 Z-65	Point k
N270 X80 Z-75	Point l
N280 Z-80	Point m
N290 X116	Point n
N300 X126 Z-85	Point o
N310 G0 G40 G52 X.. Z.. G97 S900 M05	
$0 INTERNAL ROUGHING	
N320 T05 D05 M06 (BORING TOOL R0.8)	
N330 S900 M40 M03	
N340 X10 Z10	Tool position at the start of roughing
N350 G92 S3000	
N360 G96 S70	
N370 G95 F0.1	
N380 G63 N420 N520 X16 Z5 EU50 EZ-92 P2 EB110 EC70 ER0.2 Q1 EQ0.5 EF0.1	
N390 G0 Z5	Tool retraction
N400 G0 G52 X.. Z..	
$0 INTERNAL FINISHING	
N410 T07 D07 M06 (BORING TOOL R0.4)	
N420 G41 X50 Z5	Point a’
N430 G96 S90	
N440 G01 X30 Z-5 F0.07	Point b’
N450 Z-20	Point c’
N460 X40 Z-25	Point d’
N470 Z-45 EB2	Point e’
N480 G02 X40 Z-55 I40 K-50 EB2	Point f’
N490 G01 Z-75	Point g’
N500 X30 Z-80	Point h’
N510 Z-85	Point i’
N520 X16 Z-92	Point j’
N530 G0 Z5	Point k’, Tool retraction
N540 G40 G52 X.. Z.. G97 S900 M05	
N550 M02	

4.11 Breaks in Sequence

4.11.1 Unconditional Branch to a Subroutine or Sequence of Blocks with Return

G77	Unconditional branch to a subroutine or sequence of blocks with return.

Subroutines included in the main programme or separate from it are called by addresses H.. and/or N.. N.. associated with the function.



Syntax

N.. **G77** [H..] [N.. N../N..] [P..] [S..]

G77	Unconditional branch to a subroutine or sequence of blocks with return (maximum 8 nesting levels).
H..	Number of a subroutine external to the main programme.
N.. N..	Number of the first and last block called (If both N.. have the same number or if only one block is programmed, only one block is called).
P..	Number of the contour created by the PROFIL function (see PROFIL Function User's Manual).
S..	Number of times the subroutine or sequence of blocks is to be repeated (default = 1, maximum 99 repetitions).

Property of the Function

Function G77 is nonmodal.

Cancellation

Function G77 is cancelled at the end of the block.

Notes

Subroutines called by address H end with «X OFF» and do not include an M02 block (end of programme).

A subroutine called by addresses N.. N.. can be located between M02 and «X OFF».

If argument S is programmed in a block containing other instructions, it must immediately follow the subroutine branch.

If the subroutine branch is defined by two sequence numbers in reverse order (e.g. G77 N200 N10), the system executes the programme in the normal sequence from N10 to N200 and no error is displayed.

Context

The calling programme context can be saved at the beginning of the called programme. It can then be restored at the end of execution of the called programme. The programmed status access symbols are used for save and restore (see Sec. 6.7).

Inhibiting display of subroutines being executed

Display on the programme page (PROG) of a subroutine and its internal subroutines during execution can be inhibited.

Placing the character «:» after the subroutine number inhibits display. Only the subroutine call block is displayed.

Example:

Main programme %10 calling subroutine %110: itself containing an internal subroutine %210.

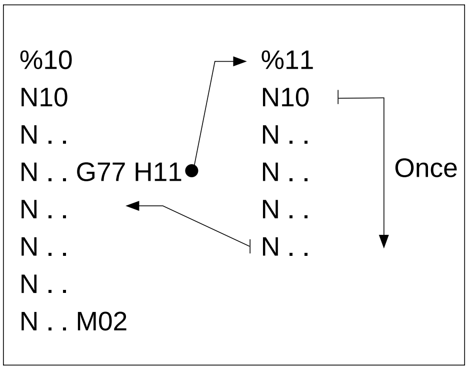
Only block N50 of programme %10 is displayed during execution of subroutines %110 and %210.

```
%10               %110:             %210
N10               N10               N10
N..               N..               N..
N50 G77 H110      N80 G77 H210      N..
N..               N..               N..
```

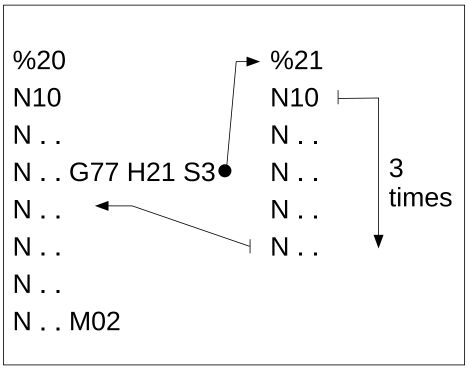
Examples

Branches to external subroutines from the main programme

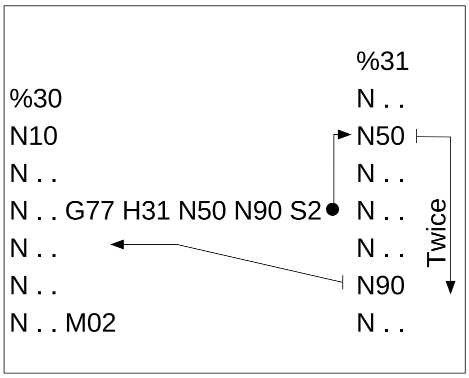
Branch to subroutine %11 from main programme %10.



Branch to subroutine %21 from main programme %20 three times.



Branch to blocks N50 to N90 of subroutine %31 from main programme %30 twice.



4

Branches to sequences in the programme

Branch to block N70 located upstream in programme %40.

The block can be called by:

`N.. G77 N70 N70`

or

`N.. G77 N70`

```
%40
N10
N . .
N . .
N70
N . .
N . . G77 N70 N70
N . .
N . . M02
```

Branch to blocks N80-N140 located upstream in the programme %50 three times.

```
%50
N10
N . .
N . .
N80
N . .     3 times
N140
N150 G77 N80 N140 S3
N . .
N . . M02
```

Branch to blocks N500-N550 located downstream in programme %60, between M02 and X OFF, four times.

```
%60
N10
N . .
N . . G77 N500 N550 S4
N . .
N400 M02
N500
N . .      4 times
N550
```

Subroutine nesting

Subroutine %5263 and the sequence of blocks N20 to N100 of main programme %123 are executed three times (S3).

Main programme

```
% 123
N10
N20
N..
N..
N50 G77 H252
N60
N..
N..
N100
N110
N..
N..
N180 G77 N20 N100 S3
N190
N..
N210
N..
N260
N..
N300 G77 N500 N550
N310
N..
N..
N390 M02
N..
N500
N..
N..
N550
```

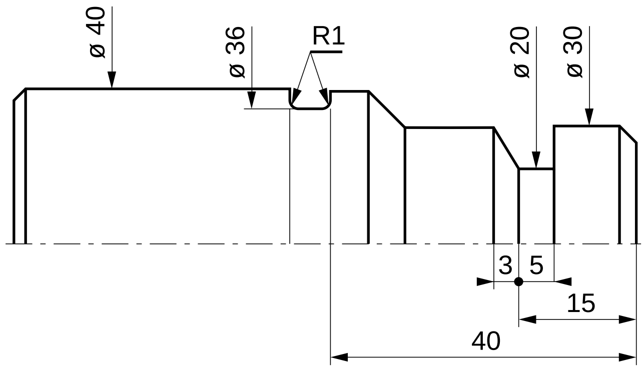
Subroutine 1

```
% 252
N10
N20
N..
N..
N80 G77 H5263 S3
N90
N..
N..
N160
```

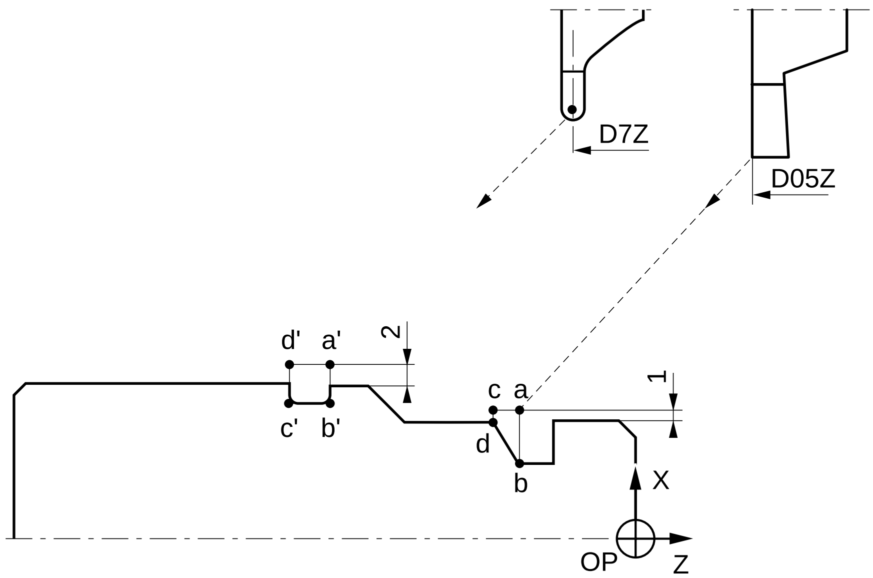
Subroutine 2

```
% 5263
N600
N610
N..
N..
N650
N660
N..
N..
N700 G77 N610 N650
N710
N..
N..
N920 G77 H123 N210 N260
N930
N..
N1290
```

Cutting grooves by calling two subroutines



Machining paths



Main programme

%30	
N10 G0 G52 X0 Z0	
$0 EXTERNAL MACHINING	
N..	
N..	
N110 G0 G40 G52 X.. Z.. G97 S900	
$0 VEE-GROOVE	
N120 T05 D05 M06 (TOOL WIDTH 5)	
N130 X32 Z-15	Point a, approach
N140 G96 S80	
N150 G77 H35	Call to subroutine %35
N160 G77 N110	Call to sequence N110
$0 GROOVE WITH RADIUS	
N170 T07 D07 M06 (TOOL RADIUS 1)	
N180 G0 G42 X44 Z-40	Point a’, approach
N190 G96 S50	
N200 G77 H40	Call to subroutine %40
N210 G77 N110 M05	Call to sequence N110
N220 M02	

Subroutine 1

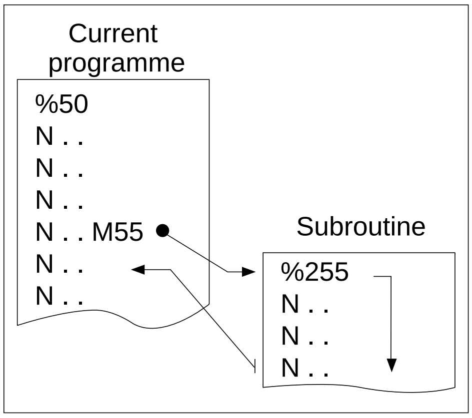
%35	
N10 G91 G01 X-6 G95 F0.08	Point b, incremental dimensions
N20 X6	Outfeed to point a
N30 G0 Z-3	Point c
N40 G01 X-1 F0.3	Point d
N50 X-5 Z3 F0.05	Point e
N60 X6 F0.08	Point a
N70 G90	

Subroutine 2

%40	
N10 G91 G01 X-4 G95 F0.06	Point b’, incremental dimensions
N20 Z-5 F0.08	Point c’
N30 X4 F0.3	Point d’
N40 G90	

4.11.2 Subroutine Branch by M Function

M..	Subroutine branch by M function.



The M.. function branches to a subroutine whose number is assigned by the machine manufacturer (see machine parameter P35).

Syntax

N.. **M..**

M.. Subroutine branch by M function.

Property of the Function

Function M.. programmed is modal.

Notes

A subroutine called by an M function cannot itself call another subroutine by an M function. However, it can be nested with another type of call (by G function or automatic control function), but in no case may two calls of the same type be nested.

A block including a subroutine call by M function may also include a positioning command.

Example: N.. M200 G00 X100

In this case, the block is processed in the following order:
- M200 sent to the PLC,
- Movement X100 executed,
- Call to the subroutine defined in P35.

Return to current programme

After execution of the subroutine, none of the data programmed earlier are restored.

The following data must be reprogrammed if necessary:
- modal preparatory G functions,
- modal technological S functions and miscellaneous M functions,
- correction D even if the tool was not changed,
- programme variables L.

Context

The calling programme context can be saved at the beginning of the called programme. It can then be restored at the end of execution of the called programme. The programmed status access symbols are used for save and restore (see Sec. 6.7).

Specific Feature of a Call to a Subroutine Number Between %11001 to %11999

When the subroutine number declared in P35 is between %11001 and %11999, the subroutine is called before execution of the block containing the M function and block concatenation is forced. It should noted that within the subroutine, execution of the functions is enabled by G998 and/or G997 as in subroutine calls by Gxxx function.

Example:

Execution of spindle indexing during axis movement in a block containing the programming: N.. X.. Y.. Z.. M19.

REMARK *For information on most of the functions used in the example below, refer to Sections 4.14.17, 6.7 and 7.3 of this manual as well as to the supplementary programming manual.*

```
%11019 (SPINDLE INDEXING DURING MOVEMENT)
VAR    [br_pos] = [.BM62] *2+ [.BM65] *2+ [.BM64] - [.BM63] /2&3+93524
       [M998] = [.BM999] - [.BM997] + 998
       [tix(9)] [i]
ENDV
M997 (forced block sequencing)
FOR [i] = 1 TO 9 DO [tix(i)] = [.IBX(i)]
   BCLR [.IBX(i)] (stop movement)
ENDF
G997 (exit M19)
G999
FOR [i] = 1 TO 9 DO
   IF [tix(i)] = 1 THEN
      BSET [.IBX(i)] (enable movement)
   ENDI
ENDF
G997 (start movement)
(wait until spindle is in position)
WHILE E [br_pos] = 0 DO G4 F.1
ENDW
M[M998]
```

Example

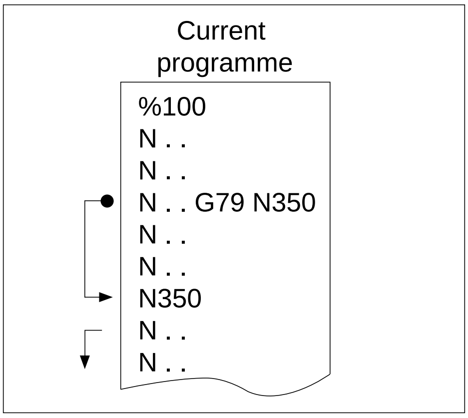
General case: call by function M55 of subroutine %255.

Current programme	Subroutine %255
%50	%255
N10 ...	N10 ...
N..	N..
N80 M55	N..
N..	N120
N..	
N950 M02	

4.11.3 Branch to a Sequence without Return

G79	Conditional or unconditional branch to a sequence without return.

A conditional or unconditional branch is made to sequence number N.. associated with the function..



Syntax

N.. **G79** [L../E.. > = < Number] N..

G79	Conditional or unconditional branch to a sequence (an upstream or downstream branch is allowed).
L.. or E..	Variable L or parameter E tested in the condition (see Secs. 6.1 and 6.2).
> = <	Comparison symbols for the condition (up to two symbols can be used).
Number	Numerical expression of the condition.
N..	Mandatory argument defining the sequence number to which the jump is to be made.

Property of the Function

Function G79 is nonmodal.

Cancellation

Function G79 is cancelled at the end of the block.

Notes

If the jump is conditional, the condition must be located between G79 and N..

Examples

Unconditional jump

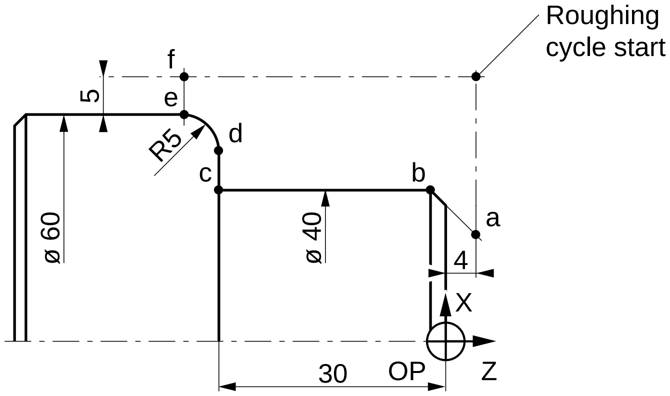
```
%25
N10 ...
N..
N50 G79 N220          Jump to sequence N220
N60
N..
N190
N..
N220
N..
N250 G79 N60          Jump to sequence N60
N..
```

Conditional branch

Counting the number of machining operations performed and branching when the specified number is reached.

```
%75
N10 L5 = 20                 Variable L5 set to 20
N20 G52 Z..
N30 T01 D01 (TOOL)
N40 S800 M40 M03
N50 G00 X.. Y.. Z..
N60 G01 Z..
N70
N..
N..                         Programming for one machining
N..                         operation
N..
N..
N..
N140
N150 L5 = L5 -1             Count each operation down
N160 G79 L5 = 0 N180        Condition: If L5=0, branch to block N180
N170 G79 N50                Branch to N50
N180 G52 Z.. M05
N190 M02
```

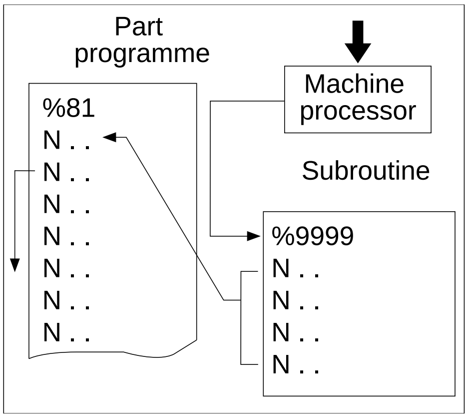
Roughing cycle with unconditional branch




```
%82
N10 G00 G52 X.. Z..
N20 T01 D01 M06 (BIDIRECTIONAL TOOL)
N30 S900 M40 M04
N40 X70 Z4                              Roughing start point
N50 G92 S3000
N60 G96 S200
N70 G79 N140                            Branch to block N140
N80 G00 X30 Z4                          Point a
N90 G01 X40 Z-1                         Point b
N100 Z-30                               Point c
N110 X50                                Point d
N120 G03 X60 Z-35 R5                    Point e
N130 G01 X70                            Point f
N140 G64 N80 N130 I0 K0 P2 G95 F0.2
N150 X70 Z-35
N160 Z4
N170 X30
N180 G80 G52 X-50 Z-100 G97 S900
N190 ...
```

4.11.4 Subroutine Call by Automatic Control Function

During execution of a part programme, the automatic control function causes a branch to subroutine %999.



Condition for the branch to subroutine %9999

Part programme being executed in mode:
- automatic (AUTO),
- single step (SINGLE),
- dry run (DRYRUN).

A subroutine called by the automatic control function cannot itself call another subroutine by the automatic control function. However, it can be nested with another type of call (by M function or G function), but in no case may two calls of the same type be nested.

Start of subroutine %9999

If a part programme is already being executed, the branch to subroutine %9999 is only made at the end of an interruptible block or the end of execution of an uninterruptible block.

Definition: An uninterruptible block is a block containing a G function causing a branch to a subroutine (G31, G45, G81, etc.) or a block for which certain parameters must be known in order to execute the next block (sequencing of three blocks in PGP mode).

If no part programme is being executed when subroutine %9999 is called, the CYCLE field goes out at the end of execution of the subroutine (the part programme is not executed).

Functions G01 and G40 are set at the start of the subroutine.

During execution of the subroutine %9999

The PLC ignores a new call or maintaining of the call to subroutine %999 during execution of this subroutine.

End of the subroutine %9999

At the end of execution of the subroutine, the system does not make a handshake with the m/c processor. It is the subroutine's responsibility to report on its execution by function M or external parameter E.

The subroutine should end with X OFF unless a return to the current programme is not desired. In this case, function M02 (end of programme) is programmed at the end of the subroutine.

Return to current programme

After execution of the subroutine, none of the data programmed earlier is restored.

The following data must be reprogrammed if necessary:
- modal preparatory G functions,
- modal technological S functions and miscellaneous M functions,
- correction D even if the tool was not changed,
- programme variables L.

Context

The calling programme context can be saved at the beginning of the called programme. It can then be restored at the end of execution of the called programme. The programmed status access symbols are used for save and restore (see Sec. 6.7).

Structure of subroutine %9999

When several functions can be processed by a subroutine, the PLC must specify the function called. This may be done by external parameter E40000 (see Sec. 6.2).

Example:

Method 1:

Each function is the subject of a specific subroutine (%a, %b, %c, etc.). In this case, subroutine %9999 is a single block used to re-direct control.

```
%9999
G77  H E40000 M..
N..
```

Parameter E40000 contains the number of the subroutine called (a, b, c, etc.) and the M function is used as a handshake (CRM).

This method has the drawback of creating an additional subroutine nesting level.

Method 2:

All the functions are written in subroutine %9999. The first block is a jump to a sequence number contained in parameter E40000 (i.e. Na.., Nb.., Nc..).

%9999	
G79 N E40000 M..	Wait for start by CRM and the branch to the sequence
Na	Processing of function 1
N..	
N..	
G79 Nz	Jump to the last sequence
Nb	Processing of function 2
N..	
N..	
G79 Nz	Jump to the last sequence
Nc	Processing of function 3
N..	
N..	
Nz	End of subroutine

Subroutine Call by Automatic Control Function with Axis Multigroups

See Sec. 4.17 (special programming for axis multigroups).

4.11.5 Block Interrupt

G10	Block interrupt.

G10 blocks are designed to be interrupted by a hardware input or by satisfying a position threshold for a particular axis

After the interrupt, a branch may be arranged to a designated block. If no interrupt is detected, the next block is executed.

Programme

```
%300
N . .
N . .
N . .
N200 X . . G10 @7 < 50.3 N250
N . .
N . .
N250
N . .
```

Syntax

N.. [G40] [G04 F..] [G00/G01/G02/G03] X.. Z.. **G10** [:n] [+X.. or F..] [@n < > Value] N.. [+ Number] [EF..]

G40	Tool radius offset cancel.
G04 F..	Interruptible dwell.
G00/G01/G02/G03	Interruptible interpolations.
X.. Z..	Interruptible axes.
G10	Block interrupt function.
:n	Numerical argument (number of interrupts from 1 to 99) significant only for hardware interrupts (on-the-fly dimension measurement). The block is not interrupted (forcing an end of block) until the nth hardware interrupt.
+X.. or F..	Arguments defining a movement or dwell to be made after the interrupt(s).

X..: A further movement to be made after the interrupt(s) before deciding whether to branch (possible on all the axes of the system, both measured and servoed).

F..: Dwell in seconds during which the block may be interrupted.

@n < > Value	Argument defining a condition for comparison with a threshold. @n: Physical address of the axis concerned by the test (axis address n is between 0 and 31). <>: Mandatory comparison symbol. Value: Comparison threshold in the same programme unit as the other axes of the group (mm or inches).
N.. + Number	Branch to block N.. after the interrupt, possibly followed by a further number to reach unnumbered blocks.
EF..	Maximum feed rate after an interrupt (see Sec. 4.11.5.1).

Property of the Function

Function G10 is nonmodal.

Cancellation

Function G10 is cancelled at the end of the block.

Notes

All the arguments for function G10 are optional.

Analysis of the interrupt

When function G10 is programmed, the end point programmed in the block may be modified by setting of the limit switch bit (ARBUT) or by a CNC interrupt generated by an IT input (part gauging function). Either of these inputs stops movement and replaces the dimensions of the required position by the dimensions of the current position.

When the programming movement is finished, the system jumps to the programmed sequenced or by default to the next sequence.

On-the-fly dimension measurement (hardware interrupt)

The hardware interrupt is generated on an interrupt input.

If movement is stopped by a gauging interrupt, the automatic control function processes this IT and informs the CNC. When interrupt IT is generated, the dimensions of the axes in the group are stored in external parameters E7x001 (axis position reference where x = physical address of the axis, see Sec. 6.2). The interrupt is taken into account by function G10 of the part programme that can then cause a branch.

The new values can be detected by dynamic operators and stored in an indexed addressing table.

REMARK *With axis multigroup programming (see Sec. 4.17), when an interrupt on one group is set at the same time as an interrupt on another group (groups 1 to 8), the interrupt on the group with the lowest number has priority.*

Examples

G10 block with a jump to an unnumbered block

Jump to the third block after N150 if a hardware interrupt is set.

```
N.. ...
N80 G01 G41 X.. Z.. F200
N90 G40 X.. G10 N150 +3     Jump to N150
N100
N..
N150
N..
N..       + 3 blocks
N..
N..
```

Interruptible block including comparison with a threshold and jump to a block once the condition is satisfied

Condition: Jump to block N240 when the measurement on axis 7 is less than 50.3.

```
N.. ...
N..
N200 X.. Z.. G10 @7 < 50.3 N240     Jump to N240
N..
N..
N240
N..
```

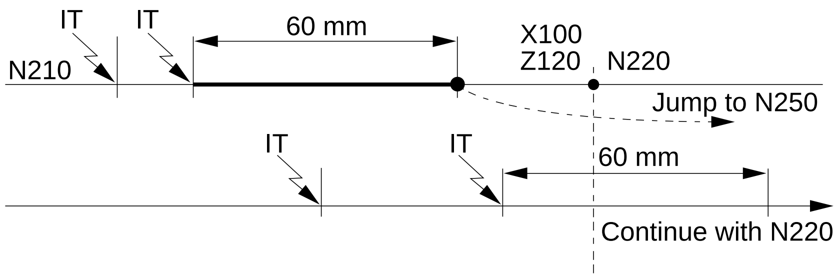
Interruptible block including comparison with a threshold and jump to a block if the condition is satisfied

```
N..
N200 G00 G40 X50
N210 G01 X100 Z120 G10 : 2 + X60 N250      Jump to N220 or N250
N220      Condition not satisfied
N230
N240
N250      Condition satisfied
N..
```

If, after the interrupt, the end point is beyond the original programmed target:

- the jump condition is not satisfied and the programme continues with the next block,
- no updates are made to the E7x001 positions.

Diagram:



4.11.5.1 Special Use of Sequence Interrupt

Block Sequencing without Stopping Movement

In state M999 (see Sec. 4.14.9), the analysis of the following blocks is suspended until the interrupt is detected, whether this interrupt is from the PLC (programmed with A.15B) or a hardware interrupt or a conditional interrupt determined by comparing an axis measurement to a threshold.

When the interrupt is detected, if the current interpolated position (to which is added the +X value programmed after G10) is beyond the programmed end point, it should be noted that:

- analysis of the following blocks is resumed immediately,
- in case of possible processing by parametric programming, the current block is considered ended, thereby authorising read or write of external parameters E (see Sec. 6.2) without effectively stopping current movements (this allows calculation of a positioning point according to the interrupt point and taking into account the geometric transformations made downstream of the interpolator),
- as soon as the next block is ready, the feed rate at the end of the current block is updated according to whether the path has changed or not.

REMARK *In state M999, if the block was not interrupted, a subroutine call by the PLC is accepted at the end of execution of the block.*

Feed Rate Limiting After an Interrupt

Before the interrupt, the speed of movement in block G10 is the modal feed rate F programmed before G10. After the interrupt, the feed rate at the end of the current block may be different if argument EF and its value (less than F) are programmed in that block or an earlier block.

Example

```
N..
N.. G01 X30 G94 F300
N..
N.. G00 X..
N100 G01 X0 F150 G10 +X2 EF100        Feed rate 100 mm/min after the
                                      interrupt
N..
```

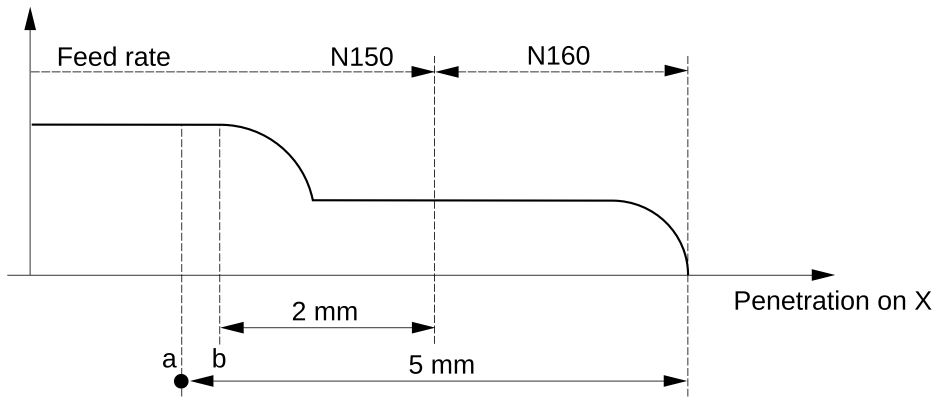
Example

The example below is defined for a system using inclined axis programming.

For the specific data used in the example, refer to the additional information given in this manual for:
- Inclined axes
- External parameters E...
- Symbolic variables [...].

```
%398
N..
N..
(Compute the offset between the machine dimension and programme dimension in the
programme reference system)
[K] = CE69001 [DECX] = UE70000 * [K] - [.IRX(1)]
N150 G01 X0 F200 G10 +X2 EF100
(Go to 5 mm from the interrupt point)
N160 [COTE] = [K] * UE70001 - [DECX] - 5 X[COTE] F100
N..
```

In the figure:
- point a: point measured on the interrupt
- point b: current position interpolated at the time of the interrupt.



4.11.6 Temporary Suspension of Next Block Preparation

G79 +/-	Temporary suspension of next block preparation in a sequence with movements.

Programming this function in a block with movements temporarily suspends preparation of the next block. A lead or lag (expressed in millimetres or seconds) at the beginning of the current block or at the end of execution of the current block is programmed on the restart of block processing.

Syntax

N.. [G00/G01/G02/G03] X.. Y.. Z.. **G79 +/-** X.. / F..

G00/G01/G02/G03	Linear or circular interpolation.
X.. Y.. Z..	End point.
G79	Temporary suspension of next block preparation in sequences with movement.
+/-	+: Lag with respect to the start of the current block. -: Lead with respect to the end of execution of the current block
X.. / F..	X..: Lag or lead in millimetres F..: Lag or lead in seconds.

Properties of the function

Function G79 is nonmodal.

Cancellation

Function G79 is cancelled at the end of the block.

Notes

Restart of processing after programming of the function

As soon as analysis of the next block is restarted, it should be noted that:
- in case of possible processing by parametric programming, the current block is considered ended, thereby authorising read or write of external parameters E (see Sec. 6.2) without effectively stopping movements,
- as soon as the following block is ready, the feed rate at the end of the current block is updated,
- this allows write of external parameters E and discrete outputs or read and test of these parameters without stopping movements.

Example

Example with timeout at the start of the current block

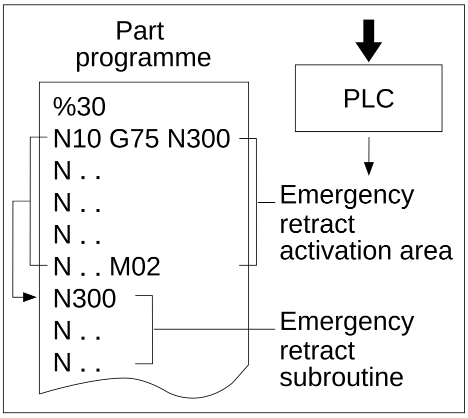
```
N..
N110 G00 X0 Z5
N120 G01 X100 G79+F4          Movement along X with 40-second
N130 G79 E10000=1 N300        timeout before analysis of block N130
N140 Y20
N..
N300 G01 X200
N..
```

Example with anticipation at the end of the current block

```
N..
N110 G00 X50 Z5               Movement along Y with anticipation of
N120 G01 Z100 G79-X10         10 mm before analysis of block N130
N130 ...
```

4.11.7 Emergency Retract

G75	Declaration of an emergency retract subroutine.



The emergency retraction subroutine is activated by a request from the PLC.

Activation stops the current programme and causes a branch to address N.. of the last emergency retract subroutine declared.

4

Syntax

N.. **G75** N..

G75	Declaration of an emergency retract subroutine.
N..	Mandatory argument associated with the function and giving the address of the emergency retract subroutine start sequence number.

Properties of the Function

Function G75 is nonmodal. Argument N.. associated with the function is modal.

Cancellation

Declaration of a subroutine G75 N.. is cancelled by:
- cancellation function G75 N0,
- function G75 N.. with a different subroutine number,
- end of programme function M02,
- a reset.

Notes

<u>Condition for activation of the emergency retract subroutine</u>

Part programme being executed in one of the following modes:
- automatic (AUTO),
- single step (SINGLE),
- manual data input (MDI),
- dry run (DRYRUN).

An emergency retract subroutine can be activated:
- after reading the block in which it is programmed,
- as long as the programme or subroutine in which it is declared is not completed,
- as long as a new emergency retract subroutine has not been declared,
- as long as emergency retract has not been cancelled by G75 N0.

If emergency retract is activated but no emergency retract subroutine has been declared in the part programme, this instruction has the same effect as pressing the «ARUS» (cycle stop) key.

Activation of an emergency retraction is sent by the PLC function with the emergency retraction request «C_DGURG».

In automatic mode, the emergency retraction programme remains active until encountering a function M00 or M02.

Emergency retraction on PLC axis groups

For information, see Sec. 4.18 (programming specific to PLC axes).

Examples

Declaration of emergency retract subroutines from a main programme

Programme	
%30 N10 N.. N40 G75 N1000	
N.. N.. N180	If emergency retract is activated in this area of the programme, branch to N1000
N190 G75 N2000	
N.. N.. N290 N300 M02	If emergency retract is activated in this area of the programme, branch to N2000
N1000 N.. N.. N1050 M02	Emergency retraction subroutine available for N40 to N180
N2000 N.. N.. N.. N2080 M00	Emergency retraction subroutine available for N190 to N290
N2090 N.. N.. N2120	Part programme resumption sequences

4

Declaration of emergency retract subroutines from a programme including subroutine branches

If an emergency retract subroutine is activated during execution of subroutine %4 called by N200 of %2, branch to N300 of %2.

If an emergency retract subroutine is activated during execution of subroutine %4 called by N40 of %3, branch to N200 of %1.

```
%1
N10 G75 N200     Déclaration de
N..              dégagement
N..              d'urgence
N..

N100 G77 H2
N..
N150 G77 H3
N..
N180 M02
N200
N..              Programme de
N..              dégagement
N..              d'urgence
N.. M02
```

```
%2
N10 G75 N300     Déclaration de
N..              dégagement
                 d'urgence
N200 G77 H4
N..
N.. G79 N390
N300             Programme de
N..              dégagement
N.. M02          d'urgence
N390
N..
```

```
%3
N10
N..
N40 G77 H4
N..
```

```
%4
N10
N..
N..
```

4.11.8 Branch to Automatic Homing Subroutine

Homing can be carried automatically on each of the machine axes by running subroutine %9990.

Conditions for running subroutine %9990

Subroutine %9990 is run by pressing the cycle start button:
- if the system is in homing mode,
- if no other programme is being executed.

During execution of subroutine %9990

As soon as subroutine %9990 or one of its subroutines starts, the machine axes can be moved without affecting homing.

End of execution of subroutine %9990

At the end of execution of subroutine %9990 (by programming of M02), the programme that was active when %9990 was called is resumed.

Automatic homing with multiple axis groups

See Sec. 4.17 (programming specific to multiple axis groups).

4.11.9 Subroutine Branch on a Reset

On a reset, the NC groups can execute subroutine %11000.

Conditions for running subroutine %11000

Subroutine %11000 is run:
- if bit 3 of machine parameter P7 word 2 is a 1,
- if this subroutine is present in the memory.

Start of execution of subroutine %11000

Subroutine %11000 begins approximately 100 microseconds after the reset pulse (S RAZ or E_RAZ) goes high. It should be noted that this pulse remains high until the subroutine ends.

Contents of subroutine %11000

A subroutine %11000 must not include any executable functions such as:
- Movements on the axes
- Dwells (G04 F..)
- Miscellaneous functions (M..), etc.

Only the following operations are allowed in subroutine %11000:
- read and write of external parameters E and programme variables L (see Chapter 6). It should be noted that at the end of execution of the subroutines, variables L0 to L19 are reset but the others preserve their values,
- declaration of dynamic operators and enabling of interaxis calibration (see Dynamic Operators Manual),
- declaration and initialisation of symbolic variables [...]. It should be noted that they must be saved by the SAVE function to be available for use after execution of the subroutine (see Supplementary Programming Manual).

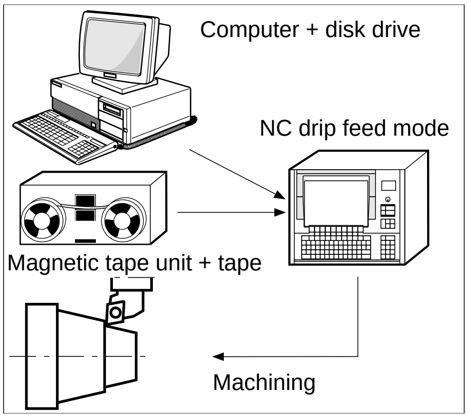
Subroutine branches on a reset in multiple axis groups

See Sec. 4.17 (programming specific to multiple axis groups).

4.11.10 Restrictions Related to Drip Feed Mode

Drip feed mode allows a programme to be checked or to be executed as the programme is read from a reader or transmitted from a host (DNC1).

Drip feed mode is used for execution of large programmes which cannot be stored in the NC (for the procedure, see operator's manual).



Notes

Use of drip feed mode causes allocation of a buffer space of up to 32,000 characters.

When the space available is less than this value but more than 1 KB, the system takes all the remaining space.

If the space available is less than 1 KB, drip feed mode is refused and error message 36 is reported.

Restrictions related to drip feed mode

It is impossible to use functions referring to block numbers in the programme:
- jump to a sequence (G79 N..),
- block interrupt (G10 N..),
- branch to sequences of the main programme (G77 N.. N..),
- activation of an emergency retract subroutine (G75 N..).

Branches to subroutines stored in the memory are allowed in drip feed mode (G77 H.. or G77 H.. N.. N..).

4.11.11 Call to Subroutine Return Block

G77 -i	Call to subroutine return block.

This function allows a subroutine to call and execute the instructions of the return block of the calling subroutine.

Syntax

N.. **G77 -** i

G77	Call to subroutine return block.
-i	Immediate value (or variable) giving the nesting level of the programme containing the block called with respect to the nesting level in which G77 -i is programmed (the return block called in this way is automatically postincremented).

Properties of the Function

Function G77 -i is nonmodal.

Cancellation

Function G77 -i is cancelled at the end of the block.

Notes

If the block called is in the «displayable» programme, it is displayed on the programme page (PROG) even if the subroutine that called it is not displayable.

It should be noted that:
- if the nesting level requested is not accessible, the system returns error message 26
- the blocks called by G77 -i must not include branches or other calls (G79, G77, etc.) or the system returns error message 26.

Examples

General Example

```
%1              %10             %100
N..             N..             N..

N..             N..             N..

N3 G77 H10      N30 G77 -1      N300 ...

N4 ...          N40 G77 H100    N400 G77 -2

N5 ...          N50 ...         N500 ...

N6 ...          N60 G77 -1      N600 ...

N7 ...          N70 ...         N700 ...

N8 ...          ...             ...
```

Specific Example

REMARK *For information on the functions used in the example below, refer to Sections 4.14.17, 6.7 and 7.7 of this manual as well as to the supplementary programming manual.*

```
%1              %10100
N...            ... (processing of the calling block)
N...            WHILE [.NOG80] = 100 DO G999 G77 -1
G100 ...        ... (processing of the blocks located between G100
...             ... and G80 G997 (execution)
...             ENDW
G80
...
```

4.11.12 ISO Programme or Block Creation/Deletion

G76+/- ISO programme or block creation/deletion.

4.11.12.1 General

Programming of function G76+/- provides the following capabilities:
- Programme creation
- Programme deletion
- Block insertion in a programme
- Block deletion from a programme.

This function is especially useful for creating part programmes automatically (e.g. after teach-in by gauging of dimensions).

This function can only be used to create and delete programmes and blocks located in part programme area zero.

REMARK *Before execution of a function G76+ (creation, insertion), the system makes sure no other programme editor or graphic display is active. If so, programme execution is suspended as long as these conditions remain present.*

With G76, the programme number can be indexed (e.g. H123.2). This is accepted providing no other programme in any of the programme areas has the same number.

If there is not enough memory space for programme creation or block insertion, the system returns error message 266.

If the block including function G76 has a format error, the system returns error message 1.

Properties of the Function

Function G76 is nonmodal and is cancelled at the end of the block.

4.11.12.2 Creating a Programme

The syntax below is used to create a programme in area zero.

Syntax

N.. **G76+** H..

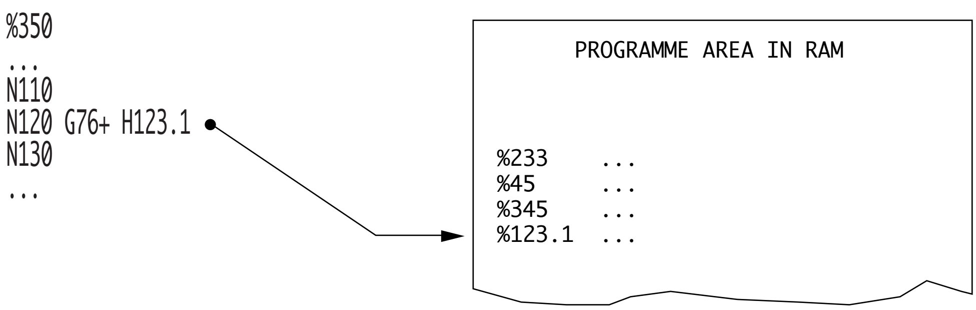
G76+	Programme creation function.
H..	Number of the programme to be created.

Notes

In programme creation mode, the programme number must be the last word in the block.

Example

Programme creation in RAM programme area (area zero).



4.11.12.3 Deleting a Programme

The syntax below is used to delete a programme located in area zero.

Syntax

N.. **G76-** H..

G76-	Programme deletion function.
H..	Number of the programme to be deleted.

Notes

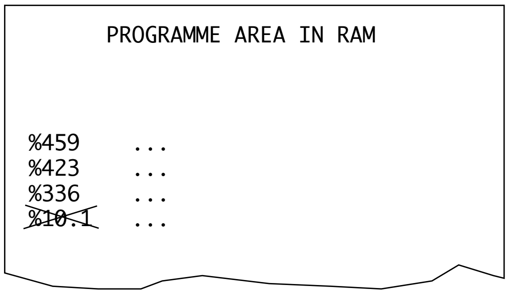
The programme number must be the last word in the block.

Several cases can occur, depending on the area where the programme is located:
- If a programme in area zero has the number specified in the function, this programme is deleted
- If a programme in an area other than area zero has the number specified in the function, deletion is refused and the system returns error message 266
- If no programme in any area has the number specified with the function, the command is accepted (but no programme is deleted).

Example

Programme deletion in RAM programme area (area zero).

```
%1050
...
N190
N200 G76- H10.1
N210
...
```



4.11.12.4 Inserting a Block

The syntax below is used to insert a block in an existing programme.

Syntax

N.. **G76**+ [H..] N..[+number] ISO block

G76+	Function defining block insertion.
H..	Specifies the programme number where the block is to be inserted (optional: default H..: the ISO block is inserted in the programme including function G76).
N.. +number	N..: Block number (mandatory). The insertion is made after this block unless "+number" is specified (note: the first block of programme %.. is block N0). +number: (optional) defines the number of lines (after the block number specified) where the insertion is to be made.
ISO block	Block to be inserted consisting of ISO functions (see list under Notes).

Notes

The following ISO functions are accepted:

- N.. (block number)
- G.. (preparatory functions; several G functions are accepted in the block)
- X.. Y.. Z.. U.. V.. W.. A.. B.. C.. (axes and dimensions)
- I.. J.. K.. (circle centre coordinates)
- P.. Q.. R.. (material vector for 3D radius correction)
- /.. (polynomial coefficients)
- L=.. to L9=.. (programme variables).

The values associated with the above functions are immediate integer or fractional values, programme variables (L..), symbolic variables [sym] or parameters E. When they are parameters E, the values are specified in the internal system unit (these data are edited in the programme as immediate values, signed if negative, and in the internal unit defined for linear or rotary axes).

The block to be inserted must not have more than 120 characters or the system returns error message 1.

If there is not enough memory space for block insertion, the system returns error message 266.

If the block to be inserted does not exist, the system returns error message 25.

Example

Block insertion in programme %336 located in the RAM programme area (area zero).

```
%36
N.. ...
N290 ...
N300 G76+ H336 N100 +2  N125 G01 X50 Z20
N310 ...
...


%336
N.. ...
N100 ...
N110 ...     N100 +2
N120 ...
             Block insertion
N130 ...
...
```

4.11.12.5 Deleting a Block

Syntax

N.. **G76-** [H..] N..[+number]

G76-	Block deletion function.
H..	Specifies the programme number from which the block is to be deleted (optional: default H..: the ISO block is deleted from the programme including function G76).
N.. +number	N..: Block number (mandatory). It is this block that is deleted unless +number is programmed (note: the first block of programme %.. is block N0). +number: (optional) specifies the line to be deleted (after the block number specified).

Notes

If the block to be deleted does not exist, the system returns error message 25.

Example

Deletion of a block in programme %567 located in the RAM programme area (area zero).

```
%222
N.. ...
N90 ...
N100 G76- H567 N200 +4
N110 ...
...

%567
N.. ...
N.. ...
N200 ...
N210 ...
N220 ...      N200 +4
N230 ...      Block deletion
N240 ...
...
```

NUM
1020/1040/1060T

PROGRAMMING MANUAL

VOLUME 2

0101938820/5

06-97 **en-938820/5**

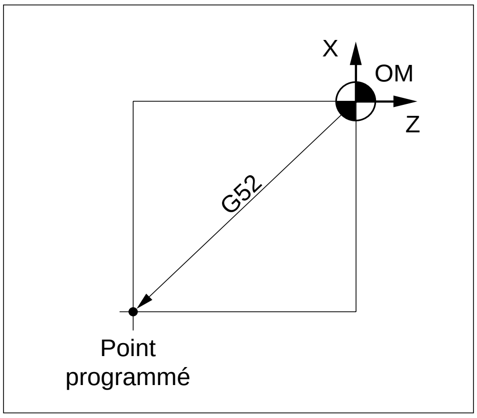
4.12 Movement Origin Selection

4.12.1 Programming of Movements in Absolute Coordinates Referenced to the Measurement Origin

G52	Programming of movements in absolute coordinates referenced to the measurement origin.

