NUM
1020/1040/1060M

PROGRAMMING MANUAL

VOLUME 1

0101938819/5

05-97 **en-938819/5**

Despite the care taken in the preparation of this document, NUM cannot guarantee the accuracy of the information it contains and cannot be held responsible for any errors therein, nor for any damage which might result from the use or application of the document.

The physical, technical and functional characteristics of the hardware and software products and the services described in this document are subject to modification and cannot under any circumstances be regarded as contractual.

The programming examples described in this manual are intended for guidance only. They must be specially adapted before they can be used in programs with an industrial application, according to the automated system used and the safety levels required.

© Copyright NUM 1997.
All rights reserved. No part of this manual may be copied or reproduced in any form or by any means whatsoever, including photographic or magnetic processes. The transcription on an electronic machine of all or part of the contents is forbidden.

© Copyright NUM 1997 software NUM 1000 family.
This software is the property of NUM. Each memorized copy of this software sold confers upon the purchaser a non-exclusive licence strictly limited to the use of the said copy. No copy or other form of duplication of this product is authorized.

Table of Contents

1 Review			1 - 1
	1.1	System Overview	1 - 3
	1.2	Machine Overview	1 - 5
2 Structure of a Programme			2 - 1
	2.1	Word Format	2 - 4
	2.2	Block Format	2 - 7
	2.3	General Structure of a Programme	2 - 9
	2.4	Classification of Preparatory G Functions and Miscellaneous M Functions	2 - 18
3 Axis Programming			3 - 1
	3.1	General	3 - 3
	3.2	Programming the Independent Secondary Axes	3 - 4
	3.3	Programming of Carrier/Carried Parallel Axis Pairs	3 - 5
	3.4	Programming Rotary Axes Modulo 360 Degrees	3 - 6
	3.5	Programming Rotary Axes with Limited Travel	3 - 7
	3.6	Programming of Axes A, B or C Declared as Nonrotary	3 - 7
4 ISO Programming			4 - 1
	4.1	Choice of the Programming System	4 - 7
	4.2	Plane Selection	4 - 10
	4.3	Spindle Control	4 - 12
	4.4	Rapid Positioning	4 - 23
	4.5	Programming Movements	4 - 26
	4.6	Path Sequencing Conditions	4 - 60
	4.7	Feed Rate	4 - 62
	4.8	Programming of Tools	4 - 76
	4.9	Basic Cycles	4 - 109
	4.10	Other Cycles	4 - 146
	4.11	Breaks in Sequence	4 - 193
	4.12	Movement Origin Selection	4 - 229
	4.13	Spline Curve Interpolation	4 - 247
	4.14	Other Functions	4 - 256
	4.15	Special Programming for Multi-axis Groups	4 - 294
	4.16	Special Programming of PLC Axes	4 - 304
	4.17	Special Features of Mixed Machines (MX)	4 - 308
	4.18	Message Transmission	4 - 314

5 Profile Geometry Programming			5 - 1
	5.1	Profile Geometry Programming (PGP)	5 - 3
	5.2	PROFIL Function	5 - 24
6 Parametric Programming			6 - 1
	6.1	Programme L Variables	6 - 3
	6.2	External E Parameters	6 - 20
	6.3	Address Equivalences	6 - 58
	6.4	Transfer of the Current Values of L Variables and E Parameters into the Part Programme	6 - 59
	6.5	Message Display with Wait for an Operator Response	6 - 61
	6.6	Display of Messages with Parametric Value	6 - 63
	6.7	Reading the Programme Status Access Symbols	6 - 64
	6.8	General Diagrams of Parametric Programming	6 - 68
7 Programme Stack - L Variables and Symbolic Variables			7 - 1
	7.1	Programme Stack	7 - 3
	7.2	Saving and Restoring L Variables	7 - 3
	7.3	Symbolic Variables	7 - 6
8 Programming of Error Numbers and Messages			8 - 1
	8.1	General	8 - 3
	8.2	Creating Error Messages	8 - 3
Appendix A Function Summary Tables			A - 1
	A.1	G Function Summary Table	A - 3
	A.2	M Function Summary Table	A - 18
	A.3	Additional Function Summary Table	A - 23
Appendix B External Parameter E Summary Tables			B - 1
	B.1	Parameters in the PLC Memory	B - 3
	B.2	Parameters in the NC Memory	B - 3
Appendix C Word Format Summary Table			C - 1

Appendix D List of Errors		D - 1
D.1	Miscellaneous Errors and Machine Errors	D - 3
D.2	Parametric Programming Errors	D - 5
D.3	Profile Geometry Programming (PGP) Errors	D - 6
D.4	Miscellaneous Errors	D - 7
D.5	Request for Movements Outside the Machine Travel Limits	D - 8
D.6	Structured Programming Errors	D - 8
D.7	Axis Errors	D - 8
D.8	Errors in Pocket Cycles	D - 9
D.9	Axes Not Identified on the Bus	D - 10
D.10	Dynamic Operators in C	D - 10
D.11	Spline Curve Interpolation Errors	D - 10
D.12	Errors in Numaform	D - 11
D.13	Cycle Programming Errors	D - 12

Record of Revisions

DOCUMENT REVISIONS		
Date	**Revision**	**Reason for revisions**
01-92	0	Document creation (conforming to software index B)
09-93	1	Update to conform to software index D Manual revisions: - Classification of G preparatory functions and M miscellaneous functions. - Special programming of axis multigroups. - Processing of blocks and programmed G and M functions (with G997 to G999). - Programming of error numbers and messages. - 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. - Control and measurement of 4 spindles. - Creation of external parameter E41004. Software index D: - Spline curve interpolation. - Rigid tapping. - 3D tool correction with 3 and 5 axes. - Creation of external parameters E42000 to E42127, E79003, E79004, E41005, E941xx, E960xx, E961xx, E962xx, E963xx.
07-94	2	Update to conform to software index F Added a paragraph concerning access to the Profil function (see Sec. 5.2). Manual revisions: - Pocket and facing cycles with any contours (G46) - 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) - Unconditional branch to a sequence by G77 N.. - Pocket cutting direction (G45) defined by EG2 or EG3

		Added changes Software at index E: - Polar programming - Feed rate in fillets EB+ and chamfers EB- - 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 - Added a paragraph concerning special characteristics of mixed machines - New arguments with cycles G81 to G89
02-95	3	Update to conform to software index G Manual revisions: - External parameters E11012, E11013, E11017, E11018, E41006, E935xx, E980xx and E981xx
04-96	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 - external parameters E32002, E32003, E32004, E32005, 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**
05-97	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 - Axis assignment by external parameter E69003 - 3D correction with cylindrical tool (G43) - Axes programmed by variables L or parameters E defined by symbolic variables Added changes: Software index J and K: - 3D curve smoothing (G104)

Structure of the NUM 1020/1040/1060 Documentation

User Documents

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

NUM M/W	NUM T	NUM M	NUM T	NUM G
OPERATOR'S MANUAL	OPERATOR'S MANUAL	PROGRAMMING MANUAL VOLUME 1 VOLUME 2	PROGRAMMING MANUAL VOLUME 1 VOLUME 2	CYLINDRICAL GRINDING PROGRAMMING MANUAL
938821	938822	938819	938820	938930

OEM Documents

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

NUM 1060	NUM 1020/1040	NUM	NUM	NUM
INSTALLATION AND COMMISSIONING MANUAL	INSTALLATION AND COMMISSIONING MANUAL	PARAMETER MANUAL	AUTOMATIC CONTROL FUNCTION PROGRAMMING MANUAL LADDER LANGUAGE	DYNAMIC OPERATORS
938816	938938	938818	938846	938871

NUM	NUM G	NUM H/HG	NUM	NUM GS
PROCAM DESCRIPTION MANUAL	CYLINDRICAL GRINDING COMMISSIONING MANUAL	GEAR CUTTING AND GRINDING MANUAL	SYNCHRONISATION OF TWO SPINDLES	SURFACE GRINDING MANUAL
938904	938929	938932	938854	938945

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

938873

NUM
T

PROCAM TURN
INTERACTIVE
PROGRAMMING

938874

NUM

DUPLICATED
AND
SYNCHRONISED
AXES

938875

NUM

PROFIL
FUNCTION
USER'S
MANUAL

938937

NUM
GS

PROCAM GRIND
INTERACTIVE
PROGRAMMING

938931

NUM
G

PROCAM GRIND
INTERACTIVE
PROGRAMMING

938952

NUM
M

PROCAM
MILL
TECHNOLOGICAL
DATA

938958

NUM
T

PROCAM
TURN
TECHNOLOGICAL
DATA

938959

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.. Y.. 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 - 23
G01	Linear interpolation at programmed feed rate	4 - 26
G02	Clockwise circular interpolation at programmed feed rate	4 - 31
G03	Counterclockwise circular interpolation at programmed feed rate	4 - 31
G04	Programmable dwell	4 - 256
G06	Spline curve execution command	4 - 247
G09	Accurate stop at end of block before going to next block	4 - 60
G10	Interruptible block	4 - 208
G12	Overspeed by handwheel	4 - 260
G16	Definition of tool axis orientation with addresses P, Q, R	4 - 79
G17	XY plane selection	4 - 10
G18	ZX plane selection	4 - 10
G19	YZ plane selection	4 - 10
G23	Circular interpolation defined by three points	4 - 45
G29	3D tool correction (3 axes or 5 axes)	4 - 99
G31	Thread chasing cycle	4 - 137
G40	Tool radius offset (cutter compensation) cancel	4 - 86
G41	Left tool radius offset (cutter compensation)	4 - 85
G42	Right tool radius offset (cutter compensation)	4 - 85
G43	3D correction with cylindrical tool	4 - 107
G45	Simple pocket cycle	4 - 146
G46	Pocket or facing cycles with any contours	4 - 155
G48	Spline curve definition	4 - 247
G49	Spline curve deletion	4 - 247

Code	Description	Page
G51	Mirroring	4 - 283
G52	Programming of movements in absoluted dimensions with reference to the measurement origin	4 - 229
G53	DAT1 and DAT2 offset cancel	4 - 232
G54	DAT1 and DAT2 offset enable	4 - 232
G59	Programme origin offset	4 - 235
G70	Inch data input	4 - 262
G71	Metric data input	4 - 262
G73	Scaling factor cancel	4 - 279
G74	Scaling factor enable	4 - 279
G75	Emergency retraction subroutine declaration	4 - 215
G76	Transfer of the current values of «L» and «E» parameters into the part programme	6 -59
G76+/-	ISO programme or block creation/deletion	4 - 224
G77	Unconditional branch to a subroutine or block sequence with return	4 - 193
G77 -i	Call of a subroutine return block	4 - 222
G78	Axis group synchronisation	4 - 300
G79	Conditional or unconditional jump to a sequence without return	4 - 203
G79+/-	Temporary suspension of next block preparation in a sequence with movements	4 - 213
G80	Canned cycle cancel	4 - 112
G81	Centre drilling cycle	4 - 113
G82	Counterboring cycle	4 - 115
G83	Peck drilling cycle	4 - 117
G84	Tapping cycle	4 - 120
G84	Rigid tapping cycle	4 - 122
G85	Reaming cycle	4 - 126
G86	Boring cycle with indexed stop and clearance at hole bottom	4 - 128

Code	Description	Page
G87	Drilling cycle with chip breaking	4 - 130
G88	Boring and facing cycle	4 - 133
G89	Boring cycle with dwell at hole bottom	4 - 135
G90	Programming in absolute dimensions with respect to the programme origin	4 - 7
G91	Programming in incremental dimensions with respect to the start of the block	4 - 7
G92	Programme origin preset	4 - 233
G92 R	Programming of the tangential feed rate	4 - 72
G93	Feed rate expressed as inverse time (V/L)	4 - 66
G94	Feed rate expressed in millimetres, inches or degrees per minute	4 - 62
G95	Feed rate expressed in millimetres or inches per revolution	4 - 70
G97	Spindle speed expressed in revolutions per minute	4 - 14
G104	3D curve smoothing	4 - 292
G997	Enabling and execution of all the functions stored in state G999	4 - 289
G998	Enabling of execution of part of the functions processed in state G999	4 - 289
G999	Suspension of execution and forcing of block concatenation	4 - 289

M Fonctions

Code	Description	Page
M00	Programme stop	4 - 267
M01	Optional stop	4 - 269
M02	End of programme	2 - 11
M03	Clockwise spindle rotation	4 - 12
M04	Counterclockwise spindle rotation	4 - 12
M05	Spindle stop	4 - 12
M06	Tool change	4 - 76
M07	Coolant 2 on	4 - 266
M08	Coolant 1 on	4 - 266
M09	Coolant off	4 - 266
M10	Clamp	4 - 264
M11	Unclamp	4 - 264
M12	Programmed feed stop	4 - 258
M19	Spindle index	4 - 17
M40 to M45	Spindle speed ranges	4 - 16
M48	Enable overrides	4 - 274
M49	Disable overrides	4 - 274
M61	Release of current spindle in the axis group	4 - 299
M62 to M65	Control of spindles 1 to 4	4 - 19
M66 to M69	Measurement of spindles 1 to 4	4 - 21
M997	Forced block sequencing	4 - 273
M998	Reactivation of edit (EDIT) and manual data input (MDI) modes and subroutine calls by the automatic control function	4 - 271
M999	Programmed cancellation of the edit (EDIT) and manual data input (MDI) modes and subroutine calls by the automatic control function.	4 - 271

Other Functions

Code	Désignation	Page
$0	Message transmission to the display	4 - 314
$1 to $6 $9	Message transmission to the PLC function or a remote server or a peripheral	4 - 316
/	Block skip	4 - 275
T	Tool number	4 - 76
D..	Call to tool correction	4 - 81
ED..	Programmed angular offset	4 - 241
EG..	Programmed acceleration modulation	4 - 277
EM-/+	Outside dimensions of the part in 3D graphic display	4 - 287
M	Conversion of the internal unit of rotary axes	6-5 and 6-22
U	Conversion of the internal unit of linear axes	6-5 and 6-22

1 Review

1.1	**System Overview**		1 - 3
	1.1.1	Overview of Modes	1 - 3
	1.1.2	Defining a Programme	1 - 3
	1.1.3	Preparing a Programme	1 - 4
1.2	**Machine Overview**		1 - 5
	1.2.1	Review of Axis Definition and Direction	1 - 5
	1.2.2	Machine Overview	1 - 6
	1.2.3	Definition of Travels and Origins	1 - 7
	1.2.4	Offset Definitions	1 - 9
	1.2.5	Definition of Tool Dimensions	1 - 14
	1.2.6	Definition of Dynamic Tool Corrections	1 - 15

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 Preparing 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 milling machine:
- the Z axis is the axis of the main spindle when this axis is parallel to one of the slideways,
- positive movement along the Z axis increases the distance between the part and tool,
- the X axis is perpendicular to the Z axis and corresponds to the largest excursion,
- the Y axis is perpendicular to the X and Z axes.

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 axis is limited by software limits whose values are defined by the manufacturer.

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 DAT1/2 values for each axis from the part programme.

Programme origin offset (OP/Op) = DAT2

Offsets on the Z axis

Offsets on the X axis

Offsets on the Y 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
Y_{PA}	$Y_{MA} = Y_{PA}$ + Y DAT1 + Y DAT2
Z_{PA}	$Z_{MA} = Z_{PA}$ + Z DAT1 + Z DAT2

Programmed shifts can be added to the programme dimensions.

Special Case of Milling Machines Equipped with Rotary Tables

The concept of part origins only applies to the two axes affected by the rotation.

The centre of rotation of the table (OC) has a special role.

Offset to the centre of rotation (OC/OM) = DAT1 (axes affected by rotation)

Offset to the part origin (OP/OC) = DAT3 (axes affected by rotation)

REMARK *For axes other than those affected by rotation, the original definitions of DAT1 and DAT2 still apply.*

Example: rotary axis B

Rotation is around an axis parallel to the Y axis. Therefore, the axes affected by the rotation are Z and X.

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 coordinates (with respect to OP)	Measurements coordinates (with respect to OM)
X_{PA}	$X_{MA} = X_{PA}$ + XDAT1 (+ XDAT2) + ΔX with ΔX = XDAT3 x cos B - ZDAT3 x sin B
Y_{PA}	$Y_{MA} = Y_{PA}$ + YDAT1 + YDAT2
Z_{PA}	$Z_{MA} = Z_{PA}$ + ZDAT1 (+ ZDAT2) + ΔZ with ΔZ = ZDAT3 x cos B + XDAT3 x sin B

1.2.5 Definition of Tool Dimensions

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

Tool radius = R
Tool length = L

Cutter tip radius = @

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 dimensions on a part.

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

Dynamic tool radius correction = DR
Dynamic tool length correction = DL

The system takes into account the corrected tool dimensions:

Corrected radius = R + DR
Corrected length = L + DL

2 Structure of a Programme

2.1	**Word Format**		2 - 4
	2.1.1	General Word Format	2 - 4
	2.1.2	Special Features of the Dimension Word Format	2 - 4
	2.1.2.1	Internal System Unit for Linear Axes	2 - 5
	2.1.2.2	Internal System Unit for Rotary Axes	2 - 5
2.2	**Block Format**		2 - 7
2.3	**General Structure of a Programme**		2 - 9
	2.3.1	General	2 - 9
	2.3.2	Branches and Subroutine Calls	2 - 11
	2.3.3	Programme Numbering	2 - 12
	2.3.4	Characteristics of the ISO and EIA Codes	2 - 13
2.4	**Classification of Preparatory G Functions and Miscellaneous M Functions**		2 - 18
	2.4.1	Classification of Preparatory G Functions	2 - 18
	2.4.1.1	Modal G Functions	2 - 18
	2.4.1.2	Nonmodal G Functions	2 - 18
	2.4.1.3	G Functions Incompatible with the State of the Programme	2 - 18
	2.4.1.4	G Functions Associated with Arguments	2 - 19
	2.4.2	Classification of Miscellaneous M Functions	2 - 21
	2.4.2.1	Modal M Functions	2 - 21
	2.4.2.2	Nonmodal M Functions	2 - 21
	2.4.2.3	«Pre» M Functions	2 - 21
	2.4.2.4	«Post» M Functions	2 - 21
	2.4.2.5	Encoded M Functions	2 - 22
	2.4.2.6	Decoded M Functions	2 - 22

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

A block defining a tool change and calling up the tool correction

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

Review of the 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	-			●		●	•	●		●
Digits	0			●	●		•			
	1	●		●	●		•			●
	2	●		●	●		•		●	
	3			●	●		•		●	●
	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	●	●			●	•	●	●	●
Miscellaneous parameters	P		●		●		•			
	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	-		●				•			
Digits	0			●			•			
	1						•			●
	2						•		●	
	3				●		•		●	●
	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		●				•	●		●
Miscellaneous parameters	o		●				•	●	●	
	p		●		●		•	●	●	●
	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. 4.18).

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 Sec. A.1).

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

Example:

`N.. G00 X.. Y..`	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.. G18 G41 X.. Y..`	Syntax correct, ZX plane selection (G18), followed by radius offset (G41)
`N..`	
`N.. G41 G18 X.. Y..`	Syntax incorrect, change of plane 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 F200`	Syntax correct
`N.. G04 T03 F2 F200`	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 (G96: constant surface speed for a mixed machine)

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.. Y..          Feed in mm/min
N..
N.. X.. Y.. 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 Sec. 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 Y50 M08`	Coolant function M08 is executed before the movements on X and Y

2.4.2.4 «Post» M Functions

Functions executed after the axis movements programmed in the block.

Example:

`N.. X50 Y100 M09`	The coolant off function (M09) is executed after movements on X and Y

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 report 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 an M function report (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**	3 - 3
3.2	**Programming the Independent Secondary Axes**	3 - 4
3.3	**Programming of Carrier/Carried Parallel Axis Pairs**	3 - 5
3.4	**Programming Rotary Axes Modulo 360 Degrees**	3 - 6
3.5	**Programming Rotary Axes with Limited Travel**	3 - 7
3.6	**Programming of Axes A, B or C Declared as Nonrotary**	3 - 7

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:

XMx (machine dimension) = XPx (programmed dimension) + xDAT1 + xDAT2 + Lx

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:

XMu (machine dimension) = XPu (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 Programming of Carrier/Carried Parallel Axis Pairs

Consider movement of the axis pair with respect to the part.

The W axis is the machine bed and the Z axis is the quill.

Representation of machine bed approach, programmed by WP2

Calculation of WM2, knowing that ZP2 = WP2

WM2 = WP2 + (∑ DAT1 + DAT2 + L) - ZM1

Representation of quill retraction, programmed by ZP3

Calculation of ZM3, knowing that ZP3 = WP3.