Movements programmed with this function are referenced to the measurement origin (OM).

All the axes are programmable with reference to the measurement origin.



Syntax

N.. [G40] [G90] [G00/G01] **G52** X.. Z.. C.. [F..]

G40	Radius offset cancel.
G90	Programming in absolute dimensions.
G00/G01	Linear interpolations at high or programmed speed.
G52	Programming of movements in absolute coordinates referenced to the measurement origin.
X.. Z.. C..	End point referenced to the measurement origin.
F..	Feed rate.

Property of the Function

Function G52 is nonmodal.

Cancellation

Function G52 is cancelled at the end of the block.

Notes

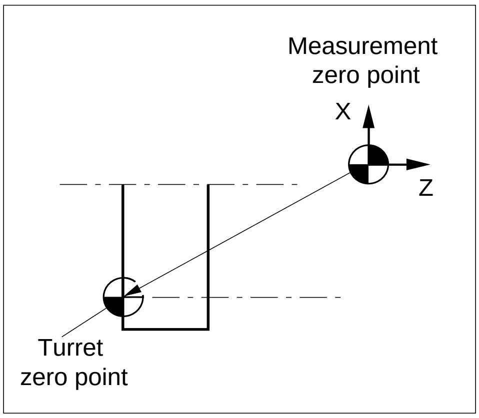
Programming function G52 in a block suspends the following data:
- tool data,
- DAT1,
- DAT2,
- programme origin offset (G59),
- angular offset (ED..),
- scaling factor (G74).

Function G52 must:
- precede programming of the axes in the block,
- be programmed with the system in G40 mode (radius offset cancel) or the system returns error message 27,
- be programmed in absolute coordinates (G90).

Examples

Programming G52 on the X and Z axes with reference to the measurement zero point before a tool change

```
%10
N10 G00 G52 X0 Z0
N20 T03 D03 M06
N..
```



Programming of G52 on axes Z, X, C:
- -50 mm from the measurement zero point on the Z axis
- -100 mm from the measurement zero point on the X axis
- 180 degrees from the measurement zero point on the C axis.

```
N.. ...
N220 G00 G52 Z-50
N230 G52 X-100 C180
N..
```

Programming of the carrier/carried axes

A parallel carrier/carried axis pair can be programmed in G00 with reference to the measurement origin. In all other cases, programming in G00 is prohibited.

Example:

DAT1 and DAT2 do not apply to the values programmed with the Z and W axes.

```
N..
N.. G00 G52 Z.. W..
N..
```

4.12.2 Datum Shift DAT1 and DAT2 Cancel/Enable

G53	DAT1 and DAT2 cancel.

G54	DAT1 and DAT2 enable. .

These functions are used to enable or cancel DAT1 and DAT2 shifts entered on the SHIFT page.

CANCEL / ENABLE

Scale E1000/1000

DAT1	DAT2
X - 100	+ 0
Z - 200	+ 20
C + 0	+ 0

Syntax

N.. **G53/G54**

G53	DAT1 and DAT2 cancel.
G54	DAT1 and DAT2 enable.

Properties of the Functions

Functions G53 and G54 are modal.

Function G54 is the default function.

Cancellation

Functions G53 and G54 cancel one another.

Notes

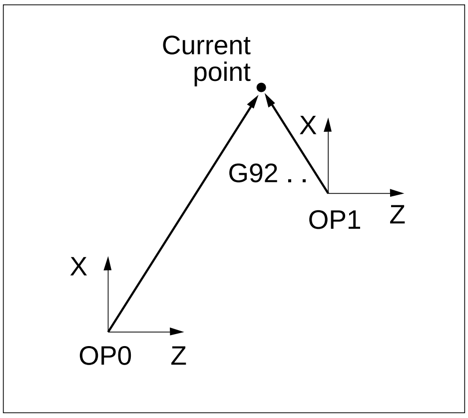
The tool offsets are not affected by function G53.

4.12.3 Programme Origin Preset

G92	Programme origin preset.

This function, associated with one or more axes and their values, defines the current point with reference to a new programme origin.

DAT1 values are redefined for the programmed axes.



Syntax

N.. **G92** X.. Z..

G92	Programme origin preset.
X.. Z..	Position with respect to the new programme origin.

<u>Calculation for programme origin preset on an axis:</u>

New DAT1 = Previous DAT1 + Previous current point/OP - Value programmed with G92

or

New DAT1 = Current point/OM - Value programmed with G92 - Tool length (in the axis) - DAT2

This operation is not performed until completion of the block before that containing function G92.

CAUTION

The new offsets remain active at the end of the programme.

Programme origin preset function G92:
- applies to all the axes, both carried and independent,
- is rejected if the last movement was programmed with respect to the origin, error message 2,
- is ignored in test (TEST) and sequence number search (SEARCH) modes,
- suspends analysis of blocks until execution of the previous block is completed,
- cannot be programmed with radius offset; the system must be in state G40,
- cannot be programmed in a PGP (Profile Geometry Programming) sequence.

Example

Value entered for Z DAT1 = -300

Value entered for Z DAT 2 = 20

Tool length L (correction D9) = 80

Programme origin preset G92 Z60

```
N..
N150 G00 D9 G40 X0 Z40
N160 G92 Z60                          Preset
N170 G00 Z..
N..
```

Analysis of block N150 yields:
- current point/OM Z = -160

Application of the first equation

New Z DAT1 = -300 + 40 - 60 = -320

Application of the second equation

New Z DAT1 = -160 - 60 - 80 - 20 = -320

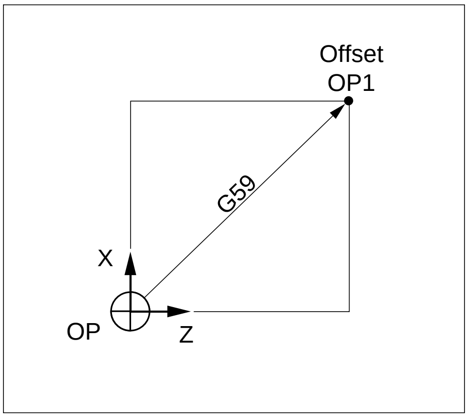
4.12.4 Programme Origin Offset

G59	Programme origin offset.

This function, associated with one or more arguments, axes and values, offsets the programme origin (OP).

An origin offset can be assigned to each system axis.

No movement is caused by the function and its arguments.



4

Syntax

N..[G90/G91]**G59** X.. Z.. U.. W.. C..[I.. K.. ED..]

G90/G91	Programming in absolute or incremental dimensions.
G59	Programme origin offset.
X.. Z.. U.. W.. C..	The axes programmed are the arguments associated with the function. They must immediately follow the function and at least one axis must be programmed.
I.. K..ED..	- I.. K..: Arguments defining the centre of rotation of an angular offset programmed with ED in the plane with respect to the initial programme origin (see Fig. 1). - ED..: Angular offset.

Properties of the Function

Function G59 is nonmodal. The arguments associated with the function are modal.

Cancellation

A programmed offset G59.. is cancelled by:
- programming G59 with axis arguments assigned zero values in absolute dimensions (G90),
- the end of programme function (M02),
- a reset.

Notes

For simplification and understanding of the programme, it is recommended to be in state G90 (absolute programming) before programming an origin offset.

Function G59 programmed in absolute dimensions (G90):

The origin offset G59 ... is referenced to DAT1 + DAT2. A new origin offset G59 ... replaces a previous one.

Function G59 programmed in incremental dimensions (G91):

The first movement programmed after G59 ... is translated by the value of the programme origin offset. The absolute position is offset by the sum of all the incremental G59 ... functions programmed earlier.

The functions below, when included in a programme, must be programmed in the following order:
- G59 ... Programme origin offset,
- ED.. Angular offset,
- G51 ... Mirror function,
- Scaling factor.

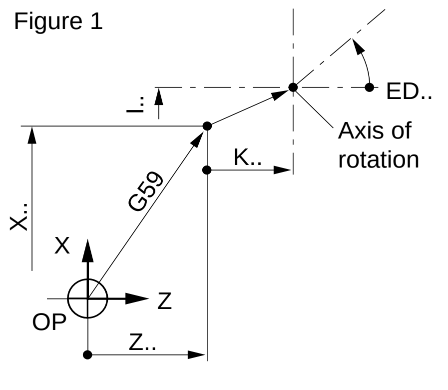
Notes on arguments I, K

Machining programmed with reference to a programme origin (OP) can be translated and oriented according to the angle programmed with ED (see Sec. 4.12.5).

```
N.. ...
N.. G59 X.. Z.. I.. K.. ED..
N.. ED..
N..
```

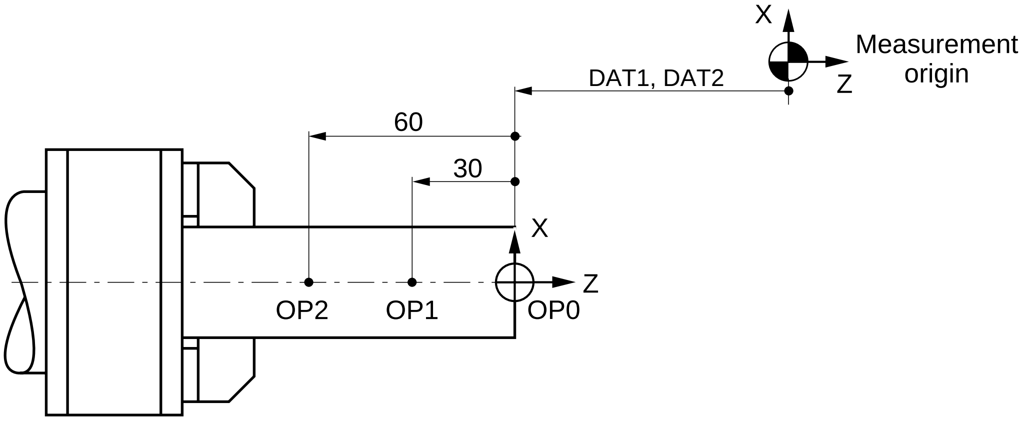
An offset on the X and Z axes is not mandatory for programming I and K.

Figure 1



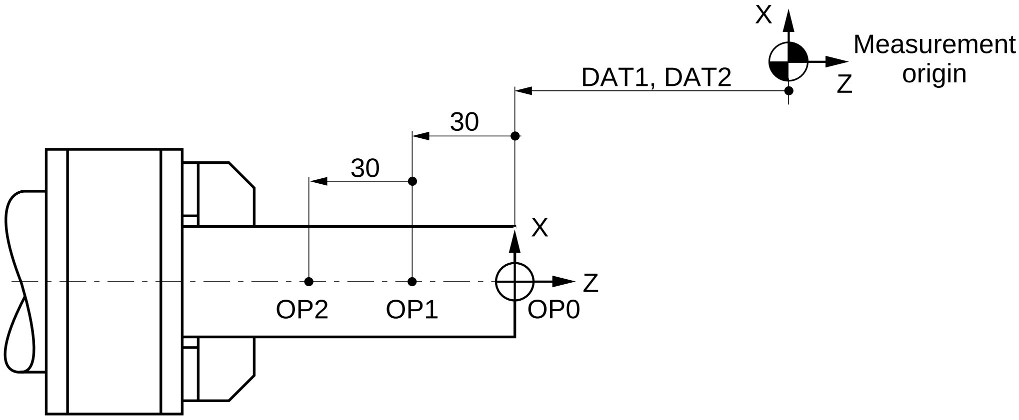
Examples

Origin offsets on the Z axis in absolute programming (G90)




```
%60
N10
N..
N50
N..
N90
N..
N120 G90 G59 Z-30               Offset 1
N.. G77 N50 N90                 Machining
N..
N230 G59 Z-60                   Offset 2
N.. G77 N50 N90                 Machining
N..
N350 G59 X0                     Offset cancelled
N..
```

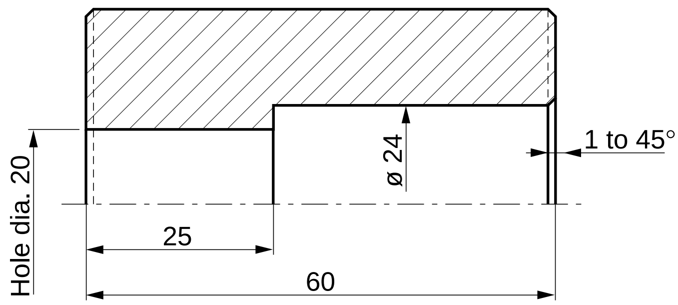
Origin offsets on the Z axis in incremental programming (G91)



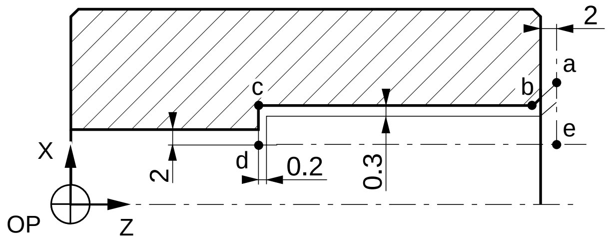

```
%70
N10
N..
N50
N..
N90
N..
N120 G91 G59 Z-30                  Offset 1
N.. G77 N50 N90                    Machining
N..
N230 G59 Z-30                      Offset 2
N.. G77 N50 N90                    Machining
N..
N350 G91 G59 Z-60 ou G90 G59 X0    Cancellation possible with G90 or G91
N..
```

Machining allowance by origin offset

Bore roughing, finishing. Machining allowance: X = 0.3, Z = 0.2



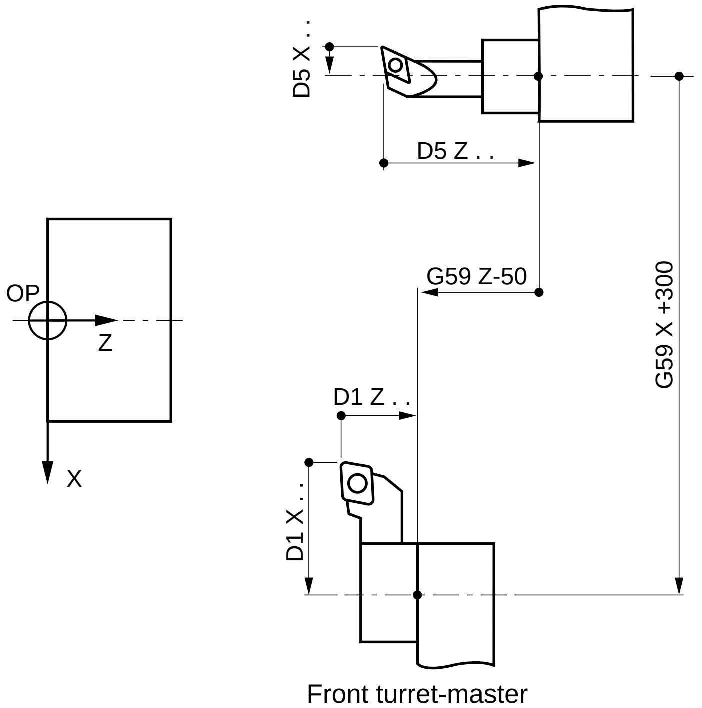
Machining paths



N.. ...	
N80 G00 G52 X.. Z.. G97 S500	
N90 T07 D07 M06 (ROUGHING BORING TOOL)	
N100 G95 F0.2	
N110 S900 M04 M40	
N120 G59 X-0.3 Z0.2	Origin offsets
N130 G00 G41 X30 Z62	Point a
N140 G96 S80	
N150 G01 X24 Z59	Point b
N160 Z25	Point c
N170 X16	Point d
N180 G00 Z62	Point e
N190 G59 X0 Z0	Offsets cancellede
N200 G40 G77 N80 N80	
N210 T09 D09 M06 (FINISHING BORING TOOL)	
N220 G77 N130 N180	Finishing
N..	

Use of origin offset on a lathe with two turrets

On lathes equipped with two turrets (front and rear), the tool dimensions of the secondary turret can be identified with respect to the reference of the master turret by programming an origin offset.



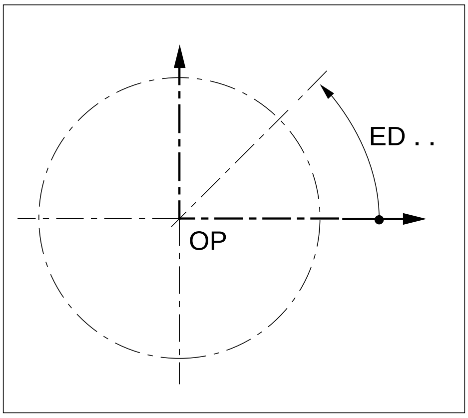

```
% 88
$0 MACHINING WITH FRONT MAIN TURRET
N10 G00 G52 X0 Z0
N20 T01 D01 M06 (EXTERNAL TOOL)
N..
N..
N340 G00 G52 X0 Z0
$0 MACHINING WITH REAR TURRET
N350 T05 D05 M06 (INTERNAL TOOL)
N360 G90 G59 X300 Z-50
N..
N500 G59 X0 Z0
N..
```

4.12.5 Angular Offset

ED..	Programmed angular offset.

Function ED associated with a value defines an angular rotation with respect to the programme origin.

The angular offset applies to the axes of the plane programmed in the blocks following the function.



4

Syntax

N.. [G90/G91] **ED..**

G90/G91	Programming in absolute or incremental dimensions.
ED..	Value of the angular offset in degrees and thousandths of a degree (format ED+034).

Property of the Function

Function ED.. is modal.

Cancellation

Angular offset ED.. is cancelled by:
- reprogramming function ED with a zero value (ED0) in absolute dimensions (G90),
- the end of programme function (M02),
- a reset.

Notes

Angular offset ED applies to:
- all the elementary cycles (G81 ...),
- radius offset (G41, G42),
- PGP (Profile Geometry Programming) unless ED.. is programmed between two blocks that are not completely defined,
- carried or independent secondary axes (U, W).

4.13 Spline Curve Interpolation

4.13.1 General

Spline curve interpolation is a mathematical curve fitting method. Spline curves are continuous curves connecting a series of points.

Spline curve interpolation insures continuity of the tangent and a constant acceleration in each of the points specified on the programmed paths.

Machining of a spline curve is programmed by:
- defining the points on the curve,
- specifying a curve execution order.

A spline curve can be deleted by programming.

4.13.2 Programming

G48	Spline curve definition.

A spline curve definition includes several instructions:
- the definition function,
- the curve number,
- the blocks defining the points on the curve.

G06	Spline curve execution command.

A spline curve execution command is given by a block containing the execution command followed by the number of the curve to be executed.

G49	Spline curve deletion.

The system allows memory space to be freed by deleting curves that have already been executed.

A curve is deleted by programming the delete function followed by the number of the curve to be deleted.

4.13.2.1 Spline Curve Interpolation

G48	Spline curve definition.

Syntax

N.. **G48** NC.. H../N.. N..

G48	Spline curve definition function.
NC..	Argument defining the curve number.
H..	Number of the subroutine in which are defined the points on the curve (optional).
N.. N..	Numbers of the first and last block defining the points on the curve.

Property of the Function

Function G48 is nonmodal.

Cancellation

Function G48 is cancelled at the end of the block.

Notes

On a turning machine whose X (or U) axis is programmed on diameter (see Parameter Manual: machine parameter P4), programming a spline curve results in the following requirements:

- the block containing the spline curve definition function (G48 ...) must be preceded by external parameter E11005=0 (forcing programming on radius, see Sec. 6.2)
- the points in the curve definition (N.. to N..) must be programmed on radius
- when the curve is completed (G06 NC..), programming on radius can be cancelled by E11005=1 (forcing programming on diameter).

Blocks defining the points of a curve

The first and last block defining a curve must include the origin and end tangents. If the tangents are not known, these blocks must be empty.

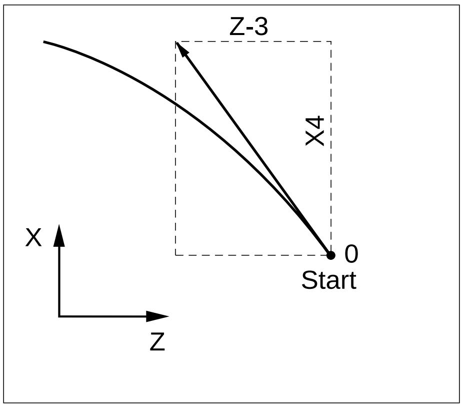
All the definition blocks other than the first and last (tangent to the start and end points) must include points on the curve (no empty lines). Otherwise, the curve plotted would be different from the one desired.

The first point definition block must include all the axes concerned by spline curve interpolation. If an axis is not programmed in this block, it is not included in spline curve interpolation even if it is programmed in the following blocks. In this case, it is interpolated linearly. It is necessary to programme the same position in this first block as in the block preceding function G06.

Blocks defining the tangents and points of a spline curve.

The projections along the X and Z axes of the tangent to the curve origin have X4, Z-3 as relative values.

```
N.. X4 Z-3      Tangent to the origin
X.. Z..
X.. Z..
X.. Z..         Points on the curve
X.. Z..
X.. Z..
N.. X.. Z..     Tangent to the end point
N..
```



The number of points defining a spline curve is limited to 255. If this value is exceeded, the system returns error message 196.

The points in a spline curve must be defined by ISO programming.

A spline curve must be defined by at least three points. If it is defined by less than three points, the system returns error message 604.

Function G48 must be programmed in state G40, i.e. without radius offset (G41 or G42). Otherwise, the system returns error message 140.

The following functions can be programmed on the curve definition blocks:
- miscellaneous M functions,
- technological F, S or other functions.

Special characters such as $, (), etc. should not be programmed in the curve definition blocks.

It is not recommended to programme modulo 360 degree axes with spline curves (problem of sign and no movement).

A closed curve is not processed automatically unless the tangents to the start and end points are identical.

Saving spline curve data

The curve coefficients are stored in tables of the programme stack. When the programme stack is full, the system returns error message 195. In this case, the stack size can be extended (see Chapter 7).

The following data are stored for each spline curve defined:
- 1 table with 5 entries,
- 3 tables with P x Q entries, where:
  P: number of axes concerned
  Q: number of points in the profile
- 1 table with P entries.

4.13.2.2 Spline Curve Execution Command

G06	Spline curve execution command.

Syntax

N.. **G06** NC..

G06	Function forcing execution of a spline curve.
NC..	Number of the curve to be executed.

Property of the Function

Function G06 is nonmodal.

Cancellation

Function G06 is cancelled at the end of the block.

Notes

The following functions cannot be programmed in the block containing function G06 NC..:
- F: feed rate,
- S: spindle speed,
- T: tool change.

The curve execution command G06 forces the polynomial interpolation function (see Supplementary Programming Manual).

Error Numbers and Messages for Spline Curve

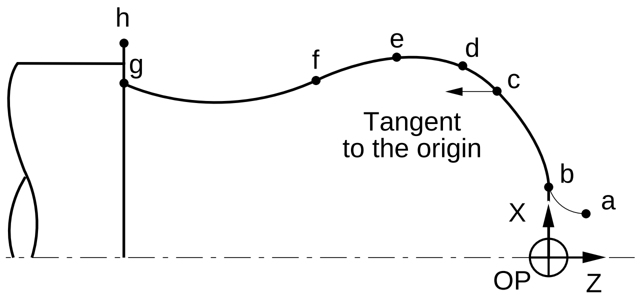
The errors are in error category 600 (See Appendix D).

4.13.2.3 Programming Examples

Programming of a spline curve defined in the ZX plane with a call to definition blocks in the main programme.

`%70`	
`N10 ...`	
`N..`	
`N..`	
`$0 FINISHING OF THE SPLINE PROFILE`	
`N180 G0 G52 X0 Z0 G97 S900 M05`	
`N190 G79 N220`	Jump to block N220
`N200 X0 Z-1`	Tangent to the origin of the curve
`G01 X35 Z-10 F0.15`	
`X37 Z-13`	
`X40 Z-25`	Points in the curve (c, d, e, f, g)
`X35 Z-50`	
`X40 Z-80`	
`N210`	No tangent to the end point
`N220 T01 D01 M06 (FINISHING TOOL)`	
`N230 S900 M03 M40`	
`E11005=0`	Forcing programming on radius
`N240 G48 NC1 N200 N210`	Spline curve definition
`N250 G0 G42 X10 Z5`	Point a
`N260 G96 S300`	
`N270 G95 F0.3`	
`N280 G02 X15 Z0 I15 K5`	Point b
`N290 G03 X35 Z-10 R30`	Point c
`N300 G06 NC1`	Curve execution order
`N310 G01 G40 X45`	Point h
`E11005=1`	Forcing programming on diameter
`N320 G77 N180 N180`	
`N330 M02`	

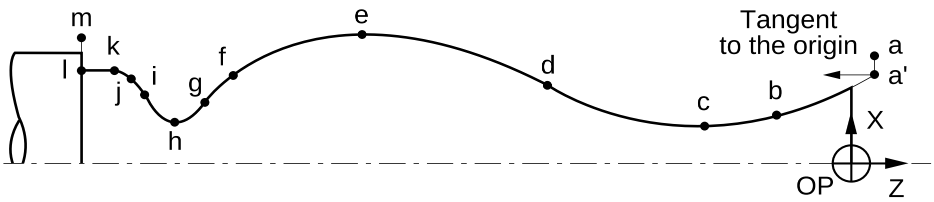
Representation of machining



Programming of a spline curve defined in the XZ plane with call to the definition blocks in a subroutine.

%150	
N10	
N..	
N200 G0 G52 X0 Z0 G97 S500 M05	
N210 T03 D03 M06 (FINISHING TOOL)	
N220 S2000 M40 M04	
E11005=0	Forcing of programming with reference to the radius
N230 G48 NC1 H151 N10 N130	Curve definition
E11005=1	Programming with reference to the diameter
N240 G0 X30 Z5	Point a
N250 G92 S3000	
N260 G96 S250	
E11005=0	Forcing of programming with reference to the radius
N270 G01 G42 X11 G95 F0.3	Point a'
N280 G06 NC1	Execution order
E11005=1	Programming with reference to the diameter
N290 G01 Z-105	Point l
N300 G40 X30	Point m
N310 G77 N200 N200	
N320 M02	
%151	
N10 X0 Z-1	Tangent to the origin
N20 X11 Z5 F0.1	Curve start
N30 X6 Z-10	
N40 X5 Z-20	
N50 X12 Z-40	
N60 X17 Z-60	
N70 X12 Z-80	Points on the curve (a to k)
N80 X8 Z-84	
N90 X5 Z-88	
N100 X9 Z-91	
N110 X11 Z-93	
N120 X12 Z-95	
N130 X0 Z-1	Tangent to the end point

Representation of machining



4.13.2.4 Freeing Memory by Deleting a Spline Curve

G49	Spline curve deletion.

This function is used to free the memory space occupied by curves already executed.

Syntax

N.. **G49** NC..

G49	Spline curve deletion.
NC..	Number of the curve to be deleted.

Property of the Function

Function G49 is nonmodal.

Cancellation

Function G49 is cancelled at the end of the block.

Notes

Function G49 must be programmed in state G40, i.e. without radius offset (G41 or G42). Otherwise the system returns error message 140.

Example

Definition and execution of a spline curve followed by deletion of the curve

```
%300
N10 ...
N.. G79 N200
N110 X.. Z..
N..                          Points in curve 1
N..
N150 X.. Z..
N..
N200 G48 NC1 N110 N150       Curve 1 definition
N300 G06 NC1                 Curve 1 execution
N..
N..
N450 G49 NC1                 Curve 1 deletion
N..
```

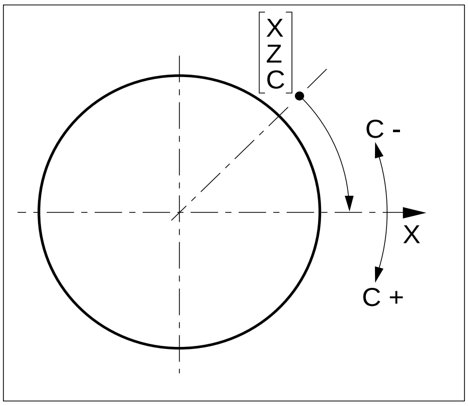
4.14 Coordinates Systems with C Axis

4.14.1 Programming in Polar Coordinates

G20	Programming in polar coordinates (X, Z, C).

This function is used to programme the linear axes X and Z and control a rotary axis C, modulo 360 degrees.

X/Z programming is the same as programming a system without a C axis.



N.. [G40] **G20** [G00/G01][X.. Z.. C..][F..]

G40	Radius offset cancel.
G20	Programming on axes X, Z and C (angular value C.. with reference to the origin).
G00/G01	Interpolations permitted.
X.. Z.. C..	End point.
F..	Feed rate.

Properties of the Function

Function G20 is modal and is the default function.

Cancellation

Function G20 is cancelled by functions G21 and G22.

Notes

Programming of function ER is excluded in G20.

Notes on declaration of the C axis

Control of the C axis requires the use of:
- a servo-drive
- a position encoder
- a special drive arrangement.

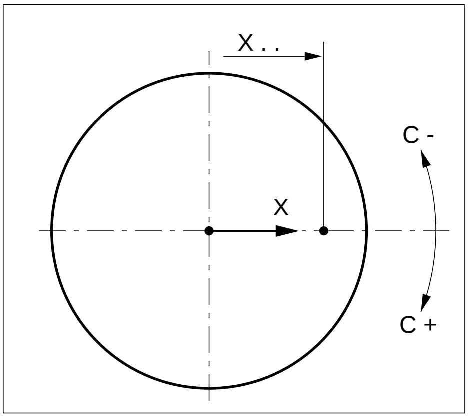
Example

`N.. ...`	
`N1050 G40 G21 (or G22) ...`	Machining in cartesian or cylindrical coordinates
`N..`	
`N1220 G40 G20 G01 X.. Z..`	Return to initial state
`N..`	

4.14.2 Definition of the Start X for Interpolation on the C Axis

G98	Definition of the start X for interpolation on the C axis.

When the system is in G20 mode, this function is used to compute the feed rate on the C axis in mm/min.



Syntax

N.. **G98** X..

G98	Definition of the start X for interpolation on the C axis.
X..	Start X.

Property of the Function

Function G98 is nonmodal.

Cancellation

Function G98 is cancelled at the end of the block.

Notes

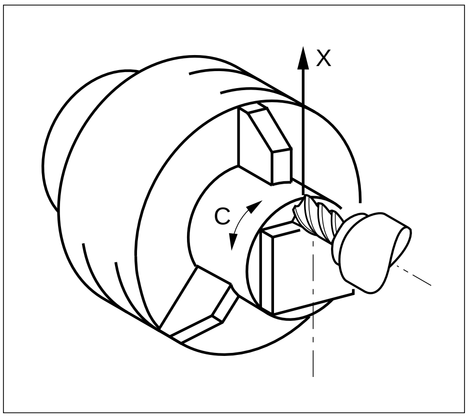
Programming G98 X.. does not cause any movement.

When function G98 X.. is not programmed, the last value declared with the X axis is used for the computation.

When the C and X axes are interpolated simultaneously, the system performs its feedrate calculation from the mean radius.

4.14.3 Programming in Cartesian Coordinates

G21	Programming in cartesian coordinates (X, Y, Z).



The system performs cartesian/polar coordinate conversion (X-Y converted to X-C).

Interpolation of the X and C axes allows milling in the plane perpendicular to the spindle axis.

The tool is driven by an auxiliary spindle.

Syntax

N.. [G40] **G21** [G00/G01] [G41/G42] [X.. Y.. Z..] [G94 F..]

G40	Radius offset cancel.
G21	Programming in cartesian coordinates.
G00/G01	Interpolations.
G41/G42	Radius offset to the left or right of the profile.
X.. Y.. Z..	End point.
G94 F..	Feed rate in mm/min.

Property of the Function

Function G21 is modal.

Cancellation

Function G21 is cancelled by function G20.

Notes

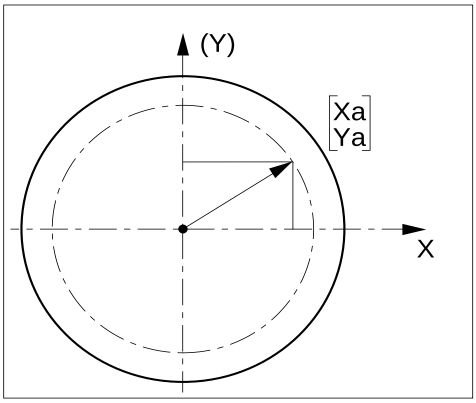
Programming

Function G21 cannot be programmed during a canned cycle. The use of a cycle causes generation of error message 76.

G21 is used for programming the X, Y and Z addresses by linear interpolation (the Z address is incompatible with circular interpolation G02 and G03).

Programming a point in G21:

```
N.. ...
N.. G21 Xa Ya
N..
```



During transition from G20 to G21:
- the system must be in G40 mode (radius offset cancel),
- if the last block in G20 includes an X demand, it must have a positive value,
- the system must not be programmed in PGP (Profile Geometry Programming).

During cancellation of G21 by G20, the system must be in G40 mode (radius offset cancel).

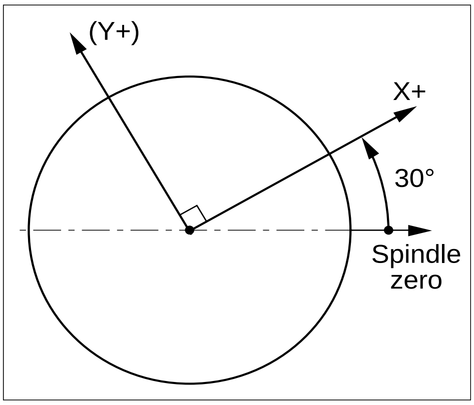
Failure to comply with the transition and cancellation conditions causes generation of error message 75.

Offsets

Programming of offsets is allowed:
- G59 X.. Y.. or G59 C..: Programmed origin offset,
- ED..: Programmed angular offset,
- G51..: Mirroring.

An angular offset programmed on the C axis causes a rotation of the X and Y coordinates, e.g.:

```
N.. ...
N.. G59 C30
N..
```



4

Tool corrections

No change of tool corrections are authorised in G21. The corrections must be programmed and modified in G20.

Feed rate limiting

When the tool is moving toward the centre of the part, the constant feed rate along the path is achieved by increasing the angular feed rate on the C axis and decreasing the linear feed rate on the X axis.

The system may therefore have to decrease the tangential machining feed rate so as not to exceed the maximum rotation speed on the C axis.

The system automatically limits the minimum rotation speed on the C axis. When the minimum speed is reached, the system returns error message 34.

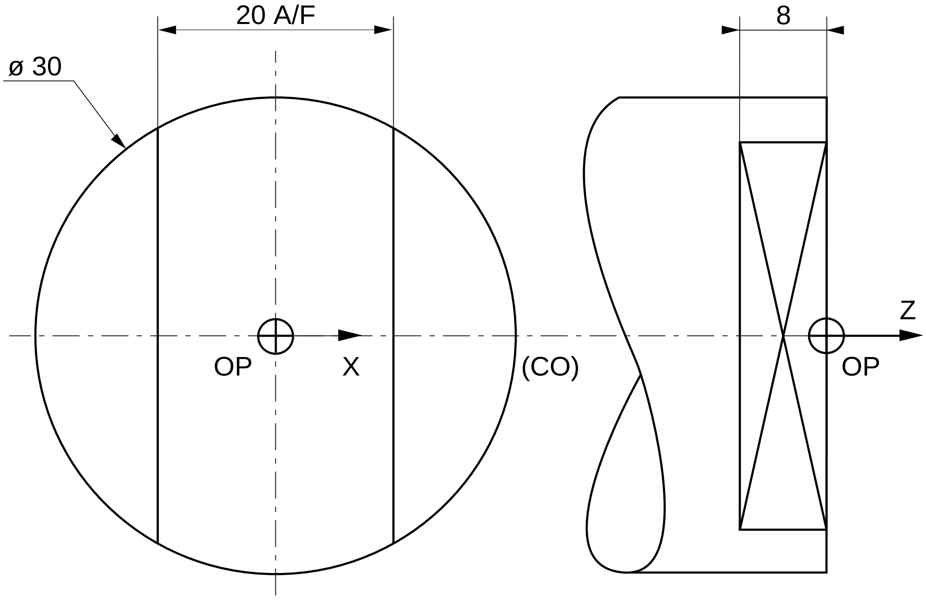
The minimum radius R is calculated by the following equation:

$$R = \frac{344000}{VCmax}$$

Where:
- R is the distance between the cutter centre and the C axis centre in µm,
- VCmax is the maximum speed on the C axis expressed in degrees/minute (see machine parameter P30).

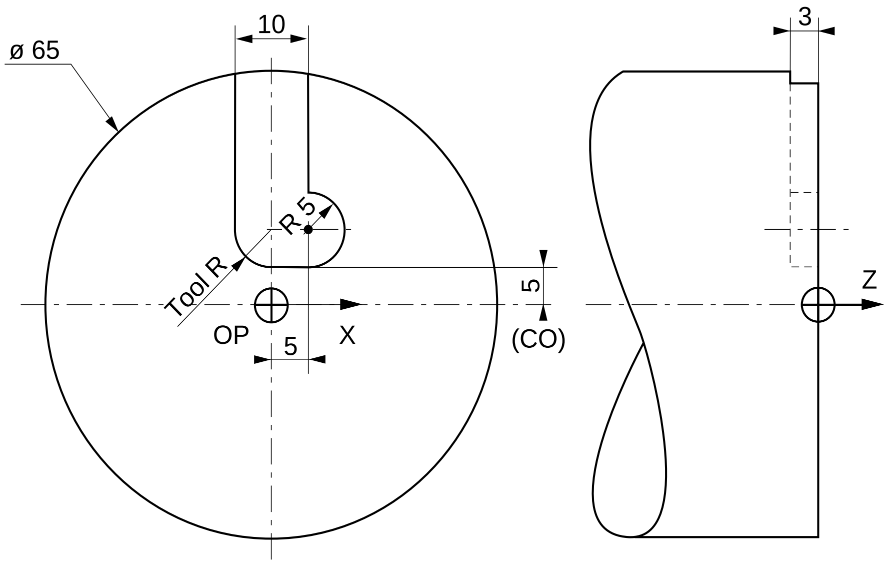
Example

Execution of two flats




```
N..
N200 G00 G40 G52 X.. Z.. M05
N210 T12 D12 M06 (CUTTER D=10)
N220 G97 S1000 M.. $0 AUXILIARY SPINDLE ROTATION
N230 X50 Z2                              Tool approach along XZ
N240 G94 F..
N250 G21 G01 G42 X10 Y11.18
N260 G00 Z-8                             Pass setting on Z
N270 G01 Y-11.18                         Execution of flat 1
N280 G00 X-10
N290 G01 Y11.18                          Execution of flat 2
N300 G00 Z3
N310 G40 G20 X100
N320 ...
```

Execution of a shape




```
N..
N200 G00 G40 G52 X.. Z.. M05
N210 T8 D8 M06 (CUTTER D=8)
N220 G97 S1200 M.. $0 AUXILIARY SPINDLE ROTATION
N230 X70 Z2                              Tool approach along XZ
N240 G94 F..
N250 G92 R1
N260 G21 G00 G41 X-5 Y35
N270 G00 Z-3                             Pass setting on Z
N280 G01 Y5
N290 X5
N300 G03 X5 Y15 R5
N310 G01 Y35
N320 G00 G40 G20 X200
N330 G52 X0 Z0
N340 ...
```

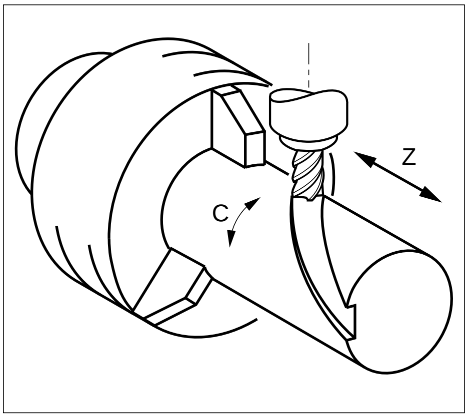
4.14.4 Programming in Cylindrical Coordinates

G22	Programming in cylindrical coordinates (X, Y, Z).

The system performs cylindrical/polar coordinate conversion (Y-Z converted to Z-C).

Interpolation of the C axis allows milling on the developed cylinder with radius X.

The tool is driven by an auxiliary spindle.



Syntax

N.. [G40] **G22** [G00/G01] [G41/G42] [X.. Y.. Z..] [G94 F..]

G40	Radius offset cancel.
G22	Programming in cylindrical/polar coordinates.
G00/G01	Interpolations.
G41/G42	Radius offset to the left or right of the profile.
X.. Y.. Z..	End point.
G94 F..	Feed rate in mm/min.

Property of the Function

Function G22 is modal.

Cancellation

Function G22 is cancelled by function G20.

Notes

Programming

G22 is used for programming addresses:
- X, Y and Z by interpolation G01
- Y, Z, J and K by interpolations G02 and G03.

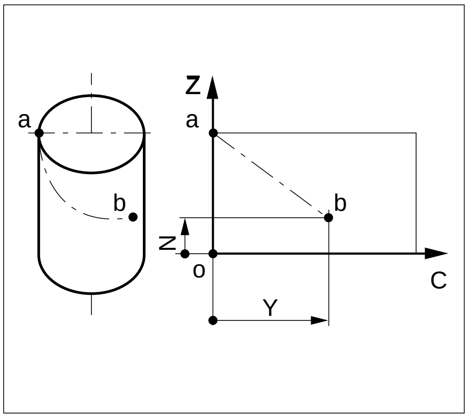
When the C axis rotates in the positive direction, the Y axis also increases positively.

Position Y0 coincides with position C0 of the C axis.

With G22, canned cycles are not authorised. The use of a canned cycle causes generation of error message 76.

Programming of a point in G22:

```
N.. ...
N.. Xa Za
N..
N.. G22 Yb Zb
N..
```



During transition from G20 to G22:
- the system must be in G40 mode (radius offset cancel),
- if the last block in G20 included a movement along X, it must be programmed with a positive value,
- the system must not be programmed in PGP (Profile Geometry Programming).

During cancellation of G22 by G20:
- the system must be in G40 mode (radius offset cancel),
- the system must not be programmed in PGP.

Failure to comply with the transition and cancellation conditions causes generation of error message 75.

Offsets

No offsets are allowed in G22 mode.

In G20 mode, origin offsets G59 are possible:
- Before the call to function G22, the start point on the C axis can be modified, e.g. N.. G59 C..
- During the programme, the position of the Z axis can be modified, but this requires a return to state G20, e.g. N.. G59 Z..

Tool corrections

No changes in the tool corrections are authorised in G22. The corrections must therefore be programmed and modified in G20.

Feed rate limiting

The feed rate is limited by the maximum feed rate authorised on the C axis and the current value of the positioning radius.

For machining small diameters, the system may have to decrease the feed rate so as not to exceed the maximum rotation speed on the C axis.

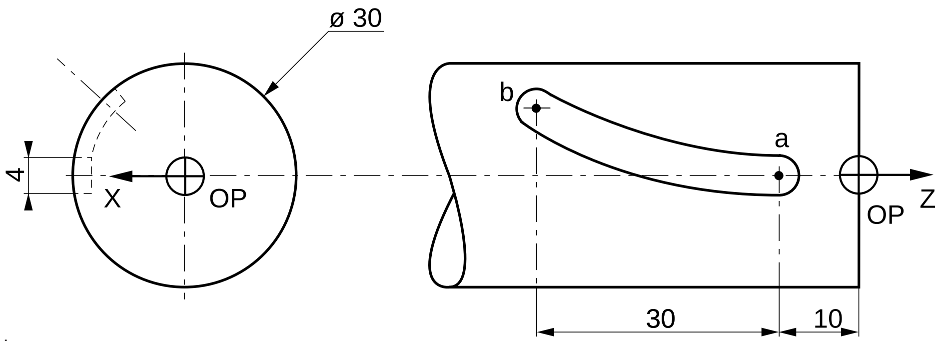
The system automatically limits the minimum feed rate when it is incompatible on the programmed path. When the minimum speed is reached, the system returns error message 34.

Initialisation of the cartesian position on the Y axis

Positioning to the first point programmed in G22 is carried out by computing the function start value on the Y axis from the value on the X axis and on the C axis, interpreted between -180 and +180, giving:

$Y = \theta \, . \, X$ (θ = angle on the C axis between + and - π).

Example



N..	
N200 G00 G40 G52 X.. Z.. M05	
N210 T08 D08 M06 (CUTTER D=4)	
N220 G97 S2500 M.. $0 AUXILIARY SPINDLE ROTATION	
N230 X32 Z-10	Point a, tool approach along XZ
N240 C0	C axis in zero point
N240 G94 F50	
N250 G01 X28	Cut application in X
N260 G22 Y-10 Z-40 F..	Point b
N270 G20 X32	
N280...	

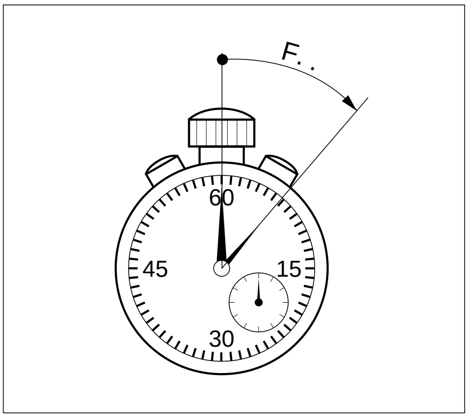
REMARK *Homing is required on the C axis.*

4.15 Other Functions

4.15.1 Dwell

G04	Programmable dwell.

Continuation of the programme is interrupted for the time programmed with argument F.



Syntax

N.. **G04** F..

G04	Programmable dwell.
F..	Dwell in seconds (0.01 to 99.99 seconds, format F022). Mandatory argument F must be programmed immediately after the function.

Property of the Function

Function G04 is nonmodal.

Cancellation

Function G04 is cancelled at the end of the block.

REMARK *Function G04 is cancelled before the end of the block when it is programmed with function G10.*

Notes

Function G04 F.. does not cancel the feed rates programmed with F in the previous block(s).

Example:

N.. ...	
N50 G01 Xa Za F0.2	Feed rate 0.2 mm/rev
N60 G04 F1.5	Dwell 1.5 seconds
N70 Xb Zb	After 1.5 seconds Machining resumes at 0.2 mm/rev
N..	

If G04 is programmed at the start of a block containing a move, the dwell will occur at the end of the block.