ZM3 = ZP3 + (∑ DAT1 + DAT2 + L) - WM2

3.4 Programming Rotary Axes Modulo 360 Degrees

Rotary axis B programmed in absolute dimensions (G90)

The angular value assigned to the axis is the end point with respect 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 used to reach this point.

Example

a : Start point.

b : End point.

Positive movement

```
N.. ...
N.. G90 B+270
N..
```

Negative movement

```
N..
N.. G90 B-270
N..
```

Rotary axis B programmed in incremental dimensions (G91)

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

Example

a : Start point.

b : End point.

Positive rotation

```
N.. ...
N.. G91 B+240
N..
```

Negative rotation

```
N.. ...
N.. G91 B-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 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 by more than one revolution in absolute dimensions (G90).

Rotation by more 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.

4 ISO Programming

4.1	**Choice of the Programming System**		4 - 7
	4.1.1	Programming by Absolute or Incremental Dimensions	4 - 7
4.2	**Plane Selection**		4 - 10
4.3	**Spindle Control**		4 - 12
	4.3.1	Direction of Rotation	4 - 12
	4.3.2	Spindle Speed Control	4 - 14
	4.3.3	Spindle Range	4 - 16
	4.3.4	Indexed Spindle Stop	4 - 17
	4.3.5	Spindle Control Selection	4 - 19
	4.3.6	Spindle Measurement Selection	4 - 21
4.4	**Rapid Positioning**		4 - 23
4.5	**Programming Movements**		4 - 26
	4.5.1	Linear Interpolation	4 - 26
	4.5.2	Circular Interpolation	4 - 31
	4.5.3	Helical Interpolation	4 - 39
	4.5.4	Circular Interpolation Defined by Three Points	4 - 45
	4.5.5	Polar Programming	4 - 47
	4.5.5.1	Polar Programming of a Line	4 - 48
	4.5.5.2	Polar Programming of a Circle	4 - 50
	4.5.5.3	Defining a Circle by the Arc Angle	4 - 54
	4.5.6	Programming Fillets and Chamfers	4 - 58
	4.5.6.1	Fillet Between Two Interpolations	4 - 58
	4.5.6.2	Chamfer Between Two Linear Interpolations	4 - 59
4.6	**Path Sequencing Conditions**		4 - 60
4.7	**Feed Rate**		4 - 62
	4.7.1	Feed Rate Expressed in Millimetres, Inches or Degrees per Minute	4 - 62
	4.7.2	Inverse Time Feed Rate Coding (V/D)	4 - 66
	4.7.3	Feed Rate Expressed in Millimetres or Inches per Revolution	4 - 70
	4.7.4	Tangential Feed Rate	4 - 72
	4.7.5	Feed Rate Specific to Fillets EB+ and Chamfers EB-	4 - 74
4.8	**Programming of Tools**		4 - 76
	4.8.1	Tool Change	4 - 76
	4.8.2	Tool Axis Orientation	4 - 79
	4.8.3	Tool Correction Call	4 - 81
	4.8.4	Positioning the Tool with Respect to the Part	4 - 85
	4.8.5	3D Tool Correction (3 Axes or 5 Axes)	4 - 99
	4.8.5.1	3D Tool Correction with Toroid or Spherical Tool	4 - 99
	4.8.5.2	3D Tool Correction with Cylindrical Tool	4 - 107

4.9 Basic Cycles			4 - 109
	4.9.1	Cycle Overview	4 - 109
	4.9.2	Cancellation of a Canned Cycle	4 - 112
	4.9.3	Centre Drilling Cycle	4 - 113
	4.9.4	Counterboring Cycle	4 - 115
	4.9.5	Peck Drilling Cycle	4 - 117
	4.9.6	Tapping Cycles	4 - 120
	4.9.6.1	Tapping Cycle	4 - 120
	4.9.6.2	Rigid Tapping Cycle	4 - 122
	4.9.7	Reaming Cycle	4 - 126
	4.9.8	Boring Cycle with Indexed Spindle Stop at the Bottom of the Hole	4 - 128
	4.9.9	Drilling Cycle with Chip Breaking	4 - 130
	4.9.10	Boring and Facing Cycle	4 - 133
	4.9.11	Boring Cycle with Dwell at the Bottom of the Hole	4 - 135
	4.9.12	Thread Chasing Cycle	4 - 137
	4.9.13	Table Summarising Cycles G81 to G89	4 - 141
	4.9.14	Examples of Programming Cycles G81-G89	4 - 142
4.10 Other Cycles			4 - 146
	4.10.1	Simple Pocket Cycle	4 - 146
	4.10.2	Pocket and Facing Cycles with Any Contours	4 - 155
	4.10.2.1	General	4 - 155
	4.10.2.2	Blocks Specific to Cycle Programming	4 - 156
	4.10.2.3	Notes on Programming Contour Geometric Definition Blocks	4 - 157
	4.10.2.4	Notes on Programming the Machining Orders	4 - 159
	4.10.2.5	Geometric Definition Header Block	4 - 161
	4.10.2.6	Pocket and Island Introduction Blocks	4 - 163
	4.10.2.7	Facing and Cavity Introduction Blocks	4 - 166
	4.10.2.8	Facing and Wall Introduction Blocks	4 - 168
	4.10.2.9	Geometric Definition End Block	4 - 170
	4.10.2.10	Initial Drilling Orders	4 - 171
	4.10.2.11	Roughing Order	4 - 173
	4.10.2.12	Finishing and Semi-Finishing Orders	4 - 175
	4.10.3	Examples of Programming of Cycles with Any Contours	4 - 178

4.11 Breaks in Sequence			4 - 193
	4.11.1	Unconditional Branch to a Subroutine or Sequence of Blocks with Return	4 - 193
	4.11.2	Subroutine Branch by M Function	4 - 200
	4.11.3	Branch to a Sequence without Return	4 - 203
	4.11.4	Subroutine Call by Automatic Control Function	4 - 205
	4.11.5	Block Interrupt	4 - 208
	4.11.5.1	Special Use of Sequence Interrupt	4 - 212
	4.11.6	Temporary Suspension of Next Block Preparation	4 - 213
	4.11.7	Emergency Retract	4 - 215
	4.11.8	Branch to Automatic Homing Subroutine	4 - 219
	4.11.9	Subroutine Branch on a Reset	4 - 220
	4.11.10	Restrictions Related to Drip Feed Mode	4 - 221
	4.11.11	Call to Subroutine Return Block	4 - 222
	4.11.12	ISO Programme or Block Creation/ Deletion	4 - 224
	4.11.12.1	General	4 - 224
	4.11.12.2	Creating a Programme	4 - 224
	4.11.12.3	Deleting a Programme	4 - 225
	4.11.12.4	Inserting a Block	4 - 226
	4.11.12.5	Deleting a Block	4 - 228
4.12 Movement Origin Selection			4 - 229
	4.12.1	Programming of Movements in Absolute Coordinates Referenced to the Measurement Origin	4 - 229
	4.12.2	Datum Shift DAT1 and DAT2 Cancel/ Enable	4 - 232
	4.12.3	Programme Origin Preset	4 - 233
	4.12.4	Programme Origin Offset	4 - 235
	4.12.5	Angular offset	4 - 241
	4.12.6	Table Offset by DAT3	4 - 245
4.13 Spline Curve Interpolation			4 - 247
	4.13.1	General	4 - 247
	4.13.2	Programming	4 - 247
	4.13.2.1	Spline Curve Interpolation	4 - 248
	4.13.2.2	Spline Curve Execution Command	4 - 251
	4.13.2.3	Programming Examples	4 - 252
	4.13.2.4	Freeing Memory by Deleting a Spline Curve	4 - 255

4.14	**Other Functions**		4 - 256
	4.14.1	Dwell	4 - 256
	4.14.2	Programmed Feed Stop	4 - 258
	4.14.3	Feed Enhancement	4 - 260
	4.14.4	Programming in Inches or Metric Data	4 - 262
	4.14.5	Axis Clamping and Unclamping	4 - 264
	4.14.6	Coolant	4 - 266
	4.14.7	Programme Stop	4 - 267
	4.14.8	Optional Stop	4 - 269
	4.14.9	Cancellation of MDI and EDIT modes	4 - 271
	4.14.10	Forced Block Continuation	4 - 273
	4.14.11	Potentiometer Inhibit	4 - 274
	4.14.12	Block Skip	4 - 275
	4.14.13	Acceleration Reduction	4 - 277
	4.14.14	Scaling Factor	4 - 279
	4.14.15	Mirror function	4 - 283
	4.14.16	Overall Dimensions of the Part in 3D Graphic Display	4 - 287
	4.14.17	Processing of Blocks and Programmed G and M Functions	4 - 289
	4.14.18	3D Curve Smoothing	4 - 292
4.15	**Special Programming for Multi-axis Groups**		4 - 294
	4.15.1	Programme Declaration	4 - 294
	4.15.2	Programming Notes	4 - 294
	4.15.3	Subroutine branches for Multi-Axis Groups	4 - 296
	4.15.3.1	Branch to Automatic Homing Subroutine	4 - 296
	4.15.3.2	Subroutine Call by a Reset	4 - 296
	4.15.3.3	Subroutine Call by the Automatic Control Function	4 - 297
	4.15.3.4	Subroutine Call by M Function	4 - 297
	4.15.4	Programming the Spindles	4 - 298
	4.15.5	Releasing the Current Spindle in the Axis Group	4 - 299
	4.15.6	Synchronising the Axis Groups	4 - 300
4.16	**Special Programming of PLC Axes**		4 - 304
	4.16.1	Programme Declaration and Storage	4 - 304
	4.16.2	Programming of the PLC Axes	4 - 305
	4.16.2.1	Emergency Retraction on a PLC Axis Group	4 - 305
	4.16.3	Editing the Programmes	4 - 306
	4.16.4	Exchanging Axes between Groups	4 - 306
	4.16.5	Exchanging Spindles Between Groups	4 - 307

4.17	**Special Features of Mixed Machines (MX)**		4 - 308
	4.17.1	Features Related to the Machine Axes	4 - 308
	4.17.1.1	Axis Declaration	4 - 308
	4.17.1.2	X and U Axes with Reference to Diameter or Radius	4 - 308
	4.17.1.3	Axis Orientation and Definition of the Programme Origin	4 - 309
	4.17.1.4	Assignment of Tool Dimensions to the Axes	4 - 310
	4.17.2	Notes on Programming the ISO Functions	4 - 312
	4.17.3	Interactive Programming on Mixed Machines	4 - 313
4.18	**Message Transmission**		4 - 314
	4.18.1	Message Transmission to the Display	4 - 314
	4.18.2	Transmission to Automatic Control Function or Remote Server or Peripheral or PC	4 - 316

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 referenced to the programme origin (OP).

G91	Incremental dimensions with respect to the start of the block.

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

The value is equal to the movement to be performed.

Syntax

N.. **G90/G91** X.. Y.. Z.. A.. B.. C..

G90	Absolute dimensions.
G91	Incremental dimensions.
X.. Y.. Z.. A.. B.. C..	End points.

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

Examples

Absolute dimensions (G90)

Tool positioned at a (start).

Programming point b in absolute dimensions (end point coordinates).

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

Programming in incremental dimensions (G91)

Tool positioned at a (start).

Programming point b in incremental dimensions (movement from a to end point b).

```
N.. (G90) ...
N.. Xa Ya
N.. G91 Xb Yb
N..
```

Programming in absolute dimensions (G90)

Coordinates of points a, b, c, d given with respect to the programme origin (OP) located in the centre of the part.

```
N.. (G90) ...
N.. X20 Y-15
N.. Y20
N.. X-15
N.. X-20 Y-10
N..
```

Programming in incremental dimensions (G91)

Incremental movements from point a to points b, c, d.

```
N.. (G90) ...
N.. X20 Y-15
N.. G91 Y35
N.. X-35
N.. X-5 Y-30
N..
```

4.2 Plane Selection

G17/G18/G19
Plane selection for circular interpolation and radius correction..

One of these G functions is programmed to define the radius correction and circular interpolation plane.

Syntax

N.. **G17/G18/G19**

G17	XY plane selection.
G18	ZX plane selection.
G19	YZ plane selection.

Properties of the Functions

Functions G17, G18 and G19 are modal.

G17 is the default function.

Cancellation

Functions G17, G18 and G19 cancel one another.

Notes

A change of plane must be programmed:
- with the system in state G40 (no radius offset by G41, G42) or the system returns error message 138 (See Sec. 4.8.4),
- after a sequence completely defined in «PGP» (Profile Geometry Programming, see Chapter 5) or the system returns error message 137.

Example

```
N.. ...
N50 G17...                    XY plane selection
N..
N160 G18 ...                  ZX plane selection
N..
N250 G17 ...                  XY plane selection
N..
```

4.3 Spindle Control

4.3.1 Direction of Rotation

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 ...(MILLING CUTTER WITH R/H HELIX)   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.

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) in a mixed machine.

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

Notes

Spindle speed format

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

Example

```
N.. ...
N130 G97 S636 M03                    Spindle rotation
N..
```

Reminder

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

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

Cutting speed V = 20 m/min.
Tool diameter D = 10 mm.

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

$$N = \frac{1000 \times 20}{3.14 \times 10}$$

N = 636.9 rpm, i.e. S636

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

S..	RPM spindle speed (with G97).
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 in a position defined with reference to a fixed point.

4

Syntax

N.. [S..] [M03/M04] [M40 to M45] EC±.. **M19**

S..	RPM spindle speed (with G97).
M03/M04	Spindle direction of rotation.
M40 to M45	Spindle speed ranges.
EC±..	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 M03 M42`	Spindle rotating
`N130 EC90 M19`	Indexed spindle stop
`N..`	

4.3.5 Spindle Control Selection

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

When the machine is equipped with several spindles, these functions set the commands for the spindle servo-drives.

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

Syntax

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

S..	Spindle speed in rpm (with G97).
M03/M04	Spindle direction of rotation.
M40-M45	Spindle speed ranges.
M62	Spindle 3 control selection.
M63	Spindle 4 control selection.
M64	Spindle 1 control selection.
M65	Spindle 2 control selection.

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 a programme (M02), each spindle is assigned to the axis group with the same number (M64 is initialised if there is only one axis group).

Notes

A spindle receives the functions of the axis group to which it is assigned:
- speed G97 S..,
- direction of rotation or spindle stop (M03, M04, M05),
- spindle speed ranges (M40-M45),
- spindle indexing (M19 EC..),
- spindle speed override enabled or disabled (M48 or M49).

The spindle of a group is released by:
- selecting a new spindle (M62-M65),
- release function M61 (see Sec. 4.15.5).

When released from a group, a spindle preserves all the characteristics it had when released (see above) but new functions for the group are no longer addressed to it. They are addressed to the new spindle assigned to the group.

For details on programming spindles with axis multigroups, see Sec. 4.15.

Example

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

4.3.6 Spindle Measurement Selection

M66/M67/M68/M69
Spindle 1 to 4 measurement selection.

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

The spindle characteristics are defined in 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 measurement of each spindle is assigned to the axis group with the same number (example: M66 is assigned to axis group 1). If there is no spindle with the same number as the group, spindle 1 is assigned by default (M66).

Notes

Each axis group can use the measurement of any spindle.

Several groups can use the measurement of the same spindle.

If the measurement of a spindle is used by an axis group for thread cutting, speed override by potentiometer for this spindle is inhibited during the thread cutting cycle (value forced to 100 percent).

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

For programming of spindles with axis multigroups, see Sec. 4.15.

Example

N.. ...	
N180 M67	Assignment of the spindle measurement to group 2
N190 G95 F0.15	Feed in mm/rev related to spindle 2
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, Y, Z primary axes,
- U, V, W secondary axes,
- A, B, C rotary axes.

Syntax

N.. [G90/G91] **G00** [R-/R+] X.. Y.. 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.. Y.. 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 or G03.

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 S600 M40 M03
N40 G00 Xa Ya Za
N50 Zb
N..
```

Rapid retraction after machining

```
N.. ...
N120 G00 Za
N130 Xb Yb
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 read tool radius (see Sec. 4.8.3).

Positioning is only applied to the axes of the plane.

Example

Programming to stop at a 2 mm approach distance in the XY plane (G17).

```
N.. ...
N.. D..
N140 G00 X100 Y30 Z-10
N150 R- X42

N160 ...
N..
```

Tool correction (see Sec. 4.8.3)
Point a
Point b, stop before the programmed point

4.5 Programming 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, Y, Z primary axes,
- U, V, W secondary axes,
- A, B, C rotary axes.

Syntax

N.. [G90/G91] **G01** [R+/R-] X.. Y.. 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.. Y.. 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 interpolations can be executed as combined 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 on the Z, X and Y axes

Groove depth 1.5 mm on machining path a, b.

Programming in absolute dimensions (G90)

```
%20
N10 G00 G52 Z..
N20 ...
N30 S600 M40 M03
N40 X20 Y20 Z2
N50 G01 Z-1.5 F50
N60 X50 Y40 F120
N..
```

Rapid to tool change
Tool call

Approach point a
Penetration in Z
Point b

Programming in Incremental Dimensions (G91)

```
%25
N10 G00 G52 Z..                    Rapid to tool change
N20 ...                            Tool call
N30 S600 M40 M03
N40 X20 Y20 Z2                     Approach point a
N50 G91 G01 Z-3.5 F50              Penetration in Z
N60 X30 Y20 F120                   Point b
N70 G90 ...
```

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

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

Positioning is only applied to the axes of the interpolation plane.

Programming to stop at a 2 mm approach distance in the XY plane (G17).

```
N.. ...
N.. D..
N140 G00 X100 Y30 Z-10
N150 G01 R- X42 F800

N..
```

Tool correction (see Sec. 4.8.3)
Point a
Point b, stop before the programmed point

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 two controlled linear axes depend on the choice of the interpolation plane:

- X (or U) and Y (or V) axes for G17,
- Z (or W) and X (or U) axes for G18,
- Y (or V) and Z (or W) axes for G19.

Syntax (XY plane)

N.. [G17] [G90/G91] **G02/G03** X.. Y.. I..J.. / R.. [F..]

G17	XY plane selection
G90/G91	Programming in absolute or incremental dimensions
G02	Clockwise circular interpolation
G03	Counterclockwise circular interpolation
X.. Y..	End point. - Coordinates of the end point with G90. - Value of the movement with G91.
I.. J..	Location of the circle centre in the XY plane (I along X, J along Y). - With reference to the programme origin with G90. - With reference to the interpolation start point with G91.
R..	Radius of the circle.
F..	Feed rate (see Sec. 4.7).

Syntax according to the plane selected: G17/G18/G19:

Main interpolation plane	Function	Interpolation	Syntax
XY	G17	G02/G03	X.. Y.. I.. J.. or R..
ZX	G18	G02/G03	X.. Z.. I.. K.. or R..
YZ	G19	G02/G03	Y.. Z.. J.. K.. or R..

Properties of the Functions

Functions G02 and G03 are modal.

Cancellation

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

Notes

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

The third axis in the plane can be programmed with circular interpolation to give helical interpolation (see Sec. 4.5.3).

In a block programmed in G02 or G03, all the addresses used for the interpolation must be specified even if they are zero (I0, J0; Plane G17) or unchanged with respect to the previous block (X and Y; Plane G17).

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 J, plane G17)

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 in the plane selected on the following primary and secondary axis pairs:

Function	Main interpolation plane	Axis pairs
G17	XY	XY, UV, UY, XV
G18	ZX	ZX, WU, WX, ZU
G19	YZ	YZ, VW, YW, VZ

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.

Examples

Circular interpolation on paths a, b, a', by programming in absolute dimensions (G90) and in the XY plane (G17)

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 J).

Machining paths

```
%23
N10 G00 G52 Z..                          Rapid to tool change
N20 ...                                  Tool call
N30 S300 M40 M03
N40 X50 Y30 Z2                           Approach point a
N50 G01 Z-2 F50                          Penetration in Z
N60 G02 X100 Y70 R40 F150                Point b
N70 G03 X50 Y30 I60.02 J68.726           Point a'
N80 G00 G52 Z.. M05
N90 M02
```

Circular interpolation on paths a, b, c, by programming in incremental dimensions (G91) and in plane YZ (G19)

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

Machining paths

```
%35
N10 G00  G52 Z..
N20 ...
N30 S600 M40 M03
N40 G19
N50 G16P+
N60 X2 Y15 Z55
N70 G01 X-1.5 F50
N80 G91 G02 Y20 Z-20 J0 K-20 F120
N90 G03 Y20 Z-20 J20 K0
N100 G90 G00 G52 X.. M05
N110 M02
```

Rapid to tool change
Tool call

YZ plane selection
Tool axis orientation (see Sec. 4.8.2)
Approach point a
Penetration in X
Point b
Point c

4.5.3 Helical Interpolation

Helical interpolation is a combination of circular and linear interpolation. The tool axis describes a helix with a constant pitch.

Helical interpolation is carried out in the three planes and applies to the primary and secondary axes.

Syntax (XY plane)

N.. [G17] [G90/G91] **G02/G03** X.. Y.. Z.. I.. J.. / R.. K.. [F..]

G17	G17 XY plane selection
G90/G91	Programming in absolute or incremental dimensions
G02	Clockwise helical interpolation
G03	Counterclockwise helical interpolation
X.. Y..	End point in the XY plane. End point coordinates with G90. Value of the movement with G91.
Z..	Point to be reached on the helix axis (XY plane) with G90 or G91.
I.. J..	Position of the interpolation centre in the XY plane (I along X, J along Y), with reference to the programme origin with G90 or the interpolation start point with G91.
R..	Radius of the circle to be interpolated
K..	Approximate helix pitch in Z (unsigned value)
F..	Feed rate in mm/min or inches/min.

Syntax according to the plane selected: G17/G18/G19:

Interpolation plane	Interpolation	Syntax	Pitch
G17(XY)	G02/G03	X.. Y.. I.. J.. or R..	K
G18 (ZX)	G02/G03	Z.. X.. K.. I.. or R..	J
G19 (YZ)	G02/G03	Y.. Z.. J.. K.. or R..	I

Notes

With helical interpolation:
- The helix pitch applies only to the axis normal to the interpolation plane of the basic reference system.
- Programming of the pitch allows the system to determine the number of revolutions required so that the real pitch is as close as possible to the programmed pitch considering the position of the start and end points.

Programming of additional axes and carrier/carried axis pairs

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

Example:
```
N..
N60 G17
N70 G02 X.. V.. W.. A.. I.. J.. K..
N..
```
- X and V linear axes interpolated circularly
- Linear axis W and rotary axis A interpolated linearly
- Helix pitch K applied to the W axis.

Example:
```
N..
N80 G18
N90 G03 X.. Y.. U.. V.. W.. I.. J..
N..
```
- X and W linear axes interpolated circularly
- Y, U, V linear axes interpolated linearly
- Helix pitch J applied to the W axis.

This type of programming is only possible when the X, U, Y, V axes are independent.

Example

Helical interpolation in the XY plane with programming in absolute dimensions (G90).

```
%55
N10 G00 G52 Z..                          Rapid to tool change
N20 ...                                  Tool call
N30 S200 M40 M03
N40 G00 Xp Yp Zp                         Approach point p
N50 G01 Xa Ya Za                         Start point a
N60 G02 Xb Yb Zb I0 J0 K.. F150          End point
N.. G00 X..
N..
```

Determination of the helix depth

Helix depth in G02 with different helix pitches:

Assume difference between a and b in Z: Δ Z = 20 mm.

Δ Z / K = Approximate number of revolutions.

Pitch K greater than or equal to Z:

Pitch K = 40: The machining from a to b is carried out in 3/4 revolution.

Pitch K = 20: The machining from a to b is carried out in 3/4 revolution.

Pitch K less than Z:

Pitch K = 10: The machining from a to b is carried out in 1 3/4 revolution.
Real helix pitch = 11.4 mm.

Pitch K = 5: The machining from a to b is carried out in 3 3/4 revolutions.
Real helix pitch = 5.33 mm.

Pitch K = 3.8: The machining from a to b is carried out in 5 3/4 revolutions.

REMARK *In the example, Pitch K = 20 is meaningless for the system since it is equal to Δ Z (no error reported).*

Example

Helical interpolation in the XY plane with programming in absolute dimensions (G90).

Execution of a helix on a cylindrical part with a diameter of 200 mm (radius = 100):
- Helix depth = 2 mm
- Real helix pitch = 120 mm
- Positioning clearance in point p on Z = 5 mm
- Start point a: Position Z = 0, angular position = -30 degrees
- End point b: Position Z = -155

Angular distance from a to b: 360° x 155 : 120 = 465° (i.e. one revolution and 105°)

Distance from end point b to point a: -105° + (-30°) = -135°

Position of point p: 360° x 5 : 12 = 15° i.e. p = 30° + 15° = -15°

```
%58
N10 G00 G52 Z..                          Rapid to tool change
N20 ...                                  Tool call
N30 S200 M40 M03
N40 G00 X110 Y110 Z5                     Approach position
N50 G01 G41 X94.661 Y25.364              Point p, (start) in G41 (see Sec. 4.8.4)
N60 G02 X-69.296 Y-69.296 Z-155 I0 J0    Point b, (end)
K120 F300
N70 G01 G40 X-110 Y-110
N..
```

4.5.4 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.. Y.. I.. J.. [F..]

G90/G91	Absolute or incremental programming.
G23	Clockwise or counterclockwise circular interpolation.
X.. Y..	End point.
I.. J..	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.. Y.. F.. I.. J.. Programming incorrect

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

Syntax According to the Plane Selected: G17/G18/G19:

Main interpo-lation plane	Function	Intermediate point	Syntax
XY	G17	I J	G23 X.. Y.. I.. J..
ZX	G18	K I	G23 Z.. X.. K.. I..
YZ	G19	J K	G23 Y.. Z.. J.. K..

Example

Circular interpolation from a through b to c in the XY plane (G17)

```
N..
N130 Z..                          Position on Z
N140 G01 Xa Ya F150               Point a, approach
N150 G23 Xc Yc Ib Jb F100         Circular interpolation
N160 G01 X.. Y.. F150
N..
```

4.5.5 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: XY, ZX or YZ (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).

The polar angle is defined with respect to the reference axis of the interpolation plane:
- X in the XY plane (G17)
- Z in the ZX plane (G18)
- Y in the YZ plane (G19).

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

4.5.5.1 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 (XY plane)

N.. [G17] [G90/G91] G00/G01 EA.. EX.. [F..]

G17	XY 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 G17
N200 X60 Y10 (Point o)
N210 G01 X40 Y10
N220 EA30 EX35
N..
```

Line programmed in incremental dimensions

```
N..
N.. G90 G17
N120 X60 Y10 (Point o)
N130 G01 X40 Y10
N130 G91 EA120 EX15
N..
```

4.5.5.2 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 (XY plane)

N.. [G17] [G90/G91] G02/G03 EA.. EX.. EA.. EI.. [F..]

G17	XY 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 G19 G01 Ya Za
N.. G02 EAb EXb EAc EIc
or
N.. G90 G19 G01 Ya Za
N.. G02 EAb EXb Jc Kc
or
N.. G90 G19 G01 Ya Za
N.. G02 Yb Zb EAc EIc
or
N.. G90 G19 G01 Ya 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 XY plane (G17).

```
N.. G90 G17 G01 Xa Ya
N.. G02 EAb EXb G91 EAc EIc
```

```
N.. G90 G17 G01 Xa Ya
N.. G02 Xb Yb G91 EAc EIc
```

```
N.. G90 G17 G01 Xa Ya
N.. G91 G02 EAb EXb G90 Ic Jc
```

```
N.. G90 G17 G01 Xa Ya
N.. G91 G02 EAb EXb G90 EAc EIc
```

4.5.5.3 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 (XY plane)

N.. [G17] [G90/G91] G02/G03 I.. J.. EA.. [F..]

G17	XY plane selection.
G90/G91	Programming of the circle centre in absolute or incremental dimensions.
G02/G03	Circular interpolation.
I.. J..	Cartesian coordinates of the circle centre in the XY plane (I along X and J along Y): - 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 X to Y, 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 G17 Xa Ya
N.. G03 I.. J.. EA+75
```

```
N.. G90 G17 Xa Ya
N.. G03 I.. J.. EA-75 (or EA+285)
```

```
N.. G90 G17 Xa Ya
N.. G02 I.. J.. EA-75
```

```
N.. G90 G17 Xa Ya
N.. G02 I.. J.. EA+75 (or EA-285)
```

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.

Syntax (XY plane)

N.. [G17] [G90] G02/G03 EA.. EI.. EA.. [F..]

G17	XY plane selection.
G90	Circle centre programmed in absolute dimensions.
G02/G03	Circular interpolation.
EA..	Angle of line EI.
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 X to Y, 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 G17 Xa Ya
N.. G03 EA.. EI.. EA+70
```

```
N.. G90 G17 Xa Ya
N.. G03 EA.. EI.. EA-70 (or EA+290)
```

```
N.. G90 G17 Xa Ya
N.. G02 EA.. EI.. EA-70
```

```
N.. G90 G17 Xa Ya
N.. G02 EA.. EI.. EA+70 (or EA-290)
```

4.5.6 Programming Fillets and Chamfers

4.5.6.1 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.. Y.. I.. J.. / R.. [F..] **EB+..** [EF..]

G01 / G02 / G03	Linear or circular interpolations.
X.. Y..	Programmed intersection point.
I.. J.. / 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. 4.7.5 (feed rate specific to fillets and chamfers).

4.5.6.2 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.. Y.. [F..] **EB-..** [EF..]

G01	Linear interpolation.
X.. Y..	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. 4.7.5 (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.

Syntax

N.. **G09** [G00/G01/G02/G03] X.. Y.. 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.. Y.. 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 Ya F500
N110 G09 Xb Yb
N120 G09 Xc Yc
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 programmed rotary axes alone.

Syntax

N.. **G94** F.. G01/G02/G03 X.. Y.. Z.. A.. B.. 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.. Y.. Z..	End point on linear axes.
A.. B.. C..	Angular end point on rotary axes.

Properties of the Functions

Function G94 is modal. It is the default function.

Reminder

A default value of 1000 mm/min (F1000) is assigned to address F at power on (for further information, see machine parameter P7 in the Parameter Manual).

Cancellation

Function G94 is cancelled by one of functions G93 and 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 G70 (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 state G94, address F.. can be programmed alone in a block).

Example


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

Programming additional axes and carrier/carried axis pairs

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 of a modulo rotary axis programmed alone

Example:

```
N.. G91 G94 F40 G01 B30
```

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

Programming of modulo rotary axis and a linear axis

Example:

```
N.. G91 G94 F100 G01 X10 B30
```

The feed rate on the X axis is expressed in millimetres per minute. The feed rate on the B 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 B axis is equal to 30 deg/6 seconds, i.e. 5 deg/s.

Programming of two modulo rotary axes

Example:

```
N.. G91 G94 F100 G01 A30 B40
```

The feed rates on A and B axes are expressed in degrees per minute.

Execution time : $t = \sqrt{\Delta A^2 + \Delta B^2} / F$ i.e. t = 30 seconds.

The feed rate on each axis is equal to:

- Feed rate on A = $\Delta A/t$, i.e. 60 deg/min
- Feed rate on B = $\Delta B/t$, i.e. 80 deg/min

Reminder

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

Feed rate Fr = N x fz x Z

- Number of teeth of the milling cutter
- Feed rate per tooth
- Rotation speed

Example:
N = 800 revolutions per minute,
fz = 0.05 mm,
Z = 4 teeth

Fr = 800 x 0.05 x 4 = 160 mm/min i.e. F160

Example

Grooving in a path from a to e.

```
%40
N10 G00 G52 Z..                    Rapid to tool change
N20 ...                            Tool call
N30 S800 M40 M03
N40 G94 F100                       Feed rate in mm/min
N50 G00 Xa Ya Z-5
N60 G01 Xb
N70 G02 Xc Yc R20 F160             Change in feed rate
N80 G03 Xd Yd R10 F70              Change in feed rate
N90 G01 Xe Ye F100                 Change in feed rate
N..
```

4.7.2 Inverse Time Feed Rate Coding (V/D)

G93	Inverse time feed rate coding (V/D).

The feed rate is programmed in inverse time when the numerical control cannot calculate the length of a path.

Example: A rotary axis programmed alone or with linear axes.

Syntax

N.. **G93** F.. G01 X.. Y.. Z.. A.. B.. C..

G93	Function setting inverse time feed rate coding: min^{-1}.
F..	Mandatory argument associated with the function defining the feed rate.
G01	Linear interpolation at the programmed feed rate.
X.. Y.. Z..	End point on the linear axes.
A.. B.. C..	Angular end point on the rotary axes.

Property of the Function

Function G93 is modal.

Cancellation

Function G93 is cancelled by one of functions G94 and 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.

Reminder

Determination of the feed rate in V/D = $\frac{1}{\text{time}}$

Feed rate F in V/L = $\frac{V}{D}$ ⟵ feed rate in mm/min
D ⟵ Path length in mm

⚠ CAUTION

Inverse time feed rate coding is prohibited for circular and helical interpolation (no error message).

When the feed rate unit is changed, the G.. function defining the new feed rate unit must be followed by argument F (if the system is already in G93 mode, the F.. address can be programmed alone in a block).

Example

```
N..
N140 G00 X.. Y.. Z..
N150 G94 F200 G01 X.. Y..
N160 Y.. F100
N..
N210 G00 Y.. Z..
N220 G93 F50 G01 U.. C..

N.. C.. F30
N..
```

Feed rate in mm/min

Feed rate in V/D on a secondary axis and a rotary axis

Example

Determination of the feed rate (F) in V/D mode (Velocity/distance) for a groove machined by rotation of the B axis (Required cutting rate, (V) = 150 mm/min)

Groove depth = 5 mm. Part diameter = 200 mm.

Determination of V/D

	Length in mm of the developed arc	Real arc length L in mm	Feed rate in V/D
	$B = \frac{\pi \times D \times ß}{360}$	$L = \sqrt{B^2 + h^2}$	F = V/D
a b	B1 = 87.266	L1 = 118.386	F1 = 1.26
b c	B2 = 43.633	L2 = 43.633	F2 = 3.43
c d	B3 = 130.899	L3 = 134.293	F3 = 1.12

```
%50
N10 (G94)                          Feed rate initialised in mm/min
N20 ...                            Tool call
N30 S500 M40 M03
N50 G00 X0 Z150 B0                 Approach position
N60 Y120 Z102 B20                  Point a
N70 G01 Z95 F150                   Penetration in Z
N80 G93 F1.26 Y40 B70              Point b, feed rate in V/D
N90 B95 F3.43                      Point c, feed rate in V/D
N100 Y10 B170 F1.12                Point d, feed rate in V/D
N110 G00 Z300                      Retraction on Z
N120 Y300 B0 M05
N130 M02
```

4.7.3 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.. Y.. 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.. Y.. Z..	End point on the linear axes.

Property of the Function

Function G95 is modal.

Reminders

- G94 (mm/min) is the default function.
- A default value of 1000 mm/min is assigned to address F (F1000).

Cancellation

Function G95 is cancelled by one of functions G93 and G94.

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 G95 mode, 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 (see Sec. 4.14.4, programming in inches).

Example

```
N..
N.. G00 X.. Y..
N140 G94 F200 G01 Z..                Feed rate in mm/min
N150 X.. Y.. F100
N..
N240 G00 X..  Y..
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..
```

4.7.4 Tangential Feed Rate

G92 R	Programming the 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 3 mm.

```
%22
N10 G00 G52 Z..                          Retraction for tool change
N20...                                   Tool call
N20 T12 D12 M06 (CUTTER DIA=10)
N30 S600 M40 M03
N40 G00 G41 Xa Ya                        Point d, left radius offset
N50 Z-9
N60 G92 R3                               Tangential feed rate limit
N70 G01 Xb F200
N80 G03 Xc Yc R10                        Feed rate applied to point of tangency
N90 G01 Yd
N100 G02 Xe Ye R2                        Feed rate applied to tool centre
N110 G01 Yf
N120 G03 Xg Yg R6.5
N130 G01 Yh
N140 G92 R0                              Tangential feed rate cancelled
N150 G00 G40 G52 Z..
N160 M02
```

4.7.5 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.

Example

Finishing a contour with feed rate EF in the chamfers and fillet (XY plane).

Undimensioned radii = cutter radius

When executing the contour, the linear and circular interpolations are carried out at modal feed rate G94 F120.

%37	
N10 G00 G52 Z..	Tool spindle positioning
N20 ...	Tool call
N30 S800 M40 M03	
N40 G92 R1	
N50 X0 Y-10	Point a
N60 Z-5	
N70 G94 F120	Feed rate for G01, G02 or G03 interpolations
N80 G1 Y15 EB-3 EF90	Point b (feed rate 90)
N90 X20 EB-10 EF70	Point c (feed rate 70)
N100 Y35 EB+5	Point d (feed rate 70)
N110 X40	Point e (feed rate 120)
N120 X50 Y15 EB+20 EF90	Point f (feed rate 90)
N130 X70 EB+7 EF70	Point g (feed rate 70)
N140 G02 X80 Y35 R40 EB+4	Point h (feed rate 70)
N150 G03 X100 Y15 R30	Point i (feed rate 120)
N160 G1 X120	Point j (feed rate 120)
N170 ...	

4.8 Programming of Tools

4.8.1 Tool Change

M06	Tool change

This function calls for a tool to be fitted in the spindle.

The tool is placed in the spindle either automatically or manually.

Syntax

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

T..	The tool is selected by function T followed by a number. The number corresponds to the location of the tool in the machine magazine.
M06	Tool change.
$0 or (...)	Possible message or comment concerning the tool characteristics (see Sec. 4.18).

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 placing the tool in the spindle:
- with reference to the programme origin (OP), or
- with reference to the measurement origin programmed with function G52 (see Sec. 4.12.1).

Example:

```
N..
N120 G00 G52 Z.. or G00 Z..
N130 T09 M06 (CUTTER DIAMETER=25)
N..
```

Examples

Possibilities of placing tools in the spindle according to the type of machine (examples given for reference).

Automatic tool change

```
N..
N100 T05 M06 (CUTTER DIAMETER=20)
N..
```

Tool change by subroutine

```
N..
N80 T06 (CUTTER DIAMETER=50)
N90 G77 H9000
N..
```

Tool change with spindle preselection and indexing

```
N..
N20 M19
N30 T02 M60 (CUTTER DIAMETER=30)
N40 M06 D02
N..
```

4.8.2 Tool Axis Orientation

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

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

The tool axis can be oriented in six different positions on machines with an interchangeable head or a right angle attachment.

Syntax

N.. **G16** P±/Q±/R±

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

Properties of the Function

Function G16 followed by one of its arguments P, Q, R is modal.

Function G16 followed by R+ is the default function.

Cancellation

Function G16 followed by one of arguments (P, Q or R) different from the one already programmed cancels the former state G16.

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 independent 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 Z.. (G17 G16 R+)        Orientation initialised along Z+
N20 T08 ... M06                     Tool change
N30 S400 M40 M03
N..
N..
N170 G00 G52 X.. Y.. Z..
N180 G16 P+                         Axis orientation along X+
N190 G00 Y.. Z..
N200 G01 X.. F..
N..
```

Machine equipped with a right angle head.

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 as a triplet of corrections on the «TOOL CORRECTION» page:
- L = Tool length
- R = Tool radius
- @ = Tool tip radius.

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

Syntax (G17 plane)

N.. [G17] [G16 R+] **D..** [G40/G41/G42] X.. Y.. Z..

G17	XY plane selection.
G16R+	Tool axis orientation along Z+.
D..	Correction number (1 to 255 corrections).
G40	Tool radius offset cancel.
G41/G42	Tool radius offset.
X..Y.. 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 correction triplets (L, R, @). If the number assigned to the correction is higher than 255, the system returns error message 8.

Tool length correction (L)

The tool length correction is assigned to the tool axis orientation defined by G16... (see Sec. 4.8.2).

The tool length declared is taken into account by programming:
- a correction number D...,
- a movement on the axis parallel to the tool axis orientation.

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

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

The maximum dimension of L 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.

Length corrections L.. of tool T02 are taken into account during the first movement on the Z axis programmed after D02 and D12.

```
%55
N10 G16 R+ G17
N20 T02 D02 M06                    Call of tool T02 and correction D02
N30 S180 M40 M03
N40 G00 X100 Y20
N50 Z30                            Taking into account length L.. of D02
N..
N100 D12 Z30                       Taking into account length L.. of D12
N..
```

2D (R) and 3D (@) tool radius correction

REMARK *The call to a 3D tool correction (G29) follows the same rules as a call to a 2D correction R (see Sec. 4.8.5).*

The tool radius correction is assigned to one of the interpolation planes defined by G17, G18 and G19.

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 cancelling 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 9999.999 mm.

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

Example

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

Radius corrections R.. of tool T05 are 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 G17
N20 T05 D05 M06                    Call of tool T05 and correction D05
N30 S180 M40 M03
N40 G00 G41 (or G42) X100 Y50      Activation of radius offset R in D05
N50 Z50
N..
N90 G00 G40 Z60                    D05 radius offset cancel
N100 G41 (or G42) X100 Y50 D15     Activation of radius offset R in D15
N..
N200 G00 G40 Z0                    D15 radius offset cancel
N..
```

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 (XY plane)

N.. [G17] [D..] [G00/G01/G02/G03] **G41/G42** X.. Y..