Example:

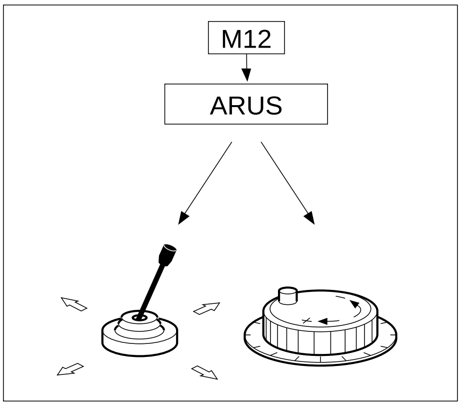
N.. ...	
N.. G01 G04 F5 X100 Z100	Dwell after movement
N..	

4.15.2 Programmed Feed Stop

M12	Programmed feed stop.

This function requires operator action after the feed stop (ARUS).

The axis jogs or handwheel are made available.



Syntax

N.. **M12** [$0...]

M12	Programmed feed stop.
$0...	Message for the operator (see Sec. 4.19).

Properties of the Function

Function M12 is a decoded modal «post» function.

Cancellation

The function is cancelled by pressing the CYCLE key on the machine panel.

Notes

M12 is ignored by the system unless bit 1 of word 1 of machine parameter P7 is set.

When function M12 enables the axis jogs or handwheel:
- the operator can only make movements in continuous jog mode (J.ILL),
- during axis movements, the system remains in the current execution mode: automatic (AUTO) or single step (SINGLE).

When the operator cancels the function after any axis movements, the programme is resumed from the new position (axis recall not required).

Function M12 is ignored in test (TEST) and sequence number search (SEARCH) modes.

Example

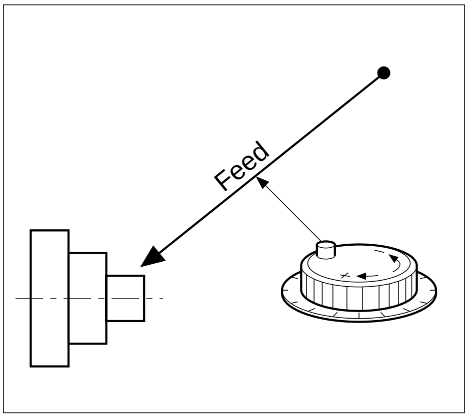
```
N.. ...
N300 G00 Z80 M12 $0 MOVEMENT ON X, THEN CYCLE START
N310 ..
N..
```

4.15.3 Feed Enhancement

G12	Feed enhancement by handwheel.

When the machine is equipped with handwheels, this functions is used to increase the speed of movement on the linear or circular paths programmed in the block.

Overspeed is applied to the first handwheel.



Syntax

N.. [G01/G02/G03] **G12** X.. Z.. [F..] [$0 ...]

G01/G02/G03	Linear or circular interpolation.
G12	Enabling of rapid feed by handwheel.
X.. Z..	End point.
F..	Feed rate.
$0 ...	Message for the operator (see Sec. 4.19).

Property of the Function

Function G12 is nonmodal.

Cancellation

Function G12 is cancelled at the end of the block.

Notes

Continuation to the next block is carried out when the programmed position is reached.

The overspeed coefficient applied by function G12 is defined in parameter P13 (see Parameter Manual).

Example

```
N.. ...
N60 G01 G12 X.. Z.. F0.2 $0 ACTUATE THE HANDWHEEL
N..
```

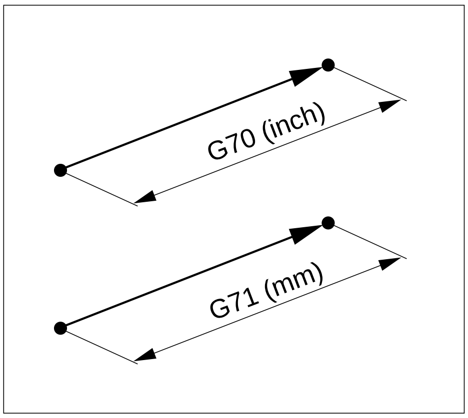
4.15.4 Programming in Inches or Metric Data

G70	Inch data input

This function is used to programme data in inches.

G71	Metric data input.

This function is used to programme data in metric units.



Syntax

N.. **G70/G71**

G70	Inch data input.
G71	Metric data input.

Properties of the Functions

Functions G70 and G71 are modal.

Function G71 or G70 is the default function.

Cancellation

Functions G70 and G71 cancel one another.

Notes

Programming can be changed from inches to metric units and vice versa in machine parameter P7 (see Parameter Manual).

It should be noted that the choice of the displayed unit (inches or metric units) is made by the automatic control function (see the C_UNIT variable in the Automatic Control Function Programming Manual).

Formats specific to inch data input (G70)

Addresses	Formats
X, Y, Z, U, V, W, I, J, K	+044
P, Q, R, ER etc...	+044

Inch data input (G70) for programme variable «L» and external parameters «E» (see Chapter 6)

The programming must be suited to the operations carried out on the dimensions and external parameters «E» expressed in their own units.

Examples

Programming of a dimension using programme variable L

```
N.. ...
N.. G70
N.. L1 = 10
N.. G01 XL1                          L1 is equal to 10 inches
```

Programming of a dimension using an external parameter E

```
N.. ...
N.. G70
N.. E80000 = 100000
N.. G01 XE80000                      E80000 is equal to 10 inches
                                     (format +044)
```

Modification of a tool dimension by the programme

External parameter E50001 represents the length of tool 1.

Its value is always held in mm.

```
N.. ...
N.. G70
N.. L10 = E50001/25400               L10 = length of tool 1 converted to
                                     inches
N.. L2 = 100 + L10                   L2 = length of tool 1 + 100 inches
N.. E50001 = L2 * 25400              Modification of the length of tool 1 with
                                     conversion in mm
```

4.15.5 Axis Clamping and Unclamping

M10	Clamp.

This function is used to clamp those axes with clamps that are not required to move.

M11	Unclamp.

This functions cancels axis clamping.

Syntax

N.. [G00/G01/ G02/ G03] **M10/M11** X.. Z.. C..

G00/G01/G02/G03	Linear or circular interpolation.
M10	Clamp.
M11	Unclamp.
X.. Z.. C..	End point.

Properties of the Functions

Function M10 is a decoded modal «post» function.

Function M11 is a decoded modal «pre» function.

Cancellation

Functions M10 and M11 cancel one another.

Notes

The axes recognised as able to be clamped by function M10 are declared in machine parameter P8 (see Parameter Manual).

When function M10 is programmed, the system generates a timeout followed by a handshake (CRM) before executing the movements in the next block.

4.15.6 Coolant

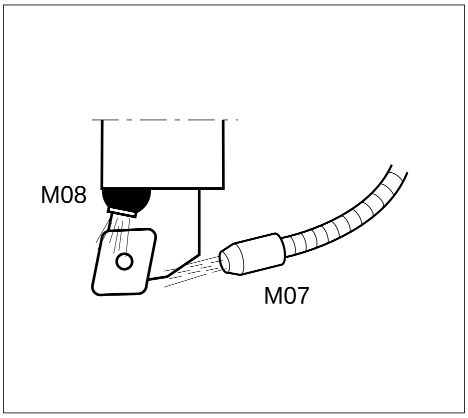
M08	Coolant 1 on.

M07	Coolant 2 on.

These functions turn on the coolant pumps.

M09	Coolant off.

This function stops the coolant pumps.



Syntax

N.. **M08/M07**

M08	Coolant 1 on.
M07	Coolant 2 on.

N.. **M09**

M09	Coolants 1 and 2 off.

Properties of the Functions

Functions M07 and M08 are decoded modal «pre» functions.

Function M09 is a decoded modal «post» function. It is the default function.

Cancellation

Functions M08 and M07 are cancelled by functions M09 or M02.

Example

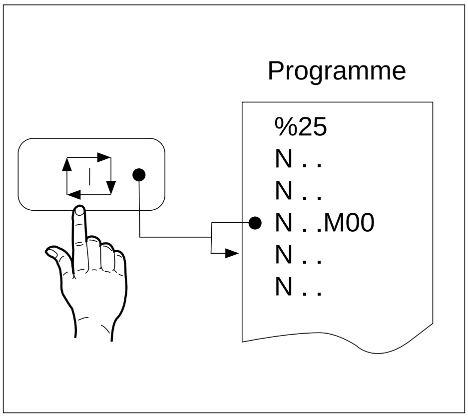
`N.. ...`	
`N40 G00 X.. Z.. M08`	Coolant 1 on
`N50 G01 Z.. M07`	Coolant 2 on
`N..`	
`N230 G00 G52 Z-100 M05 M09`	All coolant off
`N..`	

4.15.7 Programme Stop

M00	Programme stop.

This function stops execution of the current programme.

After intervention or a check, the operator restarts the programme.



Syntax

N.. [G40] **M00** [$0 ...]

G40	Radius offset cancel.
M00	Programme stop.
$0 ...	Message for the operator (see Sec. 4.19).

Properties of the Function

Function M00 is a decoded nonmodal «post» function.

Cancellation

This function is cancelled by pressing the CYCLE key.

Notes

When function M00 is read in a block:
- continuation to the next block is interrupted and spindle rotation is stopped,
- the contents of the field in the status window are modified and the indicator CYCLE is replaced by M00.

After action or check by the operator, pressing the CYCLE key restarts the programme. The indicator M00 is replaced by CYCLE.

M00 must be programmed with the system in G40 mode (radius offset cancel).

Spindle rotation must be reprogrammed after a stop programmed by M00.

Transfer of function M00 to the PLC

Function M00 is transmitted to the PLC at the end of execution of the block in which it is programmed, but before waiting for axis group synchronisation if required (G78..) in a system with multiple axis groups (in this case, stopping of the spindle is taken into account when synchronisation is completed).

Example

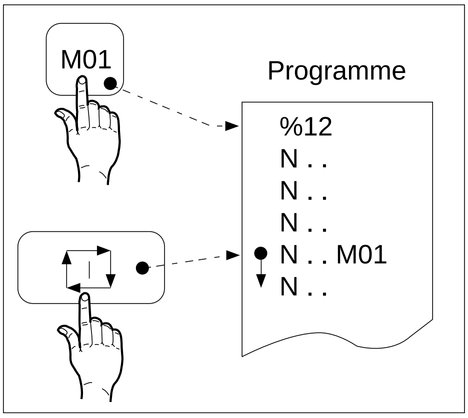
`N.. ...`	
`N.. G01 G41 X.. Z.. F0.3 M08`	
`N..`	
`N190 G00 G40 Z200 M09`	Tool retraction before intervention
`N200 M00 $0 REMOVE CHIPS BEFORE FINISHING`	Stop and message
`N210 G00 G41 X.. Z.. M03`	Continuation of the programme
`N..`	

4.15.8 Optional Stop

M01	Optional stop.

When confirmed by the operator M01 causes a stop in programme execution.

After intervention or check, the cycle is restarted by the operator.



Syntax

N.. [G40] **M01** [$0 ...]

G40	Radius offset cancel.
M01	Optional stop.
$0 ...	Message for the operator (see Sec. 4.19).

Properties of the Function

Function M01 is a decoded nonmodal «post» function.

Cancellation

The function is cancelled by pressing the CYCLE key.

Notes

When function M01 is read in a block (M01 confirmed):

- continuation to the next block is interrupted and spindle rotation is stopped,
- the contents of the field in the status window are modified and the indicator CYCLE is replaced by M00.

After action or check by the operator, pressing the CYCLE key restarts the programme. The indicator M01 is replaced by CYCLE.

Function M01 must be programmed with the system in G40 mode (radius offset cancel).

Spindle rotation must be reprogrammed after a stop programmed by M01.

Transfer of function M01 to the PLC

Function M01 is transmitted to the PLC at the end of execution of the block in which it is programmed, but before waiting for axis group synchronisation if required (G78..) in a system with multiple axis groups (in this case, stopping of the spindle is taken into account when synchronisation is completed).

Example

N.. ...	
N.. G01 G42 X.. Z.. F0.2 M08	
N..	
N290 G00 G40 Z200 M09	Tool retraction before intervention
N300 M01 $0 CHECK DIMENSION 50	Stop if M01 is confirmed
EVERY 5 PARTS	and message
N310 G00 G42 X.. Z.. M03	Continuation of the programme
N..	

4.15.9 Cancellation of MDI and EDIT modes

M999	Programmed cancellation of the edit (EDIT) and manual data input (MDI) modes and subroutine calls by the automatic control function.

When this function is programmed, the operator cannot call the EDIT and MDI modes and the PLC cannot call subroutines.

M998	Reactivation of the edit (EDIT) and manual data input (MDI) modes and subroutine calls by the automatic control function.

Syntax

N.. **M998/M999**

M999	Programmed cancellation of the edit (EDIT) and manual data input (MDI) modes and subroutine calls by the PLC function.
M998	Reactivation of the edit (EDIT) and manual data input (MDI) modes and subroutine calls by the automatic control function.

Properties of the Functions

Functions M998 and M999 are decoded modal «pre» functions.

Function M998 is the default function.

Cancellation

Functions M998 and M999 cancel one another (M998 and M999 can also be cancelled by M997 and M02).

M999 is cancelled by a reset.

Notes

Programming M999 allows:
- manual movements (JOG or MANUAL),
- use of enhanced feed by handwheel (G12),
- use of stop on switch (G10).

Programming M999 inhibits:
- switch to sequence number search (SEARCH) mode during execution of a sequence of blocks «masked» by programming M999 (ignored by the NC),
- action by the PLC or the operator during execution of a sequence of blocks.

Programming M999 allows the use of variables L100 to L199, L900 to L999 and symbolic variables [...] in the same way as variables L0 to L19 (see Chapters 6 and 7).

Writing to these variables or transfering current values into the part programme are not normally carried out until the end of the previous block (M999 allows anticipated execution of these operations).

Example

Use of variables L

In programme %1, block N100 is not prepared and executed until block N90 is completed.

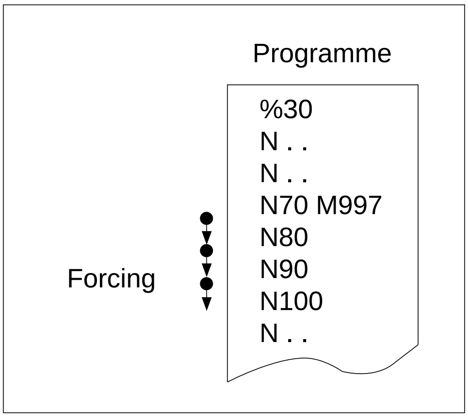
In programme %2, block N100 is prepared before execution of N80 and there is no stop at the end of block N90.

```
%1                          %2
N.. ...                     N..
N.. ...                     N80 M999
N90 X.. Z..                 N90 X.. Z..
N100 L110 = 1 + L110        N100 L110 = 1 + L110
N110 XL110                  N110 XL110
N120 L120 = L110 + L10      N120 L120 = L110 + L10
N130 ZL120                  N130 ZL120
N..                         N140 M998
N..                         N..
```

4.15.10 Forced Block Continuation

M997 Forced block continuation.

Blocks following this function are executed automatically in sequence until a cancelling function is read.



Syntax

N.. **M997**

M997 Forced block sequencing.

Properties of the Function

Function M997 is a decoded modal «pre» function.

Cancellation

Modal function M997 is cancelled by functions M998, M999 and M02.

Notes

Use of function M997 in single step (SINGLE) mode

If the operator starts programme execution in single mode, reading M997 during the programme causes sequencing of the following blocks as though the system were in continuous mode.

Use of function M997 with a subroutine call by M function requested in manual data input (MDI) mode

Programming this function causes execution of the subroutine in continuous mode.

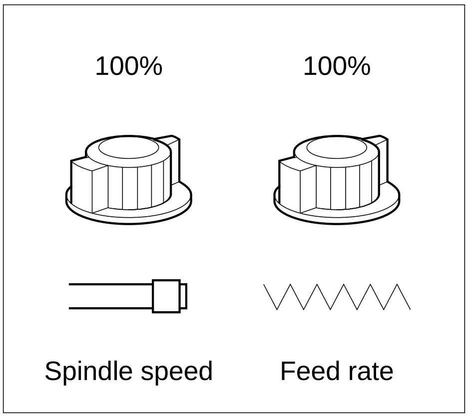
In either case, M997 can only be cancelled by a cancellation function.

4.15.11 Potentiometer Inhibit

M49	Spindle speed and feed rate potentiometer inhibit.

During execution of the programme, the operator cannot override the spindle speed and feed rate. The potentiometers are forced to 100 percent.

M48	Spindle speed and feed rate potentiometer enable.



Syntax

N.. **M49/M48**

M49	Spindle speed and feed rate potentiometer inhibit.
M48	Spindle speed and feed rate potentiometer enable.

Properties of the Functions

Function M48 is a decoded modal «pre» function. It is the default function.

Function M49 is a decoded modal «pre» function.

Cancellation

Functions M48 and M49 cancel one another.

Notes

Programming M49 causes:
- forcing of a feed rate of 100 percent (with M48, feed rate override possible from 0 to 120% of the value programmed with F),
- forcing of a spindle speed of 100 percent (with M48, spindle speed override possible from 50 to 100% of the value programmed with S).

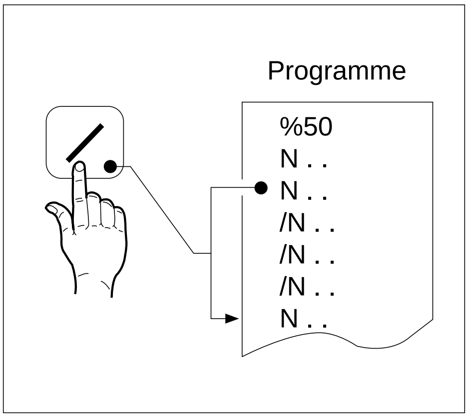
Display on the information (INFO) page is not affected by programming M49. The real percentage corresponding to the potentiometer setting is displayed.

To stop programme execution while the potentiometers are inhibited, press the feed stop (ARUS) key then the reset (RAZ) key.

4.15.12 Block Skip

/	Block skip.

A block preceded by a slash «/» is ignored when block skip is enabled by the operator.



Syntax

/ N.. (Contents unimportant)

/	Block skip (/).
N..	Block number.

Cancellation

By inhibiting block skip.

Notes

Block skip «/» is active when confirmed by the operator (the «/» indicator is displayed in the status window).

It is only possible to enable block skip «/» when the system has been reset or after M00 or M01.

Examples

Programming of «/»

If block skip is enabled, blocks N300, N310, N320 are ignored and the programme goes from block N290 to block N330.

```
N.. ...
N290 ...
/N300 G00 X.. Z..
/N310 Z..
/N320 G01 Y.. Z.. F0.3
N330
N..
```


Programming of «/» with M01

If block skip is inhibited, the blocks preceded by «/» are read by the system and M01 enabled is active.

```
N.. ...
N.. D11
/N150 G00 G41 X.. Z..
/N160 Z..
/N170 G01 X.. F0.15
/N180 G00 G40 Z150
/N190 M01 $0 CHECK DIMENSION 20 AND CORRECT D11 IF NECESSARY
N200
N..
```

4.15.13 Programmed Acceleration Reduction

EG..	Programmed acceleration reduction.

This function followed by a value sets the maximum acceleration tolerated on the programmed movements. This allows the stresses due to movement of heavy loads to be limited.

Syntax

N.. **EG**..

EG..	Programmed acceleration reduction. A positive integer giving the percentage between 1 and 100(%) of the value set by machine parameter P32 (see Parameter Manual).

Property of the Function

Function EG.. is modal.

Function EG is forced to 100 percent at power on.

Cancellation

Function EG is cancelled by:
- programming a new value (EG..),
- the end of programme function (M02),
- a reset.

Notes

On the information (INFO) page:
- EG1 to EG99 is displayable,
- EG100 is not displayable..

During programmed interpolated movements, the reduced acceleration is always taken into account by the interpolators except in the case of a feed stop (ARUS) or activation of the feed safety device.

Acceleration reduction is ignored in the following modes or functions:
- axis jog (JOG),
- homing (HOME),
- action after a programme stop (M12).

Example

```
N.. ...
N40 EG50 G00 X..                    Acceleration reduced to 50% on X
N..
```

4.15.14 Scaling Factor

G74	Scaling factor enable.

This function allows execution of scaled parts from a single programmed part or form. The scaling ratio can be entered from the keyboard or programmed.

G73	Scaling factor cancel.

Syntax

N.. [G40] **G74/G73**

G74	Scaling factor enable. The scaling ratio declared can be between 1 and 9999 over 1000 (0.001 to 9.999) and it must be an integer.
G73	Scaling factor cancel.

Properties of the Functions

Functions G73 and G74 are modal.

Function G73 is the default function.

Cancellation

Modal functions G73 and G74 cancel one another.

Function G74 is cancelled by the end of programme function (M02).

Notes

Scaling is about the programme origin (OP).

The scaling ratio can be entered from the alphanumeric keyboard or programmed by external parameter E69000 (see Sec. 6.2).

Functions G73 and G74 must be programmed:
- with the system in G40 mode (radius offset cancel),
- in a block that does not contain any circular interpolation,
- outside a sequence of incompletely defined PGP (Profile geometry Programming) blocks.

Scaling affects:
- the values programmed for the primary and secondary axes (X, Y, Z, U, V, W),
- the programme origin offset (G59).

Scaling does not affect:
- the values programmed for the rotary axes (A, B, C),
- the part origin DAT1,
- the shift between the part origin and the program origin (DAT2),
- tool dimensions (X, Z, R),
- programming with reference to the measurement origin (G52),
- the clearance dimension for machining cycles G81 to G89.

Entry of the scaling factor from the alphanumeric keyboard

Refer to the operator’s manual.

Examples

Programming the scaling factor by external parameter E69000

External parameter E69000 is a read/write parameter. The value programmed must be an integer.

`N.. ...`	
`N40 E69000 = 250`	Ratio 250/1000, i.e. reduction to 0.25
`N50 G74 G00 X.. Z..`	
`N..`	
`N200 G73`	Scaling factor cancel
`N..`	

The scaling factor can be tested by a conditional branch in the programme.

`N.. ...`	
`N.. G79 E69000 = 300 N210`	If the ratio is equal to 300, jump to block N210
`N..`	
`N..`	
`N210`	
`N..`	

4.15.15 Mirror Function

G51	Mirror function.

This function allows symmetrical machining of programme sections defining a quarter or half of the part.

The mirror is enabled or cancelled according to the axis and algebraic sign arguments programmed with the function.

Syntax

N.. **G51** X- (Y-) Z-

G51	Mirror function.
X- (Y-) Z-	The minus sign (-) enables the mirror function on the X, Y, Z axes.

Properties of the Function

Function G51 is nonmodal. The axis arguments X, (Y), Z associated with the function are modal.

Cancellation

Mirroring on the programmed axes is cancelled by:
- function G51 followed by one or more arguments X+, (Y+) or Z+ cancelling the former state G51,
- the end of programme function (M02),
- a reset.

Notes

When G51 is programmed:
- it must be programmed with at least one argument (axis and sign),
- it must be programmed alone with its arguments in the block,
- several axes can be enabled or inhibited in the same block,
- the axes to which the mirror function applies are displayable on the information page (INFO),
- if the mirror function is applied to a carried axis, it is also automatically applied to the carrier axis.

The mirror function affects:
- the sign of the designated axes. The sign change is carried out with respect to the programme origin defined by DAT1 and DAT2,
- the origin offsets programmed (G59),
- tool radius offset (G41, G42),
- the direction of rotation with circular interpolation (G02, G03).

The mirror function does not apply to:
- the part origin DAT1,
- the shift between the part origin and the programme origin (DAT2),
- the tool dimensions (X, Z, R),
- programming with respect to the measurement origin (G52).

When the mirror function is enabled on the axis corresponding to the tool orientation axis, a new orientation must be programmed (G16 ...). Example:

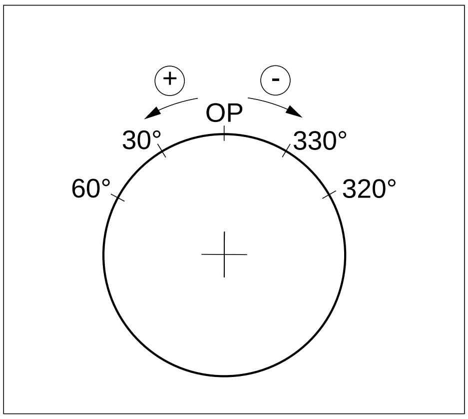
```
N.. ...
N30 G16 P-                    Tool axis orientation along X-
N..
N80 G51 X-                    Mirror on X
N90 G16 P+                    Tool axis orientation along X+
N..
```

Use of mirror function on A and C rotary axes

The function reverses the direction of rotation of the rotary axes and takes the complement to 360 degrees.

If C30 is programmed: the axis moves to 330 degrees in the negative direction.

If C300 is programmed: the axis moves to 60 degrees in the positive direction.



4

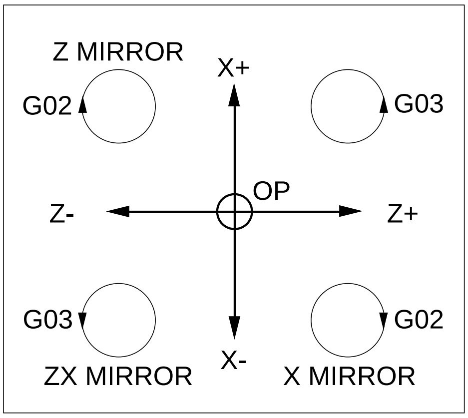
REMARK *When the mirror function is applied to a rotary axis, it should be checked by a test before machining that the axis rotates in the required direction.*

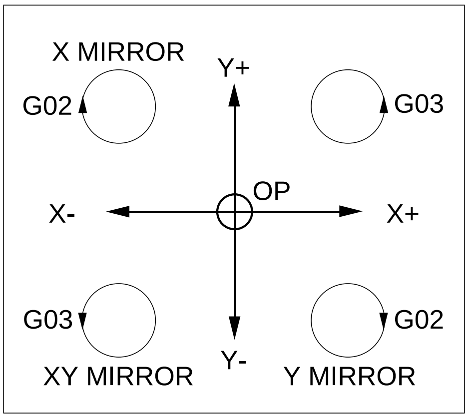
Influence of the mirror function on circular interpolation

The function reverses the clockwise (G02) and counterclockwise (G03) rotations.

Programming in polar coordinates (G20: XZ)

Programming in cartesian coordinates (G21: XYZ)





4.15.16 Processing of Blocks and Programmed G and M Functions

G999	Suspension of execution and forcing of block concatenation.

The blocks programmed after G999 are concatenated. The movements are no longer made on the axes and functions M, S and T are no longer processed.

G998	Enabling of execution of the blocks and part of the functions processed in state G999.

The movements and functions processed in state G999 are enabled and executed with the exception of certain functions which are only stored (decoded M «post» functions, timeouts and synchronisation functions for multiple axis groups).

G997	Enabling and execution of all the functions stored in state G999.

All the functions programmed (without exception) are enabled and executed, including those processed in state G999.

Syntax

N.. **G999** / **G998** / **G997**

G999	Suspends execution and forces block concatenation.
G998	Enables execution of the blocks and part of the functions processed in state G999.
G997	Enables and executes all the functions stored in state G999.

Properties of the Functions

Functions G999, G998 and G997 are modal.

Cancellation

Functions G999, G998 and G997 cancel one another.

Notes

Notes on Function G999

After execution has been suspended and the blocks have been concatenated by G999, analysis of the following blocks continues and the values encountered in these blocks of the part programme are stored in special symbolic variables.

These variables can be used to perform computations or read programmed values, but the axes are not moved and the M, T and S functions are not processed.

A subroutine call by function Gxxx (see Supplementary Programming Manual) implicitly sets function G999. This function then has to be cancelled by including functions G998 and/or G997 in the subroutine.

During the return to the part programme, state G999 is systematically reset as long as the subroutine call function (Gxxx) remains present and active (no G80).

Notes on function G998

Function G998 enables and executes:
- movements on the axes,
- auxiliary functions such as spindle speed (S), tool number (T), decoded «pre» M functions.

Function G998 stores the following functions but does not enable them:
- decoded «post» M functions (e.g. M05, M09, etc.),
- dwells (G04 F..),
- axis multigroup synchronisation functions (G78 ...).

Example

Sequencing of functions G999, G998 and G997

%55	
N.. ...	
N.. G999 S2500 M03	
N.. G00 X100 G04 F5	Blocks concatenated by G999
N.. G01 Z10	
N.. ...	
N.. M05	
N.. G998	
N.. ...	Enabling of S2500 M03 («pre» M function) and movements on the axes to X100 and Z10
N.. ...	
N.. X200 T02 M04	Movement to X200 and enabling of T02 M04
N.. ...	
N.. G997	Enabling of G04 F4 (dwell) and M05 («post» M function)

4.16 «Inclined Axis» or «Inclined Wheel» State on a Grinder

General

On a grinding machine, the «inclined axis» or «inclined wheel» state can be declared by programming the corresponding external parameter E (see Sec. 6.2):
- parameter E69001 for the «inclined axis» state,
- parameter E 69002 for the «inclined wheel» state.

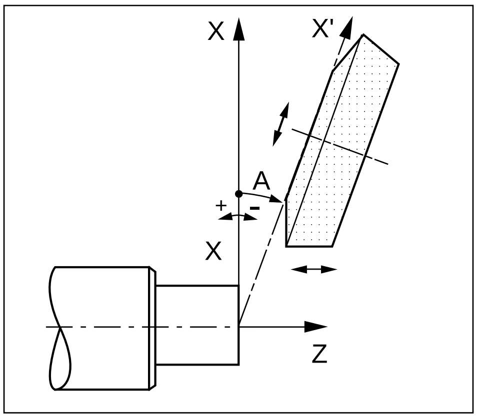
In both cases, the angle of inclination declared:
- can be positive or negative,
- must be between -80 and +80 degrees (or error message 94 is returned),
- is expressed in thousandths of a degree.

It should be noted that the «inclined axis» and «inclined wheel» states cannot be declared together in an axis group (otherwise error message 95 is returned) and that the angle of inclination is saved after a reset.

4.16.1 «Inclined Axis» State

Angle Declaration in the «Inclined Axis» State

The inclination angle can be declared:
- by programming external parameter E690001. Example: E69001 = -20000 (angle = -20 degrees), or,
- by DNC1000 (request to write segment 135. See Automatic Control Function Programming Manual).



The axis inclination is the value of angle A between the penetration axis along X and the normal to the workpiece-holder spindle.

In the «inclined axis» state, movements programmed on the axis are parallel to the axis inclination angle.

REMARK *A declaration programmed in DNC1000 is rejected if linear or circular interpolation is activated.*

Check of Travels

The travels declared in the machine parameters are relative to the physical axes. With linear or circular interpolation (system in state G20) it is checked that the path does not go beyond the soft limits on the X and Z axes.

Axis Movement

When an axis inclination is entered, the following elements can be defined in cartesian coordinates:

- linear and circular movements (the system performs the conversion of these movements for the servo-system)
- machining parameters such as origin offset, machine travels and tool corrections.

The pivot point of the inclined axis must be referenced to the measurement origin.

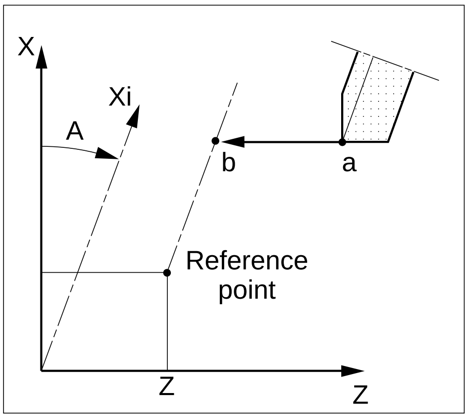
4.16.1.1 Initial Tool Positioning Before Machining on an Inclined Axis

G07	Initial tool positioning before machining on an inclined axis.

This functions defines the location on the Z axis where it intersects with the line parallel to the inclined axis going through the reference point whose position is programmed with the function.

In the figure:

- point a: path start,
- point b: intersection with the line parallel to the inclined axis.



Syntax

N.. [G90] [G00/G01] **G07** X.. Z..

G90	Programming in absolute dimensions (always).
G00/G01	Linear interpolation.
G07	Initial tool positioning before machining on the inclined axis.
X.. Z..	Reference point.

Property of the Function

Function G07 is nonmodal.

Cancellation

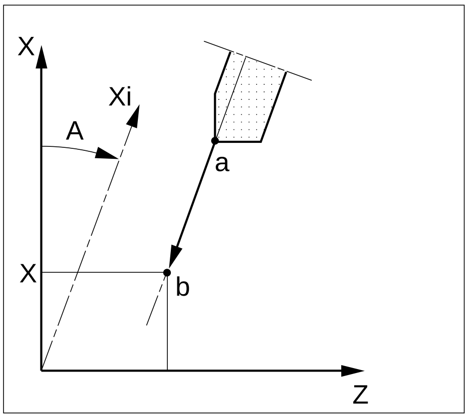
Function G07 is cancelled at the end of the block.

4.16.1.2 Movement Along the Inclined Axis

G05	Movement along the inclined axis.

This function defines movement on the X axis parallel to the inclined axis up to the cartesian end point programmed.

In the figure:
- point a: path start point,
- point b: end point.



Syntax

N.. [G90/G91] [G00/G01] **G05** X..

G90/G91	Programming in absolute or incremental dimensions.
G00/G01	Linear interpolation.
G05	Programming of a movement along the inclined axis.
X..	End point.

Property of the Function

Function G05 is nonmodal.

Cancellation

Function G05 is cancelled at the end of the block.

Notes

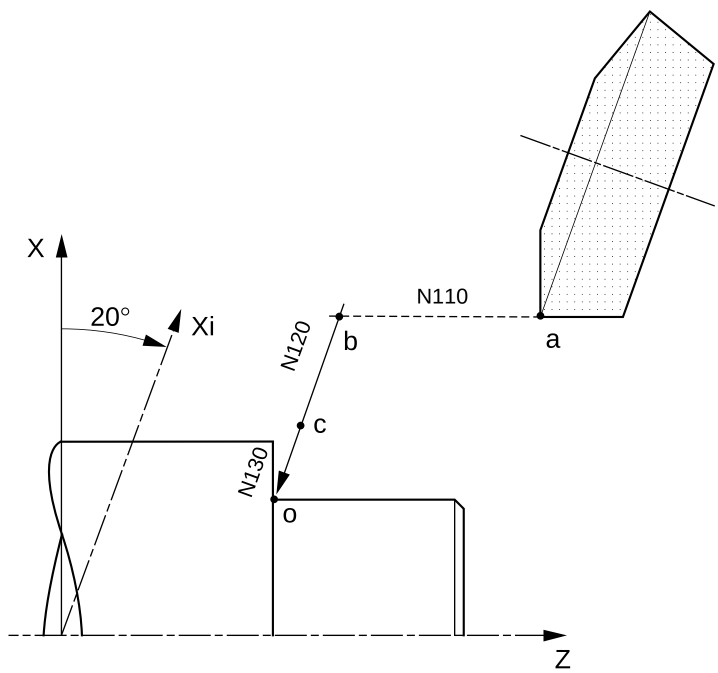
Function G07 must always be programmed before function G05.

For functions G05 and G07, failure to comply with the following programming rules cause generation of error message 7:
- the functions must be programmed in plane G20,
- the interpolation must be programmed by G00 or G01,
- function G07 must be followed by X and Z,
- function G05 must be followed by X.

Example

Positioning in Z and X in absolute dimensions (G90) with axis inclined by -20 degrees.

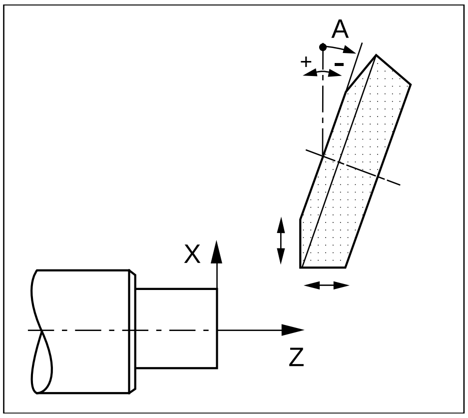
N..	
N.. E69001=-200000	Axis inclination by parameter E
N.. G97 S300 M03 M40	
N.. G95 F0.1	
N..	
N.. G00 X60 Z70	Point a, position before the branch to G07
N110 G07 X10 Z25	Definition of the reference point (o) and positioning in point b on Z
N120 G05 X30	Point c, positioning on X
N130 G01 G05 X20 F0.2	Point o, positioning on X
N..	



4.16.2 «Inclined Wheel» State

Angle Declaration in the «Inclined Wheel» State

The angle is declared by programming external parameter E69002. Example: E69002 = -15000 (angle = -15 degrees).



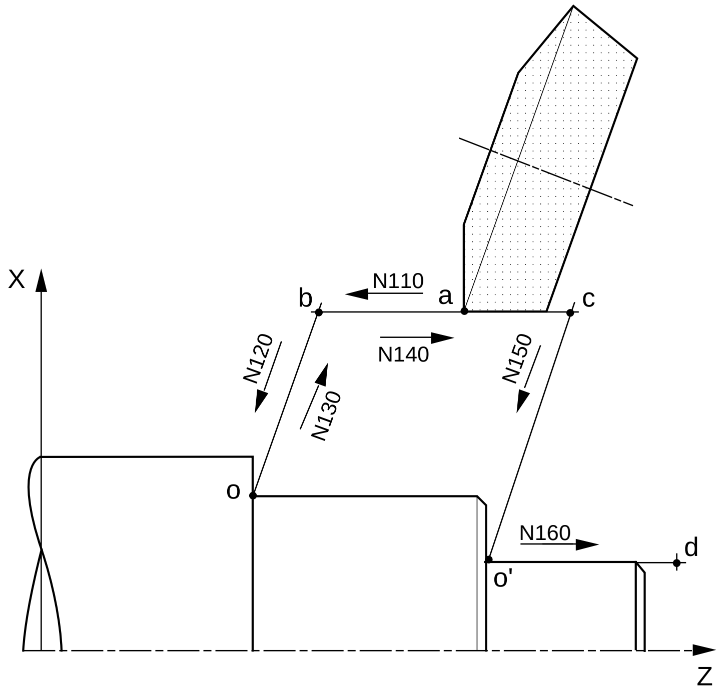
In the «inclined wheel» state, wheel movements programmed on X and Z are orthogonal.

Functions G05 and G07 (see Secs. 4.16.1.1 and 4.16.1.2) are used in the same way as in the «inclined axis» state (see example below).

Example

Positioning in Z and X in absolute dimensions (G90) with wheel inclined by -15 degrees.

`N..`	
`N.. E69002=-150000`	Wheel inclination by parameter E
`N.. G97 S300 M03 M40`	
`N.. G95 F0.1`	
`N..`	
`N100 G00 X50 Z50`	Point a, position before the branch to G07
`N110 G07 X20 Z25`	Definition of the reference point (o) and positioning in point b on Z
`N120 G01 G05 X20 F0.2`	Point o, approach along X
`N130 G05 X50`	Point b, return along X
`N140 G07 X5 Z30`	Definition of the reference point (o') and positioning in point c on Z
`N150 G05 X10`	Point o', approach along X
`N160 Z50`	Point d, movement along Z
`N..`	



X
b
N110
a
c
N120
N140
N150
N130
o
N160
d
o'
Z

4.17 Special Programming for Multiple Axis Groups

4.17.1 Programme Declaration

A multiple axis group machining programme is a combination of the programmes corresponding to each axis group.

Each programme name has a common radical followed by the index designating the axis group, e.g.:

%61.1 Programme for group 1

%61.2 Programme for group 2

Format of the programme number: %05.1 (indices .1 to .8).

The programmes corresponding to all the groups must be loaded in the NC. Otherwise, the machining programme cannot be started (unless one or more groups are inhibited).

4.17.2 Programming Notes

The use of the programme start address «%» must not appear in any comments located in a programme.

Programmable axes

The NC can control 32 axes that can be distributed in a maximum of 8 axis groups.

Each group can include up to 9 axes declared in machine parameter P9 by the machine manufacturer (see parameter manual).

Example: 5-axis lathe, 2 axis groups

Group 1: axes X Z C

Group 2: axes U W

Axes belonging to different groups can have the same address names. In the above example, axes U and W of group 2 could have been called X and Z.

It should be noted that for correct operation of a lathe with multiple axis groups, the secondary U and W axes must be declared as carried by the X and Z axes (see parameter P64 in the Parameter Manual).

Each axis group is enabled by the interface. It is possible to inhibit one or more axis groups by a switch on the machine panel (see manufacturer’s documentation).

T Function

A tool number T.. of the programme with index .1 can have the same number as a tool in other programmes without necessarily corresponding to the same tool. Example:

```
%10.1                    %10.2
N.. ...                  N.. ...
N.. T05 M06 (DRILL)      N.. T05 M06 (TURNING TOOL)
N..                      N..
```

This is not so for programmes with indices .2 to .8, whese the tool numbers used must be different.

For more details on processing of the T function, refer to the Automatic Control Function Programming Manual.

Tool corrections

The tool correction table is shared by all the programmes.

On a secondary axis (U and/or W), the tool dimensions can only be taken into account if the axis is declared carried (see machine parameter P64).

Programme variables

Variables L can be used in each programme.

L0-L19, L100-L199 and L900-L959: these variables can be used differently by each axis group (no interaction between the programmes).

External Parameters

Parameters E are common to all the programmes (except for parameters E50000, E51000, E6x000 and E7x000 which must each be used for a single axis group).

4.17.3 Subroutine branches for Multi-Axis Groups

Subroutines which are not part of the machining programme must always include the index of the calling group. For instance: %xxx.i (i = axis group number).

4.17.3.1 Branch to Automatic Homing Subroutine

Homing can be carried out automatically on each of the axes of the group by running subroutine %9990.i (i = axis group number).

Requirements for running subroutine %9990.i

Action on the cycle start button in homing mode runs subroutine %9990.i for the group if no other programme is being executed.

The automatic homing programme is run simultaneously on all the existing CNC axis groups.

Subroutine %9990.i can also be called as subroutine in another programme while preserving the property of being able to move the machine axes without affecting their homing. This possibility can be used for homing a PLC axis group. If such a programme can be used in homing mode or as a subroutine, it can also end as follows:

```
IF [.IRH(1)] = 9990.i THEM M02
ENDI
```

If it is not a subroutine, then set M02

For information on the above programming, refer to the Supplementary Programming Manual.

4.17.3.2 Subroutine Call by a Reset

On a reset, the NC groups can execute subroutines with numbers %11000.i (i = group number).

Aside from the fact that the subroutine numbers include the index (i) of each NC group, the branch is the same as the subroutine branch by reset for a single axis group (see Sec. 4.11.9).

4.17.3.3 Subroutine Call by the Automatic Control Function

The automatic control function can call and execute a subroutine with number %9999.i (i = axis group number).

The call is the same as a subroutine call with a single axis group (see Sec. 4.11.4) except for the CNC axis group number (i).

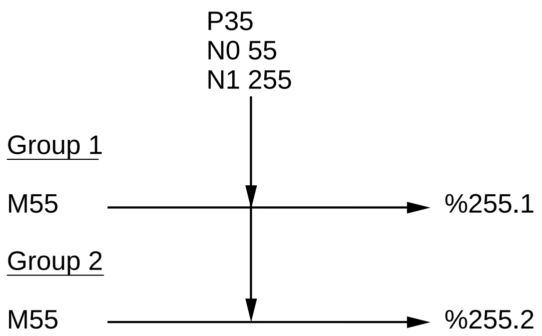
4.17.3.4 Subroutine Call by M Function

An M.. function can call and execute a subroutine with number %xxx.i (i = axis group number).

The call is the same as a subroutine call with a single axis group (see Sec. 4.11.2) except for the CNC axis group number (i).

Example

With two axis groups (1 and 2), call of two subroutines %255.1 and %255.2 with different contents by function M55.



4.17.4 Spindle Programming

The spindle characteristics are defined in parameter P6 (see parameter manual).

A spindle can be:
- controlled by an axis group,
- independent.

Spindle controlled by an axis group

At any one time, a spindle can be controlled by only one axis group and an axis group can control only one spindle.

In an axis group, selection of another spindle by one of the functions M62 to M65 is only accepted if the spindle is not currently controlled by any other axis group (otherwise, the system returns error message 38).

Independent spindle

An independent spindle is a spindle which is not controlled by any axis group or a spindle that has been released by the group which controlled it.

The release of a spindle by a group is carried out by function M61 (see Sec. 4.17.5).

Only an independent spindle can be selected by a group for control by that group (see Sec. 4.17.5).

Spindles not assigned to a group become independent.

Spindle Control

For features corresponding to spindle control (see Sec. 4.3.5).

Spindle Measurement

For features corresponding to spindle measurement (see Sec. 4.3.6).

Reminder

When a lathe is equipped with a milling spindle with indexing performed by the NC:
- the milling spindle is indexed by function EC..,
- the turning spindle is indexed by positioning on the C axis.

4.17.5 Current Spindle Release by an Axis Group

M61	Current spindle release by an axis group.

This function is used to release the current spindle in the group for control by another group.

Syntax

N.. **M61**

M61 Current spindle release by an axis group.

Properties of the Function

Function M61 is a decoded modal «post» function.

Cancellation

Function M61 is cancelled by functions M62 to M65.

Notes

After release of the spindle by M61, the spindle can be selected for control by one of the functions M62 to M65 in the programme of another axis group.

If the spindle to be controlled was not released, the system generates error message 38.

Example

Group 1 programme	Group 2 programme	
%40.1	%40.2	
N.. ...	N.. ...	
N..	N.. G97 S500 M04 M65 M40	
N..	N..	
N..	N.. M61	Release of spindle 2
N..	N..	
N.. G97 S1000 M03 M41 M65	N..	
N..	N..	

4.17.6 Axis Group Synchronisation

G78	Axis group synchronisation.

This function is used to manage the synchronisation of steps in the execution of each programme.

Syntax

N.. **G78** Q.. / Pj.i

G78	Axis group synchronisation.
Q..	Declaration of a marker in the current axis group.
Pj.i	Wait for a marker in another axis group.

Argument P contains two digits separated by a decimal point:
- j is the marker number,
- i is the index of the group in which the marker is to be tested.

Property of the Function

Function G78 is nonmodal.

Cancellation

Function G78 is cancelled at the end of the block.

Argument Q associated with the function is initialised with zero (Q0) at power on.

Reset of Q.. markers can be programmed by including instruction G78 Q0 in a programme.

Notes

Function G78 can be followed by several arguments, but at least one is mandatory.