G17	XY plane selection.
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.. Y..	End point.

G40	Radius offset cancel.

The programmed paths are applied to the centre of the tool.

Syntax

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

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

Properties of the Function

Functions G40, G41 and G42 are modal.

Function G40 is the default function.

Cancellation

Functions G41 and G42 cancel one another.

Function G40 cancels functions G41 and G42 as well as function G29 (3D tool correction).

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.

A change of plane (G17/G18/G19) must always be programmed in G40 mode (tool radius offset cancel). For example:

```
N.. ...
N100 G17 G40 X..
N..
N320 G18 G41 X.. Z..
```

Change of plane before calling the radius correction

```
N..
```

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.

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 (programme variables, see Sec. 6.1),
- E800XX and E8X999 (external parameters, see Sec. 6.2).

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.

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 Ya Za
N50 G01 G41 Xb Yb F..
N60 G03 Xc Yc Io Jo F..
N70 G02 Xd Yd Io' Jo'
N..
```

Engagement on an inside circle in G03

```
N.. ...
N.. D05
N50 G41 Xa Ya Za
N60 G03 Xb Yb Io Jo F..
N70 G03 Xc Yc Ia Ja
N..
```

Contouring a profile with radius correction in the XY plane (G17)

Undimensioned radius = Tool radius

Tool paths (finishing)

```
%30
N10 G00 G52 Z..
N20 T01 D01 M06 (CUTTER DIAMETER=20)
N30 S300 M03 M40
N40 G00 G41 X-15 Y17                    Point a, left radius offset
N50 Z-3                                 Tool position in Z
N60 G01 X10 F100 M08                    Point b
N70 X23 Y35                             Point c
N80 X25 Y17                             Point d
N90 X45                                 Point e
N100 G03 X65 Y17 R12 F50                Point f
N110 G01 X70 Y35 F100                   Point g
N120 X90 Y17                            Point h
N130 G40 X100                           Point i, radius offset cancel
N140 G00 G52 Z.. M05 M09                Retraction
N150 M02
```

Contouring a profile and a groove with radius correction in the XY plane (G17), with tool retraction and repositioning

Tool paths

```
%70
N10 G00 G52 Z..
N20 T03 D03 M06 (CUTTER DIAMETER=9)
N30 S700 M40 M03
$0 PROFILE
N40 G00 G41 X15 Y-5                    Point a in XY, left radius offset
N50 Z9 M08                             Tool position in Z
N60 G01 X5 Y5 F120                     Point b
N70 Y25                                Point c
N80 X60 Y21                            Point d
N90 Y30                                Point e
N100 G02 X75 Y15 R15                   Point f
N110 G03 X85 Y5 R10                    Point g
N120 G00 G40 Z15 M09                   Retraction, radius offset cancel
$0 GROOVE
N130 X40 Y-7                           Point h in XY, left radius offset
N140 Z10 M08                           Tool position in Z
N150 G41 X45                           Point i
N160 G01 Y5                            Point j
N170 X60                               Point k
N180 G03 X60 Y15 R5 F50                Point l
N190 G01 X35 F120                      Point m
N200 Y-10                              Point n
N210 G00 G52 Z.. M05 M09               Retraction
N220 G40 G52 X.. Y..                   Radius offset cancel
N230 M02
```

Facing with radius correction in the XY plane (G17) with change from G41 to G42

```
%72
N10 G00 G52 Z..
N20 T25 D25 M06
N30 S600 M40 M03
N40 G00 G42 X70 Y35
N50 Z15
N60 G01 X-5 F200 M08
N70 Y25
N80 G41 X65
N90 Y15
N100 G42 X-10
N110 G00 Z250 M09
N120 G40 G52 X.. Y.. M05
N130 M02
```

Point a, right radius offset
Tool position in Z
Point b
Point c
Point d, left radius offset
Point e
Point f, right radius offset
Radius offset cancel

Spot facing with radius correction applicable to positioning with stop at programmed feed rate (R+/R-)

Cutter programmed: diameter = 16. Cutter declared in correction D12: diameter = 20 (i.e. radius R = 10).

```
%13
N10 G00 G52 Z..
N20 T12 D12 M06 (CUTTER DIAMETER 16)
N30 S300 M40 M03
N40 G00 X50 Y50 Z3                    Point a, approach
N50 G01 Z-5 F50 M08
N60 R- X75 F100                       Stop before programmed point
N70 G41 G03 X75 Y50 I50 J50 F150      Point b
N80 G00 G40 X50
N90 Z50
N..
```

4.8.5 3D Tool Correction (3 Axes or 5 Axes)

3D tool correction allows machining of 3D linear paths (3 or 5 axes) taking into account the dimensions of the tool used as well as the tool tip shape.

Machining possibilities:
- Toroid or spherical tool (see Sec. 4.8.5.1),
- Cylindrical tool (see Sec. 4.8.5.2).

4.8.5.1 3D Tool Correction with Toroid or Spherical Tool

G29	3D tool correction (3 or 5 axes) with toroid or spherical tool.

3D tool correction allows machining of 3D linear paths taking into account the dimensions of the toroid or spherical tool used.

3-axis tool correction

For 3-axis correction, the tool axis is parallel to one of the axes of the basic coordinate system defined by the tool axis orientation function G16 ... (see Sec. 4.8.2).

With each programmed point is associated the «material» vector « $\vec{n}$ » normal to the surface to be machined, defined by its P, Q and R coordinates.

5-axis tool correction

For 5-axis correction, the tool axis can be inclined when the machine is equipped with a double twist machining head.

With each programmed point are associated the vector « $\vec{n}$ » normal to the surface to be machined, defined by its P, Q and R coordinates, and the tool orientation vector « $\vec{o}$ », defined by its I, J and K components, plus the twist head angles where required.

Syntax

N.. [D..] [G01] **G29** X.. Y.. Z.. P.. Q.. R.. [I.. J.. K..] [A.. / B.. / C..]

D..	Correction number.
G01	Linear interpolation.
G29	3D tool correction with toroid or spherical tool.
X.. Y.. Z..	End point.
P..Q.. R..	Components of the normal vector $\vec{n}$ (material vector) whose origin is defined in the block by the X, Y, Z coordinates of the end point (mandatory in each block).
I.. J.. K..	Components of the tool orientation vector $\vec{o}$ with 5-axis correction (mandatory in each block).
A.. / B.. / C..	Twist head angles with 5-axis correction: A: angle on X, B: angle on Y, C: angle on Z.

Properties of the Function

Function G29 is modal. None of the arguments associated with the function are modal.

Cancellation

Function G29 is cancelled by function G40 or one of functions G41 or G42.

Notes

The presence or absence of vector I J K in a block allows the distinction to be made between 3-axis and 5-axis tool corrections.

For 3D correction, the two axes of the basic three-axis system other than the tool axis may be impacted by tool radii R and @, be these axes primary, secondary, carried or independent.

3D correction can be made on a single point (possibly in MDI manual data input mode).

Concept of Surface and Normal Vector

The types of surfaces machined using G29 (irregular surfaces) and the programming requirements (normal vector) mean that 3D correction can only be used with machining programmes in symbolic language.

Machining is executed by scanning «TABCYL» (word of the APT language defining a tabulated cylinder) by consecutive linear interpolations.

The consecutive programme blocks contain the coordinates of the points. The unit normal vector in each point always points outwards from the part.

The spacing between the programmed points varies according to the required machining accuracy.

For 3-axis correction, the material vector defined by components P, Q and R must have a length of 100 mm within 1 mm (i.e. +/-0.1%). Otherwise, the system returns error message 145.

The tool correction is applied along the normal vector below:

$$\sqrt{P^2 + Q^2 + R^2} = 1000 \text{ mm}$$

	X1		X2		X3
a	Y1	b	Y2	c	Y3
	Z1		Z2		Z3

With 5-axis correction, the material vector defined by P, Q and R and the tool direction defined by I, J and K can have any length (the lengths are normed by the system), but the three components of the two vectors must be programmed in each block. Otherwise, the system returns error message 146.

Toroid and Spherical Tools

For 3D correction, the three corrections to be declared are:
- L..: tool length,
- R..: tool radius,
- @..: tool tip radius.

REMARK *The tool radius «R» must not be confused with component R of normal vector $\vec{n}$ «R».*

Toroid tool

Spherical tool

Double Twist Machine Head

Head used with 5-axis correction.

The twist head angles α and β are defined by a pair of rotary axes defined in the NC, i.e.:

A and B on X and Y,

A and C on X and Z,

B and C on Y and Z.

Examples

3-axis correction

```
N..
N.. G01 ...
N240 D17
N250 G29 X.. Y.. Z.. P.. Q.. R..
N260 X.. Y.. Z.. P.. Q.. R..
N..
N..
N890 G00 G40 X.. Y..
N..
```

5-axis correction

Twist head axes: A and C.

```
N..
N.. G01 ...
N320 D15
N330 G29 X.. Y.. Z.. P.. Q.. R.. I.. J.. K.. A.. C..
N340 X.. Y.. Z.. P.. Q.. R.. I.. J.. K..
N350 X.. Y.. Z.. P.. Q.. R.. I.. J.. K..
N360 X.. Y.. Z.. P.. Q.. R.. I.. J.. K.. A.. C..
N..
N..
N620 G00 G40 X.. Y..
N..
```

Geometric Transformations with 3-Axis Correction

The system computes the tool reference position «N» according to the tangency point «M» and the vector normal to the surface « $\vec{n}$ ».

$\overrightarrow{MN} = \overrightarrow{MC} + \overrightarrow{CA} + \overrightarrow{AN}$

$\overrightarrow{MC}$: Vector with length r(@) collinear with vector $\vec{n}$.

$\overrightarrow{CA}$: Vector with length (D/2 - r) collinear with the projection of vector $\vec{n}$ on the interpolation plane (XY in the figure).

$\overrightarrow{AN}$: Vector with length (L - r) collinear with the tool axis orientation vector (defined by G16..).

Computation mode

$$\overrightarrow{MC} \left| \begin{array}{l} \dfrac{r\,.\,P}{\sqrt{P^2+Q^2+R^2}} = r\,.\,P \\ \dfrac{r\,.\,Q}{\sqrt{P^2+Q^2+R^2}} = r\,.\,Q \\ \dfrac{r\,.\,R}{\sqrt{P^2+Q^2+R^2}} = r\,.\,R \end{array} \right.$$

$$XN = XM + \frac{r.P}{1000} + \left(\frac{D}{2} - r\right)\frac{P}{\sqrt{P^2+Q^2}}$$

$$YN = YM + \frac{r.Q}{1000} + \left(\frac{D}{2} - r\right)\frac{Q}{\sqrt{P^2+Q^2}}$$

$$ZN = ZM + \frac{r.R}{1000} + (L - r)$$

$$\overrightarrow{CA} \left| \begin{array}{l} \left(\dfrac{D}{2} - r\right)\dfrac{P}{\sqrt{P^2+Q^2}} \\ \left(\dfrac{D}{2} - r\right)\dfrac{Q}{\sqrt{P^2+Q^2}} \\ 0 \end{array} \right.$$

$$\overrightarrow{AN} \left| \begin{array}{l} 0 \\ 0 \\ (L - r) \end{array} \right.$$

Reminder

In the above equations, P, Q, R are the components of the normal vector with a length of 1000 mm.

Geometric Transformations with 5-Axis Correction

The system computes the tool reference position «N» according to the tangency point «M», the vector normal to the surface « $\vec{n}$ » and the tool orientation vector « $\vec{o}$ » (same computation principle as for 3-axis correction).

M: programmed point

B: point displayed on the current position coordinate page (AXIS) with respect to OP

N: point controlled by the system

4.8.5.2 3D Tool Correction with Cylindrical Tool

G43	3D tool correction (3 or 5 axes) with cylindrical tool.

Syntax

N.. [D..] [G01] **G43** X.. Y.. Z.. P.. Q.. R.. [I.. J.. K..] [A.. B.. C..]

D..	Correction number.
G01	Linear interpolation.
G43	3D tool correction with cylindrical tool.
X.. Y.. Z..	Point programmed on the surface.
P.. Q.. R..	Components of the material vector whose norm 1000 positions the centre of the tool tip with respect to the programmed point (the offset is obtained by the vector components divided by 1000 and multiplied by the tool radius) (mandatory in each block).
I.. J.. K..	Components of the tool vector normed at 1 by the system giving the tool axis orientation (see Notes).
A.. / B.. / C..	Twist head angles with 5-axis correction: A: angle on X, B: angle on Y, C: angle on Z.

Properties of the Function

Function G43 is modal.

Cancellation

Function G43 is cancelled by function G40.

Notes

With RTCP and twist axes, the tool direction (I J K) does not need to be programmed as it is included in the offset processed by the RTCP function (if this vector is programmed, it is ignored).

Without RTCP and twist axes, if vector I J K is not specified, the tool direction is assumed to be paraxial and is given by function G16.

For information on the RTCP function, see the Supplementary Programming Manual.

If one of the components of vector PQR is missing or if only one of the components of vector IJK is specified (but not all three), the system returns error message 149.

4.9 Basic Cycles

4.9.1 Cycle Overview

G8x (and G31)	Canned cycles in the tool axis.

Axes programmable with basic cycles:
- X, Y, Z primary axes,
- U, V, W secondary axes,
- rotary axes A, B or C are used only for positioning.

Syntax (XY plane)

N.. [G17] **G8**x [X.. Y..] Z.. [ER..] [EH..] F..

G17	XY plane selection.
G8x	Canned cycle.
X.. Y..	Tool position in the plane.
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.
F..	Feed rate in the cycle.

Properties of the Functions

Functions G8x are modal.

Cancellation

Function G8x is cancelled by function G31 or another function G8x.

Notes

When a cycle (G8x and G31) 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 the secondary axis (W) 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 positioning (linear or circular) in the plane, then on EH..

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

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

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.*

Cycle sequencing

The following addresses are not modal in cycle sequences with positioning by circular interpolation:
- I.. J.. K.. : Centre of the circle
- R.. : Radius of the circle

4.9.2 Cancellation of a Canned Cycle

G80	Canned cycle cancel.

This function is used to cancel canned cycles.

Syntax

N.. **G80**

G80	Canned cycle cancel.

Property of the Function

Function G80 is modal. It is the default function.

Cancellation

Function G80 is cancelled by one of functions G31, G81-G89.

Notes

Including function G80 in a cycle subroutine makes the cycle nonmodal.

Example

```
N..
N120 G00 X.. Y.. Z..        Tool positioning
N130 G81 Z-10 F100          Drilling cycle
N140 G80 G00 Z200           Cycle cancel
N..
```

4.9.3 Centre Drilling Cycle

G81	Centre drilling cycle.

Syntax (XY plane)

N.. [G17] **G81** [X.. Y..] Z.. [ER..] [EH..] [F..]

G17	XY plane selection.
G81	Centre drilling cycle.
X.. Y..	Tool position in the plane.
Z..	End point on a machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
F..	Feed rate in the cycle.

Property of the Function

Function G81 is modal.

Cancellation

Function G81 is cancelled by one of functions G31, G80, G82-G89.

Cycle Steps

Step 1: Rapid positioning in the plane.

Step 2: Infeed at feed rate F..

Step 3: Rapid retraction in the tool axis.

Example

Two centre drilling operations (XY plane).

```
N.. ...
N50 G00 Xa Ya Za
N60 G81 Z-8 F80
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G81 Xa Ya ERa Z-8 F80
N70 Xb Yb
N80 G80 G00 Z..
N..
```

4.9.4 Counterboring Cycle

G82	Counterboring cycle.

Syntax (XY plane)

N.. [G17] **G82** [X.. Y..] Z.. [ER..] [EH..] EF.. [F..]

G17	XY plane selection.
G82	Counterboring cycle.
X.. Y..	Tool position in the plane.
Z..	End point in the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
EF..	Mandatory dwell time in seconds (maximum 99.99 s, format EF022).
F..	Feed rate in the cycle.

Property of the Function

Function G82 is modal.

Cancellation

Function G82 is cancelled by one of functions G31, G80-G81, G83-G89.

Cycle steps

Step 1: Rapid positioning in the plane.

Step 2: Infeed at feed rate F..

Step 3: Dwell at end of drilling or counterboring.

Step 4: Rapid retraction in the tool axis.

Example

Two counterboring operations (XY plane).

```
N.. ...
N50 G00 Xa Ya Za
N60 G82 Z-5 EF2 F60
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G82 Xa Ya ERa Z-5 EF2 F60
N70 Xb Yb
N80 G80 G00 Z..
N..
```

4.9.5 Peck Drilling Cycle

G83	Peck drilling cycle.

Syntax (XY plane)

N.. [G17] **G83** [X.. Y..] Z.. [ER..] [EH..] [P..] / [ES..] [Q..] [EP..] [F..] [EF..]

G17	XY plane selection.
G83	Peck drilling cycle.
X.. Y..	Tool position in the plane.
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 (optional).
EP..	Backoff after each peck (default 1 mm).
F..	Feed rate in the cycle.
EF..	Dwell at the end of each infeed.

Property of the Function

Function G83 is modal.

Cancellation

Function G83 is cancelled by one of functions G31, G80-G82, G84-G89.

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: Rapid positioning in the plane.

Step 2: First peck to depth P.. at feed rate F..

Rapid retraction to the start point in the tool axis.

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

Step 3: Second peck at feed rate F..

Rapid retraction to the start point in the tool axis.

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

Steps 4 and 5: Pecks and retractions same as step 2.

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

Step 7: Rapid retraction in the tool axis.

Possible dwell G04 F.. in the start point.

Example

Machining two holes (XY plane).

```
N.. ...
N50 G00 Xa Ya Za
N60 G83 Z-50 P10 Q5 F50
N70 Xb Yb
N80 G80 G00 Z.
N..
```

or

```
N.. ...
N60 G83 Xa Ya ERa Z-50 P10 Q5 F50
N70 Xb Yb
N80 G80 G00 Z..
N..
```

4.9.6 Tapping Cycles

4.9.6.1 Tapping Cycle

G84	Tapping cycle.

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

Syntax (XY plane)

N.. [G17] **G84** [X.. Y..] Z.. [ER..] [EH..]EF.. [F..]

G17	XY plane selection.
G84	Tapping cycle.
X.. Y..	Tool position in the plane.
Z..	End point in the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
EF..	Dwell time in seconds (maximum 99.99 s, format EF022, default 1 second).
F..	Feed rate in the cycle.

Property of the Function

Function G84 is modal.

Cancellation

Function G84 is cancelled by one of functions G31, G80-G83, G84-G89.

Notes

In this tapping cycle, the feed rate is not coupled to the spindle rotation speed. The tap must be allowed to float to compensate for position errors.

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

Calculation of the feed rate in mm/min

F.. = Tap pitch (in mm) x spindle rotation speed (RPM).

Cycle steps

Step 1: Rapid positioning in the plane.

Step 2: Infeed at calculated feed rate F..

Step 3: Reversal of the direction of rotation at depth.

Step 4: Dwell time at depth.

Step 5: Retraction at calculated feed rate F.. in the tool axis.

Example

Machining of two M8 tapped holes, pitch 1.25 mm (XY plane).

```
N.. ...
N40 S300 M41 M03
N50 G00 Xa Ya Za
N60 G84 Z-20 EF1 F375
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G84 Xa Ya ERa Z-20 EF1 F375
N70 Xb Yb
N80 G80 G00 Z..
N..
```

4.9.6.2 Rigid Tapping Cycle

G84	Rigid tapping cycle.

With this cycle, the spindle speed controls the tap feed. The feed rate is automatically calculated from the actual spindle speed and pitch.

Syntax (XY plane)

N.. [G17] [M03/M04] [S..] [M40 to M45] **G84** [X.. Y..] Z.. [ER..] [EH..] K.. [EK..]

G17	XY plane selection.
M03/M04	Spindle direction of rotation
S..	Spindle rotation speed.
M40-M45	Spindle ranges.
G84	Tapping cycle.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
K..	Tap pitch in mm (K specifies rigid tapping).
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 G31, G80-G83, G85-G89.

Notes

When the cycle is called, the tool axis is coupled with spindle rotation.

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

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.

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: Tool positioning in the hole axis (provide clearance).

Step 2: Infeed with acceleration of the spindle 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 two M10 rigid tapping operations, pitch 1.50 mm (XY plane)

```
N.. ...
N50 S200 M41 M03
N60 G84 Xa Ya ERa Z-20 K1.5 EK2
N70 Xb Yb
N80 G80 G00 Z..
```

Execution of one rigid tapping operation with several consecutive penetration depths.

```
N.. ...
N40 S400 M41 M03
N50 G00 Xa Ya Za
N60 G84 Zb K.. EK..
N70 Zc
N80 Zd
N90 G80 G00 Z..
N..
```

4.9.7 Reaming Cycle

G85	Reaming cycle.

Syntax (XY plane)

N.. [G17] **G85** [X.. Y..] Z.. [ER..] [EH..] [F..] [EF..]

G17	XY plane selection.
G85	Reaming cycle.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
EH..	Impact plane in the machining axis.
ER..	Retraction plane in the machining axis.
F..	Feed rate in the cycle.
EF..	Retraction rate (default = feed rate F..)

Property of the Function

Function G85 is modal.

Cancellation

Function G85 is cancelled by one of functions G31, G80-G84, G86-G89.

Cycle Steps

Step 1: Rapid positioning in the plane.

Step 2: Infeed at feed rate F..

Step 3: Retraction at feed rate F.. in the tool axis.

Example

Execution of two reaming operations (XY plane).

```
N.. ...
N50 G00 Xa Ya Za
N60 G85 Z-25 F80
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G85 Xa Ya ERa Z-25 F80
N70 Xb Yb
N80 G80 G00 Z..
```

4.9.8 Boring Cycle with Indexed Spindle Stop at the Bottom of the Hole

G86	Boring cycle with indexed spindle stop at the bottom of the hole.

Syntax (XY plane)

N.. [G17] **G86** [X.. Y..] Z.. [ER..] [EH..] [EC..] [EA..] [EP..] [F..]

G17	XY plane selection.
G86	Boring cycle with indexed spindle stop at the bottom of the hole.
X.. Y..	Tool position in the plane.
Z..	Machining end point.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis
EC..	Value of the indexing position (the default value of EC is the last indexing position programmed).
EA..	Angle between EC.. programmed and the physical angular position of the cutting edge
EP..	Lateral backoff in the hole bottom (default 2 mm)
F..	Feed rate in the cycle.

Property of the Function

Function G86 is modal.

Cancellation

Function G86 is cancelled by one of functions G31, G80-G85, G87-G89.

Cycle Steps

Step 1: Rapid positioning in the plane.

Step 2: Infeed at feed rate F..

Step 3: Indexed spindle stop at depth.

Step 4: Lateral backoff 2 mm (or EP..) on the indexing axis.

Step 5: Rapid retraction in the tool axis.

Example

Execution of two bores (XY plane).

```
N.. ..
N50 G00 X.. Y.. Za
N60 G86 Xa Ya Z-10 EC.. F30
N70 Xb Yb
N80 G80 G00 Z..
N..
```

4.9.9 Drilling Cycle with Chip Breaking

G87	Drilling cycle with chip breaking.

Syntax (XY plane)

N.. [G17] **G87** [X.. Y..] Z.. [ER..] [EH..] [P..] / [ES..] [Q..] [EP..] [EF..] [F..]

G17	XY plane selection.
G87	Drilling cycle with chip breaking.
X.. Y..	Tool position in the plane.
Z..	End point in 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 (optional).
EP..	Backoff between two infeeds (default no backoff, EP = 0).
EF..	Mandatory dwell time in seconds.
F..	Feed rate in the cycle.

Property of the Function

Function G87 is modal.

Cancellation

Function G87 is cancelled by one of functions G31, G80-G86, G88-G89.

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.

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 for reference. The number of steps depends on the values programmed with the cycle.

Step 1: Rapid positioning in the plane.

Step 2: Infeed to depth P.. at feed rate F..

Dwell at the end of each infeed (possible backoff by value EP..).

Steps 3 and 4: Successive infeeds and dwells (with possible backoffs by value EP..) same as step 2.

Step 5: Infeed to depth Q.. at feed rate F..

Dwell at depth.

Step 6: Rapid retraction in the tool axis.

Possible dwell G04 F.. at the start point.

Example

Execution of two holes (XY plane).

```
N.. ...
N50 G00 Xa Ya Za
N60 G87 Z-40 P10 Q5 EF1 F40
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G87 Xa Ya ERa Z-40 P10 Q5 EF1 F40
N70 Xb Yb
N80 G80 G00 Z..
N..
```

4.9.10 Boring and Facing Cycle

G88	Boring and facing cycle.

Syntax (XY plane)

N.. [G17] **G88** [X.. Y..] Z.. [ER..] [EH..] [F..]

G17	XY plane selection.
G88	Boring and facing cycle.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
F..	Feed rate in the cycle.

Property of the Function

Function G88 is modal.

Cancellation

Function G88 is cancelled by one of functions G31, G80-G87, G89.

Cycle Steps

Step 1: Rapid positioning in the plane.

Step 2: Infeed at feed rate F..

Step 3: Feed stop at depth, spindle still rotating.

Display of the message: «FACING COMPLETED? (Y):»

Step 4: Operator action for facing.

The operator answers the message by pressing Y (yes) to enable continuation to the next step (5).

Step 5: Retraction at rapid in the tool axis.

REMARK *The message «FACING COMPLETED? (Y):» included in the cycle deletes the message, if any, programmed by address «$0» (see Sec. 4.18).*

Example

Execution of two boring and facing operations (XY plane).

```
N.. ...
N50 G00 Xa Ya Za
N60 G88 Z-12 F30
N70 Xb Yb
N80 G80 G00 Z..
N.
```

or

```
N.. ...
N60 G88 Xa Ya ERa Z-12 F30
N70 Xb Yb
N80 G80 G00 Z..
N..
```

4.9.11 Boring Cycle with Dwell at the Bottom of the Hole

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

Syntax (XY plane)

N.. [G17] **G89** [X.. Y..] Z.. [ER..] [EH..] [EF..] [F..]

G17	XY plane selection.
G89	Boring cycle with dwell in the bottom of the hole.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
ER..	Retraction plane in the machining axis.
EH..	Impact plane in the machining axis.
EF..	Dwell expressed in seconds (maximum 99.99 s, format EF022, default EF = 1 second).
F..	Feed rate in the cycle.

Property of the Function

Function G89 is modal.

Cancellation

Function G89 is cancelled by one of functions G31, G80-G88.

Cycle Steps

Step 1: Rapid positioning in the plane.

Step 2: Infeed at feed rate F..

Step 3: Dwell at the end of boring.

Step 4: Rapid retraction in the tool axis.

Example

Execution of two bores (XY plane).

```
N.. ...
N50 G00 Xa Ya Za
N60 G89 Z-25 EF1 F80
N70 Xb Yb
N80 G80 G00 Z..
N..
```

or

```
N.. ...
N60 G89 Xa Ya ERa Z-25 EF1 F80
N70 Xb Yb
N80 G80 G00 Z..
N..
```

4.9.12 Thread Chasing Cycle

G31	Thread chasing cycle.

This cycle couples the tool feed rate to the spindle rotation.

Syntax (XY plane)

N.. [G17] [M03/M04] [S..] **G31** [X.. Y..] Z.. [ER..] [EH..] K.. P.. [F..] [EF..] [EC..]

G17	XY plane selection.
M03/M04	Spindle rotation.
S..	Spindle rotation speed.
G31	Thread chasing cycle.
X.. Y..	Tool position in the plane.
Z..	End point on the machining axis.
ER..	Approach or retraction dimension on the machining axis.
EH..	Impact plane in the machining axis.
K..	Thread pitch in mm. K = pitch with XY plane (G17) J = pitch with ZX plane (G18) I = pitch with YZ plane (G19).
P..	Absolute tool retraction dimension at the end of thread cutting.
F..	Number of thread starts (1 to 9, default 1).
EF..	Dwell time in seconds (maximum 99.99 s, format EF022, default value equal to two spindle rotations).
EC..	Value of the indexing position (the default value of EC is the last indexing position programmed).

Property of the Function

Function G31 is modal.

Cancellation

Function G31 is cancelled by one of functions G80-G89.

Notes

G31 needs encoder feedback from the spindle. The definition of this device (number of lines per revolution) is established by the machine manufacturer.

The crossing of an angular position (monitored by the system) starts movements for thread cutting.

With block continuation, it is possible to change the plane and machining axis.

Clearance at the end of thread cutting

Position 1 in the figure:

Clearance along the two axes at 45 degrees, in this case $X = Y = \frac{P}{\sqrt{2}}$.

Position 2 on the figure:

Clearance along the X axis. In this case, X = P.

Position 3 in the figure:

Clearance along the Y axis. In this case, Y = P.

REMARK *Movements P are the same in the three planes (XY, ZX, YZ).*

Cycle Steps

Step 1: Rapid positioning of the chaser-holder to the centre of the hole then dwell for 1.2 seconds to start spindle rotation.

Step 2: Infeed at the pitch programmed.

Step 3: Dwell or two spindle rotations at the end of thread cutting.

Step 4: Indexed spindle stop at position zero of the position encoder.

Step 5: Rapid clearance by the value P in the axis of the plane and direction defined by the machine manufacturer.

Step 6: Rapid retraction in the tool axis.

Step 7: Rapid repositioning of the tool to the hole centre and restart of spindle rotation.

REMARK *For thread cutting with several starts, the cycle is repeated after changing the angular position for starting to cut each thread.*

Example

Execution of thread chasing, pitch=3, two starts (XY plane).

```
N..
N60 T09 D09 M06 (CHASER R2)
N50 G00 X.. Y.. Za
N60 G31 Xa Ya Z-20 K3 F2 P4 EF1
N70 G80 G00 Z..
N..
```

Cycle repeat to achieve the thread cutting precision

```
N..
N.. G00 X.. Y.. Z..                          Chaser-holder approach
N100 S300 M40 M03 G31 X.. Y.. Z.. K.. P..    Cycle
N110 G00 Z200 M00 $0 CONTROLE                Retraction and spindle stop and display
                                             of a message (see Sec. 4.18)
/N120 G79 N100 G04 F0.5
N130 ...
N..
```

Retraction and spindle stop in N110

Check of thread cutting precision.

If thread cutting is correct:

Confirm by pressing the block skip key «/» on the NC and restart the cycle.

The system skips block N120 and goes to block N130 (do no forget to inhibit block skip «/» for the rest of the programme).

If thread cutting is not correct:

Adjust the chaser. Do not press the block skip key «/».

Restart the cycle. When reaching block N110, the system jumps to block N100 and repeats the cycle.

At the end of block N100, select one of the two procedures after checking thread cutting precision again.

REMARK *It is necessary to programme a dwell in N120 so that the cycle can be repeated.*

4.9.13 Table Summarising Cycles G81 to G89

CYCLES / BREAKDOWN OF MOVEMENTS		G81	G82	G83	G84	G85	G86	G87	G88	G89
		Centre drilling	Counter-boring	Peck drilling	Tapping	Reaming	Boring with indexed spindle stop	Drilling with chip-breaking	Boring and facing	Boring with dwell in bottom of hole
Downward		Work	Work	Rapid then work with n consecutive penetrations (P Q)	Work	Work	Work	Work with n consecutive penetrations (P Q)	Work	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 BOT-TOM	Dwell		Programmed by EF	Programmed by EF on combined machine used for turning	Programmed by EF					Programmed by EF
	Spindle				Rotation reversal		Indexed stop programmed by EC			
	Retraction in the plane						Fixed or programmed by EP			
Upward Rapid		Rapid	Rapid	Rapid	Work	Work rate or other feed rate if EF present	Rapid	Rapid at the end of penetration	Rapid after confirmation by the operator	Work
End of upward travel					Spindle rotation reversal		Rapid return of the X and/or Y axes in the boring axis and reset of the spindle to the initial state			

4

4.9.14 Examples of Programming Cycles G81-G89

Sequence of drilled holes in tool axis Z (or W) in the XY plane (G17).

Drilling to different depths and positioning of the tool on different levels with ER.

```
%56
N10 G00 G52 Z..
N20 T12 D12 M06 (DRILL)
N30 S1000 M40 M03
N40 G00 Xa Ya Z50                    Tool approach
N50 G81 Z-18 ER5 F80                 Drilling cycle
N60 Xb Yb                            Cycle
N70 Xc ER-7 Z-28                     Cycle
N80 Yc'                              Cycle
N90 ER12
N100 G83 Xd Yd Z-18 P7 F100          Peck drilling cycle
N110 G81 Xe ER-17 Z-38               Drilling cycle
N120 Xf Yf Z-28                      Cycle
N130 G80 G00 G52 Z..
N..
```

Sequence of drilled holes in tool axis Y (or V) in the ZX plane (G18)

Drilling to different depths and positioning of the tool on different levels with ER.

```
%68
N10 G18 G00 G52 Y..                    ZX plane selection
N20 G16 Q+                             Tool axis orientation
N30 T05 D05 M06 (DRILL)
N40 S600 M40 M03
N50 G00 Xa Y53 Za                      Tool approach
N60 G87 Y10 P8 Q5 EF1 F60              Cycle
N70 Xb Y20 Zb                          Cycle
N80 Xc Y10 Zc ER28                     Cycle
N90 G80 G00 G52 Y..
N..
```

Sequence of drilled holes in tool axis Z (or W) in the XY plane (G17)

Drilling four holes with positioning by circular interpolation inside a groove.

```
%45
N10 G00 G52 Z..
N20 T02 D02 M06 (DRILL)
N30 S1000 M40 M03
N40 G81 Xa Ya ER20 Z10 F80          Drilling cycle
N50 G02 Xb Yb I0 J0 (or R)          Cycle
N60 Xc Yc I0 J0 (or R)              Cycle
N70 Xd Yd I0 J0 (or R)              Cycle
N80 G00 G80 G52 Z..
N..
```

Sequence of drilled holes in tool axis Z (or W) in the XY plane (G17) on a machining center with rotary axis B.

Origin offset on Z.

Drilling to different depths, positioning by rotating the B axis.

```
%89
N10 G00 G52 Z..
N20 T08 D08 M06 (DRILL)
N30 S900 M40 M03
N40 G00 X0 Y0 Z3 B0
N50 G82 Z-15 EF1.5 F70          Drilling cycle
N60 B42 Z-8                     Cycle
N70 B90 Z-15                    Cycle
N80 G80 G00 G52 Z..
N..
```

4.10 Other Cycles

4.10.1 Simple Pocket Cycle

G45	Simple pocket cycle.

This cycle is used to machine circular, oblong, rectangular and square pockets.

The primary and secondary axes are programmable in absolute dimensions and define the pocket center in the plane and the pocket depth in the tool axis.

Syntax (XY plane)

N.. [G17] **G45** X.. Y.. Z.. [ER..] EX.. EY.. [EB..] P.. Q.. [I..] [J..] [EG2/EG3] EP.. EQ.. EI.. EJ..

G17	XY plane selection.
G45	Pocket cycle.
X.. Y..	Position of the pocket center.
Z..	Pocket bottom end point
ER..	Retraction plane in the tool axis.
EX..	Pocket dimension on the X or U axis.
EY..	Pocket dimension on the Y or V axis.
EB..	Radius of a circular pocket if EB is programmed alone. Radius of an oblong pocket. Fillet for other pockets.
P..	Axial roughing path.
Q..	Lateral roughing path.
I..	Axial finishing path.
J..	Lateral finishing path.
EG2/EG3	Pocket cutting direction (default EG3) - EG2: Clockwise (work against feed) - EG3: Counterclockwise (work in feed direction, by "swallowing").

EP..	Value of the axial roughing feed rate.
EQ..	Value of the lateral roughing feed rate.
EI..	Value of the axial finishing feed rate.
EJ..	Value of the lateral finishing feed rate.

Properties of the Function

Function G45 is nonmodal. None of the arguments in the cycle are modal.

Cancellation

Function G45 is cancelled at the end of the block.

Notes

When the start of spindle rotation is programmed in the block, it must be placed before function G45 and its arguments.
Example: N.. S1000 M03 M40 G45...

If tool correction D.. is missing when cycle G45 is called, the system returns error message 898.

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

When executing a cycle programmed with axial or lateral roughing and/or finishing passes, if only one feed rate (EP, EQ, EI or EJ) is programmed, this feed rate is used as default.

If no:
- axial feed rate (roughing or finishing) is programmed, the system returns error message 892,
- lateral feed rate (roughing or finishing) is programmed, the system returns error message 893.

The cycle calls the axis address equivalence table (see Sec. 6.5). Use of the table is incompatible with the cycle call (it is initialised at the beginning of the cycle and restored initialised at the end of the cycle).

In the ZX and YZ planes, the pocket dimension in the Z (or W) axis is programmed by EZ..

During machining:
- The cycle cannot be interrupted until the pocket cycle is completed on a depth level (no possibility of mode change).
- The cycle cannot be modified until it has been completely executed.

Pass setting possibilities

Roughing cycle with a single tool

Cycle with axial (P) and lateral (Q) roughing parameters.

Example:
G45 X.. Y.. Z.. ER.. EX.. EY.. P.. Q.. EP.. EQ..

Roughing Cycle Steps

Step 1: Rapid positioning of the tool to the center of the pocket.

Step 2: Rapid axial positioning in the tool axis.

Step 3: Axial penetration at the programmed feed rate to depth P.

Step 4: Lateral positioning at the programmed feed rate to value Q (along the small side).

Execution of the first pocket contouring (and subsequent contourings if any).

Step 5: Lateral positioning on the final contour by value Q.

Execution of the last contouring to the external pocket dimensions.

Step 6: Rapid repositioning to the pocket center for execution of a new plunge and a new contouring to depth P following the same steps as 3, 4, 5.

Step 7: After execution of the last contouring to depth Z, the tool is rapidly repositioned to the pocket center with a clearance of 1 mm in the tool axis then retracted to programmed position ER.

Roughing and finishing cycle with a single tool.

Cycle with axial (P) and lateral (Q) roughing and axial (I) and lateral (J) finishing parameters.

Example:

```
G45 X.. Y.. Z.. ER.. EX.. EY..
P.. Q.. I.. J.. EP.. EQ.. EI.. EJ..
```

Finishing cycle alone with a new tool

Cycle with depth (P) and lateral finishing (J).

Example:

```
G45 X.. Y.. Z.. ER.. EX.. EY..
P.. J.. EP.. EJ..
```

Oblong, square and rectangular pockets are roughed in such a way that the values of the lateral passes in the two axes are equal (a = a').

Pass setting in the plane (Q)

R=Tool radius, L=dimension of the small side
- If Q > 2R, the system returns error message 887.
- If Q = or < R and the result of division of the half-distance to be roughed by the pass set is not an integer, the system computes a new setting for pass Q.

The number of passes is equal to (L/2-R)/Q = m (rounded to the integer above if necessary).

Determination of the new pass setting: Q' = (L/2-R)/n

The value of Q' is rounded off to the next higher micrometer (µm).

In this case, the value of the last pass is:

L/2-R-(n-1)Q' (value less than Q').

If R < Q < 2R, the value of the first pass is R/2 (except for circular pockets) and the value of the subsequent passes is Q'=L/2-R/n.

If the cutter radius does not allow at least one roughing pass or the finishing pass to be cut at the programmed value, the system returns error message 896.

Pass setting in the tool axis (P)

The system computes a new pass setting P' using the same formula as for Q' if the result of division of the total depth by the pass set is not an integer.

Examples

Execution of a rectangular pocket in the XY plane (G17)

Roughing alone without finishing.

```
%55
N10 G00 G52 Z..
N20 T04 D04 M06 (CUTTER DIAMETER=8 CENTER CUT)
N30 S3000 M03 M40
N40 G45 X75 Y52 Z-24 ER2 EX100 EY50 EB20 P5 Q7 EP100 EQ500
N50 G00 G52 Z..
N..
```

Roughing and finishing with the same tool.

```
%55
N10 G00 G52 Z..
N20 T04 D04 M06 (CUTTER DIAMETER=8 CENTER CUT)
N30 S3000 M03 M40
N40 G45 X75 Y52 Z-24 ER2 EX100 EY50 EB20 P5 Q7 I0.3 J0.3 EP100 EQ500 EI70 EJ300
N50 G00 G52 Z..
N..
```

Execution of a sequence of pockets in the XY plane (G17)

The rectangular, circular and oblong pockets are executed with a single tool.

The square pocket is executed with two tools (roughing + finishing tool).