The declaration of a marker and the conditions for continuation of the programme can be programmed in the same block, e.g.:

```
N.. G78 Q3 P5.2 P6.3
```

The choice of the axis group and the related transition conditions can be enabled selectively by the PLC (see automatic control function programming manual).

The markers declared in a programme must be numbered in increasing order but not necessarily in increments of one, e.g.:

```
N.. ...
N90 G78 Q1 ...
N..
N200 G78 Q6 ...
N..
```

REMARK *A marker is crossed when the tested marker is reached.*

Example

Programme synchronisation with markers

```
%50.1                            %50.2                      %50.3
N..                              N..                        N..
N..                              N..                        N.. G78 P2.1  P1.2 Wait Q2
N..                              N..                        N..                of %50.1
N50 G78 Q1                       N70 G78 P1.1               N..                of Q1
N..                              N..                        N..                of %50.2
N..                              N..                        N..
N.. G78 Q2 P1.2  Wait Q1         N..                        N..
N..                of %50.2      N..                        N..
N..                              N..G78 Q1                  N..G78 Q6
N..                              N..                        N..
N.. G78 P6.3     Wait Q6         N..
N..                of %50.3      N..G78 P5.1  Wait Q5
N..                              N..            of %50.1
N..                              N..
N.. G78 Q5
N..
```

Sequence N70 of programme %50.2 is crossed if programme %50.1 reaches or passes sequence N50.

Programme %50.3 does not start until programme %50.1 reaches marker Q2 and programme %50.2 reaches or passes marker Q1.

If axis group 3 is inhibited by the PLC programme, the steps related to programme %50.3 are ignored in programmes %50.1 and %50.2 (wait G78 P6.3 of %50.1 is ignored) (see Automatic Control Function Programming Manual).

Programming a Meeting Point

A meeting point can be programmed with function G78 or caused by M function programming.

Programming with function G78

When a meeting point has been programmed, the programmes do not continue until the other programmes have reached their respective markers.

When the meeting point has been reached by all the groups, the markers can be reset by programming G78 Q0.

Before the markers are reset by G78 Q0, the previous block must mandatorily be a meeting point, i.e. a complete synchronisation on all the programmes.

Example:

Programming a meeting point on two groups (with complete synchronisation).

```
%63.1                           %63.2
N10                             N10
N..                             N..
N..                             N..
N..                             N.. G78 Q8 P10.1
N.. G78 Q10 P8.2                G78 Q0
G78 Q0                          N..
N..                             N..
N..                             G79 N10
N..
G79 N10
```

The end of a programme (M02) cancels the conditions concerning this programme and is equivalent to inhibiting the programme.

When all the programmes are waiting, the system generates error message 33 (synchronisation impossible). This error is detectable in test mode (TEST) or sequence number search mode (SEARCH). It is necessary to carry out a reset and to correct the programme.

Example

Programming a meeting point with G78 on 3 groups

```
%80.1                      %80.2                      %80.3
N..                        N..                        N..
N..                        N..                        N..
N..                        N..G78 Q4 P6.1 P5.3
N..                        N..                        N..
N..                                                   N..
N..                                                   N..
N..G78 Q6 P4.2 P5.3                                   N..G78 Q5 P6.1 P4.2
N..                                                   N..
```

The three programmes are resumed simultaneously when markers Q6, Q4 and Q5 are declared.

Programming with M functions

The following functions are meeting points:
- Programmed machining stop (M12),
- Programmed stop (M00),
- Optional programmed stop (M01).

When M00 and M01 (enabled) are programmed in only one of the programmes, the movements of the group are stopped and the system waits for the end of the other programmes (M02) before activating M00. After a cycle restart, only this programme is resumed.

It should also be noted that functions M00 and M01 (even not enabled) reset the markers.

When M01 is programmed in all the programmes but is not enabled, each of the M01 sequences is a meeting point. The cycle continues automatically as soon as all the programmes are on their respective M01.

4.18 Special Programming of PLC Axes

4.18.1 Programme Declaration and Storage

The programmes executed by the PLC groups are identified by the radical 9998 followed by an index defining the group to which the axis belongs, e.g.:

%9998.2 Programme executed by PLC group 2.

%9998.3 Programme executed by PLC group 3.

Programme number format: %04.1 (indexes .1 to .8).

The programmes and subroutines must be stored in a memory area with a number strictly above zero. The search for the programme is first made in area 1 and ends where applicable in area 3.

When the main programme is stored after creation or replaced after editing, the group must be reset so that its presence is taken into account.

<u>Structure of a PLC axis group programme</u>

When functions can be processed by a PLC axis group, the PLC must specify the function called, which can be done by external parameter E40000, for instance (see Sec. 6.2).

At the start of the programme:
- an M function is used as report (CRM)
- external parameter E40000 contains the sequence number of each function requested (Na, Nb, etc.).

Example:

Programme %9998.2 (PLC axis group 2) below contains several functions. The second programme block defines a jump to the sequence number contained in parameter E40000.

%9998.2	
N1 M..	Wait for start by CRM
G79 NE40000	Jump to the sequence defined
Na	Process the first function
..	
..	
..	
G79 N1	Jump to the sequence
Nb	Process the second function
..	
..	
..	
G79 N1	Jump to the sequence
..	

4.18.2 Programming of the PLC Axes

The paths on the axes of a PLC group are programmed in ISO code.

The PLC axes are programmed in the same way as the NC axes but there are restrictions concerning the use of certain functions.

Restrictions

The following functions are unavailable:
- optional stop (M01),
- programmed feed stop (M12).

4

4.18.2.1 Emergency Retraction on a PLC Axis Group

Emergency retraction (function G75...) on a PLC axis group allows a particular sequence to be executed in automatic mode at the request of the PLC when the same group is in a cycle.

It should be noted that:
- No particular condition is required as to the current mode on the other axis groups (NC and PLC)
- The emergency retraction request on a PLC axis group only affects that axis group.

Activation of emergency retraction

Function G78 N.. defines a branch to a sequence N.. when emergency retraction is activated. Activation is cancelled by declaring a new address G75 N.. or G75 N00.

The emergency retraction programme can be activated at the request of the automatic control function by setting bit «C_DGURGn» for axis groups n.

It should be noted that:
- If the axis group is executing a cycle (automatic mode AUTO or single step mode SINGLE), activation halts the programme being executed. The halt is followed by a branch to sequence N.. programmed after G75 followed by execution of the retraction programme until one of functions M00 or M02 is encountered
- Signal E_DEGURGn for the group remains set throughout the duration of the emergency retraction programme. This signal is reset by encountering function M00 or M02 which ends the emergency retraction programme.

REMARK *If emergency retraction is activated but no G75 N.. was programmed, this activation stops the axes of the group and resets the group (RAZ).*

4.18.3 Editing the Programmes

Programmes can be edited after transferring them to area 0 when the CNC is in edit (EDIT) mode.

The programme executed is the programme with the same number located in the area with a number above zero.

If an edited programme or subroutine is stored in an area whose number is below that of its original area, the edited version is executed the next time this programme or a part of it is called.

4.18.4 Exchanging Axes between Groups

The NC and PLC axis groups can exchange axes by programming. The axes are exchanged using external parameters E7x005 (x = axis number) (see Sec. 6.2).

Purely PLC axes can only be exchanged between PLC groups. An attempt to assign a purely PLC axis to an NC axis group results in error message 92.

After a reset, the group (PLC or NC) releases the axes that were not assigned to it by machine parameter and assigns them to the priority group providing the priority group is:
- in the end of programme state (M02), or
- with no programme being executed for a PLC group.

Similarly, during a reset concerning it, a group can recover the axes released by other groups that are assigned to it by machine parameter:
- after a reset,
- by programming external parameter E7x005 (see Sec. 6.2).

It should be noted that if a group is not in state M02 when its axes are released, it cannot recover them until the next reset concerning it, i.e.:
- an NC reset for an NC group,
- a reset on the group for a PLC group.

REMARK *When the cycle starts, no checks are made on assignment of the axes to the group. All the axes assigned to the group after the last reset were assigned by machine parameter P9, but certain axes assigned to it after the first initialisation may not be reassigned if they have already been assigned to another group during a cycle (it is therefore up to the programmer where required to ensure the configuration of the axes in the group in order to prevent start of the cycle).*

4.18.5 Exchanging Spindles Between Groups

When the NC is initialised, if no spindle has been assigned to a particular group, the spindles are assigned to the groups that have the same numbers, e.g.:
Spindle 1 to group 1,
Spindle 2 to group 2, etc.

A PLC group A can be assigned a spindle if the spindle was first released by its owner group B (NC or PLC). When group B, the initial owner, is reset, the spindle is reassigned to group B (even if group A is executing a cycle).

4.19 Message Transmission

The $ character followed by one or two digits sends a message from a part programme to a recipient.

Message Recipients

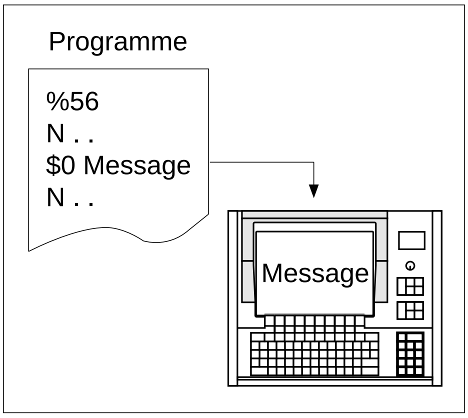
The number immediately after the $ character designates the message recipient:
- $0: System display
- $1: PLC function
- $2 or $3 or $4: Distant server
- $5 or $6: Peripheral,
- $9: PC.

The presence of a 1 after recipients $1 to $4 defines a blocking message.

4.19.1 Message Transmission to the Display

$0	Message transmission to the display.

$0 sends the message to the system display (information message on the part programme being executed).



Syntax

$0 [+] MESSAGE

$0	Message sent to the display.
+	The «+» sign extends the previous message.
MESSAGE	Message containing a maximum of 39 (alphanumeric) characters.

Cancellation
- $0 (without a message)
- End of programme (M02)
- Reset.

Notes

The message sent by $0 can be read on the display on the following screen pages:
- Summary of data on the current block (accessed by the «INFO» key)
- Current position coordinates (accessed by the «AXIS» key).

It should be noted that:
- If a message is too long, only the first 39 characters are displayed
- Only one message can be sent at a time
- Sending a new message overrides the previous one
- The 0 after $ is optional.

Extensions to the use of the $ character

The $ character can be used in parametric programming to:
- Display a message and wait for the answer by the operator (see Sec. 6.5)
- Display a message with a parametric value (see Sec. 6.6).

Examples

The message beginning in block N90 and including an extension is displayed in the following format:

```
«PROGRAMME HALT.  REMOVE CHIPS»
%30
$0 GROOVE ROUGHING FINISHING          Message
N10 ...
N..
N190 G00 Z300 $0 PROGRAMME HALT.      Message
$+ REMOVE CHIPS                       Message extension
N200 M00
N.. $0                                Message cancellation
N..
```

Use of $0 as flashing message

```
N..
N240 $0 * * * PREPARE PART INSPECTION * * *
N250 G04 F0.6
N260 $0 * * * <<<<<<<<<====>>>>>>>>>> * * *
N270 G04 F0.3
N280 G77 N240 N270 S4
N290 M00 $0 * * * OPEN DOOR * * *
N300 $0
N..
```

4.19.2 Transmission to Automatic Control Function or Remote Server or Peripheral or PC

$1 to $6 and $9	Message transmission to the automatic control function or a remote server or a peripheral or a PC.

$1 sends the message to the PLC function.

$2, $3 and $4 send the message to a remote server, i.e.:
- $2: UNI-TELWAY slave
- $3: MAPWAY
- $4: UNI-TELWAY master.

$5 and $6 send the message to a peripheral.

$9 sends the message to a PC.

Syntax

$1 to $6 $9 [1] [=] MESSAGE

$1 to $6 $9	Message transmission to the automatic control function or a remote server or a peripheral or a PC.
1	If $1, $2, $3 or $4 is followed by a 1 (i.e. $11, $21, $31 or $41), the message is «blocking» (see below).
=	The character «=» after the message recipient indicates that the message is a value or sequence of values (see notes).
MESSAGE	The message can contain: - 80 characters if «=» is not included in the syntax, - 1 to 16 values if «=» is included in the syntax.

Notes

If $1, $2, $3 or $4 is followed by a 1 (i.e. $11, $21, $31 or $41), the message is «blocking», i.e. the part programme waits for the recipient to acknowledge the message. If there is not a «1», the message is nonblocking (always the case for messages sent by $5 or $6).

If the «=» character follows the recipient identifier, the message is a value or a sequence of values (each value separated from the others by «=»). A value can be the result of a parametric expression which can include 1 to 6 values. If the «=» character is not present, the message sent includes all the characters present up to action on «Enter».

Example:

Transmission of a nonblocking message including a sequence of three values to UNI-TELWAY slave.

`$2 = 3 = E70000/1000 = L0*3/L1`	Displays the result of the parametric expression

Transmission of a blocking message including a sequence of two values to PLC function.

`$11 = E51001 = E52001`	Values displayed and wait for acknowledgement

Message acknowledgement

The messages with addresses $1 to $4 are sent to the recipient by a UNITE request. After transmission of a blocking message, the NC goes on wait for an acknowledgement which must be sent to it by a write request. The NC remains on wait and retransmits the same message every ten seconds until it receives the request for the axis group concerned (for further details, see the Automatic Control Function Programming Manual).

Answer to a message

After sending a message $1 to $4, the NC can wait for an answer as a value that it inserts in a parametric expression.

Example: L0 = $1 + ...

Notes on Message Transmission to a Peripheral by $5 and $6

$5 and $6 send messages to the serial line customisation module. Two lines can be assigned to this function by the SERIAL LINE SETTING tool with configuration Mess $5 and Mess $6 (accessed by the NC UTILITIES. For further details, see the Operator Manual.

Configurations Mess $5 and Mess $6 are used to send a message to a peripheral without imposing a protocol.

If no Mess $5 or Mess $6 configuration is specified in line customisation, the attempt to send a message by $5 or $6 causes display of error message 11.

If flow control is used (RTS/CTS or Xon/Xoff), transmission may be blocked and momentarily suspend execution of the part programme.

Specific Feature of Message Transmission to a PC by $9

$9 is used to send a simple message or a message with wait for an answer from the PC client application.

Example:

```
$9 Message ...
$9 =
```

4.20 Spindle Synchronisation

4.20.1 Spindle Acceleration Control

The CNC generates an acceleration ramp when the following functions are active.

For software at index G, the ramp is generated by:
- spindle speed build-up
- change in programmed speed (S..)
- programmed spindle stop (M05)
- spindle indexing (M19) if bit 5 of machine parameter P6, words 1 to 4 = 1.

For software at index H, management of the ramp is extended to:
- speed variations due to the spindle potentiometers
- constant surface speed
- spindle stop requested by the PLC (spindle stop requested by setting bit %W22.0 for spindle 0, %W22.1 for spindle 1, etc.).

Notes on Spindle Rotation Reversal

The notes below on spindle rotation reversal concern:
- speed-controlled spindles,
- servo-controlled spindles.

<u>Speed-controlled spindle rotation reversal</u>

For speed-controlled spindles, rotation reversal (switch from M03 to M04 or vice versa) is instantaneous so as not to modify the previous steady state. To apply an acceleration rate to reversal, it is necessary to programme a spindle stop (M05), then reprogramme the direction of rotation.

<u>Servo-controlled spindle rotation reversal</u>

For servo-controlled spindles, rotation reversal is a continuous ramp function. The ramp is defined by the acceleration declared in machine parameter P32 (first of the pair of values applied to the axis number assigned to the spindle). It should be noted that the acceleration can be modified by external parameter E9033x (x = spindle number from 0 to 3).

Note on Spindle Indexing

For spindle indexing (M19), if positioning is carried out at a constant deceleration (see P6, bit 5 of words 1 to 4), the phase during which spindle rotation is decreased down to the minimum speed is also carried out at a constant deceleration and forcing of a null speed is disabled.

4.20.2 Servo-Controlled Spindles and Synchronised Spindles

4.20.2.1 Servo-Controlled Spindles

Generally, the spindle speed is controlled directly by a programmed speed setting (S..) that may, where required, be overridden by potentiometer.

It may prove necessary to servo-control the spindles in position to couple them (or for any other reason). In this case, the setting programming by S.. is used to vary a reference position. The reference applied to the spindle servo-drive is the position error between the reference position and the measured position multiplied by a coefficient which is the servo-loop gain.

⚠ CAUTION

Spindle servo-control is preserved if the «PRES.PUIS.» signal (A.10C or %W4.4) is set. Otherwise, «position reference» = «position measurement». When signal «PRES.PUIS.» goes high again, the spindle speed is built up again according to the ramp.

Spindle Position Servo-Control Request

Position servoing of a spindle is enabled by external parameter E91024 plus the spindle number, i.e. E91024+X (x = spindle number from 0 to 3).

Example

Spindle position servo-control requests:
- Spindle 0 position servo-control request: E91024 = 1
- Spindle 1 position servo-control request: E91025 = 1
- Spindle 2 position servo-control request: E91026 = 1
- Spindle 3 position servo-control request: E91027 = 1

REMARK *To request speed control of a spindle, set E9102x = 0.*

Notes on Spindle Position Servo-Control

In order for a spindle servo-control request to be accepted:
- The spindle must be stopped (state M05 or M19)
- Homing must be completed on the spindle
- Function 22 (integrated spindle synchronisation) must be active (or error message 94 is returned).

CNC state at power on

When power is applied, the spindle speed control state is active. Subsequently, the servo-controlled or non-servo-controlled state of the spindle is preserved after a reset.

Spindle reference position read

The spindle reference position can be read in external parameter E95024+x (x = spindle number from 0 to 3).

Spindle position gain

The position gain of a spindle is given in machine parameter P45 (as for indexing). The gain is expressed in rev/min/rev and can be changed from gain to indexing by external parameter E9032x (x = spindle number from 0 to 3).

4.20.2.2 Synchronised Spindles

Spindle synchronisation applied to slave (or driven) spindle position synchronisation with master (or drive) spindles.

Spindle Synchronisation Request

When spindle synchronisation is requested, the value present with EC (in degrees) gives the requested difference between the master and slave spindle positions, i.e.: slvpos = +/+ (mas_pos + EC)

REMARK *The master and slave spindles must use the same measuring unit.*

Synchronisation of a slave spindle «s» on a master spindle «m» is programmed by external parameter E94124 plus the slave spindle «s» number and containing the address of the master spindle «m», i.e. E94124+s = 24+m (s and m = spindle numbers from 0 to 3).

Example

Request to synchronise slave spindle s = 0 to master spindle m = 2.

Programming: E94124 = 26

Desynchronisation

Synchronisation is disabled by setting programme parameter E94124+s to -1.

Notes on Spindle Synchronisation

Preparation before spindle synchronisation

Before synchronising the spindles:
- the master and slave spindles must be declared as servo-controlled (or error message 92 is returned),
- the synchronised spindle must be declared in direct or symmetric state.

In order to distinguish between direct and symmetric spindle synchronisation, the direction of rotation (M03 or M04) must already be assigned to the two spindles (slave and master). If the two spindles rotate in the same direction, synchronisation is direct. If not, it is symmetric.

Activation of spindle synchronisation

Synchronisation can be activated with the spindles rotating or stopped (S0). The slave spindle is coupled to the master spindle at constant acceleration.

Spindles synchronised

As long as a slave spindle remains synchronised, it cannot be assigned to a group by programming spindle selection functions M62 to M65 (or error message 38 is returned).

The synchronisation error (difference between the measured positions in measurement increments) is read in parameter E95244+s (s = slave spindle) defining the measurement correction value. If the error is less than the permissible tolerance (machine parameter P44 and external parameter E9031x in internal units defined by modulo), the spindle in-position signal is transmitted to the PLC by %413.B and can be read in the spindle in-position parameter E93524+s (s = slave spindle).

Management of the Synchronisation Error

Management of the synchronisation error is preserved if the spindle servo-drives are latched or off (PRES.PUIS. = 0). However (if the spindles are not mechanically interlocked), it is necessary to make sure the synchronisation error is not too large before restoring servo-control (PRES.PUIS. = 1) since the error correction is instantaneous.

The synchronisation error can be corrected gradually by applying automatic gain control to each of the spindles. The gains must be the same on all the spindles (machine parameter P45 and external parameter E9032x). This is achieved by maintaining a following error rigorously proportional to the currently requested speed and inversely proportional to the gain: Ep = Vdem/K. The correction is made by integrating the errors and assigning a parametric time constant.

The integration time constant is given in ms by external parameter E98024+x. Its programming is not accepted unless spindle x is already servo-controlled (parameter E98024+x is reset when the spindle is declared as servo-controlled by E91024+x). It is cancelled by resetting this parameter to 0.

Once the spindle is synchronised, it may be useful to preserve the correction for a certain time, after the system is stabilised, while freezing error integration. This is done by declaring a negative time constant: E98024 = -E98024.

Synchronisation Example

Spindle 0 (slave) synchronised with spindle 2 (master).

```
%..
...
WHILE E91124=1 DO M00 $ Homing on spindle 0
ENDW
WHILE E91126=1 DO M00 $ Homing on spindle 2
ENDW
E91024=1 E91026=1                    Servo-control of spindles 0 and 2
E91024=500 E91026=500                Apply an integration time constant
M64 M03 (M40) (S..)                  Select slave spindle and assign
                                     a direction of rotation
M62 M04 M41 S.. EC..                 Select master spindle and
                                     start rotation
E94124=26                            Symmetric synchronisation request
WHILE E93524=0 DO G04 F0.1 $ WAIT FOR SYNCHRONISATION OF SPINDLE 0
ENDW
E98024=E98024                        Freeze slave gain correction
...                                    Rest of programme
...
...
...
E94124=-1                            Synchronisation disabled
E94124=500                           Automatic gain control restored
...
```

REMARK *When synchronisation is disabled, spindle 0 resumes its initial speed (that before synchronisation).*

5 Profile Geometry Programming

5

5.1 Profile Geometry Programming (PGP)

5.1.1 General

The system gives the user the possibility of programming all or part of a profile consisting of geometric elements.

The system computes the undefined connecting and intersection points between elements located in the same plane.

The points can be computed between the following elements:
- line/line,
- line/circle,
- circle/circle.

Profile Geometry Programming (PGP):
- can coexist with ISO programming,
- can only be used with absolute dimensions (G90),
- applies according to the selected interpolation axes ZX (G20), XY (G21), YZ (G22) (the change of axes must be programmed on a fully defined point).
- allows programming of the tool Z axis in a block defined by cartesian coordinates XY (G21).

5.1.2 Definition of the Geometric Elements

Profile Geometry Programming (PGP) is carried out by writing a sequence of blocks.

Each block includes a geometric element which can be:
- a line segment, or
- an arc.

A geometric element can be completely or incompletely defined in a block.

Completely defined elements are:
- the end point of a line,
- the end point of an arc with the coordinates of the centre or the radius.

If an element is incompletely defined, then additional information must be contained in the next block or the next two blocks (not including fillets and chamfers).

The necessary and sufficient blocks allowing the system to compute all the coordinates of a geometric element is called a «geometric entity» (see definition).

The origin of a geometric entity is the start point of the first element.

The start point is:
- programmed in the previous block, or
- already computed by the system (the first block of an entity can also be the last block of the previous entity).

Definition of an Entity

A PGP geometric entity is a self-sufficient part of a profile.

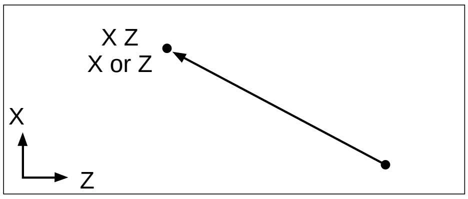
5.1.3 Definition of the Addresses Characterising PGP

The definitions of the following PGP addresses are processed according to the X and Z axes (G20). Use the X and Y axes with cartesian coordinates (G21) (see Sec. 4.13).

5.1.3.1 Addresses with Values

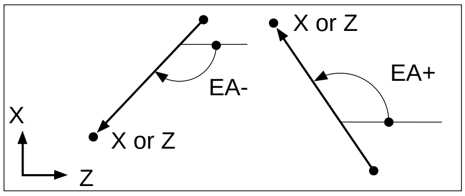
X/Z.. or **X.. Z..**

X/Z.. or X.. Z..: Coordinates of the end point of a line.



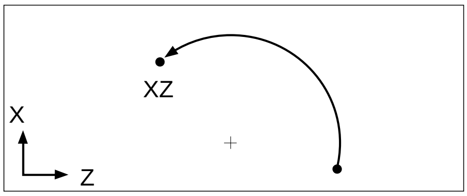
EA..

EA..: Angle element of a line.



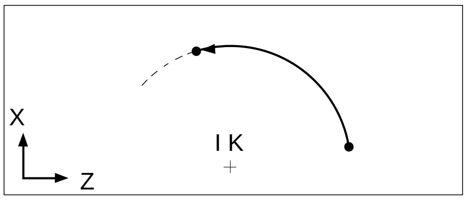
X.. Z..

X.. Z..: Coordinates of the end point of a circle.



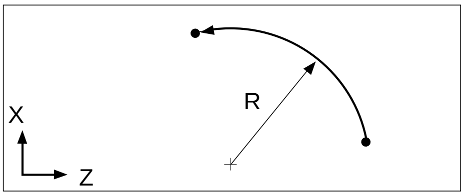
I.. K..

I.. K..: Coordinates of the centre of a circle.



R..

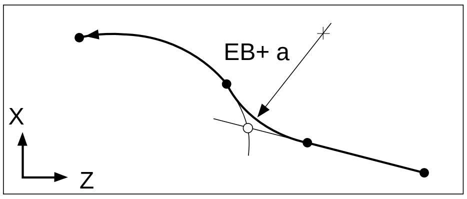
R..: Radius of a circle.



EB+..

EB+..: Fillet between two intersecting elements (e.g. line/circle).

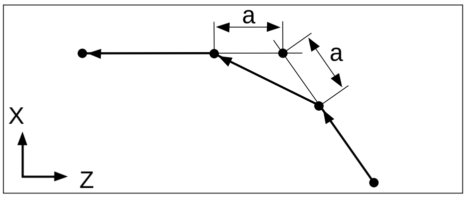
The block containing EB+.. and the next block are connected by a fillet (a=value programmed with EB+).



EB-..

EB-..: Chamfer between two intersecting lines (only).

The block containing EB-.. and the next block are connected by a chamfer (a=value programmed with EB-).

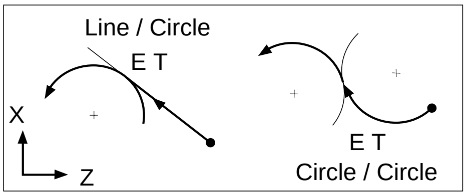


5.1.3.2 Addresses without Values

ET

ET: Tangent element.

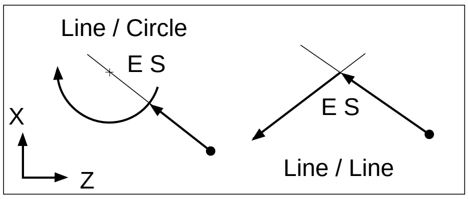
The block containing ET and the next block are tangent. ET is optional unless it is the only function characterising the element, in which case it is mandatory (see Sec. 5.3.2, Figures 10 and 14).



ES

ES: Secant element.

The block containing ES and the next block intersect. If the intersection point of two secant elements is not programmed, ES must be specified in the first block.



E+/E-

E+/E-: Discriminant.

The discriminant is used to dispel the ambiguity when the programming of one or more blocks leaves a choice between two possible solutions.

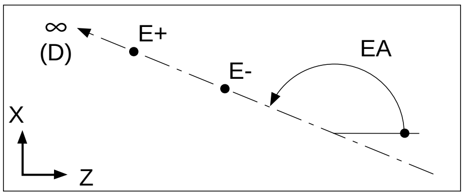
When the discriminant determines an element of an entity:
- it must be programmed in the first block of this entity,
- the + sign or - sign specifies the position of a characteristic point of one of the two solutions with respect to an imaginary line pointing in direction D.

The characteristic points can be:
- the intersection point between two secant elements,
- the point of tangency between two elements,
- the position of the centre of a circle.

The pointing line (D) is:
- the line defined by angle EA.. (if one of the elements of the entity is thus defined),
- the line connecting a known point of the first element to a known point of the last element of the entity (pointing from the first to the last). This known point is preferably the centre of a circle programmed by I and J or, lacking such, another programmed point.

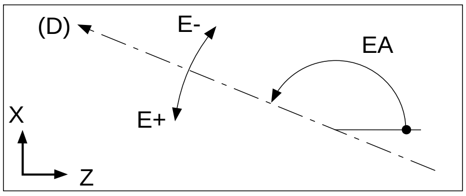
Characteristic points located on the pointing line (D).

- E+ defines the point closest to + ∞ (plus infinity) on line D.
- E- defines the point closest to - ∞ (minus infinity) on line D.



Characteristic points on either side of the pointing line (D).

- E+ defines a point to left of line D.
- E- defines a point to the right of D.



The discriminant can be combined with address ES (secant element) and address ET (tangent element). Example:

E± secant is programmed as ES+ or ES-.

E± tangent is programmed as ET+ or ET-.

Programming of the discriminant with secant elements

For line/circle or circle/circle intersecting elements, the system allows two possible solutions and it is mandatory to programme the discriminant with ES (ES+ or ES-) (see Sec. 5.3.2, Figs. 3a and 3b for instance).

Programming the discriminant with tangent elements

The system limits the number of possible solutions to two (only tangencies without backtracking are carried out by the system).

When there are two possible solutions, these solutions result in:
- the creation of an arc less than 180 degrees, or
- the creation of an arc greater than 180 degrees.

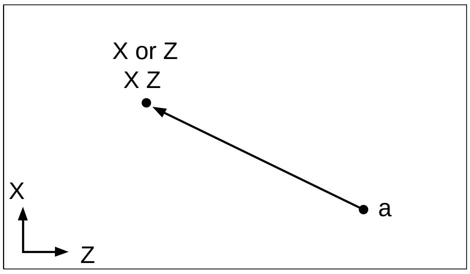
In the two cases, programming of the discriminant with ET is optional. The default solution chosen by the system is that with the smallest arc (see Sec. 5.3.2, Figures 8a and 8b, for instance).

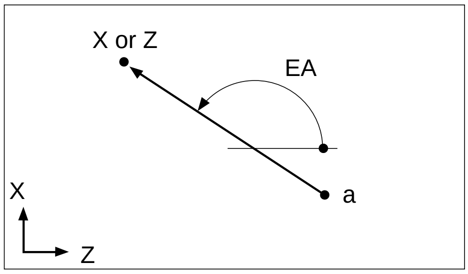
Only exception:
Circle whose centre is inside the next circle and characterised only by: the coordinates of this centre and the fact as it is tangent to the next circle (see Sec. 5.3.2, Figures 5b, 12b, 23b).

5.1.4 Programming of Geometric Elements

5.1.4.1 Programming of Fully Defined Geometric Elements

Fully defined linear geometric element (point «a» defined)

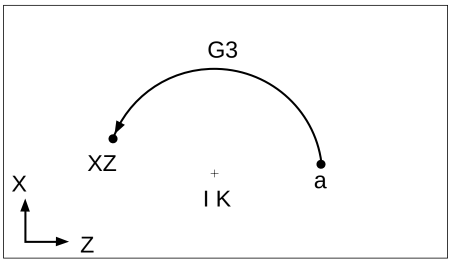


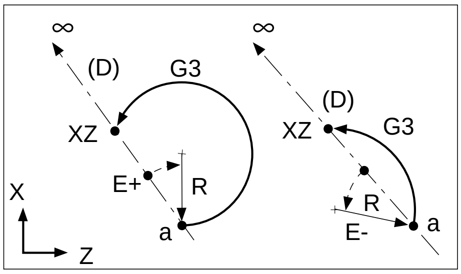



```
   N.. G01 X..
or N.. G01 Z..
or N.. G01 X.. Z..
```

```
   N.. G01 EA.. X..
or N.. G01 EA.. Z..
```

Fully defined circular geometric element (point «a» defined)






```
N.. G02/G03 X.. Z.. I.. K..
```

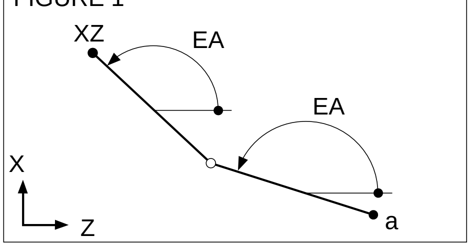
```
N.. G02/G03 X.. Z.. R.. E+/E-
```

5.1.4.2 Programming of geometric Elements not Completely Defined

Geometric elements defined using the following block(s)

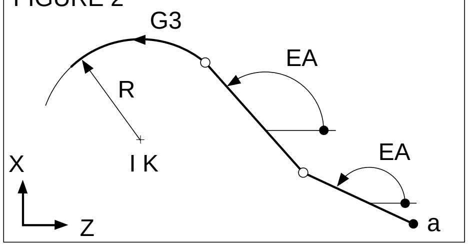
The first element is a line (start point «a» is fully defined).

FIGURE 1



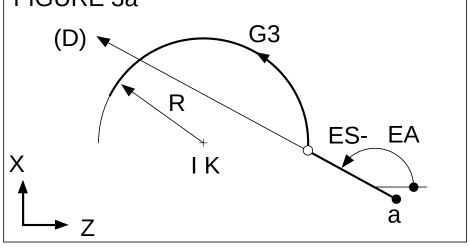

```
N.. G01 EA.. ES
N.. EA X.. Z..
```

FIGURE 2



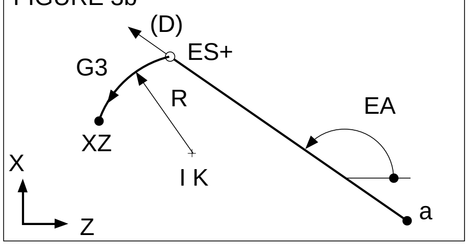

```
N.. G01 EA.. ES
N.. EA..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 3a



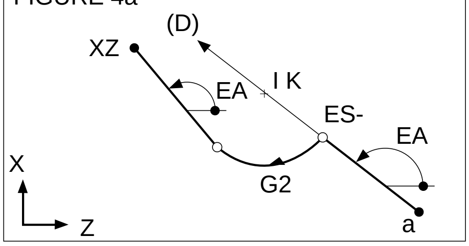

```
N.. G01 EA.. ES-
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 3b



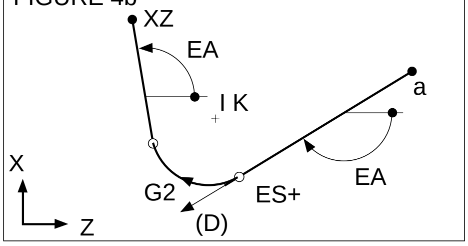

```
N.. G01 EA.. ES+
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 4a

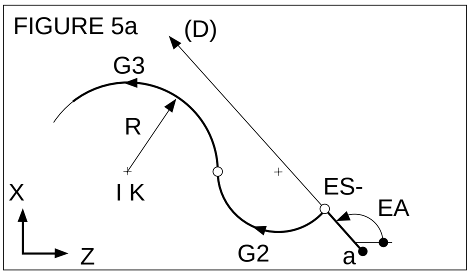



```
N.. G01 EA.. ES-
N.. G02/G03 I..K..
N.. G01 EA.. X.. Z..
```

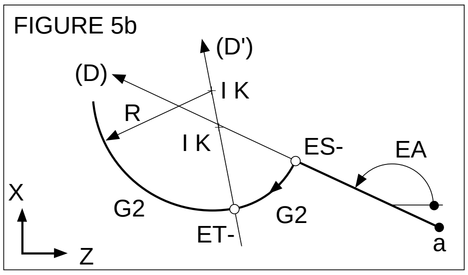
FIGURE 4b



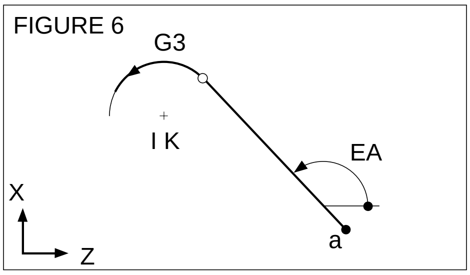

```
N.. G01 EA.. ES+
N.. G02/ G03 I..K..
N.. G01 EA.. X.. Z..
```



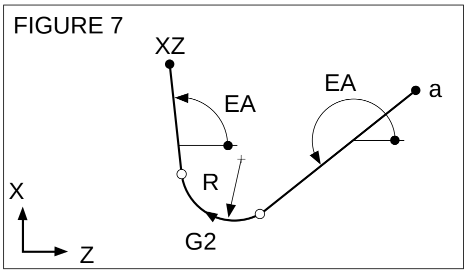

```
N.. G01 EA.. ES-
N.. G02/G03 I.. K..
N.. G02/G03 I.. K.. R../X.. Z..
```




```
N.. G01 EA.. ES-
N.. G02/G03 I.. K.. ET-
N.. G02/G03 I.. K.. R../X.. Z..
```

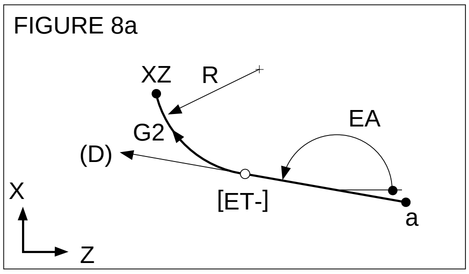



```
N.. G01 EA..
N.. G02/G03 I.. K..
```

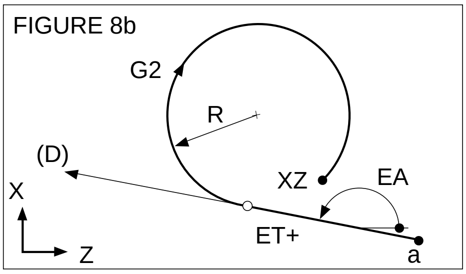



```
N.. G01 EA..
N.. G02/G03 R..
N.. G01 EA.. X.. Z..
```

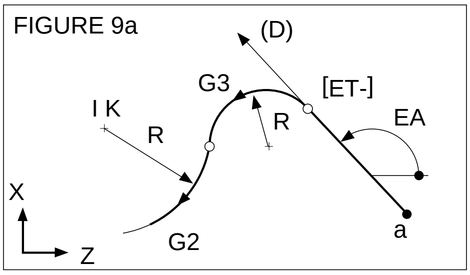
5



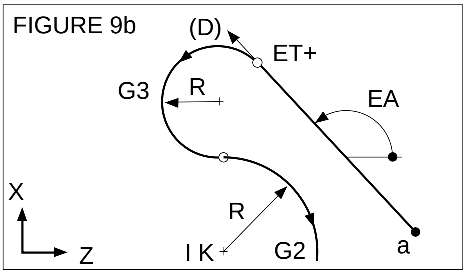

```
N.. G01 EA..
N.. G02/G03 R.. X.. Z..
```



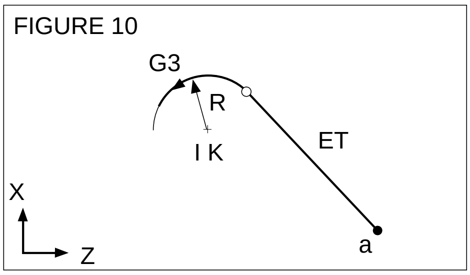

```
N.. G01 EA.. ET+
N.. G02/G03 R.. X.. Z..
```



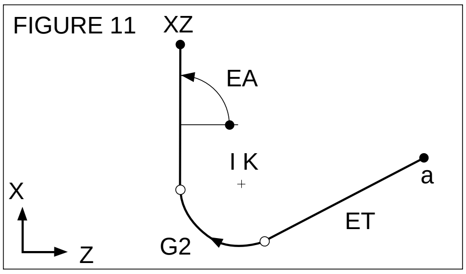

```
N.. G01 EA..
N.. G02/G03 R..
N.. G02/G03 I.. K.. R../X.. Z..
```



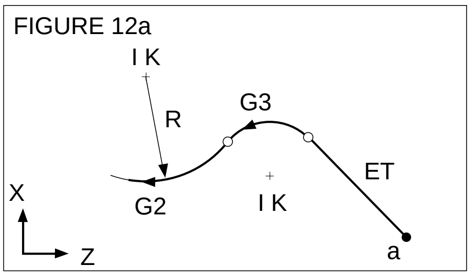

```
N.. G01 EA.. ET+
N.. G02/G03 R..
N.. G02/G03 I.. K.. R../X.. Z..
```



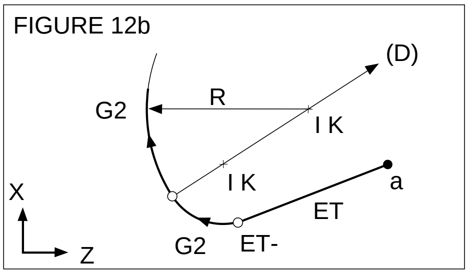

```
N.. G01 ET
N.. G02/G03 I.. K.. R../X.. Z..
```




```
N.. G01 ET
N.. G02/G03 I.. K..
N.. G01 EA.. X.. Z..
```



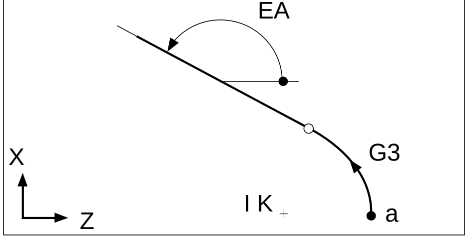

```
N.. G01 ET
N.. G02/G03 I.. K..
N.. G02/G03 I.. K.. R../X.. Z..
```




```
N.. G01 ET
N.. G02/G03 I.. K.. ET-
N.. G02/G03 I.. K.. R../X.. Z..
```

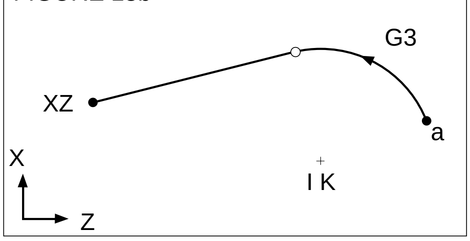
The first element is a circle (start point «a» is fully defined).

FIGURE 13a



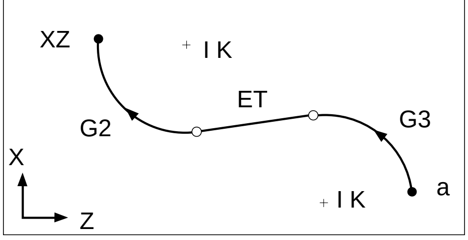
N.. G02/G03 I.. K..
N.. G01 EA..

FIGURE 13b



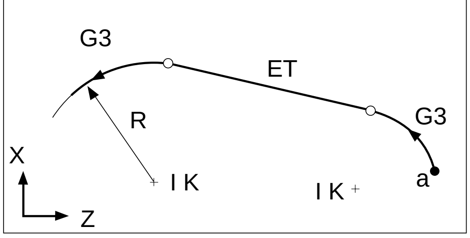
N.. G02/G03 I.. K..
N.. G01 X.. Z..

FIGURE 14a



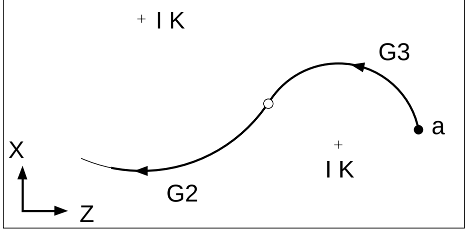
N.. G02/G03 I.. K..
N.. G01 ET
N.. G02/G03 I.. K.. R../X.. Z..

FIGURE 14b



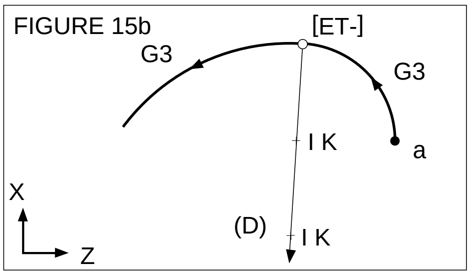
N.. G02/G03 I.. K..
N.. G01 ET
N.. G02/G03 I.. K.. R../X.. Z..

FIGURE 15a



N.. G02/G03 I.. K..
N.. G02/G03 I.. K..

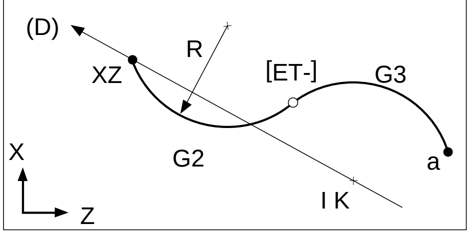
FIGURE 15b



N.. G02/G03 I.. K..
N.. G02/G03 I.. K..

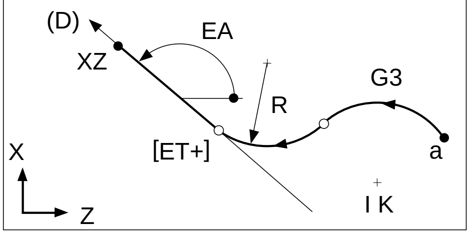
5

FIGURE 16



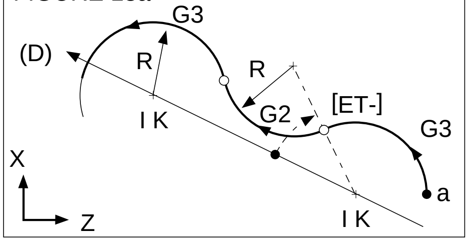
N.. G02/G03 I.. K..
N.. G02/G03 R.. X.. Z..

FIGURE 17



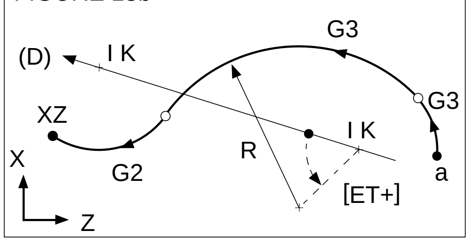
N.. G02/G03 I.. K..
N.. G02/G03 R..
N.. G01 EA.. X.. Z..

FIGURE 18a



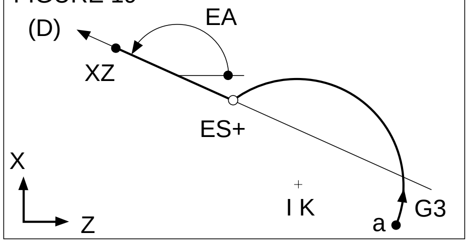
N.. G02/G03 I.. K..
N.. G02/G03 R..
N.. G02/G03 I.. K.. R../X.. Z..

FIGURE 18b



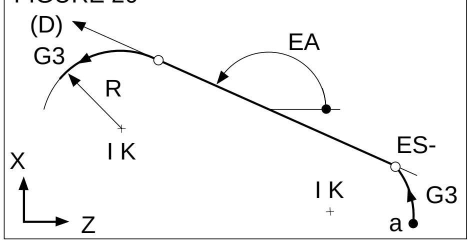
N.. G02/G03 I.. K..
N.. G02/G03 R..
N.. G02/G03 I.. K.. R../X.. Z..

FIGURE 19



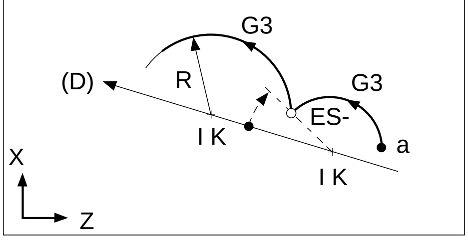
N.. G02/G03 I.. K.. ES+
N.. G01 EA.. X.. Z..

FIGURE 20



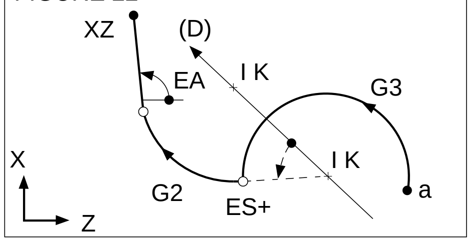
N.. G02/G03 I.. K.. ES-
N.. G01 EA..
N.. G02/G03 I.. K.. R../X.. Z..

FIGURE 21



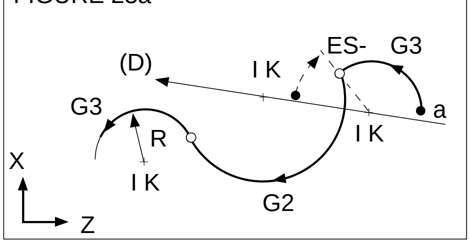