```
%206
N10 G00 G52 Z..
N20 T01 D01 M06 (CUTTER DIAMETER=10 CENTER CUT)
N30 S600 M40 M03 M08
$0 RECTANGULAR POCKET
N40 G45 X50 Y25 Z-10 ER2 EX90 EY40 EB10 P3 Q8 EP50 EQ150
$0 CIRCULAR POCKET
N50 G45 X45 Y25 Z-26 ER-8 EB10.5 P3 Q8 EP50 EQ150 J0.5 EJ200
$0 OBLONG POCKET
N60 G45 X20 Y25 Z-20 ER-8 EY30 EB9.5 P3 Q8 EP50 EQ150 I0.5 J0.5 EI70 EJ200
$0 SQUARE POCKET ROUGHING
N70 G45 X75 Y25 Z-20 ER-8 EX29.6 EY29.6 P3 Q8 EP50 EQ150
N80 G00 Z200 M05 M09
$0 SQUARE POCKET FINISHING
N90 T02 D02 M06 (CUTTER DIAMETER=6 CENTER CUT)
N100 S1000 M40 M03 M08
N110 G45 X75 Y25 Z-20 ER-8 EX30 EY30 EB3.8 P9.5 I0.5 J0.5 EP50 EI50 EJ100
N120 G00 G52 Z.. M05 M09
N130 M02
```

4.10.2 Pocket and Facing Cycles with Any Contours

G46	Pocket and facing cycles with any contours.

This cycle is used for machining one or more pockets or facings of varied shapes, with or without islands and walls.

4.10.2.1 General

The cycle is programmed by:
- a geometric definition order,
- a selection of three machining orders.

Geometric Definition Order

This order includes several dedicated blocks:
- a header block characterizing the tool data and tool scan geometry (pass setting, machining allowance, tool diameter, etc.),
- a segmenting block introducing each type of contour (pocket, island, facing, cavity or wall),
- the geometric contour definition blocks,
- a geometric contour definition end block.

When several contours are programmed in succession, each contour defined is segmented by a special block introducing the machining.

Machining Orders

Three machining orders are available:
- Initial drilling order (for tool penetration at the beginning of the cycle),
- Pocket (or facing) roughing order,
- Pocket (or facing) finishing and/or semi-finishing order.

Each of the three orders is defined by a special block to allow a possible tool change between orders (three types of drilling cycles are available).

4.10.2.2 Blocks Specific to Cycle Programming

The specific blocks of the cycle are identified by function G46 followed by argument NU with which is associated the number defining the type of block or order.

Function G46 NU.. must mandatorily be programmed at the beginning of each block.

Function G46 is nonmodal (i.e. it is cancelled at the end of the block).

Geometric definition and tool data block

N.. G46 NU0 ... : Geometric definition header block

Definition segmenting blocks introducing the machining geometry

N.. G46 NU1 ... : Segmenting block introducing a pocket

N.. G46 NU2 ... : Segmenting block introducing an island

N.. G46 NU3 ... : Segmenting block introducing a facing

N.. G46 NU4 ... : Segmenting block introducing a cavity in a facing

N.. G46 NU5 ... : Segmenting block introducing a facing (associated with a wall)

N.. G46 NU6 ... : Segmenting block introducing a wall (associated with a facing)

Block defining the end of a contour

N.. G46 NU9 ... : Geometric definition end block

Blocks defining the machining orders

N..G46 NU10 ... : Initial drilling order

N..G46 NU15 ... : Pocket (or facing) roughing order

N..G46 NU20 ... : Finishing (or semi-finishing) order

4.10.2.3 Notes on Programming Contour Geometric Definition Blocks

Blocks G46 NU1 to G46 NU9 do not contain a pocket (or facing) number NP.. since they are always associated with a block G46 NU0 mandatorily containing the pocket (or facing) number.

The contours programmed after each introduction block G46 NU1 to G46 NU6 must precede any machining orders.

The finishing and semi-finishing orders are programmed with the same machining order number (G46 NU20).

The contour definition blocks are programmed after each segmenting block with the following standard functions:
- ISO programming with or without PGP (see Chapter 5),
- Programming of L programme variables (see Sec. 6.1),
- G77.. subroutine branches (see Sec. 4.11).

Programming Notes

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

Only the first block of a contour can be programmed in G00, for instance:

```
N.. ...
N110 G46 NU1 (POCKET)
N120 G00 X.. Y..
N130 G01 X..
```

When two consecutive blocks in G01 are identical (for instance with the same X and Y values), the system returns an error message.

The first block of a PGP contour can be programmed in G00 or G01.

PGP blocks can contain dimensions programmed perpendicular to the interpolation plane (e.g. Z.. in the XY plane). In this case, these dimensions are ignored (the interpolation plane is the modal plane defined in the definition header block).

When contour machining is called by subroutine (G77...), it should be noted that the machining operations of the main programme must not have the same NP.. numbers as those called in the subroutine.

External parameters E are available for read (see Sec. 6.2) but must be programmed carefully (they should not be used for write).

Machining start and end coordinates depending on the interpolation plane

The machining start and end coordinates are optional, but may be programmed in the following blocks:
- G46 NU0 ...: Definition header block,
- G46 NU1 ...: Pocket introduction block,
- G46 NU2 ...: Island introduction block.

Interpolation plane	G17	G18	G19
Machining start	LX LY	LZ LX	LY LZ
Machining end	EX EY	EZ EX	EY EZ

The argument syntax must be complied within the block, e.g. in G18: LZLX and not LXLZ.

Use of the cycle on a machine with multiple axis groups

Function G46 can be programmed on a machine with multiple axis groups, but is only active on one group at a time. If two cycles are programmed simultaneously on two axis groups, the system returns error message 260 (working memory occupied).

It should be noted that the function cannot be executed by a PLC axis group.

Restrictions

During contour definition, the following functions are ignored:
- Programmed offset (G59)
- Angular offset (ED)
- Mirroring (G51)
- Scaling (G74)
- Programming in inches (G70)
- Table offset (DAT3).

4.10.2.4 Notes on Programming the Machining Orders

Functions Programmable with the Machining Orders

The following functions can be used with the machining orders (but they affect all the contours to be machined):
- Angular offset (ED),
- Mirroring (G51),
- Scaling (G74).

Constraints related to the Machining Orders

Most of the constraints are due to tool passing difficulties along the contours.

Constraints related to roughing tool passing

For a roughing path, it is necessary to apply a scaling factor in the definition header block (G46 NU0) between the lateral pass setting (Q) and the roughing tool diameter (ED) so as not to leave any material in the area roughed.

Factor applied: $Q \leq 0.66$ ED

If this factor is not verified, the system returns an error message.

It is recommended to use a roughing tool with the same diameter as the semi-finishing tool, subjected to collision avoidance constraints with programmed contour elements.

Constraints related to finishing or semi-finishing tool passing

For finishing, the tool diameter used must be sufficiently small to reach all the geometric elements of each of the contours and avoid collision with other programmed contours.

The same constraints apply to semi-finishing as to finishing but in addition, it is necessary to reason on a dummy tool with the same radius as the real tool to which is added the value of the machining allowance programmed with address J.

To remove the material remaining after roughing, the semi-finishing tool diameter (do) must satisfy the following relation:

$do \geq 3/2\ Q$, i.e. $Q \leq 0.66\ do$ (Q = lateral pass setting)

Narrowed Areas

Narrowed areas can be located:
- between a pocket and an island,
- between two islands,
- between two elements of a contour.

In certain cases, narrowed areas of particular types may not be machined even though they are sufficiently large to accommodate the cutter diameter.

Such areas where the material is not removed are generally located near the median line of the narrowed area. In all cases, the residual matter has a width less than the diameter of the tool used and can be removed by a single pass with that tool (3D graphic display can be used to detect such residual areas).

4.10.2.5 Geometric Definition Header Block

G46 NU0	Geometric definition header block.

Syntax (XY plane)

N.. **G46 NU0** NP.. ED.. Q.. [J..] [NR±] [R03/R04] [LX.. LY..] [EX.. EY..]

G46 NU0	Geometric definition header.
NP..	Pocket (or facing) number.
ED..	Roughing cutter diameter.
Q..	Lateral roughing pass setting.
J..	Lateral finishing allowance (default no allowance).
NR±	Work type (default NR+): - NR+: Work in feed direction (see Fig. 1), - NR-: Work against feed direction (see Fig. 2).
R03/R04	Direction of tool rotation (default R03, clockwise).
LX.. LY..	Coordinates of the drilling and/or start point for roughing (default point calculated by software).
EX.. EY..	Coordinates of roughing contour end point (default point calculated by software).

Notes

Machining allowance J

Argument J represents the minimum machining allowance after roughing. If the machining allowance must be constant over the entire contour, roughing must be followed by semi-finishing with the same value of J.

Lateral pass setting Q and roughing tool diameter ED

The system checks that the following relation is verified: Q ≤ 0.66 ED

Work type NR± and rotation direction R03

Work in feed direction
(called «swallowing) NR+ and R03.

Work against feed direction NR- and R03.

Figure 1

Figure 2

4.10.2.6 Pocket and Island Introduction Blocks

G46 NU1 and G46 NU2
Pocket and island introduction blocks.

Function G46 NU1 introduces pockets of various shapes with and without islands.

Function G46 NU2 introduces islands of various shapes in pockets or facings.

Pocket Introduction Block

Syntax (XY plane)

N.. **G46 NU1** [LX.. LY..]

N.. Contour definition.

G46 NU1	Pocket introduction.
LX.. LY..	Coordinates of the start point for contour finishing (default first contour definition point).
N.. Contour definition	Sequence of blocks programmed after block G46 NU1. If LX and LY are not programmed, the first block defines the contour start point.

Island Introduction Block

Syntax (XY plane)

N.. **G46 NU2** [LX.. LY..]

N.. Contour definition.

G46 NU2	Island introduction.
LX.. LY..	Coordinates of the start point for contour finishing (default first contour definition point).
N.. Contour definition	Sequence of blocks programmed after block G46 NU2. If LX and LY are not programmed, the first block defines the contour start point.

Notes on NU1 and NU2

When LX and LY are not programmed in blocks G46 NU1 and G46 NU2 and the contour start point is not located on the contour, the system returns an error message.

The maximum dimensions of a pocket (G46 NU1) in the two axes of the plane must not exceed 40 times the programmed lateral pass setting (Q). If this relation is not verified, the system returns an error message.

For a given pocket, the maximum number of island contours is limited to 127.

Block G46 NU2 must always be associated with definition of one of the following contours (or the system returns error message 283):
- Pocket G46 NU1,
- Facing G46 NU3,
- Facing defined as a sequence of open contours G46 NU5, G46 NU6.

Processing of the data file related to a pocket order

A long time is required to process pocket definitions.

To enhance performance, the computations are made once and for all the first time the pocket is cut and the results are stored for reuse the following times.

The preliminary computations are made directly after:
- loading the programme,
- editing pocket definition data in the programme.

The computations can only be made using the following functions:
- test mode (TEST),
- graphic drawing (direct graphic mode or in PROCAM).

If the cycle is started in automatic (AUTO), single-step (SINGLE), dryrun (DRYRUN) or sequence number search (SEARCH) modes (directly after loading or editing the programme), the system returns error message 263.

When the computations are performed, the system creates a file associated with the active part programme for:
- the pocket or facing definitions,
- the results of the definition computations.

If the programme contains several pockets, the data for the different pockets are contained in the same file.

The file is managed like a part programme by the system. It has the number of the part programme with which it is associated followed by index 9 (e.g. part programme number %100, file number %100.9).

The file created can be displayed on the directory page (DIR.) or in edit mode (EDIT) but only includes the line corresponding to its number (%n.9).

The file created is permanent, i.e. it is not deleted after execution of the programme. It can be deleted in EDIT mode like any part programme (see Operator Manual). It is automatically deleted by deletion of the part programme with which it is associated.

This file cannot be transferred by DNC1 or punched out on tape.

Memory space used for the file:
- a pocket data file associated with a part programme occupies a maximum memory space of 128 bytes per elementary block created,
- immediately after loading or editing the part programme, an additional memory space of approximately 40 Kbytes must be temporarily available in the working area (although the actual space used will be less).

4.10.2.7 Facing and Cavity Introduction Blocks

G46 NU3 and G46 NU4
Facing and cavity introduction blocks.

Function G46 NU3 is used to introduce the limits of contours to be faced with or without islands or cavities.

Function G46 NU4 is used to introduce the limits of cavities located in facing paths.

Facing Introduction Block

Syntax

N.. **G46 NU3**

N.. Contour definition.

G46 NU3	Facing introduction.
N.. Contour definition	Sequence of blocks programmed after block G46 NU3.

Cavity Introduction Block

Syntax

N.. **G46 NU4**

N.. Contour definition.

G46 NU4	Cavity introduction.
N.. Contour definition	Sequence of blocks programmed after block G46 NU4.

Notes on NU3 and NU4

Blocks G46 NU3 and G46 NU4 cannot include a finishing start point (LX LY) (facings and cavities do not include contour finishing).

Cavity G46 NU4 is an unmachined section in a facing whose limits are defined like a pocket.

Block G46 NU4 must mandatorily be associated with definition of one of following contours (or the system returns error message 283):
- pocket G46 NU1,
- facing G46 NU3,
- facing defined as a sequence of open contours G46 NU5, G46 NU6.

4.10.2.8 Facing and Wall Introduction Blocks

G46 NU5 and G46 NU6
Facing and wall introduction blocks.

Function G46 NU5 is used to introduce facing limits including walls.

Function G46 NU6 is used to introduce the walls associated with a facing.

G46 NU5 and G46 NU6 are always associated.

Facing Introduction Block

Syntax

N.. **G46 NU5**

N.. Contour definition.

G46 NU5	Facing introduction.
N.. Contour definition	Sequence of blocks programmed after block G46 NU5.

Wall Introduction Block

Syntax

N.. **G46 NU6**

N.. Contour definition.

G46 NU6	Wall introduction.
N.. Contour definition	Sequence of blocks programmed after block G46 NU6.

Notes on NU5 and NU6

Blocks G46 NU5 and G46 NU6 defining a contour are always associated. They must never be separated by blocks G46 NU.. of another type.

The sequence of introduction blocks G46 NU5 or G46 NU6 each define a machining operation with an open contour, but the sequence must define a closed contour.

Blocks G46 NU5 and G46 NU6 do not include a start point (LX LY). The contour is open and finishing therefore begins at one of the ends of the part (chosen according the type of machining).

The number of blocks G46 NU5 and G46 NU6 required to defining a closed contour is unlimited.

To close the contour, the open contour end coordinates must be the same as the start coordinates of the next open contour.

A wall can be described by several blocks G46 NU6, providing they are consecutive.

4.10.2.9 Geometric Definition End Block

G46 NU9	Geometric definition end block.

Syntaxe

N.. **G46 NU9**

G46 NU9	Geometric definition end.

Notes

Block G46 NU9 is located after the geometric definition blocks and before machining orders G46 NU10, G46 NU15 and G46 NU20.

4.10.2.10 Initial Drilling Orders

G46 NU10	Initial drilling order.

Three types of drilling orders are available.

Syntax (XY plane)

Simple Drilling Order

N.. **G46 NU10** NP.. G81 Z.. [ER..] [F..]

Peck Drilling Order

N.. **G46 NU10** NP.. G83 Z.. P.. [Q..] [ER..] [EF..] [F..]

Drilling Order with Chip Breaking

N.. **G46 NU10** NP.. G87 Z.. P.. [Q..] [ER..] [EF..] [F..]

G46 NU10	Initial drilling order.
NP..	Pocket (or facing) number.
G81/G83/G87	Drilling type.
Z..	Hole bottom end point (depending on the interpolation plane, W, Y, V, X or U).
P..	First infeed (G83 or G87).
Q..	Last infeed (G83 or G87).
ER..	Retraction plane (in the same axis as the hole bottom end point, i.e. the Z axis in the XY plane).
EF..	Dwell at the end of each infeed (G83 or G87).
F..	Drilling feed rate (in mm/min).

Notes

The drilling position is defined by the LX and LY values programmed (XY plane) in the header block (G46 NU0). The default position can also be calculated automatically by the software.

Retraction plane ER and initial Z dimension.

Dimensions ER and initial Z are also interpreted in the same way in the following machining orders:
- roughing order G46 NU15,
- finishing (or semi-finishing) order G46 NU20.

When the cycle is called, «initial Z» represents the current dimension in the axis perpendicular to the interpolation plane.

The tool infeed from dimension ER to the hole bottom Z is at the machining feed rate.

After drilling, retraction of the tool to the programmed ER value is carried at high speed.

Movement from the initial Z approach position to the programmed ER value is also carried out at high speed.

If ER is not programmed, infeed from the initial Z position to the hole bottom is carried out at the machining feed rate.

Use of a drilling order during facing

A drilling order can be used during facing if the roughing tool cannot access all the areas without retracting (it is therefore recommended to check graphically whether this order is necessary).

Omission of a drilling order

If a drilling order is omitted, drilling can nevertheless be carried out «manually» by the operator in the location chosen, but the location of the hole drilled must be defined as roughing start point in the corresponding definition order.

The above «manual» drilling procedure cannot be used when several drillings are required by the pocket and/or island configuration. In this case, it is mandatory to programme the drilling order.

4.10.2.11 Roughing Order

G46 NU15	Roughing order.

Syntax (XY plane)

N.. **G46 NU15** NP.. Z.. P.. [ER..] [EH..] [EP..] [EQ..]

G46 NU15	Roughing order.
NP..	Pocket (or facing) number.
Z..	Machining bottom end point (according to the interpolation plane, W, Y, V, X or U).
P..	Axial pass setting. Roughing is carried out by consecutive infeeds (Z/P+1). Each infeed is carried out from the initial drilling point.
ER..	Retraction plane (on the same axis as the machining bottom end point, i.e. Z axis in the XY plane).
EH..	Material impact plane.
EP..	Axial roughing feed rate (default modal F value).
EQ..	Lateral roughing feed rate (default modal F value).

Notes

The roughing start position is defined in the header block (G46 NU0) by the programmed LX and LY values (XY plane). The default point can also be computed automatically by the software.

When the cycle is executed, the system checks that the tool radius R associated with the modal corrector D is compatible with the roughing cutter diameter (ED) programmed in the definition header block (G46 NU0).

Relation verified: R = ED/2

Retraction plane ER

Plane ER is interpreted in the same way as in the initial drilling order G46 NU10. For further details, refer to the notes on G46 NU10.

Material impact plane EH and initial Z dimension

Dimension EH is used for consecutive pass settings directly related to the material plane.

Dimensions EH and initial Z are interpreted in the same way in finishing order G46 NU20.

When the cycle is called, initial Z represents the current dimension on the axis perpendicular to the interpolation plane.

When dimension EH is not programmed, the consecutive pass settings are made from dimension ER.

The tool moves from dimension ER to dimension EH at the machining feed rate.

After each pass, the tool is retracted to dimension ER at high speed.

If dimension ER is not programmed, the infeed from initial Z to EH is carried out at the machining feed rate.

If dimension ER is programmed below dimension EH, the system returns error message 276.

Constraints specific to a roughing order

A sequence of roughing contours can be programmed in any order, but the finishing operations are then carried out in the same order.

A roughing order leaves an «additional» machining allowance of variable thickness that is added to the finishing allowance programmed in the roughing order. The allowance varies from one point to another of the contour roughed, but is less than the value calculated as follows:

3/2 Q (Q = lateral pass setting)

In order to be able to finish a contour with a constant machining allowance, semi-finishing must be programmed after roughing.

4.10.2.12 Finishing and Semi-Finishing Orders

G46 NU20	Finishing or semi-finishing order.

Syntax (XY plane)

N.. **G46 NU20** NP.. Z.. P.. [ER..] [EH..] [EI..] [EJ..] [J..]

G46 NU20	Finishing or semi-finishing order.
NP..	Pocket (or facing) number.
Z..	Machining bottom end point (according to the interpolation plane, W, Y, V, X or U).
P..	Axial pass setting. Roughing is carried out by consecutive infeeds (Z/P+1). Each infeed is carried out from the initial drilling point.
ER..	Retraction plane (on the same axis as the machining bottom end point, i.e. Z axis in the XY plane).
EH..	Material impact plane.
EI..	Axial finishing feed rate (default modal F value).
EJ..	Lateral finishing feed rate (default modal F value).
J..	Lateral finishing allowance (default J = 0). The value of J must be < or = J of G46 NU0 (otherwise, the pass risks being carried out with no material removal).

Notes

When the contours are types G46 NU3 and G46 NU4 (facing and cavity introduction blocks), the use of a finishing or semi-finishing order is unnecessary (no edge). However, no error is returned if this order is programmed, it is simply ignored.

When executing the cycle, the system checks that the tool radius R associated with the modal corrector D is less than or equal to the roughing cutter radius (ED/2) programmed in the definition header block (G46 NU0).

Retraction plane ER

Plane ER is interpreted in the same way as in the initial drilling order G46 NU10. For further details, refer to the notes on G46 NU10.

Material impact plane EH

Plane EH is interpreted in the same way as in roughing order G46 NU15. For further details, refer to the notes on G46 NU15.

Constraints specific to finishing or semi-finishing orders

For semi-finishing:
- a machining allowance (J) must be programmed in the machining order (G46 NU20) and its value must be equal to the one programmed in the definition header block (G46 NU0),
- it is recommended to choose a semi-finishing tool with the largest possible diameter while making sure that it can pass between the different elements of the programmed contours.

Tool Infeed and Retraction

Infeed and retraction at the apex of an obtuse angle

Carried out linearly tangent to the start or end segments.

For infeed or retraction, the distance D is such that:

d = 5/4 (re + J)

re: Roughing tool radius

J: Machining allowance programmed in header block NU0.

Infeed or retraction at the apex of an acute angle

It is impossible to programme infeed or retraction at the apex of an acute angle. The system returns error message 288.

Infeed or retraction on a flat angle

General infeed or retraction rule:

If the action cannot be carried out linearly (Fig. 1), it can be carried out circularly using a radius of value d (see Fig. 2) or value d'.

Value of d or d':
- d = 5/4 (re + J)
- d' = 0.9 rp

re: Roughing tool radius
J: Machining allowance programmed in header block NU0.
rp: Contour radius

An infeed located elsewhere than on the end of a segment is processed in the same way as for a flat angle.

Figure 1

Figure 2

4.10.3 Examples of Programming of Cycles with Any Contours

Execution of a pocket with an island

The pocket and island contours are each defined starting in their point a. XY plane (G17).

Use of functions:
- G46 NU0
- G46 NU1
- G46 NU2
- G46 NU9
- G46 NU10
- G46 NU15
- G46 NU20

Undimensioned radii R = 5