```
N.. G02/G03 I.. K.. ES-
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 22



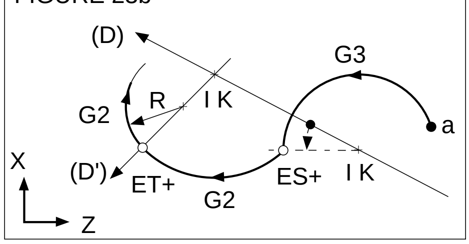

```
N.. G02/G03 I.. K.. ES+
N.. G02/G03 I.. K..
N.. G01 EA X.. Z..
```

FIGURE 23a



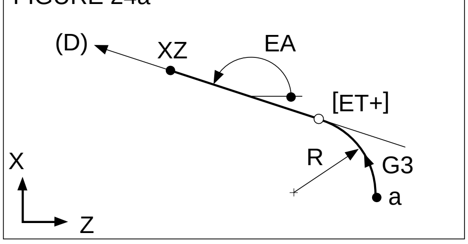

```
N.. G02/G03 I.. K.. ES-
N.. G02/G03 I.. K..
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 23b



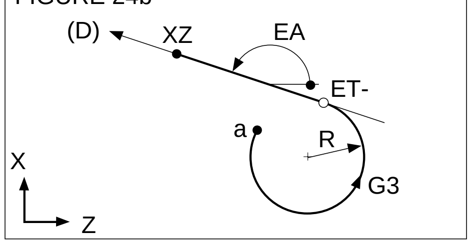

```
N.. G02/G03 I.. K.. ES+
N.. G02/G03 I.. K.. ET+
N.. G02/G03 I.. K.. R../X.. Z..
```

FIGURE 24a

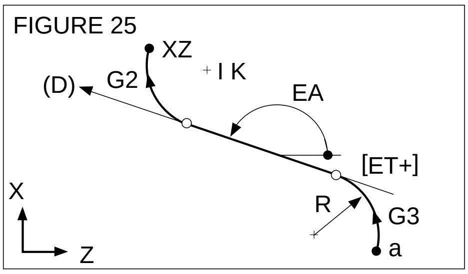


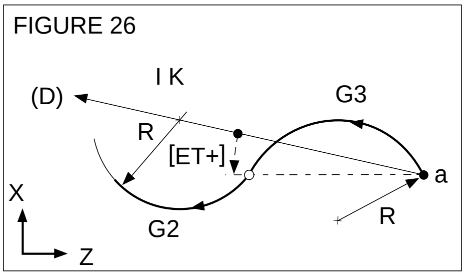

```
N.. G02/G03 R..
N.. G01 EA.. X.. Z..
```

FIGURE 24b



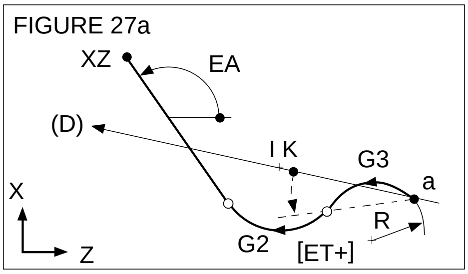

```
N.. G02/G03 R.. ET-
N.. G01 EA.. X.. Z..
```

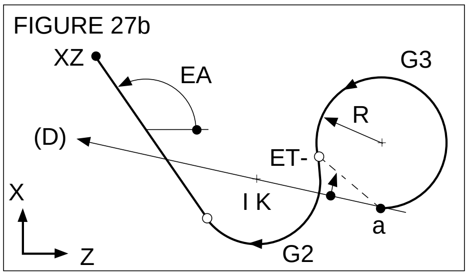





```
N.. G02/G03 R..
N.. G01 EA..
N.. G02/G03 I.. K.. R../X.. Z..
```

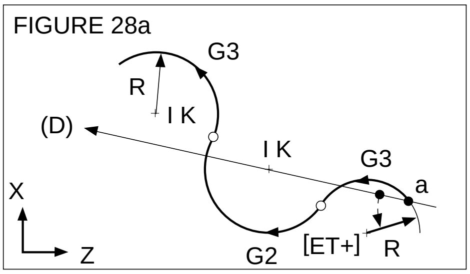
```
N.. G02/G03 R..
N.. G02/G03 I.. K.. R../X.. Z..
```

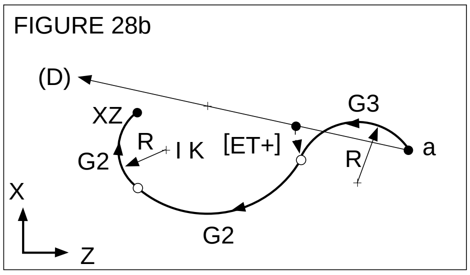





```
N.. G02/G03 R..
N.. G02/G03 I.. K..
N.. G01 EA.. X.. Z..
```

```
N.. G02/G03 R.. ET-
N.. G02/G03 I.. K..
N.. G01 EA.. X.. Z..
```

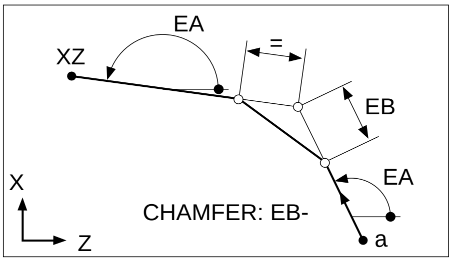


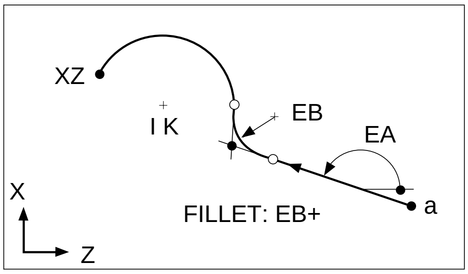



```
N.. G02/G03 R..
N.. G02/G03 I.. K..
N.. G02/G03 I.. K.. R../X.. Z..
```

```
N.. G02/G03 R..
N.. G02/G03 I.. K..
N.. G02/G03 I.. K.. R../X.. Z..
```

5.1.4.3 Programming Chamfers and Fillets Between Two Elements






```
N.. G01 EA.. ES EB-..
N.. G01 EA.. X.. Z..
```

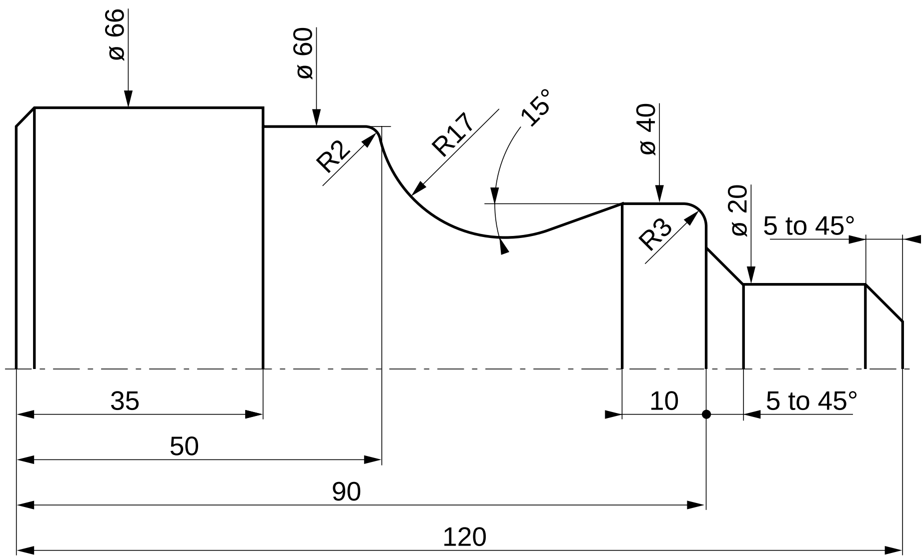
```
N.. G01 EA.. ES- EB+..
N.. G02/G03 I.. K.. X.. Z..
```

5

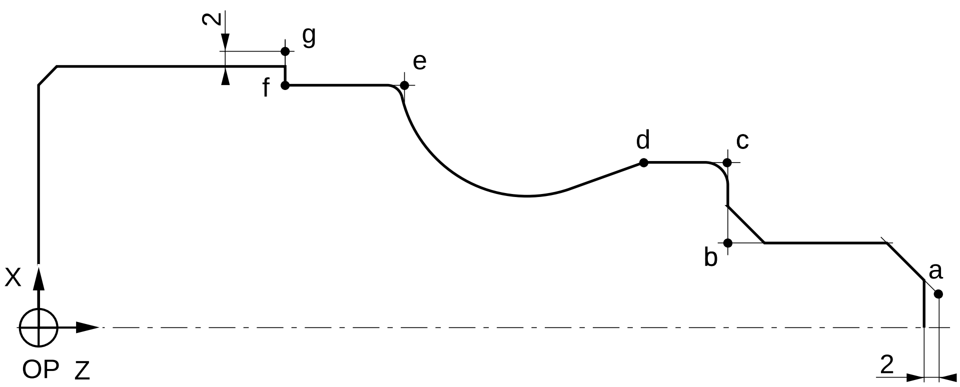
5.1.4.4 Examples of Programming in PGP

Examples

In PGP, part profile definition



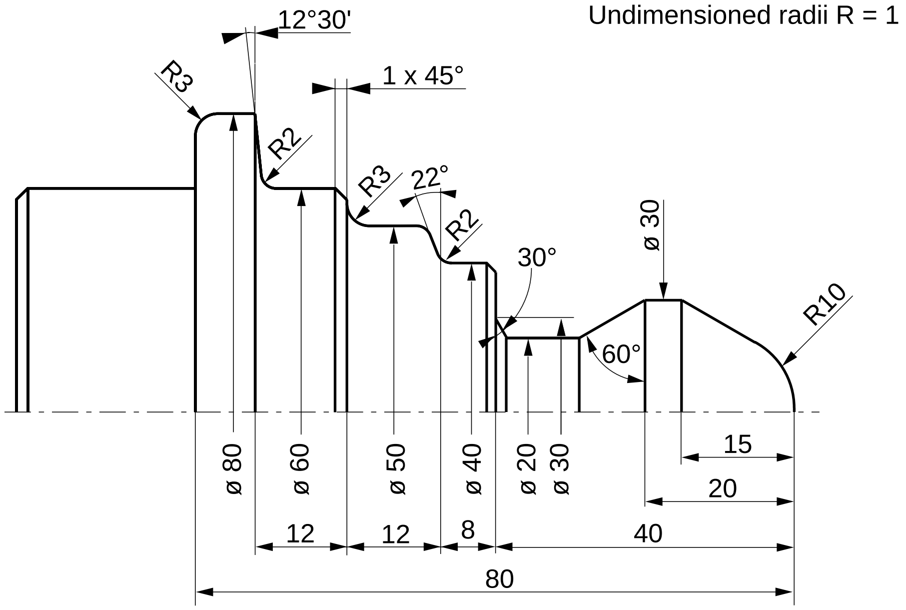
Machining paths



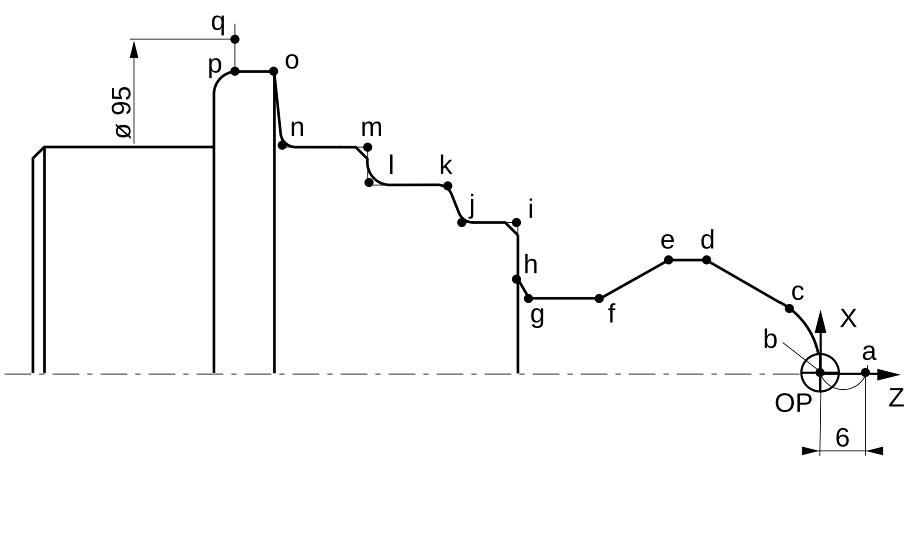
```
%88
N10 ...
N.. G92 S3000
N..
N90 G00 G52 X0 Z0
N100 T03 D03 M06 (COPYING TOOL R0.4)
N110 G97 S900 M04
N120 G00 G42 X6 Z122                    Point a, approach
N130 G96 S220
N140 G95 F0.1
N150 G01 EA135 ES
N160 EA180 X20 Z90 EB-5                 Point b
N170 X40 EB3                            Point c
N180 EA180 Z80 ES                       Point d
N190 EA195
N200 G02 X60 Z50 R17 EB2                Point e
N210 G01 Z35                            Point f
N220 X70                                Point g
N230 G00 G40 G52 X0 Z0 G97 S900
...
```

5

In PGP, part profile definition



Machining paths



```
%109
N10 ...
N.. G92 S3000
N..
N110 G00 G52 X0 Z0
N120 T05 D05 M06 (COPYING TOOL R0.4)
N130 G97 S900 M04
N140 G00 G42 X0 Z6                  Point a, approach
N150 G92 S3000
N160 G96 S150
N170 G95 F0.08
N180 G02 X0 Z0 I0 K3                Point b
N190 G03 I0 K-10                    Point c
N200 G01 X30 Z-15                   Point d
N210 Z-20                           Point e
N220 EA-150 X20                     Point f
N230 EA180 ES                       Point g
N240 EA120 X30 Z-40                 Point h
N250 X40 EB1                        Point i
N260 EA180 Z-48 EB2                 Point j
N270 EA 112 X50 EB1                 Point k
N280 Z-60 EB3                       Point l
N290 X60 EB-1                       Point m
N300 EA180 ES EB2                   Point n
N310 EA102.5 X80 Z-72 EB1           Point o
N320 Z-78                           Point p
N330 G00 X95                        Point q
N340 G40 G52 X0 Z0 G97 S900
N..
```

5.2 Profil Function

The PROFIL function is covered in a separate manual titled «PROFIL Function User's Manual».

This section reviews how to:
- access PROFIL,
- call a contour created by PROFIL.

5.2.1 Accessing Profil

PROFIL is accessed by the ISO editor background task editing function. PROFIL cannot be accessed in edit (EDIT) mode.

Requirements

Basic softkeys displayed. CNC in Auto, Single, Manual mode or no mode selected.

Actions

Select the GRAPHIC PROGRAMMING menu. ☞ PROCAM

The «GRAPHIC PROGRAMMING» menu is displayed.

Select «5 BACKGROUND EDIT». ☞ % 5 ⟵

The message «SPECIFY PROGRAMME» is displayed.

Enter the programme number in which the contour is to be described «%[programme No.]». ☞ ⟵

When the programme number is new, the CNC displays the message «CREATE NEW PROGRAMME (Y/N)?.

Confirm creation of a new programme. ☞) 0 ⟵

Display of: =%[programme No.].

REMARK *When the programme number already exists, the CNC displays it (e.g. =%50) followed by the blocks it contains.*

Enter the PROFIL access letter. ☞ P ⟵

The PROFIL main menu is displayed.

5.2.2 Calling a Contour Created by PROFIL

To be executable, a numbered contour created by PROFIL must be called. The call can be made in two ways:
- by the subroutine call function G77,
- from a machining cycle G63, G64 or G65.

The contour call syntax differs according as the contour was created inside the main part programme or in a subroutine.

5.2.2.1 Calling a Contour by Function G77

General Contour Call Syntax by Function G77

G77 [H..] [N.. N..] P.. [S..]

P.. Number of the contour created by the PROFIL function.

Example

Call by G77 of contour 1 located in subroutine %301.

```
%300 (MAIN PROGRAMME)
N..
N..
N..
N150 G77 H301 P1                    Call of contour 1
N..
N..
```

5.2.2.2 Calling a Contour from a Machining Cycle

General Contour Call Syntax from a Machining Cycle

G63/G64/G65 [H..] [N.. N..] EP.. (other arguments of the cycle)

EP.. Number of the contour created by the PROFIL function.

Example

Calling of contour 1 in subroutine %401 from cycle G64.

```
%400 (MAIN PROGRAMME)
N..
N..
N150 G64 H401 EP1 I.. K.. P..                Call to contour 1
N..
N..
```

REMARK *If limits N.. N.. are used, the contour start and end line numbers must be written by the user.*

6 Parametric Programming

Parametric programming uses functions that can be assigned to all the NC addresses in place of numerical values and that can be used as particular functions.

Functions used in parametric programming:
- Programme L variables,
- External E parameters.

6.1 Programme L Variables

6.1.1 Definition

Variables are elements that can be substituted for numerical values to provide a programming aid.

Programme variables are defined by letter address L followed by a number from one to three digits.

6.1.2 List of L Variables

- Variables L0 to L19,
- Variables L100 to L199,
- Variables L900 to L959.

Variables L0 to L19, L100 to L199, L900 to L959 have the same format and use but cause differences in programming (see Sec. 6.1.11).

6.1.3 Variable Assignments

L variables can be assigned to all the programmable NC addresses.

The assignment of an L variable to an NC address automatically results in the correct use of the unit for the programmed address.

6.1.4 Initialisation

The variables are initialised to zero:
- at power on,
- at the end of the programme (M02),
- after a reset.

6.1.5 Application

The values assigned to L variables can be:
- integer or decimal numbers (maximum 8 digits plus sign),
- fixed values or values resulting from operations.

6.1.6 Use

L variables can be used:
- to perform operations,
- for increments and decrements,
- for conditional branches (with function G79) after comparison with an expression,
- jointly with programming of external E parameters to make transfers.

6.1.7 Operations Possible with L Variables

Arithmetic Operations

Arithmetic operation	Symbol	Arithmetic operation	Symbol
Addition	+	Multiplication	*
Subtraction	-	Division	/

REMARK *Division by zero is impossible.*

Arithmetic Functions

Arithmetic function	Symbol	Arithmetic function	Symbol
Sine	S	Arc tangent	A
Cosine	C	Square root	R
Truncation	T		

Sine (S) and cosine (C)	The term following these functions is in degrees.
Truncation	Extraction of the integer value of the number following the symbol.
Arc tangent	The result of the operation is in thousandths of a degree.

REMARK *Extraction of the square root of a negative number is impossible.*

6

Logic Operations

Logic operation	Symbol	Logic operation	Symbol
AND	&	OR	!

AND and OR	The operations are performed on values from which the decimal part is truncated (truncating performed automatically by the system) and which are expressed in binary.

6.1.8 Comparison Symbols Used with L Variables

Comparison symbol		Comparison symbol	
Equal to	=	Greater than or equal to	> =
Greater than	>	Less than or equal to	< =
Less than	<	Different from	< >

6.1.9 Conversion of the Internal Unit

In parametric expressions, functions U and M are used to convert values expressed in the internal system unit (see Chapters 2 and 3) to the programming unit:
- Function U is specific to linear axes
- Function M is specific to rotary axes.

Examples

<u>Use of function U</u>

Case of a system with micrometres (µm) as internal unit for linear axes.

Reminder: Programming in inches by G70 or metric system by G71.

U254000 returns the value 254 for G71 (254 mm)

U254000 returns the value 10 for G70 (10 inches).

<u>Use of function M (rotary axes)</u>

Case of a system with 0.0001 degree as internal unit for rotary axes.

External parameter E78000 defines the reference position of axis 8 (C axis).

Conversion from 0.0001 degree (internal unit) to the programming unit in degrees. For any position on the C axis, the programming is:

L0=ME78000 then L0=[.IRX(9)]

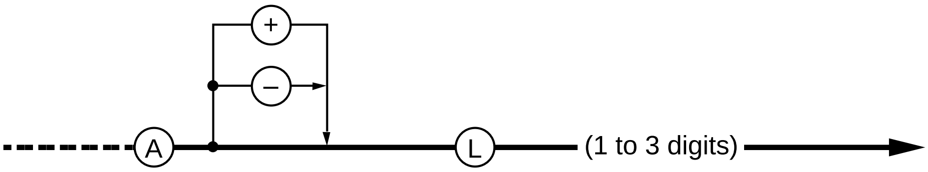
Variable L0 returns the position on the C axis in degrees.

6.1.10 Programming Syntax for L Variables

Programming syntax for L variables is given as Conway diagrams and followed by programming examples.

6.1.10.1 Assignment of a Variable to an NC function

Syntax



A	NC function.
+ / -	Sign.
L	Variable used as numerical value.

6

Example

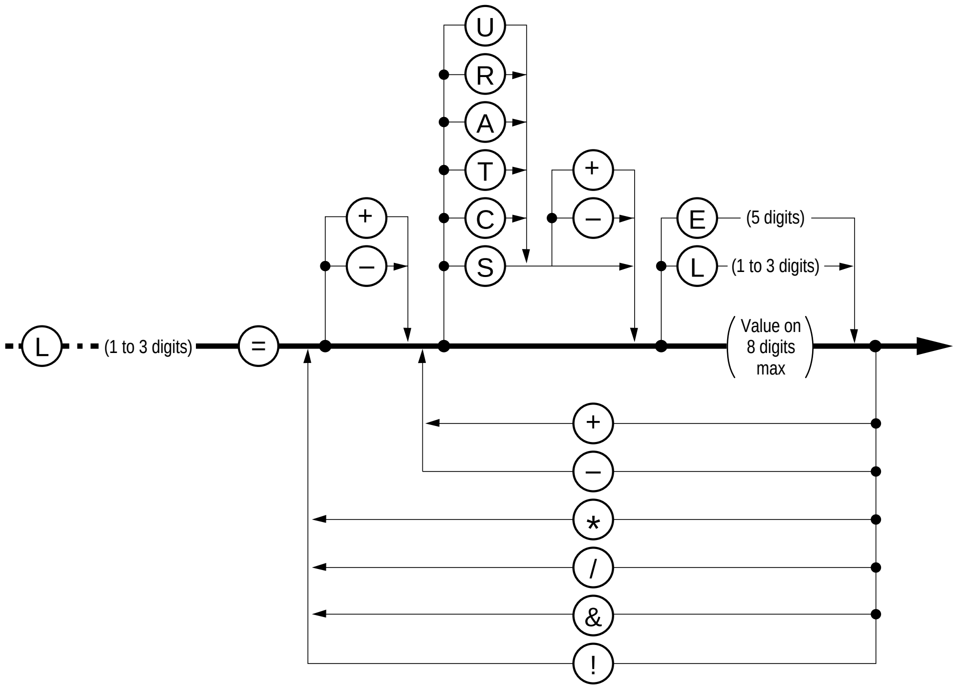
Use of a variable with NC addresses that have different units.

Assignment of variable L5 to addresses X and F.

`N..`	
`L5 = 18`	Declaration of the value of variable L5
`N..`	
`N.. G00 XL5`	L5 equals 18 mm with the X address
`N..`	
`N80 G94 G01 Z70 FL5`	L5 equals a feed rate of 18 mm/min with the F address
`N..`	

6.1.10.2 Declaration of a Variable in the Programme

Syntax



The operations given in this diagram are detailed in Section 6.1.7.

Notes

When the result of an operation giving a fractional number is assigned to an L variable, the system keeps the first eight digits and truncates the following digits (after the decimal point). When the integer part of the result exceeds eight digits, the system reports an error.

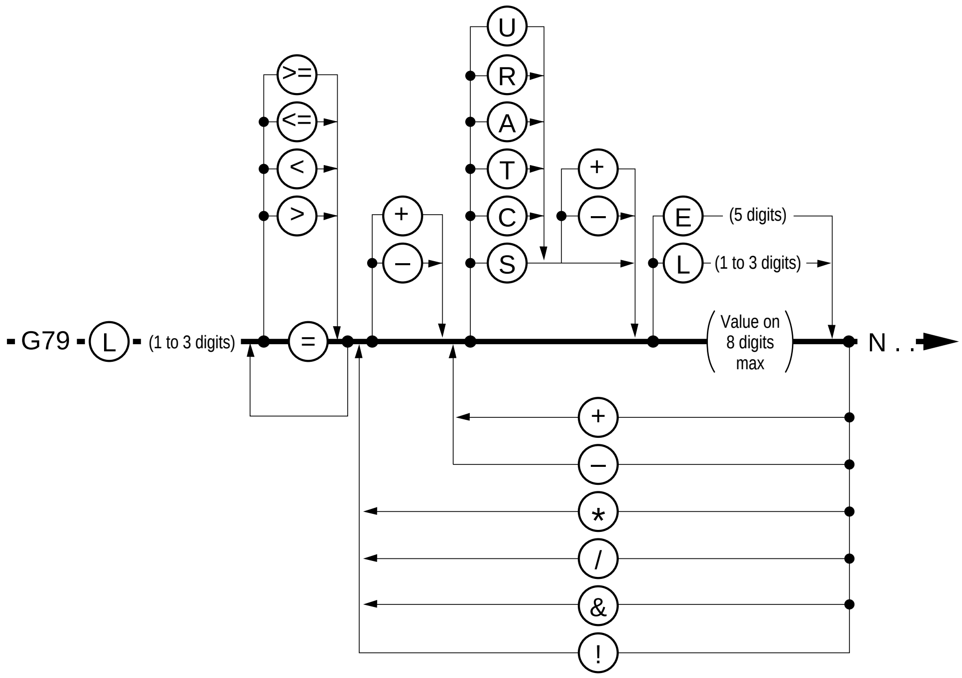
Example

Use of the variables with arithmetic operations.

`N..`	
`L1 = 5`	Declaration of the value of L1
`L2 = L1 + 5.3 * 3 * S30`	After the operation, L2 takes on the value 15.45 (sine 30° = 0.5)
`L3 = 100 / 3`	After the operation, L3 takes on the value 33.333333 (limitation to eight digits)
`N..`	
`N90 G00 XL2 Z30`	The value of L2 (15.45) is assigned to the X axis
`N100 XL3`	The value of L3 (truncated to 33.333) is assigned to the X axis
`N..`	

6.1.10.3 Test of a Variable for a Conditional Branch

Syntax



REMARK *For tests on fractions, see Sec. 6.8.2.*

Example

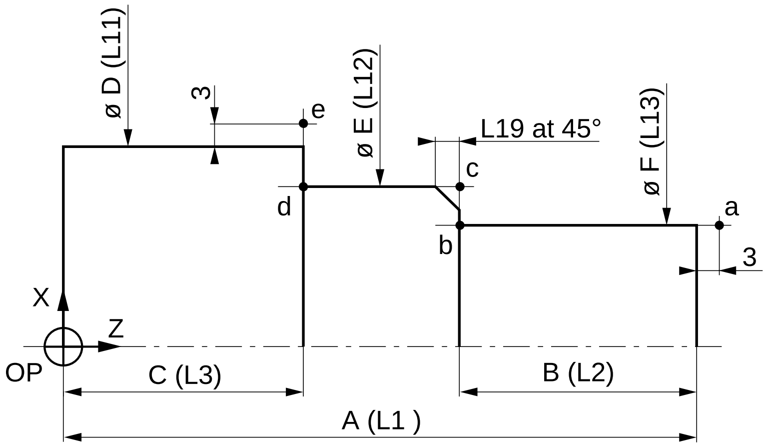
Use of a variable with a conditional test on the variable contents.

`N..`	
`N50 L1 = 0`	Initialisation of the variable with zero
`N60 L1 = L1 + 1`	Increment of the variable
`N..`	
`N..`	
`N200 G79 L1 < 10 N60`	Condition: if L1 < 10, jump to N60; else continue
`N210`	
`N..`	

6.1.10.4 Examples of Programming L Variables

Use of variables for machining parts of a family

Machining of part 2 according the table.



Variable assignment

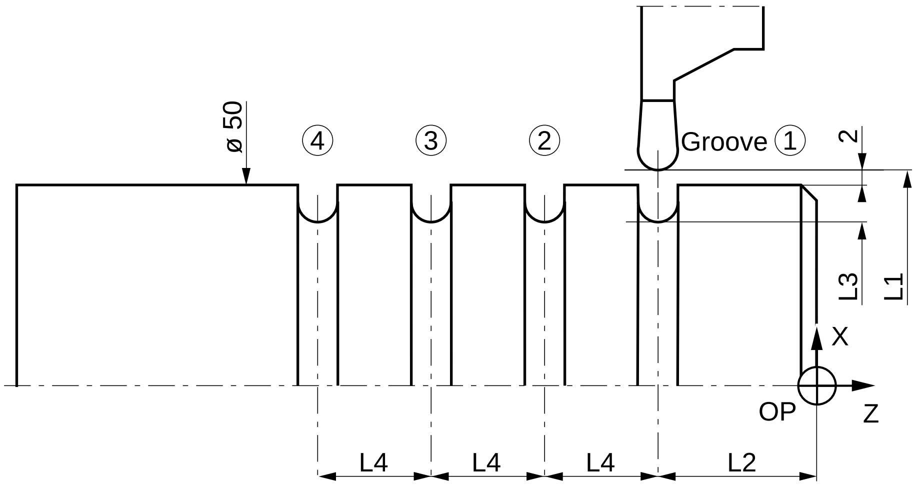
Variables	L1	L2	L3	L11	L12	L13	L19

Dimension	A	B	C	D	E	F	Chamf.
Part 1	80	30	30	50	40	30	2
Part 2	85	28	35	48	39	28	1.5

```
%18
L1=85 (LENGTH A)
L2=28 (LENGTH B)
L3=35 (LENGTH C)
L11=48 (DIAMETER D)
L12=39 (DIAMETER E)
L13=28 (DIAMETER F)
L19=1.5 (CHAMFER)
N10 G00 G52 X0 Z0
N20 T01 D01 M06 (EXTERNAL TOOL)
N30 G97 S900 M40 M03
N40 G95 F0.2
L4=L1+3                              Computation of point Za
N50 G42 XL13 ZL4                     Point a
N60 G92 S2000
N70 G96 S160
L5=L1-L2                             Computation of point Zb
N80 G01 ZL5                          Point b
N90 XL12 EB-L19                      Point c
N100 ZL3                             Point d
L6=3*2+L11                           Computation of point Xe
N110 XL6                             Point e
N120 G40 G77 N10 N10 G97 S900 M05
N130 M02
```

Use of variables with conditional test on the number of grooves

Execution of four grooves by programming a single G83 cycle.




```
%42
L1=54 (START X)
L2=20 (START Z)
L3=40 (GROOVE BOTTOM DIAMETER)
L4=15 (STEP BETWEEN GROOVES)
L10=50 ((CONSTANT SURFACE SPEED) L11=0.1 (FEED RATE))
N10 G00 G52 X0 Z0
N20 G16 P+
N30 T01 D01 M06 (GROOVING TOOL)
N40 G97 S900 M40 M03
N50 XL1 Z-L2                    Groove 1 axis position
N60 L5=L5+1                     Increment a counter
N70 G92 S2000
N80 G96 SL10
N90 G83 XL3 P4 Q1 G95 FL11      Groove in cycle
N100 G00 G80 G79 L5=4 N130      Condition: if L5=4 go
                                to block N130
N110 G91 Z-L4                   Spacing between two grooves
N120 G90 G79 N60                Branch to execute the next groove
N130 G77 N10 N10 G97 S900 M05
N140 M02
```

6.1.11 Notes on Programming Variables L100 to L199 and L900 to L959

Variables L100 to L199

Variables L100 to L199 have the same format and use as variables L0 to L19 but cause a difference in programming.

Loading (entry) of a variable L100 to L199 suspends preparation of the block to which it belongs until execution of the previous block is completed.

A block including a variable L100 to L199 can therefore not be preceded by a block whose execution requires knowledge of the following block(s) (i.e. Profile Geometry Programming (PGP) or tool radius offset (G41, G42)).
Example:
- Profile geometry programming (PGP),
- tool radius offset (G41, G42).

Programming of variables L100 to L199 with function M999

Programming M999 prohibits the operator or PLC from intervening in execution of a sequence of blocks, allowing the use of variables L100 to L199 in the same way as variables L0 to L9 (see Sec. 4.14.9).

i.e. M999 allows anticipated execution of these operations.

If these variables are used in dynamic operators as equivalents having no relation with the functions they represent, they must not be used for calculations.

Variables L900 to L959

It is not recommended to use variables L900 to L959 in a programme with canned cycles (G81, G82, etc.), since they risk being overwritten when a cycle is called.

6.1.12 Equivalence of Variables L900 to L925

Variables L900 to L925 are equivalent to addresses A to Z respectively.

Example:
A = 250 is equivalent to L900 = 250
B = 1234 is equivalent to L901 = 1234 (and so forth to L925 for Z).

6.1.13 Symbolic Addressing of Variables L900 to L925 and L926 to L951

Variables L900 to L925 and L926 to L951 can be addressed by alphabetic symbols preceded by «'« (apostrophe). The variables can be used in the left or right member of an expression.

Variables L900 to L925

Variables L900 to L925 can be addressed by symbols 'A to 'Z respectively.

Example:

'C = 'A + 'B is equivalent to L902 = L900 + L901

Variables L926 to L951

Variables L926 to L951 can be addressed by symbols 'EA to 'EZ respectively ('EA = L926, 'EB = L927, etc. down to 'EZ = L951).

Example:

'A = 'B - 'EA/'EZ is equivalent to L900 = L901-L926/L951

6.2 External E Parameters

6.2.1 Definition

External E parameters are used by the part programme to access information contained in the PLC or CNC memory.

External parameters are defined by address letter E followed by five digits.

6.2.2 Application

Depending on the external E parameters, the part programme accesses the CNC or PLC memory for:
- read only,
- read/write.

The value assigned to an external E parameter is always an integer.

6.2.3 Assignment

The assignment of an external parameter to an NC address causes the same units to be used for the E parameter and the programmed address.

6.2.4 Initialisation

The automatic control function initialises parameters E10000, E20000, E21000, E30000, E40000, E41000, E41001 and E41002. The other external parameters are never reset by the system.

6.2.5 Use

The external parameters can be used:
- for operations,
- for increments and decrements,
- for conditional branches with function G79 after comparison with an expression,
- jointly with L variables to make transfers.

Precautions for Use

The use of the external parameters obeys certain rules:
- the use of parameters E11000 is reserved for creation of special machining cycles,
- parameters E20000, E40000, E41000, E41001, E41002, E70000 are read-only parameters,
- when used for tests (with G79), parameters E20000 etc. can modify the sequence of execution of the current part programme blocks,
- an operation on an external parameter stops movement at the end of the previous block,
- a block including an external parameter cannot be preceded by a block whose execution requires knowledge of the following block(s).

Example:
- Profile Geometry Programming (PGP),
- tool radius offset (G41, G42).

6.2.6 Operations Possible with External E Parameters

Arithmetic Operations

Arithmetic operation	Symbol	Arithmetic operation	Symbol
Addition	+	Multiplication	*
Subtraction	-	Division	/

REMARK Division by zero is impossible.

Arithmetic Functions

Arithmetic function	Symbol	Arithmetic function	Symbol
Sine	S	Arc tangent	A
Cosine	C	Square root	R

Sine (S) and cosine (C) The term following these functions is in ten-thousandths of a degree.

Arc tangent The result of the operation is in thousandths of a degree.

REMARK Extraction of the square root of a negative number is impossible.

Logic Operations

Logic operation	Symbol	Logic operation	Symbol
AND	&	OR	!

AND and OR The operations are performed on binary values.

6.2.7 Comparison Symbols Used with External E Parameters

Comparison symbol		Comparison symbol	
Equal to	=	Greater than or equal to	> =
Greater than	>	Less than or equal to	< =
Less than	<	Different from	< >

6.2.8 Conversion of the Internal Unit

In parametric expressions, functions U and M are used to convert values expressed in the internal system unit (see Chapters 2 and 3) to the programming unit:

- Function U is specific to linear axes
- Function M is specific to rotary axes.

Examples

<u>Use of function U</u>

Case of a system with micrometres (µm) as internal unit for linear axes.

Reminder: Programming in inches by G70 or metric system by G71.

U254000 returns the value 254 for G71 (254 mm)

U254000 returns the value 10 for G70 (10 inches).

<u>Use of function M</u>

Case of a system with 0.0001 degree as internal unit for rotary axes.

External parameter E78000 defines the reference position of axis 8 (C axis).

Conversion from 0.0001 degree (internal unit) to the programming unit in degrees. For any position on the C axis, the programming is:

L0=ME78000 then L0=[.IRX(9)]

Variable L0 returns the position on the C axis in degrees.

6.2.9 List of External E Parameters

6.2.9.1 Exchange Parameters with the Automatic Control Function

E10000 to E10031

E10000 to E10031: List of 32 bits

Read/write parameters.

Parameters used to associate the results of tests on quantities (by G79) with the processing performed by the PLC programme.

The parameters are sent without waiting for a report by the PLC (a dwell G04 F.. can be programmed in the part programme to wait for additional data).

E20000 to E20031

E20000 to E20031: List of 32 bits

Read-only parameters.

Parameters used to read information on the state of the machine during the execution of a part programme.

E20100 to E20111

Read-only parameters addressing the machine interrupt inputs (IT).

E20100 to E20103: Address the four IT inputs of the PLC

E20104 to E20107: Address the four IT inputs of the first IT_ACIA card

E20108 to E20111: Address the four IT inputs of the second IT_ACIA card

For parameters E20104 to E20111, the absence of an IT_ACIA card generates error message 13.

E30000 to E30127

E30000 to E30127: List of 128 words encoded on 32 bits (long word)

Read/write parameters.

Parameters used to send significant signed numerical values read by the PLC programme.

E40000 to E40127

E40000 to E40127: List of 128 words encoded on 32 bits (long word)

Read-only parameters.

Parameters used to read signed numerical values which can be an end point, a shift, etc. from the machine processor.

E42000 to E42127

E42000 to E42127: List of 128 bytes located in PLC/CNC exchange area (accessed by the ladder function)

Read-only parameters, except by dynamic operators (operators: 6, 10, 11 and 15).

6.2.9.2 Programme Analysis Access Parameters

E110xx

The use of these parameters is reserved for the creation of special machining cycles.

Unless otherwise specified, these parameters are of the read/write type and are used to enable or inhibit geometric transformations and speed overrides.

They have values of 0 or 1 and belong to an axis group.

A «0» inhibits transformation and a «1» enables transformation, i.e. it is carried out if the function using it is programmed or if it is enabled by default.

Unless otherwise specified, these parameters are set to 1 by a reset.

E11000: Angular offset (function ED..) enabled

E11001: Programme origin offset (function G59) enabled

E11002: Not used

E11003: Processing of mirror function (function G51) enabled

E11005: Programming by diameter enabled

This parameter is reset to force programming by radius on the X (or U) axis. Programming by diameter is possible only with G20 (programming in X, Y, Z polar coordinates) provided it is enabled by machine parameter P4.

E11006: Tool centre programming enabled

A «0» in this parameter forces programming with reference to the tool centre whatever the orientation defined in the machining parameter.

E11007: Spindle speed potentiometer enabled

When this parameter is set to «0» for an axis group, the spindle speed used for this group is set to 100 percent.

E11008: Drawing of a Complete Circle

With circular interpolation, this parameter is set to «0» to inhibit drawing a complete circle when the start point and end point are the same.

This parameter is forced high after a reset, i.e. drawing of a complete circle is enabled.

E11012: Following error cancellation

For high precision contouring, this parameter enables cancellation of the following error. The parameter is set to enable cancellation and reset to inhibit it.
A system reset does not affect the value of parameter E11012.

E11013: Gradual acceleration

This parameter is set to enable transformation in S of the speed variation. It is reset to vary the speed linearly.
At power on, this parameter is set as declared in machine parameter P7 (bit 5, word 1).

The state of the parameter is preserved after a reset.

E11014: Addressing of the deceleration function on several blocks

The deceleration function is enabled when this parameter is set to «1» and disabled when it is set to «0».

E11015: Angle crossing management enabled

When this parameter is set to «0», the angles are not analysed when calculating the end of block speed.

E11016: Group turret type addressing

A «0» in this parameter indicates a rear turret and a «1» a front turret. The new value is saved until the next CNC reset.

E31000 and E31001

Read/write parameters used to choose a type of line for graphic display.

These parameters are set to zero by graphic initialisation of the system.

E31000: Line type for G00 in graphic mode

E31001: Line type for G01, G02 and G03 in graphic mode

Numbers assigned to the parameters:
- 0: Solid line,
- 1: Dotted line,
- 2: Dashed line,
- 3: Combined line,
- 4: No line (stylus raised).

E32000 to E32005

E32000: Minimum interpolation block execution time

Read/write parameter defining the minimum execution time for a block containing a linear or circular interpolation, expressed in milliseconds (ms).

This parameter can be modified by programming. It only affects the axis group in which it is programmed (it is initialised with the value of P51 by a reset).

E32001: Overspeed coefficient on the path in G12

Read/write parameter defining the handwheel overspeed coefficient expressed in 1/1024 (with G12). The variations of the increments on the first handwheel multiplied by this coefficient generate an overspeed applied to the programmed path.

The new value written is saved until the NC is reset.

E32002: Servo-control error tolerated on a circle

Parameter defining the servo-control error tolerated on a circle, expressed in micrometres. It is the image of machine parameter P52.

E32003: Angle crossing speed analysis angle

Parameter defining the angle above which angle crossing speed is always analysed. This parameter is set to «0» at power on. Its value in degrees must be declared by programming or this processing is always carried out. The last value programmed is saved after a reset.

E32004: Sagittal error

Parameter defining the sagittal error in micrometres. At power on, this value is initialised with 10 and can be modified by programming. The last value programmed is saved after a reset.

E32005: Number of total speed anticipation filter terms

Parameter defining the number of terms of the total speed anticipation filter. This parameter is assigned to all the axes of the group in which it is declared. It is the image of machine parameter P55 (words 8 to 15).

Write of this parameter is enabled if:
- if the high precision contouring option is present; otherwise the system returns error message 4,
- anticipation is enabled (E11012 = 0), or the system returns error message 95
- it is below 14, or the system returns error message 94.

The last value programmed is saved after a reset.

E49001 to E49128

E49001 to E49128: Operation number read

Read-only parameters giving the dynamic operator declared in steps 1 to 128.

6.2.9.3 Machine State Access Parameters

E21000 to E21255

E21000 to E21255: Presence of functions 0 to 255

Read-only parameters addressing binary values.

These parameters test for presence of functions 0 to 255.
They are set to indicate that the function is present and reset to indicate that it is absent.

E41000 to E41006 and E41102

Read-only parameters returning encoded values.

The test of parameters E41000 to E41003 is used to skip parts of programme loops.

E41000: Current mode number

This parameter is the image of operand EN.20 transmitted to the PLC.

Example: «TEST» mode E41000 = 6.

Numbers assigned to the parameter:
- 0: continuous mode,
- 1: single mode,
- 2: MDI mode,
- 3: rapid mode,
- 4: sequence number search mode,
- 5: edit mode,
- 6: test mode,
- 7: manual mode,
- 8: home mode,
- 9: shift mode,
- 10: automatic tool setting mode,
- 13: loading mode,
- 15: unloading mode.

E41001: Current axis group number

This parameter read in a programme returns the axis group number to which the programme is assigned (group 1: E41001=0, group 2: E41001=1, etc.). The axis group number used for graphic display is equal to the number of groups (for instance, for a machine with three axis groups, E41001=3 when the programme is executed in graphic mode).

When in the TEST mode, the following operations are inhibited:
- stop on switch,
- priority interrupt,
- subroutine call by the automatic control function.

When in the graphic mode, the following operations are ignored:
- write to external parameters,
- subroutine call by M function,
- message transmission to the operator by $0, to the PLC by $1,
- declaration of real-time operations with dynamic operators.

Conditional branches in the part programme can trap the system into an endless loop.

E41002: Number of machine axis groups

This parameter is also the graphic axis group number (reminder: 1 to 8 groups).

E41003: Graphic machining simulation status

The value of this parameter gives the status of use of graphic simulation:
- 0: no graphic simulation,
- 1: graphic simulation with material removal for turning,
- 2: dynamic graphic simulation for turning or milling.

E41004: Job reference, Affaire No.

Eight-decade number.

E41005: Sampling period

Value in microseconds.

E41006: Axis group position servo-loop time constant

This parameter defines the time constant in milliseconds (ms). It is the image of machine parameter P56.

E41102: Number of NC axis groups

This parameter is used to determine the number of PLC axis groups by subtracting parameter E41102 (number of NC axis groups) from parameter E41002 (number of machine axis groups).

E33xyz and E43xyz

Read/write parameters.

E33xyz: Addressing of the PLC output terminals

E43xyz: Addressing of the PLC input terminals

For these parameters:
- The hundreds digit specifies the rack number (x = 0 to 6)
- The tens digit specifies the card number (y = 0 to 9)
- The units digit specifies the channel number (z = 0 to 9).

REMARK *Parameters E33xyz and E43xyz are not accessible unless Ladder variable %Qrc3B.1 = 1 (see Automatic Control Function Programming Manual in Ladder Language).*

Parameters E33xyz can be written by dynamic operator No. 11 (see Dynamic Operators Manual).

Parameters E43xyz can be read by dynamic operator No. 6 (see Dynamic Operators Manual).

The read and write errors detected are listed under numbers 10 to 14 (see Appendix D).

E34xxy and E44xxy

Parameters E34xxy are write-only and parameters E44xxy are read-only.

The data read and written are signed values encoded on 16 bits.

E34xxy: Addressing of 8I8O analogue card analogue outputs

E44xxy: Addressing of 8I8O analogue card analogue inputs

For these parameters:
- The hundreds and tens digits specify the card number (xx = 00 to 13)
- The units digit specifies the channel number (y = 0 to 7).

Parameters E34xxy can be written by dynamic operator No. 11 (see Dynamic Operators Manual).

Parameters E44xxy can be read by dynamic operator No. 6 (see Dynamic Operators Manual).

The read and write errors detected are listed under numbers 10 to 14 (see Appendix D).

E79002 to E79004

E79002: Value of the feed rate potentiometer assigned to the group.

The value is expressed in 1/128.

These parameters are used to address the feed rate override coefficient.

E79002 = 128 corresponds to the 100 percent setting of the potentiometer.

This parameter can only be modified by dynamic operators (read/write).

E79003: Distance remaining to be travelled in the block being executed

The distance on the path programmed in the block being executed is expressed in the internal unit of the system.

This parameter can only be modified by dynamic operators (read/write).

E79004: Current feed rate on the path programmed in the block

The speed is expressed as the distance (in internal system units) per sample time.

This read-only parameter can only be modified by dynamic operators.

6.2.9.4 Machining Parameters

E50000 and E51000

Read-only parameters.

E50000: Current tool correction number

Parameter related to function D.
Example: current tool correction = Dxxx, i.e. E50000 = xxx

E51000: Current tool axis orientation

Parameter giving the physical address of the axis parallel to the tool direction.

If the tool axis orientation is negative, the value of 100 is added to the axis address, i.e.:

0 for X+	100 for X-
(1 for Y+	101 for Y-)
2 for Z+	102 for Z-

Example:

If E51000=2 (positive orientation on the Z axis), E51000=102 (negative orientation on the Z axis).

E5y001 and E5y255 - Tool Corrections

Read/write parameters.

The units are in microns (except for E56001 to E56255 and E57001 to E57255).

E50xxx: Tool X dimension of correction xxx

E51xxx: Tool Z dimension of correction xxx

E52xxx: Tool radius R of correction xxx

E53xxx: Dynamic X correction of correction xxx

E54xxx: Dynamic Z correction of correction xxx

E55xxx: Tool tip orientation of correction xxx

E56001 to E56255: Available parameters (H of the table of dynamic corrections).

Read/write parameter used for management of tool wear or other information (maximum 8 characters).

The values of H can be manipulated by the part programme. For instance, if the programme contains E56001=E56001+1, the value of H of correction D1 is incremented by 1 each time the block is read.

The values of H can be entered manually (see operator's manual).

E57xxx: Parameter defining the tool type of correction xxx
- 0: milling tool,
- 1: turning tool,
- 2: boring tool.

E6x000, E6x001 and E6x005 - Parametric Geometric Transformations

Read/write parameters where x represents the axis number in the group (0 to 8).

The units of axes X, Y, Z, U, V and W (x = 0 to 5) and axes A, B and C (x = 6 to 8) are determined by the internal system units specified for the linear and rotary axes (see Secs. 2.1 and 3.1).

E6x000: DAT1 on axis x

E6x001: DAT2 on axis x

E6x005: Offset programmed by G59 on axis x

E6x002 and E6x003 - Dynamic Machine Travel

Read/write parameters where «x» is the axis number in the group (0 to 8).

The units of axes X, Y, Z, U, V and W (x = 0 to 5) and axes A, B and C (x = 6 to 8) are determined by the internal system units specified for the linear and rotary axes (see Secs. 2.1 and 3.1).

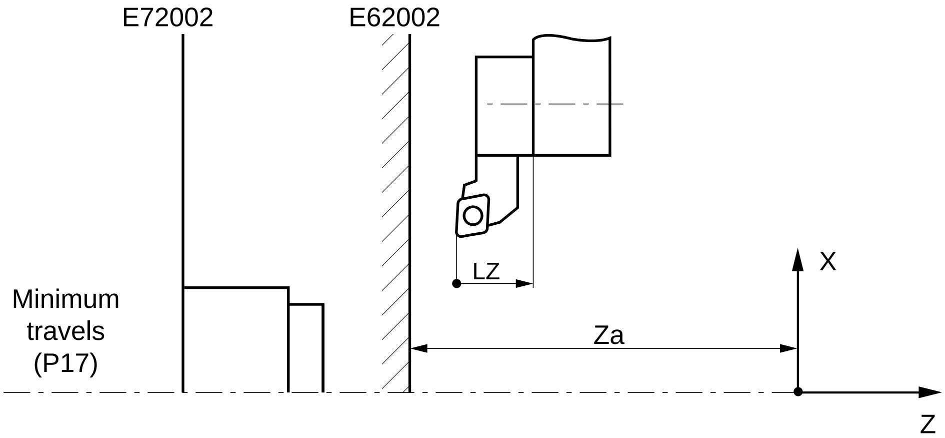
E6x002: Minimum dynamic machine travel on the x axis

E6x003: Maximum dynamic machine travel on the x axis

The machine travels are constrained by software limits entered when setting up the machine (values contained in machine parameter P17).

It may prove necessary to modify these limits according to the part to be machined or the environment (collision avoidance). Parameter E60002 is used to modify the limits inside those defined by parameter P17.

At power on, after a reset or an M02, these parameters are initialised with the value 90000000 (+ or - according as it is the minimum or maximum).



To modify these limits, the programme should contain the following:

```
%28
N..
E62002= Za
```

Declaration of the minimum dynamic travel on Z

```
N.. ...
```

REMARK *During machining, the system applies the tool corrections to remain within the limits defined. For instance, for maximum travel, the tool's cutting edge cannot exceed dimension Za-LZ.*

E69000 to E69002

E69000: Scaling factor

Read/write parameter.

The units are in thousandths.

E69001: Axis tilt on grinder

Read/write parameter.

The unit is ten-thousandths of a degree.

E69002: Axis tilt on grinder

Parameter used to declare a grinder inclination (X and Z axes orthogonal) in an axis group. The unit is ten-thousandths of a degree.

The angle of inclination is preserved after a reset.

6.2.9.5 Group Axis Access Parameters

E7x000 to E7x007 - E7x100 and E7x101

Parameters where x represents the axis number in the group (0 to 8).

For parameters E7x000 to E7x008, the units of axes X, Y, Z, U, V and W (x = 0 to 5) and axes A, B and C (x = 6 to 8) are determined by the internal system units specified for the linear and rotary axes (see Secs. 2.1 and 3.1).

E7x000: Position reference of an axis in the group

Read-only parameters (except for dynamic operators).

E7x001: Probed "on-the-fly" dimension for an axis in the group

Read/write parameters.

These parameters are used to directly read the axis positions during movement after an NC priority interrupt (PLC IT inputs) programmed with G10 (see Sec. 4.11.5).

The measurement is stored in these parameters until:
- the next interrupt is received
- a manual or programmed (M02) reset is carried out.

When G10 is read in the part programme, these parameters can be initialised to 99999999 from the programme so as to make a test after acquiring the dimension (should there be no interrupt before the programmed dimension).

E7x002: Minimum static machine travel

Read-only parameters.

This parameter is the image of machine parameter P17 (see E6x002 and E6x003).

E7x003: Maximum static machine travel

Read-only parameters.

This parameter is the image of machine parameter P17 (see E6x002 and E6x003).

E7x004: Direction of movement of the axes being interpolated

Read-only parameter.

This parameter is used only in conjunction with dynamic operators.

With linear interpolation, this parameter contains the incremental move for the current block on the axis whose physical address is identified by x.

With circular interpolation, this parameter contains the component on axis x of the vector with length R tangent to the path (applies only to linear axes in the interpolation plane).

E7x005: Axis address assignment

Read/write parameters.

Read of these parameters gives the physical address of a programme axis (a programme axis is assigned to a physical axis by machine parameter P9).

When no physical axis is assigned to the programme axis, the parameter takes on the value -1.

Examples:
- if word N2 of parameter P9 has the value 05 (programme axis 5=W of group 1 assigned to physical address 2) in the first axis group, E75005=2 (physical axis 2 assigned to programme axis 5),
- if word N8 of parameter P9 has the value 26 (programme axis 6=A of group 3 assigned to physical address 8) in the third axis group, E76005=8 (physical axis 8 assigned to programme axis 6),
- if no word of P9 has the value 13 (programme axis 3=U of group 2 is unassigned) in the second axis group, E73005=-1 (no physical axis assigned to programme axis 3).

Writing to these parameters can assign a different physical axis to a programme axis.

A physical axis must be freed before being assigned to a new programme axis (by assigning the value -1 to the parameter of the programme axis with which the physical axis was originally associated).

Example:
- in an axis group, E75005=2 (physical axis 2 assigned to programme axis 5=W),
- in the same axis group, E74005=8 (physical axis 8 assigned to programme axis 4=V),
- to assign physical axis 8 to programme axis 5 (W):
  - release physical axis 8 from programme axis 4 (E74005=-1),
  - assign physical axis 8 to programme axis 5 (E75005=8).

⚠ CAUTION

A new physical axis assigned to a programme axis must be of the same nature as the previous axis (linear or rotary axis, servo or not, modulo or with limited excursion, etc.).

A reset or end of programme (M02) restores the default conditions specified by machine parameter P9.

6

E7x006: Axis coupling

Read/write parameters.

Parameters used to create or cancel carrier/carried axis pairs.

These parameters apply to primary or secondary linear axes (programme axis number between 0 and 5).

The value «0» assigned to the parameter cancels the primary/secondary axis pairing to which the axis specified belongs and a «1» specifies that the axis is carried.

When the parameter related to one axis is modified, the parameter related to the other axis of the primary/secondary axis pair is automatically modified.

A reset reconfigures the axis pairs as specified by machine parameter P64.

Example:

E70006 = 0 Axes X and U are independent
E72006 = 1 Axes Z and W are coupled.

E7x007: Axes programmed on diameter

Read-only parameters.

Only parameters E70007 (X axis) and E73007 (U axis) can be programmed on diameter.

«0» means that these parameters are programmed on radius.
«1» means that they are programmed on diameter.

E7x100: Interpolator position references

Read-only parameters.

This parameter contains the position references from the interpolators. Interaxis calibration uses these references after transformation for computing corrections.

REMARK *The interpolated references E7x100 (with or without transformation) are copied into the references used in servo-controls (E7x000) only for the axes declared in machine parameters P2 and P3 (possibly modified by E9100x).*

E7x101: Interpolation speed limiting

Read/write parameters.

Write of this parameter limits the interpolation speed on logical axis x in the group where it is programmed according to the relation:

$$V\ ipo\ (x) = P30 * \frac{E7x101}{100}$$

However, the maximum speed on the axis is not modified.

The limit value accepted is between 0 and 100 (percent) (rounded off to the nearest integer).

These parameters are forced to 100 percent at power on. They are all reset to 100 percent after a reset.

Possible uses of this parameter:
- Typical use: reserves part of the speed for geometric transformations carried out downstream of the interpolation, for instance with RTCP with or without axis driven in NM_AUTO mode (see supplementary programming manual).
- Special use with E7x101 = 0: axis x is assigned a zero speed:
  - if it is programmed alone in a block, it is ignored
  - if it is programmed with other axes, only the movements on the other axes are carried out.

It should be noted that:
- programming others than of that a CNC axis group or programming of an non-existent axis returns error message 91
- if the programmed value is negative or above 100, the system returns error message 92.

6

6.2.9.6 Spindle Access Parameters

E79000 and E79001

Read-only parameters (except for the dynamic operators).

E79000: Reference position of the spindle whose measurement is used by the group to which it belongs

Position in ten-thousandths of a degree.

When the position is accessed by dynamic operators, it is expressed as 1/4096 revolution modulo 2^{16}.

E79001: Speed demand for the spindle controlled by its group

The value is: (speed demand/max speed of the range) x 2^{15}.

In «normal» parametric programming, this parameter is a read-only parameter and gives the value programmed by «S» as a function of the spindle speed range, but not the value calculated by the dynamic operators.

The reference value is encoded on 15 bits + sign bit.

⚠ CAUTION

When this parameter is programmed by dynamic operators, the sign bit is used by the system only in state M05. Otherwise, it is forced to insure the direction of rotation programmed by M03 or M04.

The spindle potentiometer is not active in this case.

E9010x, E9011x and E9020x

Read-only parameters (except by dynamic operators) where «x» is the spindle number (0 to 3).

These parameters allow any axis group to access any spindle.

E9010x: Spindle x position reference

(Same format as parameter E79000).

E9011x: Spindle x modulo

For additional information, see machine parameter P40.

E9020x: Spindle x speed setting

(Same format as parameter E79001).

E9030x to E9033x

Read/write parameters where «x» is the spindle number (0 to 3).

The new values set for these parameters are saved until the next NC reset.

E9030x: Maximum spindle x indexing speed

Speed in rev/min (see machine parameter P43).

E9031x: Spindle x in-position window

Value expressed in internal units (see machine parameters P40 and P44).

E9032x: Spindle x indexing gain

Value expressed in rev/min/rev (see machine parameter P45).

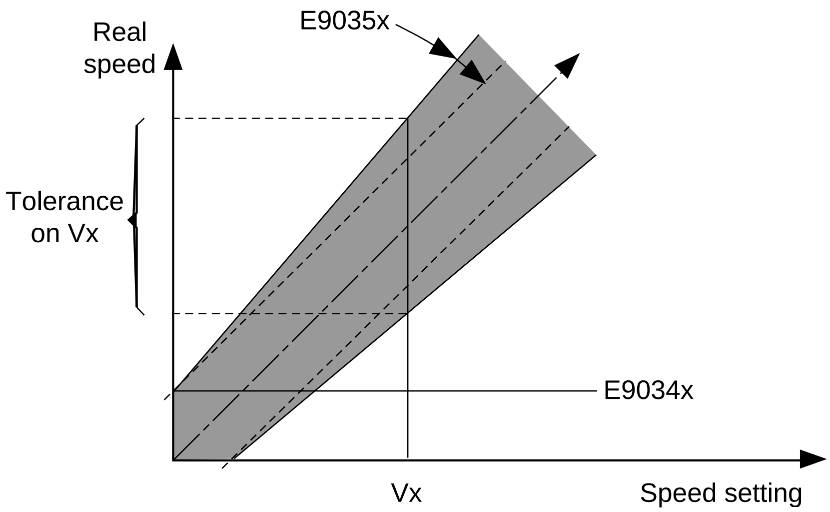
E9033x: Spindle x indexing acceleration

Value expressed in rev/s^2 (see machine parameter P32).

E9034x to E9035x

Read/write parameters where «x» is the spindle number (0 to 3).

These parameters are used to make sure the difference between the spindle speed set and the real speed remains within a tolerance interval and also, where required, to detect nonrotation of the spindle when the speed is below a minimum threshold.



E9034x: Speed threshold below which spindle x is considered stopped

Parameter defining the speed threshold below which the spindle is considered to be stopped. This parameter is expressed in rpm. It is initialised with 10 at power on and can be modified by programming. This parameter is also used as tolerance threshold and detection of the speed reached.

E9035x: Tolerance interval coefficient

Parameter defining the tolerance interval coefficient as a function of the speed. This parameter expressed as 1/256 is initialised with 13 (i.e. 5 percent) at power on and can be modified by programming.

REMARK *In PLC variables %R12.W, bits %R12.0 to %R12.3 (spindles 0 to 3) indicate to the PLC that spindle rotation is correct and bits %R12.4 to %R12.7 indicate that the spindle is stopped (i.e. at a speed below the value declared in E9034x).*

6.2.9.7 Common NC Parameters

E80000 to E80050

Local read/write data parameters.

E8x000 to E8x999

Local read/write data parameters reserved for the system use.

These parameters are related to the inter-axis calibration (see installation and commissioning manual).

E81xxx: Master axis position reference

E82xxx: Slave axis corrections with respect to the master axes

6.2.9.8 Machine Axis Access Parameters

E9yyxx

Parameters where xx represent the physical address of the axis (0 to 31).

An operation on these parameters stops movement at the end of the previous block.

E900xx, E910xx, E911xx and E920xx

Read/write parameters.

E900xx: Axis measurement

Read of measured or servo axes.

Write is not possible for servo axes.

E910xx: Servo or non-servo status of the axes

These parameters reflect the servo or non-servo status of an axis (a physical axis is declared servo or non-servo by machine parameter P3) and are used to modify this state.

A «1» in this parameter indicates that the corresponding axis is servo controlled and a «0» that it is non-servo.

A reset or end of programme (M02) reconfigures the axes as specified by machine parameter P3.

E911xx: Homing status on an axis

These parameters indicate whether the axis homing procedure has been done.

A «1» indicates that a HOME has not been completed on the axis and a«0» that it has been completed.

This parameter addresses the axes and spindles, e.g.:

If spindle 1 is measured, programming E91124 = 1 forces a NOT HOMED status for the spindle. Since homing is automatic on spindles, this makes it possible to reset the spindle position reference.

E913xx: Clampable axis enable state

This parameter enables or disables axis clamping by programming.

Example:
- E91308 = 1: axis 08 can be clamped
- E91308 = 0: axis 08 cannot be clamped.

A reset applies the settings of machine parameter P8 to the axes. The list of axes that can be clamped is sent to the PLC (by PLC variable %R24.L).

E920xx: Axis origin switch enable state

These parameters reflect the state of bit 14 of the axis port command word.

A «1» indicates that the switch is enabled and a «0» that it is inhibited.

The state of the control word can be read on all the measured axes of the machine.

Writing to this control word is possible only for non-servo axes.

A reset, end of programme (M02) or contact with the measurement sensor zero automatically cancels an enabled status of the switch.

In HOME mode, it takes on a value of «1» when the axis jog is engaged and «0» when it is released or when the switch is encountered.

E930xx to E936xx

Read-only parameters.

E930xx: State of the origin switches

These parameters reflect the state of bit 14 of the axis port status word.

A «1» indicates that the switch is active and the «0» that it is idle.

E931xx: Measured axis

A «1» indicates that the axis is measured.

This parameter is the image of machine parameter P2.

E932xx: Axis declared as rotary modulo 360 degrees

A «1» indicates that the axis is declared rotary modulo 360 degrees.

This parameter is the image of machine parameter P1 (word 1).

E933xx: Axis homing direction

A «1» indicates that the axis is moving in the negative direction in homing mode.

This parameter is the image of machine parameter P15 (word 1).

E934xx: Homing status without switch wiring

A «1» indicates that the origin switch is not wired on this axis.

This parameter is the image of machine parameter P15 (word 2).

E935xx: Axis (or spindle) in-position window

This parameter is used to define the position of an axis or a spindle. Its state is the image of the bits of %R6.B to %R9.B transmitted to the PLC. However, if physical axis numbers 24 to 27 correspond to spindles, the associated parameters, E93524 to E93527, indicate the spindle in-position status (spindle indexed by M19 or synchronised spindle) and are the image of the bits of %R13.B transmitted to the PLC.

E936xx: Measurement encoder type

Parameter defining the measurement encoder type. It is the image of bits 25 and 26 of machine parameter P34.

Values that can be assigned to the parameter:
- 0: incremental encoder
- 1: absolute encoder
- 2: combined encoder
- 3: ruler with encoded reference marks.

E940xx and E942xx

Read/write parameters.

E940xx: Assignment of a slave axis (or spindle) to a master axis (or spindle)

These parameters are related to inter-axis calibration (see installation and commissioning manual).

E941xx: Association of a slave axis (or spindle) with a master axis (or spindle)

The axes must be associated before slaving an axis to a master axis for duplication or synchronisation (see duplicated and synchronised axes manual).

The associations can be defined by machine parameter P27 (see parameter manual) or in the part programme, e.g.:

E941xx = m	The slave axis is denoted «x» and the master axis «m». If «m» = -1, the association of slave axis «x» with master axis «m» is cancelled.

In the current position coordinate page (AXIS) for an axis group, pressing the continuation «.../..» key calls a new page on which are displayed the slave axes whose master axes belong to this group. These axes are identified by the symbolic name of the master axis (X, Y, etc.) followed by their own physical address.

E942xx: Axis/spindle switching

Parameter used with programmed association or reassociation of an axis or spindle measurement system with a motor reference output (dac) located at a different address.

Example: E942xx = yy.

This setting associates motor reference output xx with axis or spindle measurement system at address yy.

In this case:
xx (00 to 31): physical address of the axis or spindle motor reference output
yy (00 to 31): physical address of the measurement system.

If addresses xx and/or yy are not recognised, the system returns error message 92.

Restriction: reassociation is impossible and inoperative on QVN axes.

Parameters E942xx are saved after a reset.

Typical example:

A spindle (@24) can be used as C axis (@4). The spindle and the C axis are driven by the same motor, with different encoders.

```
...
(USE AS SPINDLE)
E94224=24
(by precaution, programme:)
E91004=0 (not slaved)
...
...
(USE AS C AXIS)
E94224=04
(by example, programme:)
E91004=1 (C slaved)
E91104=0 (HOMING completed)...
...
```

E950xx to E952xx

6

Read-only parameters.

E950xx: Axis reference offset

These parameters can only be written by interaxis calibration (see Installation and Commissioning Manual) or by the dynamic operators (see dynamic operators manual).

E951xx: Position of the origin switch with respect to the machine origin

This parameter is the image of machine parameter P16.

E952xx: Axis measurement correction (or spindle)

E960xx to E963xx

Read/write parameters used for duplication or synchronisation (see duplicated and synchronised axes manual).

E960xx: duplicated axis in automatic modes

If this parameter is a «1», the axis is duplicated in the automatic modes. If it is a «0», it is not.

E961xx: duplicated axis in manual mode (JOG)

If this parameter is a «1», the axis is duplicated in manual mode (JOG). If it is a «0», it is not.

E962xx: synchronised axis

If this parameter is a «1», the axis is synchronised. If it is a «0», it is not.

E963xx: symmetrically driven axis

If this parameter is a «1», the axis is driven symmetrically. If it is a «0», it is not.

E970xx to E973xx

Read-only parameter.

E970xx: Maximum axis speed

Maximum speed in mm/min or deg/min.

E971xx: Axis acceleration at the work rate

Acceleration expressed in mm/s^2 or deg/s^2.

E972xx: Axis acceleration at high speed

Acceleration expressed in mm/s^2 or deg/s^2.

E973xx: Maximum authorised speed at angle crossings

This parameter is the image of machine parameter P33. Its value is specified in mm/minute.

E980xx to E983xx

Read-write parameters

E980xx: Axis servo-control coefficient

In the position servo-loop, this parameter is the proportional action coefficient applied to the axis following error to obtain its speed reference. This parameter is the image of machine parameter P21. It is expressed in 0.001 mm or degrees.

The value written is immediately applied to the servo-loop. However, after a RE_INIT (power up), the value of machine parameter P21 is reused.

E981xx: Axis acceleration anticipation time constant

When machining at very high speed, this parameter is used to programme the axis acceleration anticipation. After an initialisation, the value of machine parameter P19 is reused.

E982xx: Amplitude of the antistick pulse on reversal

This parameter is the image of machine parameter P19 (words 32 to 63). It is expressed in micrometres.

E983xx: Time constant to absorb the antistick pulse

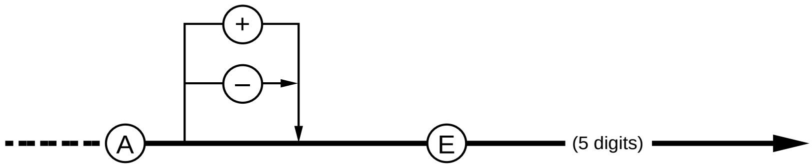
This parameter is the image of machine parameter P19 (words 64 to 95). Its value is expressed in 1/100000 s.

6.2.10 External E Parameter Programming Syntax

The programming syntax for external E parameters is given as Conway diagrams followed by programming examples.

6.2.10.1 Assignment of an External Parameter to an NC Function

Syntax



A	NC function.
+ / -	Sign.
E	External parameter used as numerical value

Notes

External E parameters have integer values. When they are assigned to a function with a decimal value, the decimal point is implicit and depends on the format of the function.

When the value of the parameter is not compatible with the format of the function (too many digits, etc.), the system reports an error.

Example

Use of an external parameter with NC addresses in different units.

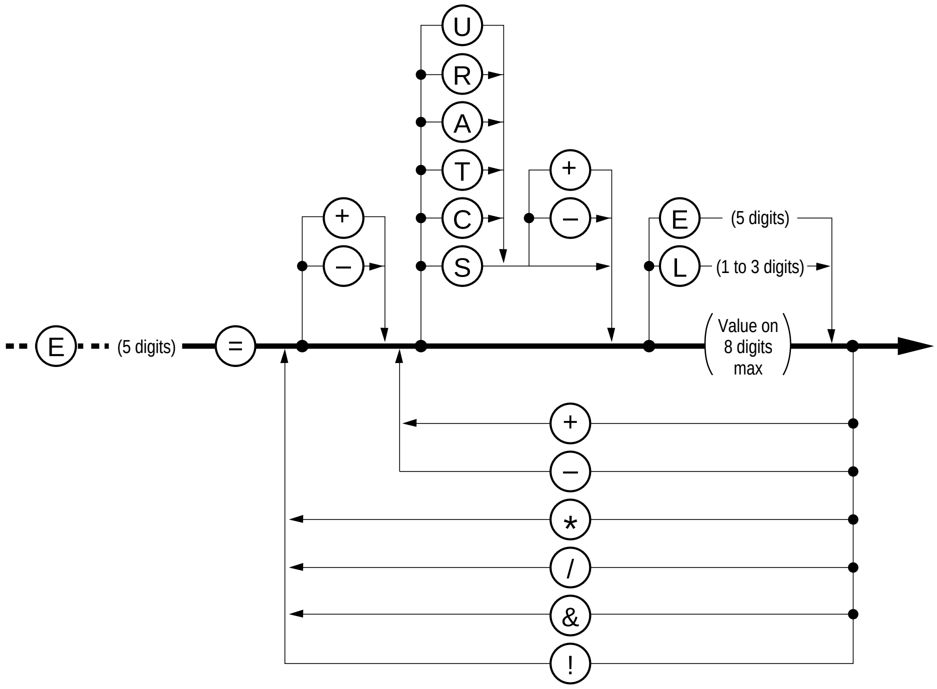
Assignment of external parameter E80000 to addresses X and C.

`N..`	
`E80000 = 18000`	Declaration of the value of the parameter
`G00 XE80000`	The value of E80000 is equivalent to 18 millimetres (internal unit = µm)
`G00 CE80000`	The value of E80000 is equivalent to 1.8 degrees (format C034, i.e. ten-thousandths of a degree)

It should be noted that if the feed rate F is programmed as FE80000 (example), the feed rate F takes on the integer value declared in parameter E (in this case, the decimal digits are excluded with F).

6.2.10.2 Declaration of an External Parameter in the Programme

Syntax



The operations in this diagram are described in detail in Sec. 6.2.6.

Notes

When an operation result that is not an integer is assigned to an external parameter, the decimal digits are truncated. To keep a result with decimal digits, this result must be assigned to a variable L (see Sec. 6.1.10.2).

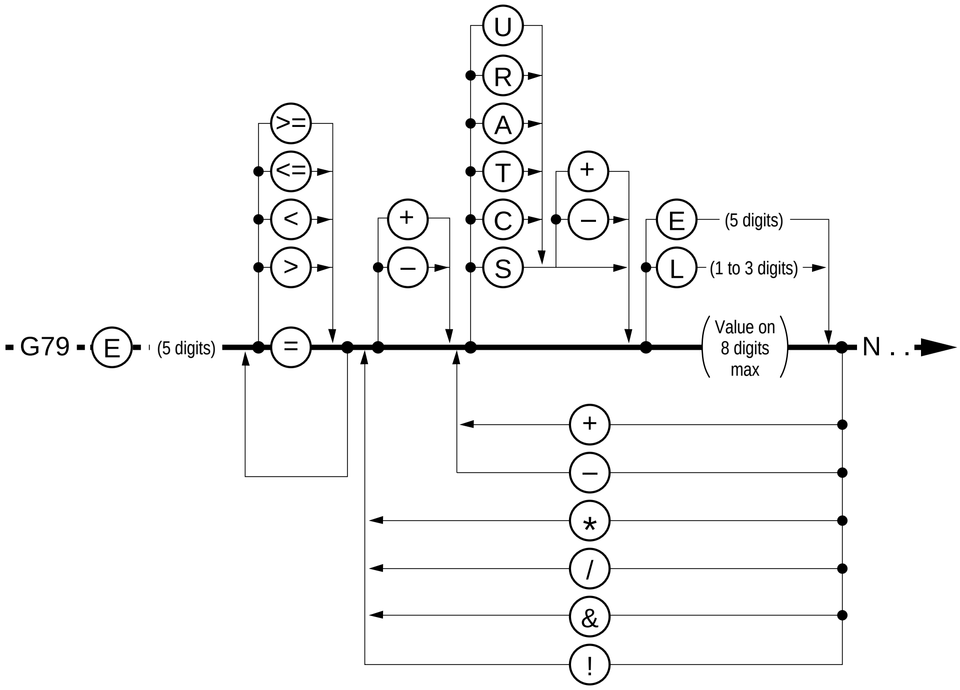
Example

Use of external parameters with arithmetic operations.

N..	
E80002 = 3150	Declaration of the value of E80002
E80016 = 2400	Declaration of the value of E80016
E80005 = E80002 / E80016	After the operation, E80005 takes on the value 1 (truncation)
L1 = E80002 / E80016	After the operation, L1 takes on the value 1.3125
N..	

6.2.10.3 Test on an External Parameter for a Conditional Branch

Syntax



REMARK *For tests on fractions, see Sec. 6.8.2.*

Example

Use of an external parameter with a conditional test.

`N.. ...`	
`N50 E56003 = E56003+1`	Increment of the parameter
`G79 E56003 = > 10 N50`	Condition: if E56003 ≥ 10, branch to N50, else continue
`N..`	

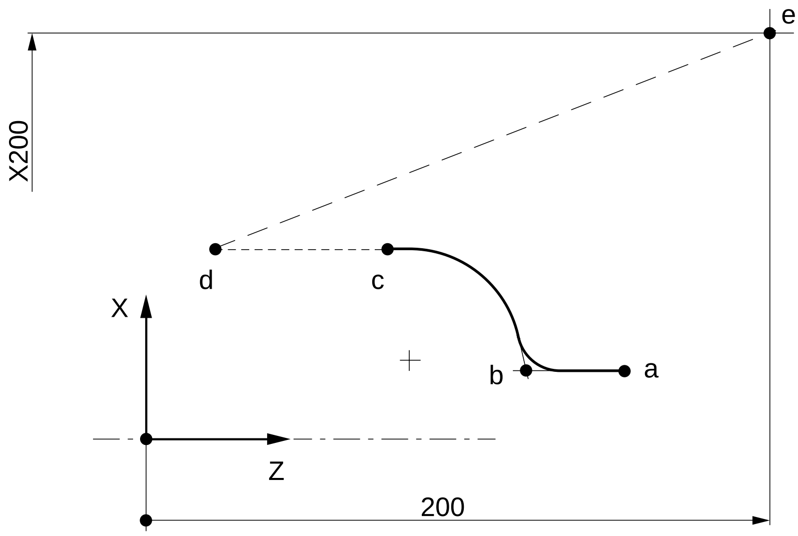
6.2.10.4 Examples of Use of External E Parameters

Examples

Use of the external parameters related to the choice of lines for graphic display (E31000 and E31001)

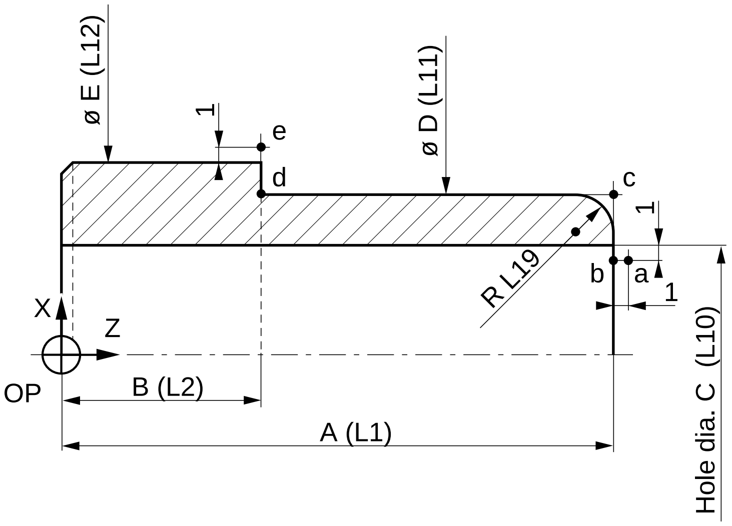
Drawing of paths using different types of lines.

```
%15
N10 E31000=4 G00 X200 Z200
N20 X50 Z150                        Point a, no trace (stylus raised)
N30 E31001=0 G01 Z125 ES+ EB10      Point b, solid line
N40 G03 X110 Z90 I75 K90 R35 ET     Point c
N50 E31001=1 G01 EA180 Z20          Point d, dotted line
N60 E31001=2 G00 X200 Z200          Point e, dashed line
N70 ...
```

Use of external parameters related to DAT1 (E6x000), tool correction (E50xxx, E51xxx, E52xxx, E55xxx) and the standard NC parameter E80000.

Execution of one of the three types of parts proposed by the programme. The part number is selected by the operator following the message in block N10 (see Sec. 6.7).



Assignment of variables

Variables	L1	L2	L10	L11	L12	L19

Code	A	B	C	D	E	Radius
Part 1	70	25	30	40	50	4
Part 2	65	30	25	38	48	3
Part 3	60	35	20	36	46	2

Main Programme

```
%100
E60000=-240000 (DAT1 ON X)
E62000=-458678 (DAT1 ON Z)
E50003=78500 (DIMENSION X) E51003=53470 (DIMENSION Z)
E52003=800 (RADIUS 0.8) E55003=1 (ORIENTATION C1)
N10 $0 ENTER PART NUMBER (1, 2 OR 3)
E80000=$0
N20 G79 E80000=1 N50
N30 G79 E80000=2 N70
N40 G79 E80000=3 N90
G79 N10                                  Return to N10 if 1, 2 or 3 not selected
N50 L1=70 L2=25 L10=30 L11=40 L12=50 L19=4 $PART 1
N60 G79 N100
N70 L1=65 L2=30 L10=25 L11=38 L12=48 L19=3 $PART 2
N80 G79 N100
N90 L1=60 L2=35 L10=20 L11=36 L12=46 L19=2 $PART 3
N100 G77 H101
N120 M02
```

Subroutine

```
%101
N10 G00 G52 X-50 Z-100
L0=E52003/1000
N20 T03 D03 M06 (EXTERNAL TOOL R0.8)
N30 S900 M40 M04
L3=-L0+1*2+L10 (Xa) L13=L1+L0+1 (Za)
N40 G42 XL3 ZL13                         Point a
N50 G92 S3500
N60 G96 S250
N70 G95 F0.2
N80 G01 ZL1                              Point b
N90 XL11 EBL19                           Point c
N100 ZL2                                 Point d
L4=L0+1*2+L12 (Xe)
N110 XL4                                 Point e
N120 G00 G40 G52 X0 Z0 G97 S900 M05
```

6

Use of the external parameters related to the machine axes (E90xxx and E93xxx)

Automatic HOME locating program on two axes: X (axis 0), Z (axis 1), zero ORPOM.

The programme below is given for reference and must be adapted to the features of the particular machine.

For use of the axis address equivalence table (see Sec. 6.3).

%9990	
G79 N100	Branch to N100
N10	
L1=90000+L0	E900xx: measurement of the axis
L2=91000+L0	E910xx: slave axis
L3=92000+L0	E920xx: origin switch enable
L4=93000+L0	E930xx: state of the origin switch
L6=91100+0	Homing status
EL6=1	Homing not performed
EL2=0 EL1=-1000 EL2=1 L5=EL1/1000	Initialisation of the measurement with -1
N20 G79 EL4=0 N30	Branch to N30 if the axis is not on the switch
G52 G00 L5=L5-1 @XL5 G79 N20	Back off the switch by 1 mm, sign depending on the HOME direction
N30 EL6=1 EL2=0 EL1=-50000000 EL3=1 EL2=1	Measurement initialisation at 50 m, sign depending on the HOME direction
G01 G52 @X0 G10 @L0>0 N40	Movement to the measurement origin until reaching a value of zero (if ORPOM is non-zero , take this into account in the comparison)
N40	
N100 G94 F1000	
@X=X L0=0 G77 N10 N40 @X=Z L0=1 G77 N10 N40	Definition of address equivalences: @X: axis name, L0: axis number
N110 M02	

Use of the external parameters related to the current mode numbers (E41000) and machine axes (E90xxx to E93xxx and E911xx)

The programme below is given for reference and must be adapted to the features of the particular machine.
Automatic HOME locating programme for the C axis (physical address 3), zero ORPOM.

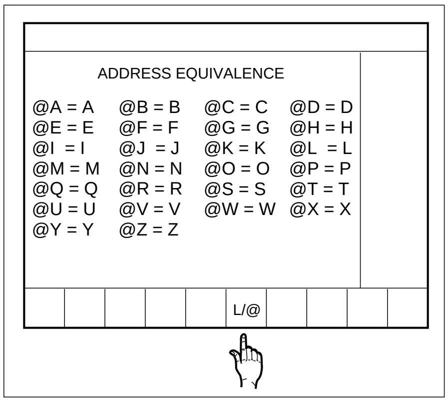
`%1000`	
`N10 S100 M03 M40`	
`N20 M997`	
`N30 G79 E41000=4 N140`	=4: Rapid mode
`N40 G79 E41000=6 N140`	=6: Test mode
`N50 G04 F1`	
`N60 M..`	Encoded M functions defined by the machine manufacturer
`N70 G04 F1`	
`N80 M64 M05`	
`N90 E91103=1`	Homing not performed
`N100 E91003=0 E92003=1`	E910xx: Axis non-servo
`E90003=990000 E91003=1`	E920xx: Enable HOME switch E900xx: Overwrite axis pos'n E910xx: Axis servo
`N110 L5=E90003/10000 +180`	E900xx: Read back pos'n + 180
`N120 G94 F200`	
`N130 G01 G52 CL5 G10 @3<50 N150`	Movement to HOME (if ORPOM was non-zero, take it into account in the comparison)
`N140 G79 N100`	
`N150`	

Programme %1001 is used to make the C axis non-servo (spindle)

`%1001`	
`N10 M997`	
`N20 G79 E41000=4 N50`	
`N30 G79 E41000=6 N50`	
`N40 M..`	Encoded M function defined by the machine manufacturer
`N50 M998`	

6.3 Address Equivalences

This table is used to specify the equivalence between two axes. For instance, it allows a programme to be written for another axis group, such as UW.



The symbol @ followed by an address (A to Z) specifies an equivalent axis.

The declaration of an equivalent address is programmed by:

$$@\begin{bmatrix}A\\ ''\\ ''\\ Z\end{bmatrix} = \begin{bmatrix}A\\ ''\\ ''\\ Z\end{bmatrix}$$

The assignment of a value to an equivalent function is programmed by:

$@\begin{bmatrix}A\\ ''\\ ''\\ Z\end{bmatrix}$ followed by the value

Example :

```
@ X = 300
```
@ X = 300 is equal to U 300

The address equivalence table is reset at power on, by a reset or by M02 (@A = A, @ B = B, ..., @ Z = Z).

The declaration of a new equivalent address ($@\begin{bmatrix}A\\ ''\\ ''\\ Z\end{bmatrix} = \begin{bmatrix}A\\ ''\\ ''\\ Z\end{bmatrix}$) suspends preparation of the block in which it is programmed until execution of the previous block is completed.

The address equivalence table is displayed by pressing the «Programme variable» key L/@ (see example in Sec. 6.2).

6.4 Transfer of the Current Values of L Variables and E Parameters into the Part Programme

G76 Transfer of the current values of L variables and E parameters into the specified programme or programme section.

This function is used to update the contents of a file called by addresses H.. and/or N.. N..

The file of L variables and E parameters is updated with the new contents of the corresponding active data.

Syntax

N.. **G76** [H..] [N.. N..]

G76	Transfer of the current values of L variables and E parameters into the programme specified.
H..	Specification of the programme into which the values are transferred.
N.. N..	Specification of the programme section into which the values are transferred.

Notes

The parameters to be transferred must be located at the beginning of blocks: L variables and E parameters located after another function in a block are ignored.

The specification of an L variable or E parameter must be followed by the sign «=» and at least ten characters (spaces, algebraic sign, numbers, decimal point) designed to be replaced by a new value.

Failure to comply with this rule returns error message 97.

Examples

<u>Transfer of the values of L and E into the current programme</u>

In the example below, only parameters L101, E80001, L4 and L6 are modified. The other parameters are ignored.

N..	
N40 G76 N100 N120	
N..	
N100 ..	
L101=.......... E80001=..........	
L4=.......... G04 E52002=..........	E52002 is not modified as it is located after G04
L6=..........	
G01 X100 L3=..........	L3 is not modified
N120	
N..	

<u>Transfer of the values of L and E into a subroutine</u>

%125	
N10 G77 H200 N50 N80	Execution of blocks N50 to N80 of %200
N..	
N..	Execution of the programme after modification of the parameters by %200
N..	
N600 G76 H200 N50 N80	Update of the file
N610 M02	

```
%200
N10 ...
N..
N50
L1=.......... E52002=..........
E80004=..........
N80
N..
```

6.5 Message Display with Wait for an Operator Response

The «$» character can be used as variable in a parametric expression. It causes the programme to wait for an operator entry.

The block containing the parametric expression can be preceded by a message indicating the type of data to be entered.

Syntax

L../E.. = Expression depending on $

L../E..	Programme variable or external parameter.
$	Variable whose value must be entered by the operator.

Cancellation
- end of programme (M02),
- reset.

6

Notes on Operation

Upon reading the «$» character associated with a variable (possibly accompanied by a message with a maximum of 39 characters), the system stops the programme and waits for the operator to type an entry.

The characters typed are echoed on the message line of one of the following pages:
- Information page (INFO.)
- Current position with respect to OP page (AXIS)
- Trace during machining page (TRACE DURING MACHINING).

The entry can be numerical characters or one alphabetic character. Each type of entry corresponds to a specific use.

Numerical entry

The number of digits entered must not exceed eight plus sign and decimal point.

The limit values are as follows:
- Integer: ≤ 99999999 or ≤ -99999999
- Positive decimal: ≤ 0.0000001 or ≤ 9999999.9
- Negative decimal: ≤ -9999999.9 or ≤ -0.0000001.

The operator must confirm the entry by pressing the Enter key on the keyboard (otherwise the system does not proceed to the next block).

REMARK *Before pressing Enter, the characters entered can be deleted by the Delete key.*

Alphabetic entry

Only the first character entered is analysed.

Any capital letter from A to Z is accepted. Each letter returns a value:
A returns 1
B returns 2
and so forth to Z = 26.

REMARK *The first character entered is analysed by the system which automatically proceeds to the next blocks (it is not necessary to confirm the entry by Enter).*

It should be noted that in a test programme, each numerical value from 1 to 26 corresponds to alphabetic characters A to Z.

Example

```
N.. ...
N50 E80000=50
$0 ENTER THE VALUE OF X =        Display of the message for the operator
N60 L0=E80000 + $                Wait for entry of the value
N..
```

6.6 Display of Messages with Parametric Value

The «$» character can be used as first term of a parametric expression whose value is displayed after being calculated by the system.

The block containing the parametric expression can be preceded by a message indicating the name of the variable displayed.

Syntax

$ = expression

$	Variable whose value is displayed.

Cancellation
- end of programme (M02),
- reset.

Notes

An expression «$=...» in a block after the branch function (G79) cannot be used as a branch condition.

Example

`N..`	
`L0=0`	Initialisation of variable L0
`N90 L0=LO+1`	Increment of L0
`$0 EXECUTION OF SUBROUTINE %`	Display of the first part of the message
`$= L0`	Display of the value following the previous message
`G77 HL0`	Call to subroutine %(L0)
`G79 L0<5 N90`	Condition: if L0 < 5, branch to block N90, else continue
`N..`	

6.7 Reading the Programme Status Access Symbols

The programme status access symbols allow all modal functions to be read. They can be used to save the programme context when a subroutine is called. The context can then be restored when execution of the subroutine is finished.

These read-only symbols are accessible by parametric programming.

These symbols can be:
- symbols to access the data in the current block
- symbols to access the data in the previous block.

The same data are accessible in the previous block as in the current block. The symbols are the same, but are preceded by two dots (instead of only one).

The only reason for accessing the data in the previous block is when execution of the current block is suspended by programming of function G999. The data are those of the last executable block (or possible the last block already executed).

6.7.1 Symbols for Accessing the Data of the Current or Previous Block

The symbols can be:
- symbols addressing Boolean values
- symbols addressing numerical values.

General Syntax

Variable = [. symbol] or [.. symbol]

Variable	L programme variable, symbolic variable [symb], E parameter.
[. symbol]	Data of the current block. Symbol between square brackets, preceded by a dot.
[.. symbol]	Data of the previous block. Symbol between square brackets, preceded by two dots.

6.7.1.1 Symbols Addressing Boolean Values

The symbols addressing Boolean values associated with programmed functions are used to determine whether the functions are active or not.

The Boolean values are defined by 0 or 1.

Addressing G Functions

[.BGxx]	G function addressing.

The symbol [.BGxx] is used to determine whether the G functions specified by xx are enabled or inhibited, e.g.:

[.BGxx]=0: function Gxx inhibited

[.BGxx]=1: function Gxx enabled

List of G Functions

[.BG00]	[.BG01]	[.BG02]	[.BG03]	[.BG17]	[.BG18]	[.BG19]
[.BG20]	[.BG21]	[.BG22]	[.BG29]	[.BG40]	[.BG41]	[.BG42]
[.BG43]	[.BG70]	[.BG71]	[.BG80]	[.BG81]	[.BG82]	[.BG83]
[.BG84]	[.BG85]	[.BG86]	[.BG87]	[.BG88]	[.BG89]	[.BG90]
[.BG91]	[.BG93]	[.BG94]	[.BG95]	[.BG96]	[.BG97]	

Addressing M Functions

[.BMxx]	M function addressing.

The symbol [.BMxx] is used to determine whether the M functions specified by xx are enabled or inhibited, e.g.:

[.BMxx]=0: function Mxx inhibited

[.BMxx]=1: function Mxx enabled

List of M Functions

[.BM03]	[.BM04]	[.BM05]	[.BM07]	[.BM08]	[.BM09]	[.BM10]	[.BM11]
[.BM19]	[.BM40]	[.BM41]	[.BM42]	[.BM43]	[.BM44]	[.BM45]	[.BM48]
[.BM49]	[.BM64]	[.BM65]	[.BM66]	[.BM67]	[.BM997]	[.BM998]	[.BM999]

6.7.1.2 Symbols Addressing Numerical Values

The symbols addressing numerical values are used to read the modal data of the current block or the previous block.

Addressing a Value

[•Rxx]	Addressing a value.

The symbol [•Rxx] is used to address a value corresponding to the elements specified by xx.

[•RF]	Feed rate (units as programmed by G93, G94 or G95).
[•RS]	Spindle speed (G97: format according the spindle characteristics declared in machine parameter P7).
[•RT]	Tool number.
[•RD]	Tool correction number.
[•RN]	Number of the last sequence (block) encountered. If the block number is not specified, the last numbered block is analysed.
[•RED]	Angular offset.
[•REC]	Spindle orientation (milling).
[•RG4]	Programmed dwell (G04 F..). This function, although nonmodal, may remain stored. Its value can therefore be read if the system is in state G999 of G998.
[•RG80]	Number of the G function branching to the subroutine. In a subroutine called by a G function, the number of the calling function is addressed by [•RG80]; in state G80, its value is zero.
[•RNC]	Value of NC (spline curve number).
[•RDX]	Tool axis orientation. Defined by the following signs and values: = +1 for G16 P+ or -1 for G16 P- = +2 for G16 Q+ or -2 for G16 Q- = +3 for G16 R+ or -3 for G16 R-
[•RXH]	Nesting level of the current subroutine (up to 8 levels). = 1: main programme = 2: first nesting level = 3: second nesting level, etc.

Example