```
%100
N10 G0 G52 Z..
N20 G52 X.. Y.. M05 M09

$0 GEOMETRIC DEFINITION HEADER
N50 G46 NU0 NP1 ED8 Q5 J0.5 NR+ R03 LX25 LY0

N100 G46 NU1 (POCKET CONTOUR DEFINITION)
N110 G01 X35 Y0 (POCKET START)
N120 Y-15 EB5
N130 X25
N140 Y-20 EB5
N150 X-15
N160 G02 X-15 Y20 I-15 J0
N170 G01 X25 EB5
N180 Y15
N190 X35 EB5
N200 Y0

N300 G46 NU2  LX15 LY0 (ISLAND CONTOUR DEFINITION)
N310 G01 X15 Y5 (ISLAND START)
N320 Y-10
N330 X-15
N340 G02 X-15 Y10 I-15 J0
N350 G01 X15

N400 G46 NU9 (END OF GEOMETRIC DEFINITIONS)

$0 MACHINING ORDERS
N500 G77 N10 N20
N510 T01 D01 M06 (CUTTER DIAMETER 10)
N520 S1500 M03 M40
N530 G46 NU10 NP1 G83 Z-9.5 P4 Q1 ER2 F300 (DRILLING ORDER)
N540 G77 N10 N20

N600 T02 D02 M06 (ROUGHING CUTTER DIAMETER 8)
N610 M03
N620 G46 NU15 NP1 Z-10 P3 ER1 EP250 EQ350 (ROUGHING ORDER)
N630 G77 N10 N20

N700 T03 D03 M06 (SEMI-FINISHING CUTTER DIAMETER 8)
N710 G77 N610
N720 G46 NU20 NP1 Z-10 P6 ER1 EI200 EJ300 J0.2 (SEMI-FINISHING ORDER)
N730 G77 N10 N20

N800 T04 D04 M06 (FINISHING CUTTER DIAMETER 8)
N810 G77 N610
N820 G46 NU20 NP1 Z-10 P10 EH0 (FINISHING ORDER)
N830 G77 N10 N20 M02
```

Execution of a pocket with seven islands of different shapes

The pocket and island contours are each defined from their starting point a. XY plane (G17). Outer dimensions of the part: 90 x 80 x 15

Use of functions:
- G46 NU0
- G46 NU1
- G46 NU2
- G46 NU9
- G46 NU10
- G46 NU15
- G46 NU20

```
%200
N10 G0 G52 Z..
N20 G52 X.. Y.. M05 M09

$0 GEOMETRIC DEFINITION HEADER
N50 G46 NU0 NP1 ED5 Q3 J0.2 NR+ R03 LX10 LY10 EX-10 EY-10

(POCKET CONTOUR DEFINITION)
N100 G46 NU1 LX0 LY-35 (LX LY FINISHING START POINT)
N110 G01 X40 Y10 (POCKET START)
N120 Y-35 EB6
N130 X-40 EB6
N140 Y35 EB8
N150 X40 EB8
N160 Y10

(ISLAND CONTOUR DEFINITION.  CENTRE SQUARE)
N200 G46 NU2

N210 G01 X0 Y8 (ISLAND START)
N220 X8 Y0
N230 X0 Y-8
N240 X-8 Y0
N250 X0 Y8

(ISLAND CONTOUR DEFINITION.  CIRCLE, DIAMETER 18)
N300 G46 NU2 LX11 LY-20 (LX LY FINISHING START POINT)
N310 G01 X29 Y-20 (ISLAND START)
N320 G02 X29 Y-20 I20 J-20

(ISLAND CONTOUR DEFINITION.  10 x 10 SQUARE)
N400 G46 NU2
N410 G01 X28 Y5 (ISLAND START)
N420 Y-5
N430 X18
N440 Y5
N450 X28

(ISLAND CONTOUR DEFINITION.  RIGHT TRIANGLE)
N500 G46 NU2
N510 G01 X8 Y18 (ISLAND START)
N520 X28
N530 X8 Y28
N540 Y18
```

```
(ISLAND CONTOUR DEFINITION.  20 x 10 RECTANGLE)
N600 G46 NU2 LX-15 LY18 (LX LY FINISHING START POINT)
N610 G01 X-5 Y18 (ISLAND START)
N620 Y28
N630 X-25
N640 Y18
N650 X-5

(ISLAND CONTOUR DEFINITION.  CIRCLE, DIAMETER 10)
N700 G46 NU2
N710 G01 X-17 Y0 (ISLAND START)
N720 G03 X-17 Y0 I-22 J0

(ISLAND CONTOUR DEFINITION.  ISOCELES TRIANGLE)
N800 G46 NU2 LX-15 LY-25 (LX LY FINISHING START POINT)
N810 G01 X-10 Y-25 (ISLAND START)
N820 X-25
N830 X-17.5 Y-15
N840 X-10 Y-25

N900 G46 NU9 (END OF GEOMETRIC DEFINITIONS)

$0 MACHINING ORDERS
N1000 T11 D11 M06 (CUTTER DIAMETER 6)
N1010 S2500 M03 M40
N1020 G46 NU10 NP1 G81 Z-3.5 ER2 F100 (DRILLING ORDER)
N1030 G77 N10 N20

N1100 T12 D12 M06 (ROUGHING CUTTER DIAMETER 5)
N1110 S3500 M03 M40
N1120 G46 NU15 NP1 Z-4 P3 ER1 EH0 EP100 EQ150 (ROUGHING ORDER)
N1130 G77 N10 N20

N1200 T13 D13 M06 (SEMI-FINISHING CUTTER DIAMETER 5)
N1210 G77 N1110
N1220 G46 NU20 NP1 Z-4 P3 ER1 EI100 EJ200 J.2 (SEMI-FINISHING ORDER)
N1230 G77 N10 N20

N1300 T14 D14 M06 (FINISHING CUTTER DIAMETER 5)
N1310 G77 N1110
N1320 G46 NU20 NP1 Z-4 P4 ER1 EH0 (FINISHING ORDER)
N1330 G77 N10 N20 M02
```

Execution of a facing including an island

The limits of the facing and the island contour are each defined from their start point a. XY Plane. The island is defined in subroutine %452.

Use of functions:
- G46 NU0
- G46 NU2
- G46 NU3
- G46 NU15
- G46 NU20

```
%451
$0 GEOMETRIC DEFINITION HEADER
N10 G46 NU0 NP1 ED10 Q7 J0.3 NR+ R03

N100 G46 NU3 (DEFINITION OF FACING LIMITS)
N110 G01 X0 Y0 (FACING START)
N120 Y-77
N130 X-77
N140 Y0
N150 X0

N200 G46 NU2 (ISLAND CONTOUR DEFINITION)
N210 G77 H452 (CALL TO ISLAND CONTOUR SUBROUTINE %452)

N300 G46 NU9 (END OF GEOMETRIC DEFINITION)

$0 MACHINING ORDERS
N400 G0 G52 Z..
N410 G52 X.. Y.. M05 M09
N420 T31 D31 M06 (CUTTER DIAMETER 10)
N430 S2500 M03 M40
N440 G46 NU15 NP1 Z-15 P5 ER2 EP250 EQ350 (ROUGHING ORDER)

N500 G46 NU20 NP1 Z-15 P5 ER2 EI200 EJ350 J0.2 (SEMI-FINISHING ORDER)

N600 G46 NU20 NP1 Z-15 P15 ER2 EH0 EI200 EJ350 (FINISHING ORDER)
N700 G77 N400 N410 M02

%452 (ISLAND CONTOUR DEFINITION SUBROUTINE)
N10 G92 R1
N20 G01 X-1 Y-15 (ISLAND START POINT)
N30 EA-90 ES+
N40 G02 I-15 J-15 R16 ES+ EB5.5
N50 G01 EA225 X-19.445 Y-37.123
N60 EA225 ES- EB5.5
N70 G02 I-52 J-52 R24 ES+ EB5.5
N80 G01 EA45 X-37.123 Y-19.445
N90 EA45 ES- EB5.5
N100 G02 I-15 J-15 R16 ES-
N110 G01 EA0 X-15 Y-1
N120 EA0 ES+
N130 G02 I-15 J-15 R16 ES-
N140 G01 EA-90 X-1 Y-15
```

Execution of a facing including a cavity and an island

The facing and cavity limits and island contour are each defined from their start point a. XY plane (G17). The part is initially preformed.

Use of functions:
- G46 NU0
- G46 NU2
- G46 NU3
- G46 NU4
- G46 NU15
- G46 NU20

```
%400
N10 G0 G52 Z..
N20 G52 X.. Y.. M05 M09

$0 GEOMETRIC DEFINITION HEADER
N50 G46 NU0 NP1 ED16 Q10 J0.5 NR+ R03

N100 G46 NU3 (DEFINITION OF FACING LIMITS)
N110 G01 X0 Y-30 (FACING START)
N120 X-160 EB20
N130 Y60 EB20
N140 X-10 EB20
N150 Y30 EB10
N160 X0
N170 G02 X0 Y-30 I0 J0

N200 G46 NU4 (DEFINITION OF CAVITY LIMITS)
N210 G01 X-130 Y10 (CAVITY START)
N220 Y30 EB5
N230 X-40 EB5
N240 Y-10 EB5
N250 X-130 EB5
N260 Y10

N300 G46 NU2 (ISLAND CONTOUR DEFINITION)
N310 G01 X20 Y0 (ISLAND START)
N320 G02 X20 Y0 I0 J0

N400 G46 NU9 (END OF GEOMETRIC DEFINITIONS)

$0 MACHINING ORDERS
N500 T07 D07 M06 (CUTTER DIAMETER 16)
N510 S2000 M03 M40

$0 FACING ROUGHING ORDER
N550 G46 NU15 NP1 Z-4.5 P4.5 ER2 EH0 EP300 EQ400
N560 G77 N10 N20
$0 FACING FINISHING ORDER
N600 T08 D08 M06 (CUTTER DIAMETER 16)
N610 M03
N620 G46 NU15 NP1 Z-5 P5 ER2 EH0 EP300 EQ400

$0 ISLAND CONTOUR FINISHING ORDER
N700 G46 NU20 NP1 Z-5 P5 ER2 EH0 EI200 EJ300
N710 G77 N10 N20
N720 M02
```

Execution of a facing with three walls and two islands

The facing contours with walls and islands are each defined from their start point a. XY plane (G17).

Use of functions:
- G46 NU0
- G46 NU2
- G46 NU5
- G46 NU6
- G46 NU9
- G46 NU15
- G46 NU20

```
%500
N10 G0 G52 Z..
N20 G52 X.. Y.. M05 M09

$0 GEOMETRIC DEFINITION HEADER
N100 G46 NU0 NP1 ED8 Q5 J0.2 NR+ R03

N150 G46 NU5 (DEFINITION OF FACING LIMITS)
N160 G01 X80 Y0 (FACING START)
N170 X25

N200 G46 NU6 (WALL 1 DEFINITION)
N210 X25 Y0 (WALL 1 START)
N220 Y10 EB-5
N230 X0

N250 G46 NU5 (DEFINITION OF FACING LIMITS)
N260 X0 Y10 (START OF SECOND PART OF FACING)
N270 Y60

N300 G46 NU6 (WALL 2 DEFINITION)
N310 X0 Y60 (WALL 2 START)
N320 X10 Y50
N330 X45
N340 Y60

N350 G46 NU5 (DEFINITION OF FACING LIMITS)
N360 X45 Y60 (START OF THIRD PART OF FACING)
N370 X60

N400 G46 NU6 (WALL 3 DEFINITION)
N410 X60 Y60 (WALL 3 START)
N420 G03 X80 Y40 I80 J60

N450 G46 NU5 (DEFINITION OF FACING LIMITS)
N460 G01 X80 Y40 (START OF FOURTH OF FACING)
N470 Y0
```

```
N500 G46 NU2 (ISLAND 1 CONTOUR DEFINITION)
N510 G01 X70 Y5 (ISLAND 1 START)
N520 X45
N530 G02 X45 Y15 R5
N540 G01 X60
N550 G01 Y20
N560 G02 X70 Y20 R5
N570 G01 Y5

N600 G46 NU2 (ISLAND 2 CONTOUR DEFINITION)
N610 G01 X30 Y30 (ISLAND 2 START)
N620 G02 X30 Y40 R5
N630 G01 X45
N640 G02 X45 Y30 R5
N650 G01 X30

N700 G46 NU9 (END OF GEOMETRIC DEFINITIONS)

$0 MACHINING ORDERS
N800 T41 D41 M06 (CUTTER DIAMETER 8)
N810 S2500 M03 M40
N820 G46 NU15 NP1 Z-6 P4 ER1 EH0 EP230 EQ400 (ROUGHING ORDER)

N900 G46 NU20 NP1 Z-6 P4 ER1 EI200 EJ350 J.2 (SEMI-FINISHING ORDER)
N910 G77 N10 N20

N1000 T42 D42 M06 (FINISHING CUTTER DIAMETER 8)
N1010 S3000 M03 M40
N1020 G46 NU20 NP1 Z-6 P6 ER1 EH0 (FINISHING ORDER)
N1030 G77 N10 N20 M02
```

Execution of facings with islands and pocket

The facing limits and island contours are each defined from their start point a. Island 1 is defined from subroutine %452 and facing with circular island 2 is defined from programme %451 (see example including programme %451 and subroutine %452. XY plane (G17)).

Use of functions:

- G46 NU0
- G46 NU1
- G46 NU2
- G46 NU3
- G46 NU9
- G46 NU10
- G46 NU15
- G46 NU20

For other part dimensions see %451. Undimensioned radii = cutter radius

```
%457
$0 BRANCH TO SUBROUTINE %451 - ISLAND 1 MACHINING
N05 G77 H451 N10 N600

N10 G G52 Z
N20 G52 XY M05 M09

$0 GEOMETRIC DEFINITION HEADER - UPPER PART
N50 G46 NU0 NP10 ED10 Q7 J0.5 NR+ R03

N100 G46 NU3 (DEFINITION OF FACING LIMITS)
N110 G77 H452 (BRANCH TO SUBROUTINE %452)

N200 G46 NU2 (ISLAND CONTOUR DEFINITION, CIRCLE 2)
N210 G01 X-5 Y-15 (ISLAND START, CIRCLE 2)
N220 G03 X-5 Y-15 I-15 J-15

N300 G46 NU9 (END OF GEOMETRIC DEFINITIONS)

$0 MACHINING ORDERS - UPPER PART
N400 T50 D50 M06 (CUTTER DIAMETER 10)
N410 S2500 M03 M40
N420 G46 NU15 NP10 Z-5 P3 ER1 EP250 EQ350 (ROUGHING ORDER)

N500 G46 NU20 NP10 Z-5 P5 ER1 EH0 (SEMI-FINISHING ORDER)
N510 G77 N10 N20

N600 T51 D51 M06 (FINISHING CUTTER DIAMETER 10)
N610 G77 N410
N620 G46 NU20 NP10 Z-5 P5 EH0 EI250 EJ350 (FINISHING ORDER)
N630 G77 N10 N20

$0 GEOMETRIC DEFINITION HEADER (OCTAGONAL POCKET)
N700 G46 NU0 NP11 ED8 Q5 J0.2
```

```
N800 G46 NU1 (OCTAGONAL POCKET CONTOUR DEFINITION)
N810 G01 X-32 Y-52 (POCKET START)
N820 Y-72 EB-11.72
N830 X-72 EB-11.72
N840 Y-32 EB-11.72
N850 X-32 EB-11.72
N860 Y-52

N900 G46 NU9 (END OF GEOMETRIC DEFINITIONS)

$0 MACHINING ORDERS (OCTAGONAL POCKET)
N1000 T52 D52 M06 (DRILL DIAMETER 10)
N1010 S1200 M03 M40
N1020 G46 NU10 NP11 G81 Z-10.5 ER0 F200 (DRILLING ORDER)
N1030 G77 N10 N20

N1100 T53 D53 M06 (ROUGHING CUTTER DIAMETER 8)
N1110 S3000 M3 M40
N1120 G46 NU15 NP11 Z-11 P3 ER-4 EP250 EQ350 (ROUGHING ORDER)

N1200 G46 NU20 NP11 Z-11 P6 ER-4 (SEMI-FINISHING ORDER)
N1210 G77 N10 N20

N1300 T54 D54 M06 (FINISHING CUTTER DIAMETER 8)
N1310 G77 N1110
N1320 G46 NU20 NP11 Z-11 P6 ER-4 EI250 EJ350 (FINISHING ORDER)
N1330 G77 N10 N20

N1500 M02
```

NUM
1020/1040/1060M

PROGRAMMING MANUAL

VOLUME 2

0101938819/5

05-97 **en-938819/5**

Despite the care taken in the preparation of this document, NUM cannot guarantee the accuracy of the information it contains and cannot be held responsible for any errors therein, nor for any damage which might result from the use or application of the document.

The physical, technical and functional characteristics of the hardware and software products and the services described in this document are subject to modification and cannot under any circumstances be regarded as contractual.

The programming examples described in this manual are intended for guidance only. They must be specially adapted before they can be used in programs with an industrial application, according to the automated system used and the safety levels required.

© Copyright NUM 1997.
All rights reserved. No part of this manual may be copied or reproduced in any form or by any means whatsoever, including photographic or magnetic processes. The transcription on an electronic machine of all or part of the contents is forbidden.

© Copyright NUM 1997 software NUM 1000 family.
This software is the property of NUM. Each memorized copy of this software sold confers upon the purchaser a non-exclusive licence strictly limited to the use of the said copy. No copy or other form of duplication of this product is authorized.

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../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

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 is then 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.

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).

Execution of two grooves by subroutine branches

Shape 1, subroutine H35: two branches for execution of two passes each with depth 2.5 mm.

Shape 2, subroutine H40: three branches for execution of three passes each with depth 3 mm.

Main programme

```
%30
N10 G00 G52 Z.. M05 M09
$0 2 SHAPES WITH S/PROG %35 ET %40
N20 T01 D01 M06 (CUTTER DIAMETER=9)
N30 S800 M40 M03
N40 G00 X15 Y10 Z3                    Point a, approach
N50 G01 Z0 F50                        Approach on Z
$0 MACHINING SUBROUTINE %35
N60 G77 H35 S2                        Branch to subroutine %35
N70 G77 N10                           Branch to sequence N10
N80 T02 D02 M06 (CUTTER DIAMETER=7)
N90 S1000 M40 M03
N100 G00 X95 Y10 Z3                   Point e, approach
N110 G01 Z0 F50                       Approach on Z
$0 MACHINING SUBROUTINE %40
N120 G77 H40 S3                       Branch to subroutine %40
N130 G77 N10                          Branch to sequence N10
N140 M02
```

Subroutine for shape 1

```
%35
$0 SUBROUTINE OF MAIN PROGRAMME %30
N10 G91 G01 Z-2.5 F40 M08             Point a, incremental infeed
N20 G90 Y40 F80                       Point b
N30 X50                               Point c
N40 G03 X50 Y10 I80 J25               Point d
N50 G01 X15                           Point a
```

Subroutine for shape 2

```
%40
$0 SUBROUTINE OF MAIN PROGRAMME %30
N10 G91 G01 Z-3 F40 M08               Point e, incremental infeed
N20 G90 Y80 F70                       Point f
N30 G02 X80 Y40 R15                   Point g
N40 G01 X95                           Point h
N50 Y10                               Point e
```

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 is then 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]
```

Subroutine Call by M Function with Axis Multigroups

See Sec. 4.15 (special programming for axis multigroups).