```
%100
N10 G00 G52 X0 Z0 G71
N..
N40 G97 S1000 M03 M41
N50 M60 G77 H9000                          Tool check subroutine call
N..
N350 M02

%9000
VAR
[GPLANE] [MRANGE] [MDRN] [GINCH] [GABS]    Symbolic variables (see Chapter 7)
[XRETURN] [YRETURN] [ZRETURN]
ENDV

$ CONTEXT SAVE
N10 [GPLANE]=20*[.BG20]
    [GPLANE]=21*[.BG21]+[GPLANE]           Interpolation plane
    [GPLANE]=22*[.BG22]+[GPLANE]
N20 [MRANGE]=40*[.BM40]                    Speed range
    [MRANGE]=41*[.BM41]+[MRANGE]
N30 [MDRN]=03*[.BM03]
    [MDRN]=04*[.BM04]+[MDRN]               Direction of rotation
    [MDRN]=05*[.BM05]+[MDRN]
N40 [GINCH]=70*[.BG70]                     Inches
    [GINCH]=71*[.BG71]+[GINCH]
N50 [GABS]=90*[.BG90]                      Absolute
    [GABS]=91*[.BG91]+[GABS]
[XRETURN]=E70000
[ZRETURN]=E72000
N60 G90 G70 G00 G52 Z0
N70 G52 X-100 Z-100 M05                    Tool check position
N80 G52 G10 Z-500
N90 G52 Z0
N100 G52 Z [ZRETURN]                       Return to Z position
N110 G52 X [XRETURN]
G[GPLANE] M[MRANGE]
M[MDRN] G[GINCH] G[GABS]                   Restore context
```

6

6.8 General Diagrams of Parametric Programming

The following symbols are used in the diagrams below:

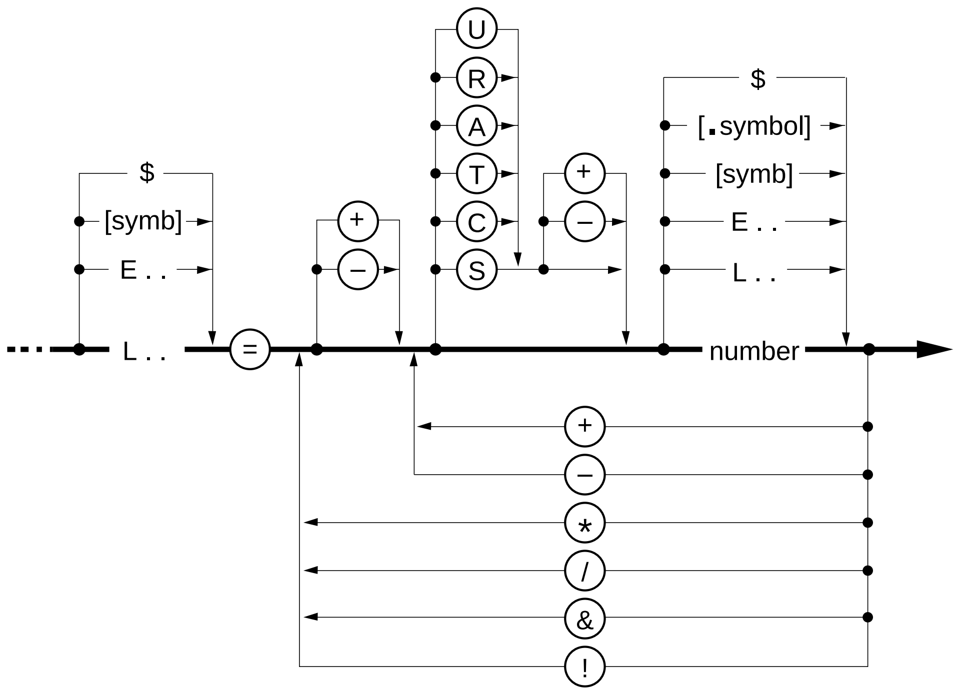
[.symbol] or [..symbol]: Programme status access symbols

[symb]: Symbolic variables

These programming tools are defined in this programming manual (Sections 6.7 and 7.3) and the supplementary programming manual (Chapter 2).

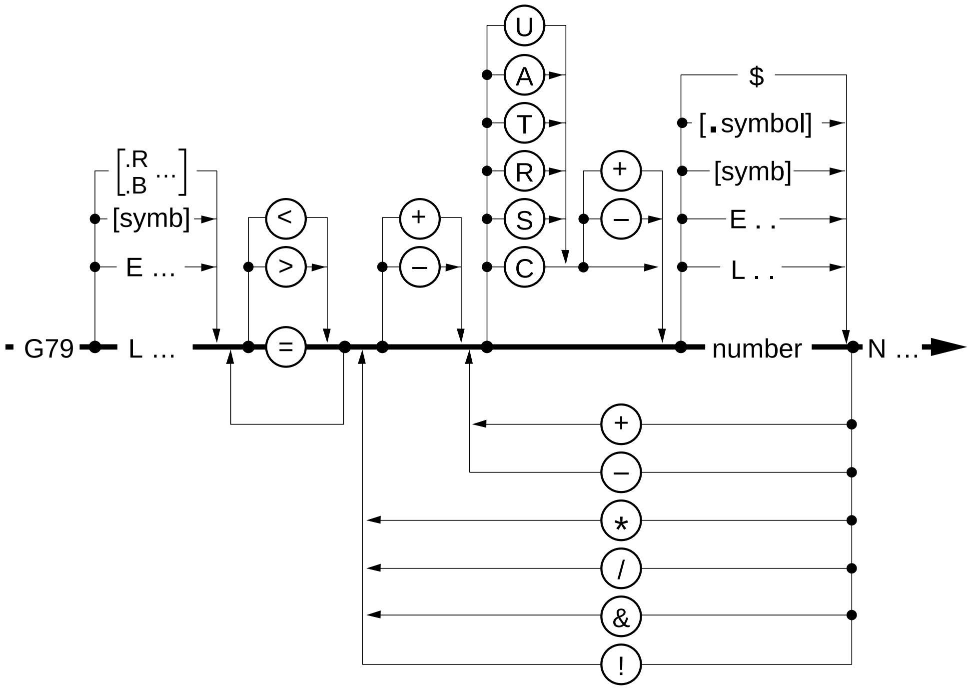
6.8.1 Load of a Parametric Expression

Syntax



6.8.2 Comparison for Conditional Branch

Syntax



Note on > = < Tests on Fractions

In tests using > = < on fractions, the test may return false whereas the system displays a true result. This is especially the case for a loop.

Example:

```
...
N10
...
...
...
L1 = L1 + 0.6 G79 L1 < 6 N10
...
```

It should be noted that after 10 loop iterations, L1 is equal to 6 (on the display) whereas the computation result gives 5.99. An additional iteration is therefore required. Since 0.6 in binary is not exact, there is always a remainder.

In hexadecimal = ,99999999 ... soit 1/2 + 1/16 + 1/256 + 1/512 + 1/4096 + 1/8192 + ...

In practice, for incrementation with fractions, always make the test on a lower value consistent with the incrementation.

Example:
For 0.6, with 10 iterations, test on 5.9
For 0.1, with 10 iterations, test on 0.96
For 0.01, with 10 iterations, test on 0.096.

7 Programme Stack - L Variables and Symbolic Variables

7

7.1 Programme Stack

Each axis group (including graphic axis groups) is allocated a stack located at the bottom of the memory.

7.1.1 Use of the Stack

The stack is used to:
- save and restore L programme variables,
- assign symbolic variables,
- save spline curve coefficients.

7.1.2 Stack Allocation

The stack size is defined by machine parameter P58 (see parameter manual).

7.2 Saving and Restoring L Variables

7.2.1 PUSH Function

PUSH	Saves the values of the L variables on the stack.

Syntax

PUSH Lm - Ln

PUSH	Function forcing save of the L variable values.
Lm - Ln	L variable numbers from m to n (inclusive) whose values are to be saved, i.e.: - 0-19, - 100-199, - 900-959.

Notes

The PUSH function must be the first word in a block (no sequence number).

Saving variables spanning different ranges of numbers by one PUSH is prohibited, e.g.:

Saving the values of L1-L19 and L100-L110

Right:

`PUSH L1 - L19`	Saves the values of L1 to L19
`PUSH L100 - L110`	Saves the values of L100 to L110

Wrong:

```
PUSH L1 - L110
```

7.2.2 PULL Function

PULL	Restores the values of the L variables.

Syntax

PULL Lm - Ln

PULL	Function restoring the values of the L variables.
Lm - Ln	L variable numbers from m to n (inclusive) whose values are restored, i.e.: - 0-19, - 100-199, - 900-959.

Notes

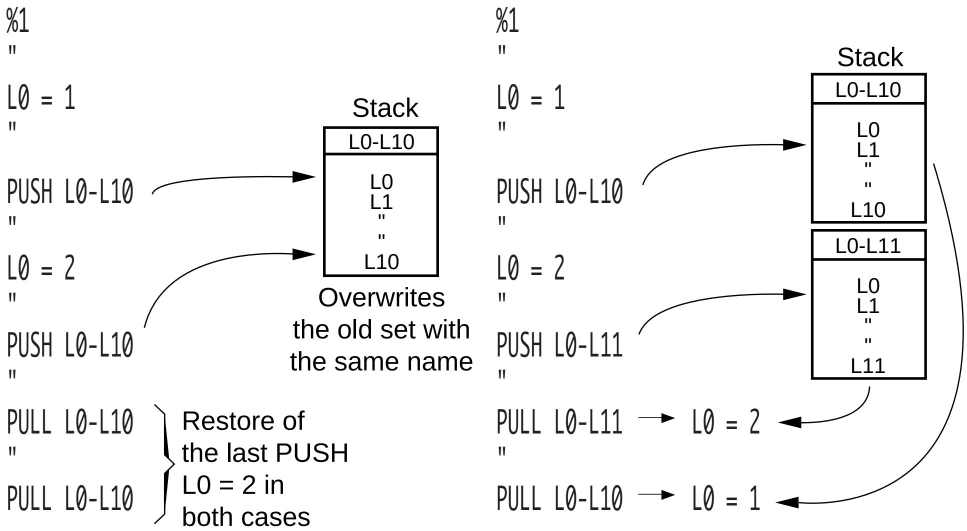
The PULL function must be the first word in a block (no sequence number).

The PULL function:
- does not delete the values from the stack,
- allows variables saved by a PUSH to be restored several times by PULL,
- can cause a stack overflow after a series of PUSH-PULL operations (in this case, the system returns error message 195).

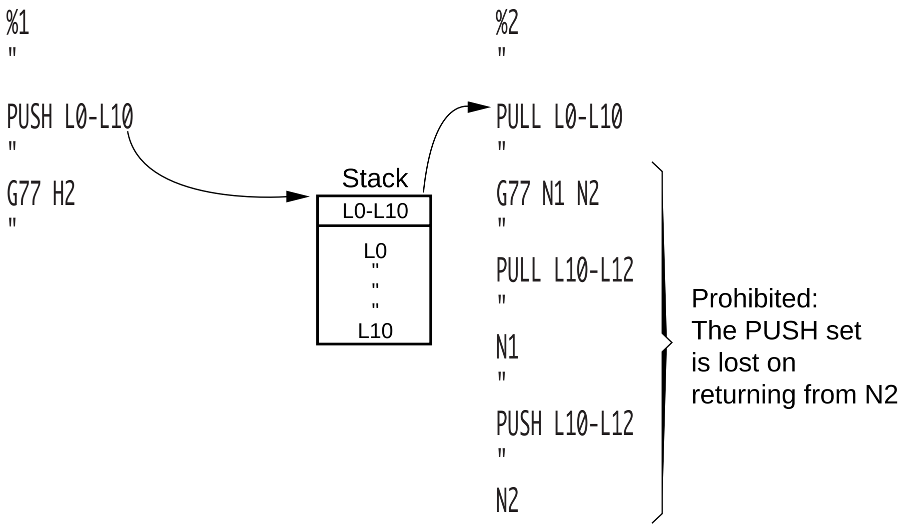
7

Example

Restore by PULL of the values of Lm to Ln saved by the last PUSH on the same set of variables.



A PULL from Lm to Ln restores only the variables saved in the main programme or current subroutine or those from lower nesting levels (to be valid, a PULL must always be preceded by a PUSH).



7.3 Symbolic Variables

Variables declared with symbolic names. The use of symbolic variables allows the number of variables used in parametric programming to be extended.

7.3.1 Declaring Symbolic Variables - VAR and ENDV Functions

Symbolic variables are declared between square brackets and are denoted [symb] in the syntax defined below.

Symbolic variables are always declared between functions VAR and ENDV.

VAR	Symbolic variable declaration.

VAR is the keyword enabling symbolic variable declaration.

ENDV	End of symbolic variable declaration.

ENDV is the keyword defining the end of symbolic variable declaration.

Syntax

VAR [symb]
ENDV

VAR	Symbolic variable declaration.
[symb]	Symbolic variables with up to 8 alphanumeric characters, the first of which is always alphabetic.
ENDV	End of symbolic variable declaration.

7

Notes

Functions VAR and ENDV must be the first words in the block (no sequence number).

VAR must be separated from the symbolic variable(s) by a space.

It should be noted that spaces are recognised as characters when writing a symbolic variable. If a variable name includes a space, the space must always be included in the name entered. Otherwise, the system returns error message 198.

ENDV must be the only word in the block.

Example:

```
VAR [INDEX] [RD12]
    [PHASE2]
ENDV                                    only word in the block
```

7.3.2 Using Symbolic Variables in ISO Programmes

A symbolic variable declaration suspends preparation of the block to which the variables belong until execution of the previous block is finished (according to the same principle as variables L100 to L199; see 6.1, programming notes).

Symbolic variables have real values.

Symbolic variables can be:
- assigned to all the ISO programming functions,
- used in parametric expressions,
- associated where applicable with L programme variables and E external parameters.

The programmer can access only the symbolic variables declared in the main programme or subroutine or those from lower nesting levels.

<u>Programming axes by variable L or parameter E defined by symbolic variable</u>

The system provides the possibility of programming axes by variable L or parameter E defined by symbolic variables.

The number of the variable L or parameter E used is assigned to symbolic variable [symb...]. Example:

```
VAR
[symb1]=80000                           Parameter E number
[symb2]=0                               Variable L number
ENDV
...
...
... XE[symb1]                           X axis programming (XE80000)
... BL[symb2]                           B axis programming (BL0)
```

REMARK *If X[symb2] or XL0 or XL[symb2] is programmed, the value specified by the symbolic variable or variable L is in mm or in degrees if the axis is rotary.*
If XE80000 or XE[symb1] is programmed, the value specified by parameter E (set or not) is defined in the internal system unit (see Sec. 2.1), by default µm or 0.001 degree if the axis is rotary.

Other Uses of Symbolic Variables

Symbolic variables can be used:
- to create tables or arrays,
- with structured branch and loop commands.

For further information, refer to the supplementary programming manual.

Example

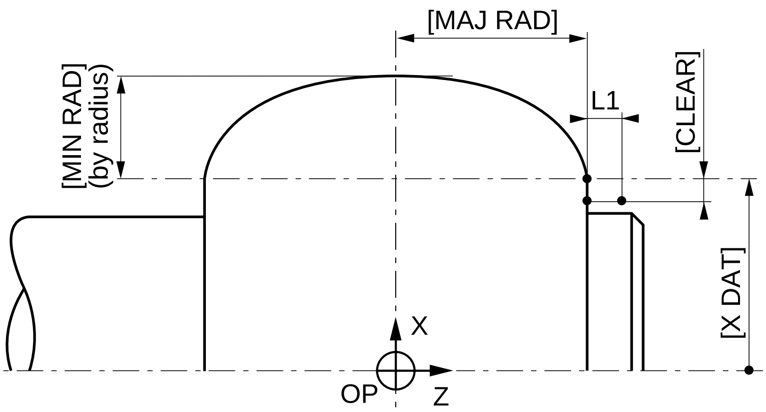
Use of symbolic variables for machining a semielliptical shape

```
%35
M999
VAR
[XSHIFT] = 50  [CLEAR] = 10
[MAJ RAD] = 40 [MIN RAD] = [MAJ RAD]
[SPSPD] = 900  [DRN] = 04  [RANGE] = 40       Variable
[FDRATE] = 0.2  [CSS] = 100                   declaration
[PITCH] = 5  [STRTANG] = 0  [ENDANG] = 180
[COS]  [SIN]
ENDV
L1 = [MAJ RAD] + [CLEAR]
N10 G0 G52 X0 Z0
N20 T1 D1 M06 (BUTTON TOOL R=3)
N30 G97  S [SPSPD]  M [DRN]  M [RANGE]
$ MACHINING
N40 G0 X- [CLEAR] ZL1
N50 G1 G42 Z [MAJ RAD]
N60 G92 S4000
N70 G96 S [CSS]
```

7

```
N80 X0 G95 F [FDRATE]
[SIN] = [MAJ RAD] * S [STRTANG]
[COS] = [MIN RAD] * C [STRTANG]
X [SIN]  Z [COS]
[STRTANG] = [STRTANG] + [PITCH]
G79 [STRTANG] = < [ENDANG] N80 + 1
N90 G1 X- [CLEAR]
N100 Z-L1
N110 G0 G40 XL1
N120 G52 X0 Z0 G97 S900 M05
N130 M02
```

<u>Representation of machining</u>



7.3.3 Acquisition of Variables from the Stack of Another Axis Group: Function VAR H.. N.. N..

VAR H.. N.. N..	Acquisition of variables from the stack of another axis group.

Function VAR H.. N.. N.. is used to read the symbolic variables declared in another programme from the stack of another axis group.

Syntax

VAR H.. :s N.. N.. +i +j

VAR	Acquisition of variables from the programme stack of another axis group.
H..	Number of the programme or subroutine from where the symbolic variables are to be fetched.
:s	Number «s» of the memory area where programme H.. is located (s can have any value between 0 and 3 after the character :) Argument :s is optional. (see details below on :s)
N.. N..	Numbers of the first and last block defining the limit between which the symbolic variables are located. (see details below on N.. N..)
+i +j	Offset with respect to blocks N.. and N.. +i with respect to the first block, +j with respect to the last block.

Notes

When a variable acquired by VAR N.. N.. has the same name as a variable already declared in the current programme or subroutine, the current one's value is overwritten.

7

Details on Area Number :s

An independent axis group (PLC group) whose programme is located in a protected memory area (area 1, 2 or 3) can read variables located in area 0 if the NC is not in an edit mode. Otherwise, execution of the independent axis group programme is suspended until the NC leaves the edit mode.

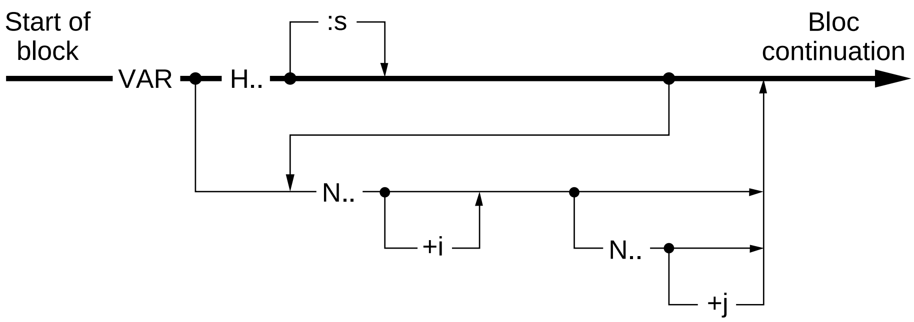
If :s is not specified, the programme is searched for:
- from area 0 to area 3 for the NC groups,
- from area 1 to area 3 for the independent groups.

Details on Blocks N.. N..

If programme number H.. is not specified, the limits from N.. to N.. are searched for in the current programme or subroutine.

If the second block number N.. is not specified, only the variables in block N.. are read.

Diagram of the Syntax



Example

%1	%2
N10 ...	N10
N..	N..
VAR H2 N50 N60 L0=[I] L1=[T(1,3)]	VAR
N..	N50 [I] = 3 [J] = 4
	[T(2,3)] = 3 , 4
	= 5 , 6
	N60 = 7 , 8
	ENDV

7.3.4 Deleting Symbolic Variables from the Stack

The variables are not deleted until the block preceding the deletion command has been executed.

7.3.4.1 Automatic Deletion of Symbolic Variables

A return from a subroutine deletes all the variables declared in the subroutine and releases the space they occupied on the stack.

The end of programme command (M02) or an operator reset deletes all the variables and reinitialises the stack.

7.3.4.2 Programmed Deletion of Variables: DELETE Function

DELETE	Deletion of symbolic variables from the programme or subroutine. Deletes the global variables from the stack.

Syntaxe

DELETE * [symb1]/[symb2]…

DELETE	Deletes the symbolic variables from the programme or subroutine. Deletion of the global variables from the stack.
*****	When the variables names are preceded by the character *****, the variables to be deleted are searched for in the entire stack. When the variable names are not preceded by the character *****, the search for the variables to be deleted is limited to the variables declared in the main programme or subroutine.
[symb1]/[symb2]…	Variables to be deleted.

7

Notes

The DELETE function:
- must be the first word in the block (no sequence number),
- must be followed by at least one space but there must be no spaces in the list of variables,
- is followed by the list of variables and arrays to be deleted; the variables are separated by the character /.

Only the first four letters of the keywords are significant for the system. For instance, DELE is equivalent to DELETE.

Example

`DELE [IX]/[TAB1]/[PROF1]`	Deletion of variables
`DELE *[IZ]/[TAB3]/[PROF2]`	Deletion of variables from the stack

8 Programming of Error Numbers and Messages

8

8.1 General

Error numbers can be included for analysis in the programmes and subroutines.

The error numbers can be accompanied by a message.

The errors detected by the programmes are processed and displayed in the same way as the errors detected by the NC software.

8.1.1 Error Numbers

The errors created can be numbered from 1 to 9999.

The error number is programmed after the address E followed by a decimal point. For instance, to create error 503, programme:

```
E.503
```

This programming causes the following display:

```
error E.503 block N..
```

8.1.2 Error Messages

When the error number is accompanied by a message, programming E.503 results in the following display:

```
Error 503 - block N..
SPINDLE NOT INDEXED.
```

8.2 Creating Error Messages

The messages are contained in a list of programmes numbered from %20000 to %29900 in 100-block increments (maximum 100 messages per programme).

Each block of a message programme (%20000 to %29900) begins with the block number N.. corresponding to the error number, possibly followed by the character $ and the message.

REMARK *There must always be a space between the symbol $ and the message.*

The third and fourth digits from the right of the error programme number are the same as the number of the error they contain. For instance, programme %2xx00 and errors Nxx00.

Example:

Programme of messages 1 to 99

```
%20000 (ERRORS 1 TO 99)
N0001 $ ...
```

to

```
N0099 $ ...
```

Etc. up to programme %29900 (see below)

Programme of messages 9900 to 9999

```
%29900 $ ... (ERRORS 9900 TO 9999)
N9900 $ ...
N..
N..
N..
N..
N9999 $ ...
```

REMARKS *When the blocks following an error number are unnumbered, they may provide additional information to the message displayed. In case of an error, these blocks are not displayed (the list of errors can be consulted).*

When an error created has the same number as a standard NUM error, the error created takes precedence.

Example

Programming with error numbers and messages

Error message programme

```
%20500 (ERRORS 500 to 599)
N501 $ DIMENSION NOT POSITIVE
N502 $ G FUNCTION MISSING
N503 $ SPINDLE NOT INDEXED
N..
N540 $ MACHINE CONDITIONS NOT SATISFIED
- OIL PRESSURE LOW
- SEE DETAIL ON PLC MESSAGE
N..
N..
N..
N..
N..
N599 ...
```

Part programme

%123	
N10 ...	
N..	
N150 G108	Cycle calling a subroutine
N..	
N.. M02	

Subroutine

%10108	
N..	
N..	
G79 L0=0 N300	Test for error 501, jump to block N300 if the condition is not satisfied and display the message
N..	
N..	
IF E10002=1 AND E10020= 0 THEN E.540	Test for error 540
ENDI	
N..	
N300 E.501	Error 501
N..	

8

Appendix A Function Summary Tables

A

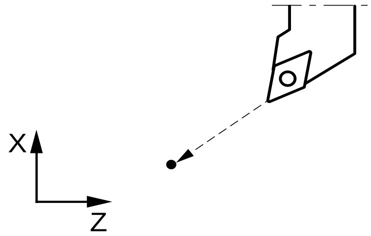
A.1 G Function Summary Table

The functions initialised at power on are identified by «*».

G00: Linear interpolation at high speed (see Sec. 4.4)

Syntax:
N.. [G90/G91] **G00** [R+/R-] X.. Z..

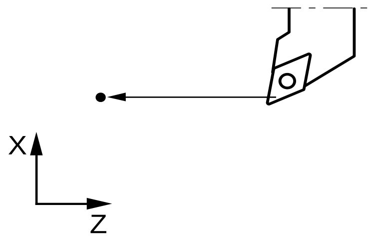
Cancellation:
G01/G02/G03/G33.



G01*: Linear interpolation at programmed feed rate (see Sec. 4.5.1)

Syntax:
N.. [G90/G91] **G01** [R+/-] X.. Z.. [F..]

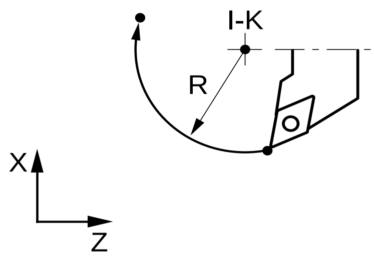
Cancellation:
G00/G02/G03/G33.



G02: Clockwise circular interpolation at programmed feed rate (see Sec. 4.5.2)

Syntax:
N.. [G90/G91] **G02** X.. Z.. I.. K.. / R.. [F..]

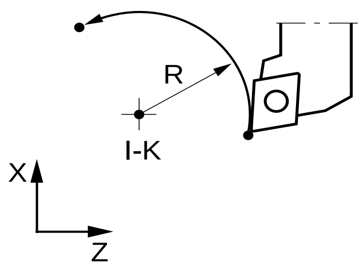
Cancellation:
G00/G01/G03/G33.



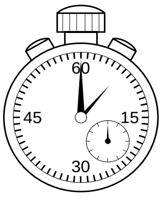
G03: Counterclockwise circular interpolation at programmed feed rate (see Sec. 4.5.2)

Syntax:
N.. [G90/G91] **G03** X.. Z.. I.. K.. / R.. [F..]

Cancellation:
G00/G01/G02/G33.



G04: Programmable dwell (see Sec. 4.15.1)



Syntax:
N.. **G04** F..

Cancellation:
End of block.

G05: Programming a movement on the inclined axis (see Sec. 4.16.2)

Syntax:
N.. [G90/G91] [G00/G01] **G05** X..

Cancellation:
End of block.

G06: Spline curve execution command (see Sec. 4.13.2.2)

Syntax:
N.. **G06** NC..

Cancellation:
End of block.

G07: Initial tool positioning before machining along an inclined axis (see Sec. 4.16.2)

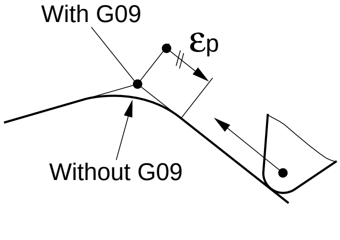
Syntax:
N.. [G90] [G00/G01] **G07** X.. Z..

Cancellation:
End of block.

G09: Deceleration at end of block before continuation on next block (see Sec. 4.6)

Syntax:
N.. **G09** [G00/G01/G02/G03] X.. Z.. [F..]

Cancellation:
End of block.

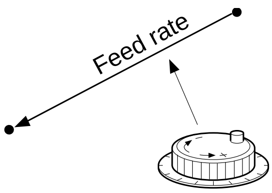


G10: Interruptible block (see Sec. 4.11.5)

Syntax:
N.. [G40] [G04 F..] [G00/G01/G02/G03] X.. Z.. **G10** [:n] [+X.. or F..] [@n < >Value] N.. [+ Number] [EF..]

Cancellation:
End of block.

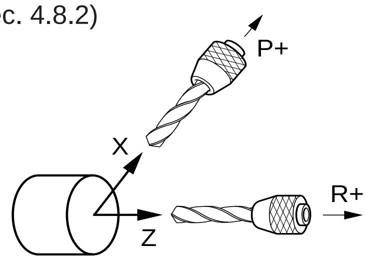
G12: Overspeed by handwheel (see Sec. 4.15.3)



Syntax:
N.. [G01/G02/G03] **G12** X.. Z.. [F..] [$0 ...]

Cancellation:
End of block.

G16*: Definition of the tool axis orientation with addresses P, R (see Sec. 4.8.2)



Syntax:
N.. **G16** P±/R±

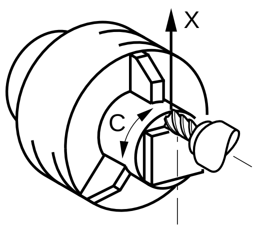
Cancellation:
G16 P±/R±.

G20*: Programming in polar coordinates (X, Z, C) (see Sec. 4.14.1)

Syntax:
N.. [G40] **G20** [G00/G01] [X.. Z.. C..] [F..]

Cancellation:
G21 and G22.

G21: Programming in cartesian coordinates (X, Y, Z) (see Sec. 4.14.3)

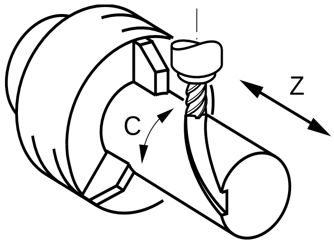


Syntax:
N.. [G40] **G21** [G00/G01] [G41/G42] [X.. Y.. Z..] [G94 F..]

Cancellation:
G20.

A

G22: Programming in cylindrical coordinates (X, Y, Z) (see Sec. 4.14.4)



Syntax:
N.. [G40] **G22** [G00/G01] [G41/G42] [X.. Y.. Z..] [G94 F..]

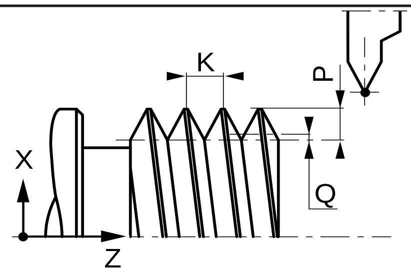
Cancellation:
G20.

G23: Circular interpolation defined by three points (see Sec. 4.5.3)

Syntax:
N.. [G90/G91] **G23** X.. Z.. I.. K.. [F..]

Cancellation:
End of block.

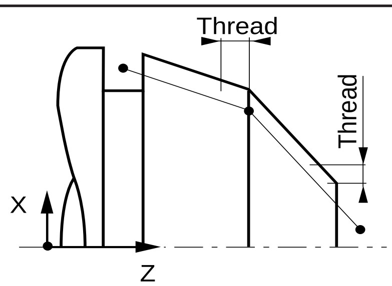
G33: Constant lead thread cutting (see Sec. 4.9.2)



Syntax:
N.. **G33** X.. Z.. K.. [EA..] [EB..] P.. [Q..] [R..] [F..] [S..]

Cancellation:
End of block.

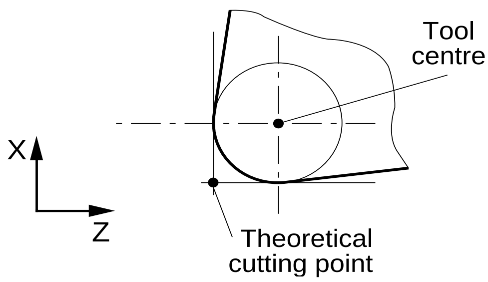
G38: Sequenced thread cutting (see Sec. 4.9.3)



Syntax:
N.. **G38** X.. Z.. K..

Cancellation:
G00/G01/G02/G03.

G40*: Tool compensation cancel (see Sec. 4.8.4)



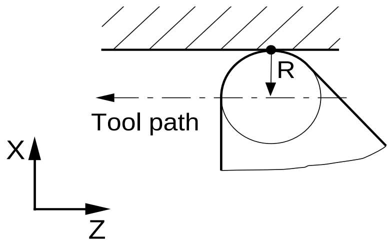
Syntax:
N.. [G00/G01] **G40** X.. Z..

Cancellation:
G41/G42.

G41: Tool compensation, left (see Sec. 4.8.4)

Syntax:
N.. [D..] [G00/G01/G02/G03] **G41** X.. Z..

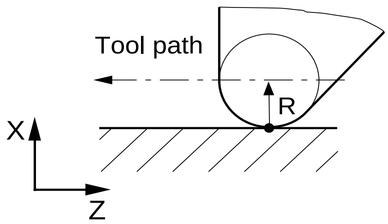
Cancellation:
G40/G42.



G42: Tool compensation, right (see Sec. 4.8.4)

Syntax:
N.. [D..] [G00/G01/G02/G03] **G42** X.. Z..

Cancellation:
G40/G41.



G48: Spline curve definition (see Sec. 4.13.2.1)

Syntax:
N.. **G48** NC.. H.. /N.. N..

Cancellation:
End of block.

G49: Spline curve deletion (see Sec. 4.13.2.4)

Syntax:
N.. **G49** NC..

Cancellation:
End of block.

G51: Mirror function (see Sec. 4.15.15)

Syntax:
N.. **G51** X- (Y-) Z-

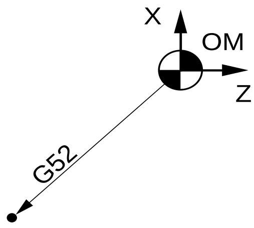
Cancellation:
G51 X- (Y-) Z- is cancelled by programming G51 X+ (Y+) Z+.

A

G52: Programming movements in absolute dimensions with reference to the measurement origin (see Sec. 4.12.1)

Syntax:
N.. [G40] [G90] [G00/G01] **G52** X.. Z.. C.. [F..]

Cancellation:
End of block.



G53: Cancellation of shifts DAT1 and DAT2 (see Sec. 4.12.2)

Syntax:
N.. **G53**

Cancellation:
G54.

G54*: Enabling of shifts DAT1 and DAT2 (see Sec. 4.12.2)

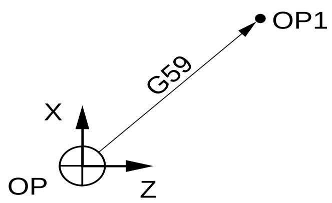
Syntax:
N.. **G54**

Cancellation:
G53.

G59: Programme origin offset (see Sec. 4.12.4)

Syntax:
N.. [G90/G91] **G59** X.. Z.. U.. W.. C.. [I.. K.. ED..]

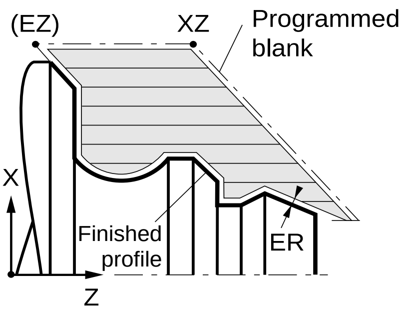
Cancellation:
Cancelled by G59 with different X.. (Y..) Z..



G63: Roughing cycle with groove (see Sec. 4.10.4)

Syntax:
N.. **G63** N.. N.. X.. Z.. EX.. / EZ.. P.. / R.. EA.. / EU.. EW..
[EB..] [EC..] [ER..] [Q..] [EQ..] [EF..]

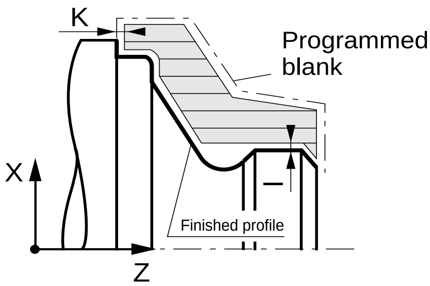
Cancellation:
End of block.



G64: Turn/Face roughing cycle (see Sec. 4.10.1)

Syntax:
N.. **G64** N.. N.. [I.. K..] P.. / R..
N.. BLANK DEFINITION

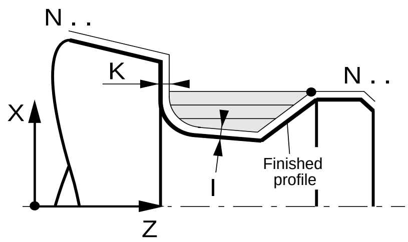
Cancellation:
G80.



G65: Groove roughing cycle (see Sec. 4.10.2)

Syntax:
N.. **G65** N.. N.. X.. / Z.. [I.. K..] [EA..] P.. / R.. [Q..] [EF..]

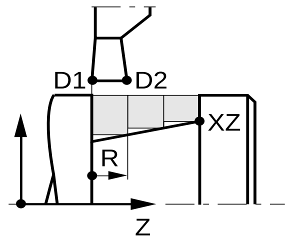
Cancellation:
End of block**.**



G66: Plunging cycle (see Sec. 4.10.3)

Syntax:
N.. **G66** D.. X.. Z.. [EA..] P.. / R.. [EF..]

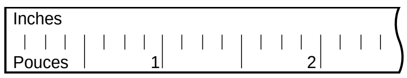
Cancellation:
End of block



G70: Inch data input (see Sec. 4.15.4)

Syntax:
N.. **G70**

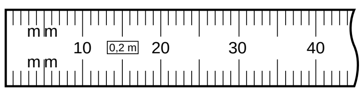
Cancellation:
G71.



G71: Metric data input (see Sec. 4.15.4)

Syntax:
N.. **G71**

Cancellation:
G70.



G73*: Scaling factor cancel (see Sec. 4.15.14)

Syntax:
N.. [G40] **G73**

Cancellation:
G74.

G74: Scaling factor enable (see Sec. 4.15.14)

Syntax:
N.. [G40] **G74**

Cancellation:
G73.

G75: Emergency disengagement subroutine enable (see Sec. 4.11.7)

Syntax:
N.. **G75** N..

Cancellation:
Cancelled by G75 N0 or G75 with different N..

G76: Transfer of the current values of L and E parameters into the part programme (see Sec. 6.4)

Syntax:
N.. **G76** [H..] [N.. N..]

Cancellation:
End of block.

G76+/-: ISO programme or block creation/deletion (see Sec. 4.11.12)

The syntax specific to each function is described below

Cancellation G76+/-:
End of block.

G76+: Programme creation (see Sec. 4.11.12.2)

Syntax:
N.. **G76+** H..

G76-: Programme deletion (see Sec. 4.11.12.3)

Syntax:
N.. **G76-** H..

G76+: ISO block insertion (see Sec. 4.11.12.4)

Syntax:
N.. **G76+** N.. [+number] ISO block

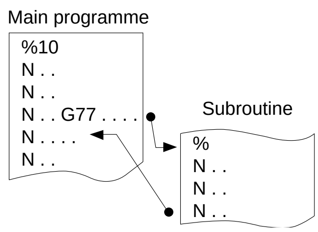
G76-: ISO block deletion (see Sec. 4.11.12.5)

Syntax:
N.. **G76-** [H..] N.. [+number]

G77: Unconditional branch to a subroutine or sequence of blocks with return (see Sec. 4.11.1)

Syntax:
N.. **G77** [H..] [N.. N..] [S..]

Cancellation:
End of block.



G77 -I: Call of the subroutine return block (see Sec. 4.11.11)

Syntax:
N.. **G77** i

Cancellation:
End of block.

G78: Axis group synchronisation (see Sec. 4.17.6)

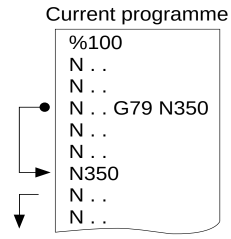
Syntax:
N.. **G78** Q.. / Pj.i

Cancellation:
Cancelled by G78 Q0 or G78 with different Q..

G79: Conditional or unconditional jump to a sequence without return (see Sec. 4.11.3)

Syntax:
N.. **G79** [L../E.. > = < Number] N..

Cancellation:
End of block.



G79+/-: Temporary suspension of next block preparation in a sequence with movements (see Sec. 4.11.6)

Syntax:
N.. [G00/G01/G02/G03] X.. Z.. **G79 +/-** X.. / F..

Cancellation:
End of block.

G80*: Machining cycle cancel (see Sec. 4.9.1)

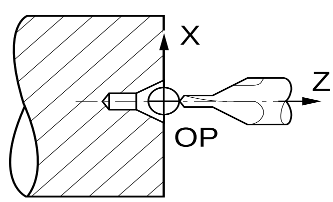
Syntax:
N**.. G80**

Cancellation:
G64/G81-G85/G87/G79.

G81: Centre drilling cycle (see Sec. 4.9.4.2)

Syntax:
N.. **G81** X.. / Z.. [ER..] [EH..]

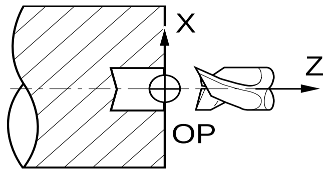
Cancellation:
G80/G64/G65/G66/G82-G85/G87 and G89.



G82: Counterboring cycle (see Sec. 4.9.4.3)

Syntax:
N.. **G82** X.. / Z.. [ER..] [EH..] EF..

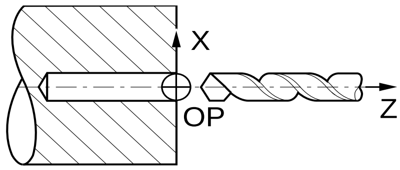
Cancellation:
G80/G64/G65/G66/G81/G83-G85/G87 and G89.



G83: Peck drilling cycle (see Sec. 4.9.4.4)

Syntax:
N.. **G83** X.. / Z.. [ER..] [EH..] [P..]/[ES..] [Q..] [EP..] [EF..]

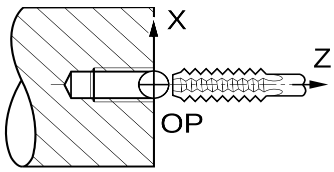
Cancellation:
G80/G64/G65/G66/G81/G82/G84/G85/G87 and G89.



G84: Tapping cycle (see Sec. 4.9.4.5)

Syntax:
N.. **G84** X.. / Z.. [ER..] [EH..] EF..

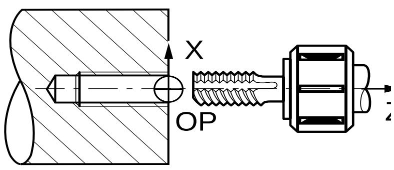
Cancellation:
G80/G64/G65/G66/G81-G83/G85/G87 and G89.



G84: Rigid tapping cycle (see Sec. 4.9.4.6)

Syntax:
N.. **G84** X.. / Z.. K.. [ER..] [EH..] [EK..]

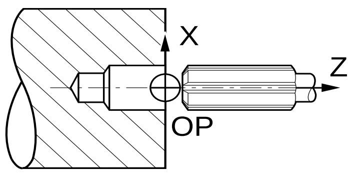
Cancellation:
G80/G64/G65/G66/G81-G83/G85/G87 and G89.



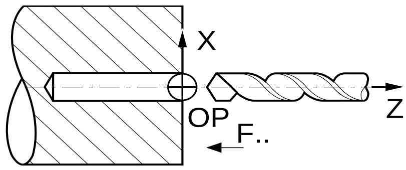
G85: Boring cycle (see Sec. 4.9.4.7)

Syntax:
N.. **G85** X.. / Z.. [ER..] [EH..] [EF..]

Cancellation:
G80/G64/G65/G66/G81-G84/G87 and G89.



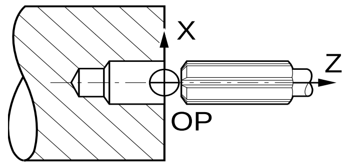
G87: Drilling cycle with chip breaking (see Sec. 4.9.4.8)



Syntax:
N.. **G87** X.. / Z.. [ER..] [EH..] [P..]/[ES..] [Q..] [EP..] [EF..]

Cancellation:
G80/G64/G65/G66/G81-G85 and G89.

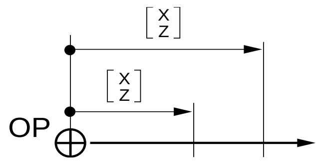
G89: Boring cycle with dwell in hole bottom (see Sec. 4.9.4.9)



Syntax:
N.. **G89** X.. / Z.. [ER..] [EH..] EF..

Cancellation:
G80/G64/G65/G66/G81-G85 and G87.

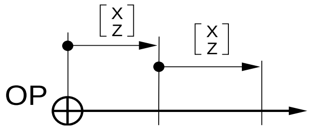
G90*: Programming in absolute dimensions with respect to the programme origin (see Sec. 4.1.1)



Syntax:
N.. **G90** X.. Z.. C..

Cancellation:
G91.

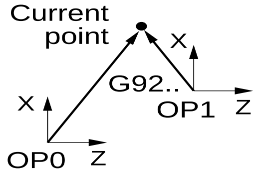
G91: Programming in incremental dimensions with respect to the start of the block (see Sec. 4.1.1)



Syntax:
N.. **G91** X.. Z.. C..

Cancellation:
G90.

G92 X Z: Programme origin preset (see Sec. 4.12.3)



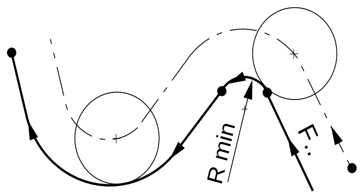
Syntax:
N.. **G92** X.. Z..

Cancellation:
End of block.

G92 R: Programming of the tangential feed rate (see Sec. 4.7.3)

Syntax:
N.. **G92** R..

Cancellation:
Cancelled by G92 R0 or G92 with different R..



G92 S: Spindle speed limiting (see Sec. 4.3.7)

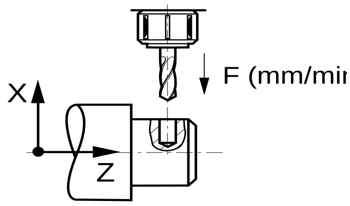
Syntax:
N.. **G92** S..

Cancellation:
Cancelled by G92 S0 or G92 with different S..

G94*: Feed rate expressed in millimetres, inches or degrees per minute (see Sec. 4.7.1)

Syntax:
N.. **G94** F.. G01/G02/G03 X.. Z.. C..

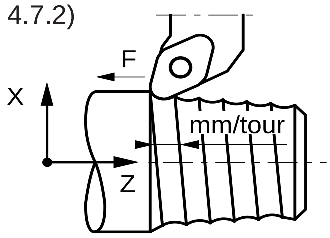
Cancellation:
G95.



G95: Feed rate expressed in millimetres or inches per revolution (see Sec. 4.7.2)

Syntax:
N.. **G95 F**.. [G01/G02/G03] [X.. Z..]

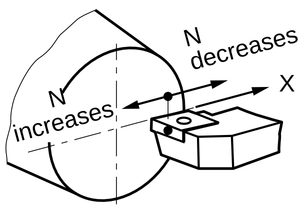
Cancellation:
G94.



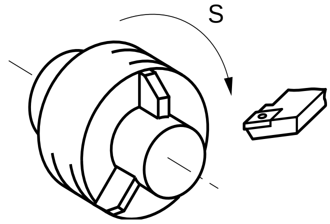
G96: Constant surface speed expressed in meters per minute (see Sec. 4.3.2.1)

Syntax:
N.. **G96** [X..] S..

Cancellation:
G97.



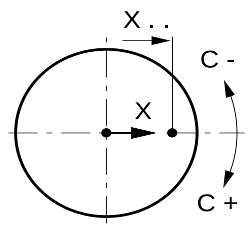
G97*: Spindle speed expressed in revolutions per minute (see Sec. 4.3.2)



Syntax:
N.. **G97** S.. [M03/M04]

Cancellation:
G96.

G98: Definition of the start X for interpolation on the C axis (see Sec. 4.14.2)



Syntax:
N.. **G98** X..

Cancellation:
End of block.

G997: Enabling and execution of all functions stored in state G999 (see Sec. 4.15.16)

Syntax:
N.. **G997**

Cancellation:
G998/G999.

G998: Enabling of execution of the blocks and part of the functions processed in state G999 (see Sec. 4.15.16)

Syntax:
N.. **G998**

Cancellation:
G997/G999.

G999: Suspension of execution and forcing of block concatenation (see Sec. 4.15.16)

Syntax:
N.. **G999**

Cancellation:
G997/G998.

A.2 M Function Summary Table

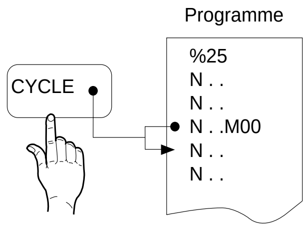
The functions initialised at power on are identified by «*».

The miscellaneous functions listed in the table are decoded functions.

M00: Programme stop (see Sec. 4.15.7)

Syntax:
N.. [G40] **M00** [$0 ...]

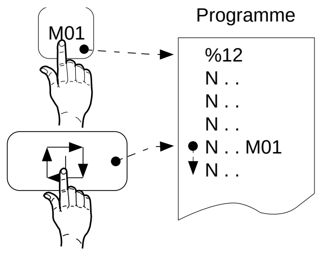
Cancellation:
Action on the machine panel CYCLE key.



M01: Optional stop (see Sec. 4.15.8)

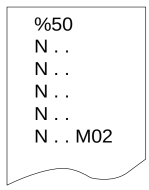
Syntax:
N.. [G40] **M01** [$0 ...]

Cancellation:
Action on the machine panel CYCLE key.



M02: End of programme (see Sec. 2.3)

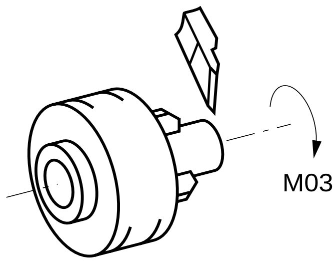
Syntax:
N.. **M02**



M03: Clockwise spindle rotation (see Sec. 4.3.1)

Syntax:
N.. **M03**

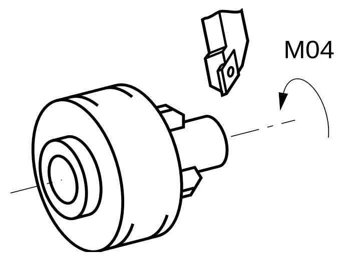
Cancellation:
M04/M05/M00/M19.



M04: Counterclockwise spindle rotation (see Sec. 4.3.1)

Syntax:
N.. **M04**

Cancellation:
M03/M05/M00/M19.



M05*: Spindle stop (see Sec. 4.3.1)

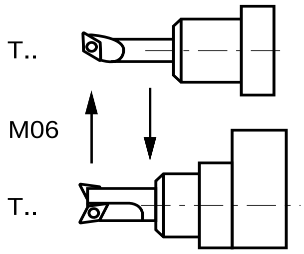
Syntax:
N.. **M05**

Cancellation:
M03/M04.

M06: Tool change (see Sec. 4.8.1)

Syntax:
N.. T.. **M06** [$0.. or (...)]

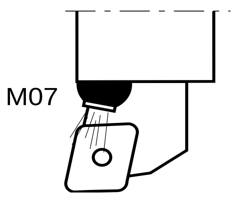
Cancellation:
M function report (CRM).



M07: Coolant 2 on (see Sec. 4.15.6)

Syntax:
N.. **M07**

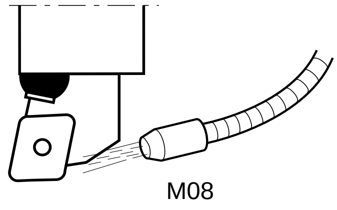
Cancellation:
M09.



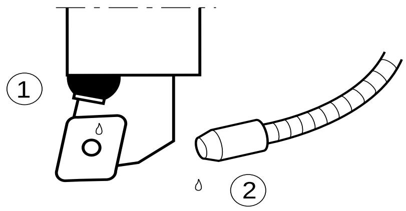
M08: Coolant 1 on (see Sec. 4.15.6)

Syntax:
N.. **M08**

Cancellation:
M09.



M09*: Coolant off (see Sec. 4.15.6)



Syntax:
N.. **M09**

Cancellation:
M07/M08.

M10: Clamp (see Sec. 4.15.5)

Syntax:
N.. [G00/G01/ G02/G03] **M10** X.. Z.. C..

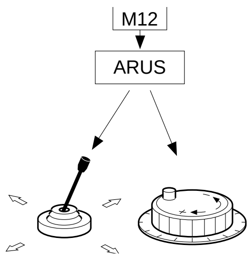
Cancellation:
M11.

M11: Unclamp (see Sec. 4.15.5)

Syntax:
N.. [G00/G01/G02/G03] **M11** X.. Z.. C..

Cancellation:
M10.

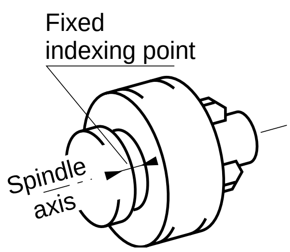
M12: Programmed feed stop (see Sec. 4.15.2)



Syntax:
N.. **M12** [$0...]

Cancellation:
Action on the machine panel CYCLE key.

M19: Spindle index (see Sec. 4.3.4)



Syntax:
N.. [G97 S..] [M40-M45] [M03/M04] C±.. **M19**

Cancellation:
M03/M04/M05.

M40-M45: Spindle speed ranges (see Sec. 4.3.3)

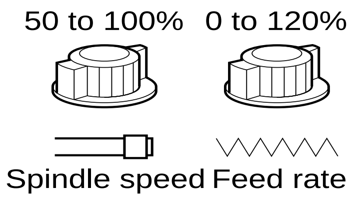
Syntax:
N.. [G97 S..] [M03/M04] **M40 to M45**

Cancellation:
Cancel one another.

M48*: Enable overrides (see Sec. 4.15.11)

Syntax:
N.. **M48**

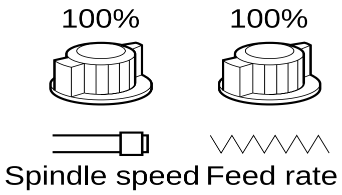
Cancellation:
M49.



M49: Disable overrides (see Sec. 4.15.11)

Syntax:
N.. **M49**

Cancellation:
M48.



M61: Current spindle release from group (see Sec. 4.17.5)

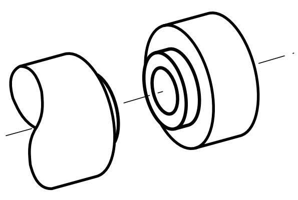
Syntax:
N.. **M61**

Cancellation:
M62-M65.

M64*, M65, M62, M63: Control of spindles 1 to 4 (see Sec. 4.3.5)

Syntax:
N.. [G97 S..] **M62/M63/M64/M65** [M40 to M45] M03/M04

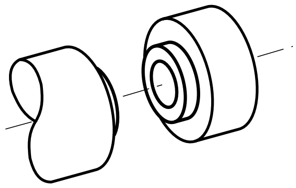
Cancellation:
Cancel one another.



M66*, M67, M68, M69: Measurement of spindles 1 to 4 (see Sec. 4.3.6)

Syntax:
N.. **M66/M67/M68/M69**

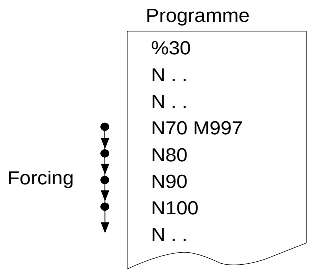
Cancellation:
Cancel one another.



M997: Forced block sequencing (see Sec. 4.15.10)

Syntax:
N.. **M997**

Cancellation:
M998/M999/M02.



M998*: Reactivation of the EDIT and MDI modes and subroutine calls by the automatic control function (see Sec. 4.15.9)

Syntax:
N.. **M998**

Cancellation:
M997/M999.

M999: Programmed cancellation of the EDIT and MDI modes and subroutine calls by the automatic control function(see Sec. 4.15.9)

Syntax:
N.. **M999**

Cancellation:
M997/M998/M02.

A.3 Additional Function Summary Table

ED: Programmed angular offset (see Sec. 4.12.5) **Syntax:** N.. [G90/G91] **ED**.. **Cancellation:** ED0 or different ED..
EG: Programmed acceleration modulation (see Sec. 4.15.13) **Syntax:** N.. **EG**.. **Cancellation:** Programming of a new value, M02, reset.
F: Feed rate, dwell, number of thread starts **Syntax:** N.. G94 **F..** (Feed rate in mm/min, deg/min and inch/min, see Sec. 4.7.1) N.. G95 **F..** (Feed rate in mm/rev and inch/rev, see Sec. 4.7.2) N.. G04 **F..** (Dwell in seconds, see Sec. 4.15.1) N.. G33 **F..** (Thread cutting, number of thread starts, see Sec. 4.9.2) **Cancellation:** G94, G95: different F..; G04, G33: end of block.
S: Rpm, meters/minute, number of subroutine repetitions, number of passes. **Syntax:** N.. G92 **S..** (Spindle rpm limiting, see Sec. 4.3.7) N.. G96 **S..** (Constant surface speed in meters/min, see Sec. 4.3.2.1) N.. G97 **S..** (Spindle speed in rpm, see Sec. 4.3.2) N.. G77 [H..] [N.. N..] **S..** (Subroutine call and repetitions, see Sec. 4.11.1) N.. G33 **S..** (Thread cutting, number of passes, see Sec. 4.9.2) **Cancellation:** S0 or different S..

T: Tool number (see Sec. 4.8.1) **Syntax:** N.. **T..** M06 (Tool change) **Cancellation:** T0 or different T..
D: Tool correction number (see Sec. 4.8.3) **Syntax:** N.. **D..** (Call of correction) **Cancellation:** D0 or different D..

Appendix B External Parameter E Summary Tables

B

Parameters E are accessible for read only or for read/write by the part programme.

The values of parameters E related to movements on the axes are expressed in the internal system units specified for linear axes and rotary axes (see Sec. 2.1). In the tables below, “IU” denotes the internal unit.

The parameters specific to milling are given in the tables for guidance and are identified by the word «Milling».

B.1 Parameters in the PLC Memory

Parameters	See Sec.	Description	Value or units	Access by the part programme
E10000 to E10031	6.2.9.1	1-bit words	0 or 1	Read/write
E20000 to E20031	6.2.9.1	1-bit words	0 or 1	Read only
E20100 to E20103	6.2.9.3	State of PLC interrupt machine inputs (IT)	0 or 1	Read only
E20104 to E20107	6.2.9.3	State of interrupt inputs (IT) on a first IT_ACIA card	0 or 1	Read only
E20108 to E20111	6.2.9.3	State of interrupt inputs (IT) on a second IT_ACIA card	0 or 1	Read only
E30000 to E30127	6.2.9.1	Long words	- 99999999 to 99999999	Read/write
E40000 to E40127	6.2.9.1	Long words	- 99999999 to 99999999	Read only
E42000 to E42127	6.2.9.1	1-byte words	0 to 1	Read only

B

B.2 Parameters in the NC Memory

Parameters	See Sec.	Description	Value or units	Access by the part programme
E11000	6.2.9.2	Angular offset (ED) enabled	0 or 1	Read/write (+ graphic)
E11001	6.2.9.2	Programme origin offset (G59) enabled	0 or 1	Read/write (+ graphic)

Parameters	See Sec.	Description	Value or units	Access by the part programme
E11003	6.2.9.2	Mirror function (G51) enabled	0 or 1	Read/write (+ graphic)
E11005	6.2.9.2	Programming on diameter	0 or 1	Read/write (+ graphic)
E11006	6.2.9.2	Tool centre programming	0 or 1	Read/write (+ graphic)
E11007	6.2.9.2	Spindle potentiometer enabled	0 or 1	Read/write (+ graphic)
E11008	6.2.9.2	Drawing of a complete circle	0 or 1	Read/write (+ graphic)
E11012	6.2.9.2	Following error cancellation	0 or 1	Read/write
E11013	6.2.9.2	Gradual acceleration	0 or 1	Read/write
E11014	6.2.9.2	Addressing of the deceleration function on several blocks	0 or 1	Read/write
E11015	6.2.9.2	Change of vector anticipation enabled	0 or 1	Read/write (+ graphic)
E11016	6.2.9.2	Front or rear turret	0 or 1	Read/write
E11017	Milling	Inclined plane function enabled	0 or 1	Read only
E11018	Milling	RTCP function enabled	0 or 1	Read only
E21000 to E21255	6.2.9.3	Presence of functions 0 to 255	0 or 1	Read only
E31000	6.2.9.2	Line type for G0 in graphic display	0 to 4	Read/write (+ graphic)
E31001	6.2.9.2	Line type for G01-G02-G03 in graphic display	0 to 4	Read/write (+ graphic)
E32000	6.2.9.2	Minimum execution time of an interpolation block	ms	Read/write
E32001	6.2.9.2	Path overspeed coefficient in G12	1 / 1024	Read/write
E32002	6.2.9.2	Servo-control error tolerated on a circle	µm	Read/write

Parameters	See Sec.	Description	Value or units	Access by the part programme
E32003	6.2.9.2	Angle crossing speed analysis angle	1/10000 degree	Read/write
E32004	6.2.9.2	Sagittal error	µm	Read/write
E32005	6.2.9.2	Number of total speed anticipation filter terms	1 to 14	Read/write
E33xyz	6.2.9.3	PLC output terminal addressing	x = 0 to 6 y = 0 to 9 z = 0 to 9	Read/write
E34xxy	6.2.9.3	Addressing of 8I8O analogue card analogue outputs	xx = 0 to 13 y = 0 to 7	Write only
E41000	6.2.9.3	Current mode number	0 to 15	Read only
E41001	6.2.9.3	Current axis group number	0 to number of groups	Read only
E41002	6.2.9.3	Number of groups in the machine	1 to 8	Read only
E41003	6.2.9.3	Graphic machining simulation status	0 to 2	Read only
E41004	6.2.9.3	Job reference image	0 to 99999999	Read only
E41005	6.2.9.3	Sampling period	µs	Read only
E41006	6.2.9.3	Axis group position servo-loop time constant	ms	Read only
E41102	6.2.9.3	Number of CNC axis groups	0 to 7	Read only
E43xyz	6.2.9.3	PLC input terminal addressing	x = 0 to 6 y = 0 to 9 z = 0 to 9	Read/write
E44xxy	6.2.9.3	Addressing of 8I8O analogue card analogue inputs	xx = 0 to 13 y = 0 to 7	Read only
E49001 to E49128	6.2.9.2	Read of dynamic operation numbers (dynamic operators)	1 to 128	Read only
E50000	6.2.9.4	Current tool correction number	0 to 255	Read only
E50001 to E50255	6.2.9.4	Tool X dimension	IU	Read/write

Parameters	See Sec.	Description	Value or units	Access by the part programme
E51000	6.2.9.4	Current tool orientation	0 to 2 or 100 to 102	Read only
E51001 to E51255	6.2.9.4	Tool Z dimension	IU	Read/write
E52001 to E52255	6.2.9.4	Tool radius	IU	Read/write
E53001 to E53255	6.2.9.4	Dynamic X correction	IU	Read/write
E54001 to E54255	6.2.9.4	Dynamic Z correction	IU	Read/write
E55001 to E55255	6.2.9.4	Tool tip orientation	0 to 8	Read/write
E56001 to E56255	6.2.9.4	Available parameters (H of the dynamic correction table)	- 99999999 to 99999999	Read/write
E57001 to E57255	6.2.9.4	Tool type	0 to 2	Read/write
E6x000 x = axis number (0-8)	6.2.9.4	DAT1	IU	Read/write (+ graphic)
E6x001 x = axis number (0-8)	6.2.9.4	DAT2	IU	Read/write (+ graphic)
E6x002 x = axis number (0-8)	6.2.9.4	Minimum dynamic travel	IU	Read/write (+ graphic)
E6x003 x = axis number (0-8)	6.2.9.4	Maximum dynamic travel	IU	Read/write (+ graphic)
E6x004 x = axis number (0-5)	Milling	DAT3 (limited to linear axes)	IU	Read/write (+ graphic)
E6x005 x = axis number (0-8)	6.2.9.4	Offset programmed by G59	IU	Read/write (+ graphic)
E69000	6.2.9.4	Scaling factor	1/1000	Read/write (+ graphic)
E69001	6.2.9.4	Inclined axis on grinding machine	1/10000 degree	Read/write (+ graphic)
E69002	6.2.9.4	Inclined wheel on grinding machine	1/10000 degree	Read/write (+ graphic)

Parameters	See Sec.	Description	Value or units	Access by the part programme
E69003	Milling	Axis assignment	XYZ axis assignment	Read/write (+ graphic)
E7x000 x = axis number (0-8)	6.2.9.5	Position reference	IU	Read only
E7x001 x = axis number (0-8)	6.2.9.5	Reference on on-the-fly dimension measurement	IU	Read/write
E7x002 x = axis number (0-8)	6.2.9.5	Minimum static travel	IU	Read only
E7x003 x = axis number (0-8)	6.2.9.5	Maximum static travel	IU	Read only
E7x004 x = axis number (0-8)	6.2.9.5	Direction of movement	IU	Read only
E7x005 x = axis number (0-8)	6.2.9.5	Axis address assignment	-1 to 31	Read/write
E7x006 x = axis number (0-8)	6.2.9.5	Carried axis	0 or 1	Read/write (+ graphic)
E7x007 x = axis number (0-8)	6.2.9.5	Axes programmed with reference to diameter	0 or 1	Read only
E7x100 x = axis number (0-8)	6.2.9.5	Position reference from the interpolator	IU	Read only
E7x101 x = axis number (0-8)	6.2.9.5	Interpolation speed limitation	0 to 100	Read/write
E79000	6.2.9.6	Position reference of the measured spindle in the group	0 to 3599999 1/10000 degree	Read only
E79001	6.2.9.6	Group spindle setting	$1/2^{15}$ of the maximum speed	Read only
E79002	6.2.9.3	Feed rate potentiometer value	1/128	Read only
E79003	6.2.9.3	Distance remaining to be travelled	IU	Read/write
E79004	6.2.9.3	Current speed on the path	IU	Read only
E80000 to E80050	6.2.9.7	Local data	- 99999999 to 99999999	Read/write
E81000 to E81999	6.2.9.7	Master axis reference positions	IU	Read/write

B

Parameters	See Sec.	Description	Value or units	Access by the part programme
E82000 to E82999	6.2.9.7	Slave axis corrections	IU	Read/write
E90000 to E90031	6.2.9.8	Axis measurement	IU	Read/write
E9010x	6.2.9.6	Spindle x position reference	0 to 3599999 1/10000 degree	Read only
E9011x	6.2.9.6	Spindle x modulo	According to spindle encoder	Read only
E9020x	6.2.9.6	Spindle x speed reference	$1/2^{15}$ of the maximum speed	Read only
E9030x	6.2.9.6	Maximum spindle x indexing speed	rev/min	Read/write
E9031x	6.2.9.6	Spindle x in-position window	IU	Read/write
E9032x	6.2.9.6	Spindle x indexing gain	rev/min/rev	Read/write
E9033x	6.2.9.6	Spindle x indexing acceleration	rev/s^2	Read/write
E9034x	6.2.9.6	Threshold below which spindle x is considered stopped	rev/min	Read/write
E9035x	6.2.9.6	Tolerance interval coefficient	1/256	Read/write
E91000 to E91031	6.2.9.8	Servo-controlled axis (or spindle)	0 or 1	Read/write
E91100 to E91131	6.2.9.8	Homing status on the axis or spindle (homing performed or not)	0 or 1	Read/write
E91200 to E91231	Milling	N/M AUTO axes	0 or 1	Read/write
E91300 to E91331	6.2.9.8	Clampable axis enable state	0 or 1	Read/write
E92000 to E92031	6.2.9.8	Axis origin switch enable state	0 or 1	Read/write
E93000 to E93031	6.2.9.8	Axis origin switch status	0 or 1	Read only

Parameters	See Sec.	Description	Value or units	Access by the part programme
E93100 to E93131	6.2.9.8	Measured axis	0 or 1	Read only
E93200 to E93231	6.2.9.8	Axis declared rotary modulo 360 degrees	0 or 1	Read only
E93300 to E93331	6.2.9.8	Homing direction on the axis	0 or 1	Read only
E93400 to E93431	6.2.9.8	Homing status on the axis without switch wiring	0 or 1	Read only
E93500 to E93531	6.2.9.8	Axis (or spindle) in position	0 or 1	Read only
E93600 to E93631	6.2.9.8	Measurement encoder type	0 or 4	Read only
E94000 to E94031	6.2.9.8	Assignment of a master axis to a slave axis	-1 to 31	Read/write
E94100 to E94131	6.2.9.8	Association of a slave axis (or spindle) master axis	-1 to 31	Read/write
E94200 to E94231	6.2.9.8	Axis/spindle switch	-1 to 31	Read/write
E95000 to E95031	6.2.9.8	Axis reference offset	IU	Read only
E95100 to E95131	6.2.9.8	Origin switch position with respect to machine origin	IU	Read only
E95200 to E95231	6.2.9.8	Axis measurement correction (or spindle)	IU	Read only
E960xx	6.2.9.8	Duplicated axis in automatic mode	0 or 1	Read/write
E961xx	6.2.9.8	Duplicated axis in manual mode (JOG)	0 or 1	Read/write
E962xx	6.2.9.8	Synchronised axis	0 or 1	Read/write
E963xx	6.2.9.8	Symmetrically driven axis	0 or 1	Read/write
E97000 to E97031	6.2.9.8	Maximum axis speed	mm/min or deg/min	Read only
E97100 to E97131	6.2.9.8	Axis acceleration at the work rate	mm/s^2 or deg/s^2	Read only

Parameters	See Sec.	Description	Value or units	Access by the part programme
E97200 to E97231	6.2.9.8	Axis acceleration at high speed	mm/s^2 or deg/s^2	Read only
E97300 to E97331	6.2.9.8	Maximum authorised angle crossing speed	mm/min	Read/write
E98000 to E98031	6.2.9.8	Axis servo-loop coefficient	1/1000	Read/write
E98100 to E98131	6.2.9.8	Axis acceleration anticipation time constant	µs	Read/write
E98200 to E98231	6.2.9.8	Antistick pulse amplitude	µm	Read/write
E98300 to E98331	6.2.9.8	Antistick pulse absorption time constant	1/100 ms	Read/write

Appendix C Word Format Summary Table

The formats of the axis words specified in the table are expressed as follows:
- For linear axes: 5 digits to the left and 3 to the right of the decimal point are allowed when the internal system unit (see Sec. 2.1) is µm
- For rotary axes: 3 digits to the left and 4 to the right of the decimal point are allowed when the internal system unit is 0.0001 degree.

For instance, for linear axes: if the system is set to 0.1 µm (internal unit), the formats are expressed with 4 decimal digits; for the X axis, the format is X+044.

For instance, for rotary axes: if the system is set to 0.001 degree (internal unit), the formats are expressed with 3 decimal digits; for the C axis, the format is C+033.

For the words related to the machining feed rate with no assigned formats (F.., EF..), refer to the machine manufacturer's manual for the maximum and minimum feed rates (maximum 8 digits and decimal point).

Format	Description
%051	Programme number (1 to 99999.9)
N05	Sequence number (1 to 32767)
G02 G03	Preparatory functions (0 to 99) Preparatory functions (100 to 250 and 997 to 999)
H05	Subroutine number (with G77, G76 and G48)
X+053	Movement on the X axis. In a cycle, end point on the machining axis
Y+053	Movement on the Y axis
Z+053	Movement on the Z axis. In a cycle, end point on the machining axis
I+053 J+053 K+053 I+053 K+053 K053 K053	For circular interpolation (G02, G03), centre of the circle With programmed origin offset (G59), centre of rotation of an angular offset (ED) For roughing cycle (G64, G65), finishing machining allowance on X For roughing cycle (G64, G65), finishing machining allowance on Z For rigid tapping cycle (G84), tap pitch For thread cutting cycle (G33, G38), thread pitch
U+053	Movement on the U axis
V+053	Movement on the V axis
W+053	Movement on the W axis

C

Format	Description
A+034	Movement on the A axis
B+034	Movement on the B axis
C+034 C+034	Movement on the C axis Spindle indexing modulo 360 degrees (M19)
E+/E-	For profil geometry programming (PGP), tangent element
EA+033 EA+033 EA+033 EA+033 EA+033 EA+033	For profile geometry programming (PGP), line angle For thread cutting cycle (G33), apex half-angle of a tapered thread For roughing cycle with groove (G63), start angle for pass setting For groove roughing cycle (G65), groove pass setting angle For plunging cycle (G66), groove bottom slope angle In polar programming, angle of the line
EB+053 EB-053 EB+033 EB+033	For contour definition and PGP, radius or fillet between two interpolations For contour definition and PGP, chamfer between two linear interpolations For thread cutting cycle (G33), tool penetration angle For roughing cycle with groove (G63), limit groove penetration angle
EC+033	For roughing cycle with groove (G63), limit end of pass angle in roughing axis
ED+034	Programmed angular offset
EF.. EF. EF.. EF.. EF.. EF022 EF022	For roughing cycle with groove (G63), penetration feed rate in material For grooving cycle (G65), penetration feed rate in material In a reaming cycle (G85), retraction rate Feed rate specific to fillets (EB+) or chamfers (EB-) Maximum feed rate after an interrupt (G10) For plunging cycle (G66), dwell in groove bottom For cycles (G83, G87), dwell at the end of each penetration
EG03	For interpolation, acceleration modulation
EH+053	In (G81 to G89) cycles, dimension of the impact plane on the machining axis
EI+053	In polar programming, length of the line (start/circle centre)
EP053 EP053 EP053	In a peck drilling cycle (G83), backoff clearance after each peck In a drilling cycle with chip breaking (G87), backoff between two infeeds In a plunging cycle (G66), value of retraction at 45 degrees at the end of the pass
EQ053	For roughing cycle with groove (G63), minimum chip size
EK01	For rigid tapping cycle (G84), penetration/retraction speed ratio

Format	Description
ER053	For roughing cycle with groove (G63), finishing allowance
ER+053	In cycles (G81 to G89), dimension of the approach or retraction plane on the machining axis
ES+/ES-	For profile geometry programming (PGP), secant element
ES02	In cycles G83 and G87, number of infeeds at constant value
ES02	In cycles (G33), number of passes at constant value
ET+/ET-	For profile geometry programming (PGP), tangent element
EU+053	For roughing cycle with groove (G63), with XZ, pass setting angle and limit of last pass
EW+053	For roughing cycle with groove (G63), with XZ, pass setting angle and limit of last pass
EX+053	For roughing cycle with groove (G63), pass end position in X
EX+053	In polar programming, angle of the line (start/end)
EZ+053	For roughing cycle with groove (G63), pass end position in Z
P053	For thread cutting cycle (G33), total depth of thread
P053	For roughing cycle (G63, G64, G65), pass depth in X
P053	For plunging cycle (G66), step in X
P053	For cycles (G83 or G87), value of the first penetration
P041	For axis group synchronisation (G78), wait for marker
P+ / P-	Tool axis orientation (G16), along X or U
Q053	For thread cutting cycle (G33), depth of last pass
Q053	For roughing cycle (G63, G65), holding of positioning before pass setting
Q053	For cycles (G83 or G87), value of the last penetration
Q04	For axis group synchronisation (G78), declaration of a marker
R+053	For circular interpolation (G02, G03), radius of the circle
R053	For thread cutting cycle (G33), length of the retraction slope at the end of thread cutting
R053	For roughing cycle (G63, G64, G65), pass depth in Z
R053	For plunging cycle (G66), step in Z
R053	For tangential feed rate programming (G92), value of the curve radius
R+ / R-	For linear interpolation (G00, G01), positioning at programmed distance
R+ / R-	Tool axis orientation (G16), along Z (or W)
F..	Feed rate in mm/min and degrees/min (G94)
F..	Feed rate in inches/min (G94)
F..	Feed rate in mm/rev (G95)
F..	Feed rate in inches/rev (G95)
F022	Dwell in seconds (G04)
F02	For thread cutting cycle (G33), number of threads (1 to 99)

Format	Description
M02 M03	Miscellaneous functions (0 to 99) Miscellaneous functions (100 to 899)
NC04	Spline curve number (1-9999) with interpolation (with G48, G06, G49)
S05 S05 S05 or S032 S02 S02	Spindle rpm limiting (G92) Constant surface speed in m/min (G96) Spindle speed in rpm (G97) In a subroutine branch (G77), number of subroutine iterations (1 to 99) For thread cuttting cycle (G33), number of passes (1 to 99)
T08	Tool number (0 to 9999999)
D03	Correction number (0 to 255)
L03	Programme variables L (0 to 19, 100 to 199 and 900 to 959)
E5	External parameters E

Appendix D List of Errors

D

D.1 Miscellaneous Errors and Machine Errors

Error No.	Meaning of the error
1	Unknown character / Axis not recognised by the system Too many digits after a function Presence of a sign after a function which does not allow signs Truncated block signalled by ? via CLOSE in drip feed mode
2	Unknown G function or a mandatory argument missing after the G
3	Attribute of a G code wrongly positioned
4	Option not enabled or option parameter conflict: Structured programming, RTCP, synchronised axes, etc.
5	Geometric option programming not enabled
6	Polynomial interpolation option missing coefficient table full
7	Error in programming movements parallel to inclined axes (grinder): - Programming is not in plane G20 - Interpolation is not in G00 or G01 - X is not programmed after G05 - X and Z are not programmed after G07
8	Tool correction number too high
9	A sequence of too many non-working blocks - Endless Loop
10	In PLC terminal access: Bus exchange error
11	In PLC terminal access: Bus initialisation error or exchange inhibited
12	In PLC terminal access: Rack parameter error
13	In PLC terminal access: No such card
14	Inclined plane option missing PLC boundary access: channel missing
15	Invalid line configuration
16	Error in RTCP activation
17	End of block in a comment - close bracket missing
18 *	Servo error: P50 too small
20	No M02 at the end of the programme Blocks not made executable in a cycle called by a G function
21	Blank definition incoherent in 3D mode
24	Error in inclined plane declaration - Function reactivated when already active - Function argument declaration incomplete - Pivot point axis does not exist or is not servo-controlled - Incoherent value in one of the matrix terms
25	Subroutine or sequence number does not exist
26	Too many subroutine nesting levels
27	Radius offset: In G52 machine origin programming / With taper threads
28	Syntax error in CCSPD or index table radius definition G96 must be followed by S / G97 must be followed by S / initial radius cannot be determined X or U not programmed in this block or a previous block

* Machine error. Caution: For this type of error, a CNC reset causes a general reset (CNC reset + PLC reset)

Error No.	Meaning of the error
29	No range programmed for CCSPD / No range compatible with S in G97: No range search option: S not included between min. and max. values of the range programmed With range search option: S does not belong to any range
30	Line error detected
31 *	PPR or PPL mode impossible with the line protocol selected
32 *	Homing error / Axis already on limit switch
33 *	All slides on wait for synchronisation
34	Minimum radius reached in G21 interpolation
35 *	Sequence number not found in SEARCH
36 *	Part programme memory full
37	Max. feed rate exceeded for thread cutting (COMAND)
38	Spindle already controlled by another axis group
39 *	Axis synchronisation error (with axis synchronisation option)
40-49 *	Excessive following error on axis 0 to 9
50-59 *	Excessive following error on axis 10 to 19
60-69 *	Excessive following error on axis 20 to 29
70 and 71 *	Excessive following error on axis 30 and 31
72	Incremental programming after an incomplete block (PGP)
75	Switch from state G20 to G21 or G22: last block in G20 incomplete as it is programmed in PGP or radius correction or with X ≤ 0 first block in G21 without X and Y or G22 without Y and Z Switch from state G21 or G22 to G20: last block in G21 or G22 incomplete or first block in G20 in mode G41 or G42: In G21 or G22, initial radius negative or zero
76	In G21, programming of a fixed turning and milling cycle
77	Tool type incompatible with the machining phase (milling or turning)
78	Syntax error in programming slide synchronisation G78 P: Maximum 4 digits, must be less than the number of slides G78 Q: Maximum 4 digits No M00, M01 or M02 with G78 P..

* Machine error. Caution: For this type of error, a CNC reset causes a general reset (CNC reset + PLC reset).

D.2 Parametric Programming Errors

Error No.	Meaning of the error
91	Parameter No. not recognised
92	Negative parameter assigned to a function which does not take a sign Parameter value higher than the maximum value of the function to which it is assigned
93	Error in parameter declaration or test expression: L function not followed by symbols =,<,>, &, ! Association with a prohibited function by a linking character -, +, *, /.
94	Operation prohibited in a parametric expression: Square root of a negative number / Division by 0
95	Attempt to write in an external input parameter or a read-only parameter
96	The block preceding the external parameter declaration is incomplete Programming of L100 ... in a contour definition in G64
97	Parameter update impossible in G76: No symbol = after the parameter number Less than 10 characters allocated for entry of the value
98	Write by an axis group of a dynamic operation already used by another group
99	Error related to the N/M AUTO function - More than 5 N/M AUTO axes defined - Non-servo-controlled axis defined as N/M AUTO - Definition of an N/M AUTO axis of another group

D

D.3 Profile Geometry Programming (PGP) Errors

D.3.1 The end point is not determined or cannot be computed from the elements in the blocks

Error No.	Meaning of the error
101	PGP: Insufficient data for programming a circle Circle programmed on two parallel axes (with R / see Error 107)
102	Line programmed by an angle and one coordinate with no way of calculating the other coordinate
106	In G02, G03, programming of the third axis without helical option
107	PGP: Circle programmed by its radius and end point, with the end point separated from the start point by more than 2 * radius Circle programmed by X, Z, I K with a start radius different from the end point (20 microns) / Helical: dimension of 3rd axis missing Circle programmed on two parallel axes (with I, J, K / see Error 101)

D.3.2 The point of tangency or intersection cannot be computed from the data in two blocks

Error No.	Meaning of the error
110	PGP: Syntax error in the first of two blocks of a PGP entity
111	PGP: Syntax error in the second block of a PGP entity
112	PGP: Line/line intersection in which: First block starting point = second block end point, or First line angle = second line angle
113	PGP: The values programmed in the two blocks do not allow determination of an intersection or tangency point
114	PGP: Intersection or tangency point not determined by ET+, ET-, ES+ or ES-

D.3.3 The points of tangency or intersection cannot be computed from the data in three blocks

Error No.	Meaning of the error
121	PGP: Syntax error in the last of the three blocks of a PGP entity
122	PGP: The first two blocks are non-intersecting lines
123	PGP: The data programmed in the three blocks do not allow determination of the tangent point
124	PGP: Tangent point of the second and third blocks not specified by ET+ or ET-

D.3.4 Fillet or chamfer definition errors

Error No.	Meaning of the error
130	Zero displacement in one of the two blocks connected by a fillet or a chamfer
131	Fillet or chamfer programmed in a block including M0, M1 or M2 Programming insufficient in a sequence of blocks, not allowing determination of the end point
135	A chamfer can only connect two straight lines

D.3.5 Miscellaneous errors in PGP

Error No.	Meaning of the error
136	More than two blocks without movement between two geometric elements whose intersection or tangency point is to be calculated
137	Change of interpolation plane with an invalid block

D.4 Miscellaneous Errors

Error No.	Meaning of the error
138	Change of interpolation plane when not in G40 (FCU)
139	Two carried parallel axes programmed in the same block outside G52 and outside G00
140	Radius correction programming error: Too many extraneous blocks between two consecutive paths The following functions cannot be programmed when radius offset is active: M00, M01, M02, access to external parameters, writing of parameters E8xxxx or L > 100
141	Carried parallel axes: Programming of a circle whose start point was programmed with one axis and whose end point was programmed with the associated parallel axis
143	Scale factor cancelled or enabled with radius offset
144	Movement of a quantified axis different from the increment
145	G29: ABS VAL (P * P + Q * Q + R * R - 1000 mm) > 1 mm (normal vector not a unit vector
146	Offset in space / G29 - At least one of dimensions P, Q or R missing - At least one of dimensions X/U, Y/V or Z/W missing
148	Number of axes programmed exceeds the maximum authorised number
149	Tool radius too large with respect to programmed path

D.5 Request for Movements Outside the Machine Travel Limits

Error No.	Meaning of the error
150	Travel overrun on the X axis
151	Travel overrun on the Y axis
152	Travel overrun on the Z axis
153	Travel overrun on the U axis
154	Travel overrun on the V axis
155	Travel overrun on the W axis
156	Travel overrun on the A axis
157	Travel overrun on the B axis
158	Travel overrun on the C axis
159	Request for programmed movement on an UN-HOMED axis

D.6 Structured Programming Errors

Error No.	Meaning of the error
190	Too many branch or loop nesting levels (maximum 15)
191	Non-compliance with the syntax in structured programming structured programming prohibited in MDI mode the index of a FOR loop must be: an L variable, a symbolic variable or a parameter E80000, E81000 or E82000 non-compliance with the syntax of PUSH and PULL instructions DO missing after WHILE programming of IF, THEN, ELSE in MDI mode
192	Keyword not recognised or prohibited in the context
193	Structure error
195	Programme stack saturated / Too many constants defined for the space allocated
196	Error in array index declaration
197	Use of a symbol not declared as VAR
198	Syntax error in variable symbol declaration
199	Incorrect variable declaration syntax

D.7 Axis Errors

Error No.	Meaning of the error
210 to 219 *	Poor signal or pulse generator complementarity error on axis 0 to 9
220 to 229 *	Poor signal or pulse generator complementarity error on axis 10 to 19
230 to 239 *	Poor signal or pulse generator complementarity error on axis 20 to 29
240 and 241 *	Poor signal or pulse generator complementarity error on axis 30 and 31
245 *	Fault on digital servo-control

* Machine error. Caution: For this type of error, a CNC reset causes a general reset (CNC reset + PLC reset)

D.8 Errors in Pocket Cycles

Error No.	Meaning of the error
260	Working memory busy
261	Programme number too high
262	NU number not among those authorised
263	Execution impossible - Test or Graphic mode mandatory after first load or after editing
264	No dimension programmed in the contouring plane or dimension outside the plane
265	First positioning block missing; contour definition must begin with G0 or G1
266	Not enough memory
267	Character not allowed in pocket syntax
268	Pocket programming block incomplete or containing illegal data
269	Contour block incomplete / Positioning block missing before pocket definition
270	Pocket definition partly or completely missing
271	Tool orientation not perpendicular to the contouring plane
272	Real tool not compatible with pocket technological data
273	Change of contouring plane between pocket definition and machining
274	Two nested pocket definitions
275	NU0 programmed with G59
276	Zero pocket depth
277	Pocket definition start point or end point coordinates incomplete
278	The spindle rotation direction is incompatible with the one required in the pocket definition
279	G function not allowed in a pocket programming block
280	First contour block incomplete
281	Discontinuity in one of the contours described
282	Pocket definition parameter error(s)
283	The external contour must be unique and must exist
284	Error in contour definition
285	Too many contours
286	Pass setting excessive with respect to the tool diameter
287	Pass setting insufficient with respect to the dimensions
288	Finishing infeed in an acute angle or an unroughed area: change the infeed point
289	Tool diameter excessive
290	Internal error
291	Finishing infeed outside the contour
292	Double positioning at the start of the contour
293	Roughing end point present during facing

D.9 Axes Not Identified on the Bus

Error No.	Meaning of the error
300 to 309 *	Axis 0 to 9 declared in P2 but not detected on the bus
310 to 319 *	Axis 10 to 19 declared in P2 but not detected on the bus
320 to 329 *	Axis 20 to 29 declared in P2 but not detected on the bus
330 and 331 *	Axis 30 and 31 declared in P2 but not detected on the bus

* Machine error. Caution: For this type of error, a CNC reset causes a general reset (CNC reset + PLC reset)

D.10 Dynamic Operators in C

Error No.	Meaning of the error
400	Loading dyn. ops in C: The size of user code is too big
401	Loading dyn. ops in C: Format error
402	Loading dyn. ops in C: Checksum error
403	The system has insufficient memory for dyn. ops in C
404	Loading dyn. ops in C: Open error
405	Loading dyn. ops in C: Read error
406	Loading dyn. ops in C: Close error
407	Loading dyn. ops in C:The directory is empty
410	Dyn. ops in C: Number of parameters passed doesn’t tally
411	Dyn. ops in C: USER ERROR from INIT: negative return
413	Unrecognised dyn. ops in C
414	Dyn. ops in C without MAIN
420	Dyn. ops in C: USER ERROR from the QUIT function
421	Dyn. ops in C: USER ERROR from the QUIT function: negative return
423	Dyn. ops in C: Range of function in C not from [0..100]

D.11 Spline Curve Interpolation Errors

Error No.	Meaning of the error
600	Curve number zero
601	N.. N.. must be programmed
602	No axes programmed in the first block of the contour
603	Curve slope undefined
604	Less than three blocks in the profile
605	Curve number unknown

D.12 Errors in Numaform

Error No.	Meaning of the error
700	Options missing
701	S.. missing at beginning of curve
702	Number of S.. different in T1 & T2
703	Minimum 2 occurrences of S in T1
704	Undefined section (in T3)
705	Plane switching outside S mark
706	Spindle stopped
707	Invalid E= function
708	E=1 or E=2: Section positioning error
709	T1 & T2 can’t have any points in common
710	P, Q must be positive
711	S different in T1 & T2
712	Undetermined tool position
713	Error: S=0 or T>3
730	F= less than or equal to zero
731	Intersection of concentric circles
732	Intersection of parallel lines
733	Limit cannot be a horizontal plane
740	F = error

D

D.13 Cycle Programming Errors

Error No.	Meaning of the error
830	Positioning not completed
831	Spindle stopped
832	End point, P and K must be programmed
833	Retraction clearance too small
834	EB value: -90 < EB < +90
835	The values of P, Q, R and K are absolute values
836	The interpolation plane must be G81 or G20
837	Bad value of F or S
862	P or R and end point to be programmed
863	End point incoherent with EA
864	Milling tool prohibited in G66
871	Finished profile limits not defined
872	No dimensions in blank definition
873	P or R not programmed
874	Blank inconsistent with finished profile
875	No intersection of EA with the profile
876	Relief angle EB incorrectly defined
880	Cycle axis unknown
881	Parameter value not compatible
882	Hole bottom dimension not programmed
883	Pitch (I J K) or clearance (P) not programmed
884	More than 9 thread starts
885	Pocket incompatible with the plane selected
886	Tool incompatible with the radius programmed
887	Cut > tool diameter
888	Dwell prohibited in this cycle
889	Syntax error
890	Tool orientation incompatible
891	Return plane = bottom of hole
892	Axial feed missing
893	Lateral feed missing
894	ER prohibited in G20
895	G21,G22 prohibited in cycle
896	Dimension incompatible with tool radius
897	Length of oblong pocket < diameter
898	Tool corrector missing
899	Spindle not assigned to this group or spindle or group incompatible

Index

Symbols

A

B

C

D

J

L

M

N

O

P