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                   Programming for one machining
N..                   operation
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
```

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.

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 subroutine %9999

The PLC ignores a new call or maintaining of the call to subroutine %999 during execution of this subroutine.

End of subroutine %9999

At the end of execution of the subroutine, the system does not make a report to the machine processor. It is the subroutine's responsibility to report on its execution by function M or external parameter E.

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 is then 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 report (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. 4.15 (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.

If an interrupt occurs, all axis positions are stored.

Programme

```
%300
N . .
N . .
N . .
N200 X . . G10 @7 < 50.3 N250
N . .
N . .
N250
N . .
```

Note

The order of the data is important.

Syntax

N.. [G40] [G04 F..] [G00/G01/G02/G03] X.. Y.. Z.. **G10** [:n] [+X.. or F..] [@n < > Value] N.. [+ Number] [EF..]

G40	Tool radius offset cancel.
G04 F..	Interruptible dwell, but without branch to designated block.
G00/G01/G02/G03	Interruptible interpolations.
X.. Y.. 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 interrupt block is not acknowledged (forcing at 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 secs 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 (Less than or greater than). Value: Comparison threshold in the same 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 axis motion 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 all 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 processed interrupt is taken into account by function G10 of the part programme that then manages the branch.

New dimensional values can be detected by dynamic operators and stored in an indexed addressing table.

REMARK *With axis multigroup programming (see Sec. 4.15), when an interrupt on a 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.. Y.. F200
N90 G40 X.. G10 N150 +3    Jump to N150
N..
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.. Y.. 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 Y120 G10 : 2 + X60 N250   Jump to N220 or N250
N220
N230          Condition not satisfied
N240
N250
N..           Condition satisfied
```

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 X200 F300
N..
N.. G00 X..
N100 G01 X0 F150 G10 +X2 EF100
```

Feed rate 100 mm/min after the interrupt

```
N..
```

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 (XY plane)

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 Y0
N120 G01 X100 G79+F4
```

Movement along X with 40-second timeout before analysis of block N130

```
N130 G79 E10000=1 N300
N140 Y20
N..
N300 G01 X200
N..
```

Example with anticipation at the end of the current block

```
N..
N110 G00 X0 Y0
N120 G01 Y100 G79-X10
```

Movement along Y with anticipation of 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.

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

Condition for activation of the emergency retract subroutine

Part programme being executed in one of the following modes:
- automatic (AUTO),
- single step (SINGLE),
- manual data input (MDI),
- dry run (DRYRUN).

In automatic mode, the emergency retraction programme remains active until encountering a function M00 or M02.

An emergency retraction programme can be activated:
- by reading of the block in which it is programmed
- as long as the programme or subroutine in which it is declared is not ended
- as long as a new emergency retraction programme has not been declared
- as long as the emergency retraction function has not been cancelled by G75 N0.

If emergency retraction is activated but no retraction programme has been declared in the part programme, it 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».

Emergency retraction on PLC axis groups

For information, see Sec. 4.16 (programming specific to PLC axes).

Examples

Declaration of emergency retract subroutines from a main 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

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 sequence N300 of %2.

If an emergency retract subroutine is activated during execution of subroutine %4 called by N40 of %3, branch to sequence N200 of %1.

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. 4.15 (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. 4.15 (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).

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).

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 ...
N230 ...
N240 ...
...
```

N200 +4

Block deletion

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.. Y.. Z.. A.. B.. 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.. Y.. Z.. A.. B.. 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,
- DAT3,
- 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 Z axis with reference to the measurement zero point before a tool change

```
%10
N10 G00 G52 Z0
N20 T03 D03 M06
N..
```

Programming of G52 on axes Z, B, X, Y:

- -50 mm from the measurement zero point on the Z axis,
- 180 degrees from the measurement zero point on the B axis,
- -100 mm from the measurement zero point on the X and Y axes.

```
N.. ...
N220 G00 G52 Z-50
N230 G52 X-100 Y-100 B180
N..
```

Programming of the additional axes and carrier/carried axis pairs

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	DAT3
X - 100	+ 0	+ 0
Y - 50.3	+ 0	+ 0
Z - 200	+ 20	+ 0
B + 0	+ 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.. Y.. Z..

G92	Programme origin preset.
X.. Y.. Z..	Position with respect to the new programme origin.

Calculation for programme origin preset on an axis:

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 reference to the measurement 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 PGP (Profile Geometry Programming).

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 Y0 Z40
N160 G92 Z60                    Preset
N170 G00 Z..
N..
```

Analysis of block N150 yields:
- current point/OM Z = -160

<u>Application of the first equation</u>

New Z DAT1 = -300 + 40 - 60 = -320

<u>Application of the second equation</u>

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.

Syntax

N..[G90/G91]G59 X.. Y.. Z.. U.. V.. W.. A.. B.. C..[I.. J.. K.. ED..]

G90/G91	Programming in absolute or incremental dimensions.
G59	Programme origin offset.
X.. Y.. Z.. U.. V.. W.. A.. B.. 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.. J.. K.. D..	I.. J.. K..: Arguments defining the centre of rotation of an angular shift programmed with ED (see Sec. 4.12.5) in the plane with respect to the initial programme origin (see figure 1). The translation, if any, of the programme origin is carried out after the rotation. ED..: Angular shift.

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. A new origin offset is applied to the next movement, but the absolute position is offset by the sum of all the G59 ... functions programmed earlier.

The functions below, when included in a programme, must be programmed in the following order:
- ED.. Angular offset,
- G59 ... Programme origin offset,
- G51 ... Mirror function,
- Scaling factor.

Features related to arguments I, J

Machining programmed with respect to the programme origin (OP) can be translated and oriented according to the angle programmed with ED (see Sec. 4.12.5).

```
N.. ...
N.. G59 X.. Y.. I.. J.. ED..
N.. ED..
N..
```

An offset on the X and Y axes is not mandatory for programming I and J.

FIGURE 1

Examples

Origin offsets on the X axis programmed in absolute dimensions (G90) in the XY plane (G17)

```
%60
N10
N..
N50
N..
N90
N..
N120 G90 G59 X50                    Offset 1
N.. G77 N50 N90                     Machining
N..
N230 G59 X100                       Offset 2
N.. G77 N50 N90                     Machining
N..
N350 G59 X0                         Cancellation
N..
```

Origin offsets on the X axis programmed in incremental dimensions (G91) in the XY plane (G17)

```
%70
N10
N..
N50
N..
N90
N..
N120 G91 G59 X50                    Offset 1
N.. G77 N50 N90                     Machining
N..
N230 G59 X50                        Offset 2
N.. G77 N50 N90                     Machining
N..
N350 G91 G59 X-100 or G90 G59 X0    Cancellation possible in G91 or G90
N..
```

Repetition of a shape with origin offsets, XY plane (G17)

```
%110
N10 G00 G52 Z0
N20 T09 D09 M06 (CUTTER DIAMETER=5)
N30 S2000 M40 M03
$0 SHAPE 1
N40 G00 G41 X3 Y-5                    Point a, approach X, Y
N50 Z-2                               Approach on Z
N60 G01 Y10 F120                      Point b
N70 G03 X3 Y19 R4.5 F80               Point c
N80 G01 X-3 F120                      Point d
N90 Y-5                               Point e
$0 SHAPE 2
N100 G59 X-40                         Absolute offset on X
N110 G77 N40 N90
$0 SHAPE 3
N120 G59 X28.5 Y-4.5                  Absolute offset on X and Y
N130 G77 N40 N90
$0 SHAPE 4
N140 G91 G59 X29.5                    Incremental offset on X
N150 G90 G77 N40 N90
N160 G59 X0 Y0                        Offset cancelled
N170 G00 G40 G52 Z0 M05 M09
N180 M02
```

Repetition of a pocket by origin offsets and angular offsets ED.. (see Sec. 4.12.5) in the XY plane.

```
%61
N10 G00 G52 Z0
N20 T11 D11 M06 (CUTTER DIAMETER=10)
N30 S800 M40 M03 M08
$0 POCKET 1
N40 G59 X25 Y40                          Origin offset
N50 G45 X0 Y0 EX30 EY20 ER1 P3 Q8 Z-3
EP50 EQ200
$0 POCKET 2
N60 G59 X65 Y60                          Pocket centre offset
N70 ED45 G77 N50 N50                     Angular offset and machining
$0 POCKET 3
N80 G59 X65 Y20                          Pocket centre offset
N90 ED-45 G77 N50 N50                    Angular offset and machining
N100 ED0 G00 G52 Z0 M05 M09
N110 M02
```

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.

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, G45, etc.),
- 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, V, W).

REMARK *If an angular offset is programmed in a plane (G17, G18 or G19), it is not cancelled by a change of plane. ED.. then applies to the values in the new plane.*

Examples

Execution of three grooves in the XY plane (G17)

Bore diameter 30 completed.

```
%60
N10 G00 G52 Z-60 M05 M09
$0 3 GROOVES ROUGHING
N20 T06 D06 M06 (CUTTER DIAMETER=8)
N30 S800 M40 M03
N40 G00 G41 X5 Y10                    Point a, groove 1 approach
N50 Z-10                              Approach on Z
N60 G01 Y20 F100 M08                  Point b
N70 G03 X-5 Y20 R5 F50                Point c
N80 G01 Y10 F100                      Point d
N90 ED+120                            Angular offset 120 degrees
N100 G77 N40 N80                      Execution of groove 2
N110 ED+240                           Angular offset 240 degrees
N120 G77 N40 N80                      Execution of groove 3
N130 ED0 G77 N10 N10
$0 3 GROOVES FINISHING
N140 T07 D07 M06 (CUTTER DIAMETER=8)
N150 G77 N30 N130
N160 M02
```

Drilling, spot facing and tapping of seven holes angularly offset in the XY plane (G17)

Movement by linear interpolation between holes. Main programme.

```
%1
$0 MAIN PROGRAMME WITH SUBROUTINES %10 %20 %30
N10 G00 G52 Z0 M05 M09
N20 T01 D01 M06 (CUTTER DIAMETER=8.5)
N30 S1000 M40 M03
N40 X50 Y0 Z27                          Point a, approach
N50 G77 H10 S7                          Branch to subroutine %10
N60 G90 ED0                             Offset cancel
N70 G77 N10 N10
N80 T02 D02 M06 (BLADE DIA=20)
N90 S250 M03
N100 X50 Y0 Z27                         Point a, approach
N110 G77 H20 S7                         Branch to subroutine %20
N120 G90 ED0                            Offset cancel
N130 G77 N10 N10
N140 T03 D03 M06 (TAP M10)
N150 S200 M03
N160 X50 Y0 Z27                         Point a, approach
N170 G77 H30 S7                         Branch to subroutine %30
N180 G90 ED0                            Offset cancel
N190 G77 N10 N10
N200 M02
```

Subroutines

```
%10
$0 DRILLING
N10 G90 G81 X50 Y0 Z-5 F90 M08          Drilling cycle
N20 G80 G91 ED45                        Angular offset in incremental dimensions

%20
$0 SPOT FACING
N10 G90 G82 X50 Y0 Z17 EF1 F50 M08      Spot facing cycle
N20 G80 G91 ED45                        Angular offset in incremental dimensions

%30
$0 TAPPING
N10 G90 G84 X50 Y0 Z-8 F300 M08         Tapping cycle
N20 G80 G91 ED45                        Angular offset in incremental dimensions
```

4.12.6 Table Offset by DAT3

DAT3 Table offset.

This function applies to rotary axes A, B or C.

The offsets can be applied by entering values:
- on the next page (.../..) of the Shift mode on the NC,
- by external parameter E,
- by processor interchange.

The DAT3 feature takes into account the theoretical shifts caused to a pair of axes when the part zero is not co-incident with the centre of rotation.

Rotary axes and axis pairs affected by DAT3

Rotary axis	Axis pairs
A	YZ
B	ZX
C	XY

REMARKS *The axes of the pair to which a DAT3 can be assigned are identified by stars on the SHIFT screen page.*

A value entered for an axis not marked by a star is ignored.

Entry of DAT3 on the SHIFT page of the NC

Refer to the operator's manual.

Entry of DAT3 by external parameters (see Sec. 6.2)

Parameters E6x004 are use to programme the offset (x = physical address of the axis). Example:

Programming offsets for the B axis affecting the XZ pair.

```
N.. ...
E60004=-50000          Negative 50 mm offset on X
E62004=70000           Positive 50 mm offset on Z
N..
```

The table below is shown with a slaved B axis (see Chapter 3).

REMARK *The representation is the same for the A and C axes.*

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 inhibited 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 (XY Plane)

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

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. 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 Y axes of the tangent to the curve origin have X-3, Y4 as relative values.

```
N.. X-3 Y4   Tangent to the origin
X.. Y..
X.. Y..
X.. Y..      Points on the curve
X.. Y..
X.. Y..
N.. X.. Y..  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 XY plane with a call to definition blocks in the main programme.

```
%350
G79 N100
N10 X1 Y Z                          Tangent to the origin
X-240 Y0 Z0 F300                    Start of the curve and definition of the
X-200 Y10 Z5 (c)                    axes to be interpolated
X-180 Y20 Z10
X-160 Y30 Z15
X-140 Y20 Z20
X-80 Y40 Z35
X0 Y60 Z40                          Points on the curve (c to n)
X80 Y40 Z35
X140 Y20 Z20
X160 Y30 Z15
X180 Y20 Z10
X200 Y10 Z5
X240 Y0 Z0 (n)
N20                                 No tangent in the end point
N100 T3 D3 M6 (TOOL R5)
N110 S3000 M40 M3
N120 G48 NC5 N10 N20                Definition of the curve
N130 G0 X-300 Y20                   Point a
N140 Z0
N150 G1 G41 Y0                      Point b
N160 X-240                          Point c
N170 G06 NC5                        Curve execution command
N180 G1 X300                        Point p
N190 G0 G40 Y20                     Point q
N200 G0 G52 Z0 M5
N210 M2
```

Representation of machining

Programming of two consecutive spline curves defined in the XY plane with call to definition blocks in two subroutines

```
%300
N10 G0 G52 Z0
N20 T1 D1 M6 (CUTTER R5)
N30 S2000 M40 M3
N40 G92 R1
N50 G48 NC1 H31 N10 N70          Curve 1 definition
N60 G42 X20 Y-10                 Point a
N70 Z-3
N80 G1 X0 F200                   Point b
N90 EA90 EB5 ES-                 Point c
N100 EA180 X-13 Y10              Point d
N110 G06 NC1                     Curve 1 execution command
N120 G48 NC2 H32 N10 N50         Curve 2 definition
N130 G06 NC2                     Curve execution command
N140 G0 G40 X20
N150 G52 Z0 M05
N160 M02

%31
N10 X-1 Y0                       Tangent to the origin
N20 X-13 Y10
N30 X-50 Y70
N40 X-78 Y75                     Points on curve 1 (d, e, f, g, h)
N50 X-104 Y55
N60 X-140 Y0
N70 X0 Y-1                       Tangent to the end point

%32
N10 X0 Y-1                       Tangent to the origin
N20 X-140 Y0
N30 X-82 Y-60                    Points on curve 2 (h, i, j)
N40 X0 Y0
N50 X1 Y5                        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..

4

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.. Y..
N..                                  } Points in curve 1
N..
N150 X.. Y..
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 Other Functions

4.14.1 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 Ya F200	Feed rate 200 mm/min
N60 G04 F1.5	Dwell 1.5 seconds
N70 Xb Yb	After 1.5 seconds machining resumes at 200 mm/min
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 Y100	Dwell after movement
N..	

4.14.2 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. 4.18).

Properties of the Function

Function M12 is a decoded nonmodal «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 (see Parameter Manual).

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 the test (TEST) and sequence number search mode (SEARCH).

Example

```
N.. ...
N300 G00 Z80 M12 $0 MOVEMENT ON X, THEN CYCLE START
N310 ..
N..
```

4.14.3 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.. Y.. Z.. [F..] [$0 ...]

G01/G02/G03	Linear or circular interpolation.
G12	Enabling of rapid feed by handwheel.
X.. Y.. Z..	End point.
F..	Feed rate.
$0 ...	Message for the operator (see Sec. 4.18).

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 machine parameter P13 (see Parameter Manual).

Example

```
N.. ...
N60 G01 G12 X.. Y.. F200 $0 ACTUATE THE HANDWHEEL
N..
```

4.14.4 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 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
```

4.14.5 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.. Y.. Z.. A.. B.. C..

G00/G01/G02/G03	Linear or circular interpolation.
M10	Clamp.
M11	Unclamp.
X.. Y.. Z.. A.. B.. 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 wait for a report (CRM) before executing the movements in the next block.

After a reset, function M10 is initialised on the axis groups which have axes that can be clamped.

Example

N.. ...	
N40 M10	Clamp.
N50 G00 X.. Y.. Z.. B..	The X, Y, Z, B axes are not clamped.
N60 G01 X.. Y.. Z.. F200	The B axis which does not move is clamped.
N..	
N210 M11	Unclamp.
N..	

4.14.6 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.. Y.. M08                Coolant 1 on
N50 G01 Z.. M07                    Coolant 2 on
N..
N230 G00 G52 Z-100 M05 M09         All Coolant off
N..
```

4.14.7 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. 4.18).

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.. Y.. F200 M08
N..
N190 G00 G40 Z200 M09                Tool retraction before intervention
N200 M00 $0 REMOVE CHIPS BEFORE      Stop and message
FINISHING
N210 G00 G41 X.. Y.. Z.. M03         Continuation of the programme
N..
```

4.14.8 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. 4.18).

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.

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.. Y.. F200 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.. Y.. M03               Continuation of the programme
N..
```

4.14.9 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 of 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
N.. ...
N.. ...
N90 X.. Z..
N100 L110 = 1 + L110
N110 XL110
N120 L120 = L110 + L10
N130 ZL120
N..
N..
```

```
%2
N..
N80 M999
N90 X.. Z..
N100 L110 = 1 + L110
N110 XL110
N120 L120 = L110 + L10
N130 ZL120
N140 M998
N..
```

4.14.10 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 continuation.

Properties of the Function

Function M997 is a decoded modal «pre» function.

Cancellation

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 forcing can only be cancelled by a cancellation function.

4.14.11 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 «post» 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.

4.14.12 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).

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.. Y..
/N310 Z..
/N320 G01 Y.. F200
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.. Y..
/N160 Z..
/N170 G01 X.. F150
/N180 G00 G40 Z150
/N190 M01 $0 CHECK DIMENSION 20 AND CORRECT D11 IF NECESSARY
N200
N..
```

4.14.13 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..
```

4.14.14 Scaling Factor

G74	Scaling factor enable.

This function allows execution of scaled parts from a single programmed part or form. The scale ratio can be entered from the keyboard or programmed.

G73	Scaling factor cancel.

Syntax

N.. [G40] **G74/G73**

G74	Scaling factor enable. The ratio declared can be between 1 and 9999 to 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

Functions G73 and G74 cancel one another.

Function G74 is cancelled by the end of programme function (M02).

Notes

Scaling is around the programme origin (OP).

The 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.

The ratio affects:
- the values programmed for the primary and secondary axes (X, Y, Z, U, V, W),
- the programme origin offset (G59).

The ratio 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),
- table off-centering (DAT3),
- tool dimensions (L, R, @),
- programming with reference to the measurement origin (G52),
- the clearance dimension for canned cycles G81 to G89.

Entry of the scaling factor from the alphanumeric keyboard

Refer to the operator's manual.

Examples

Programming of 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.. Y..
N..
N200 G73                            Scaling factor cancel
N..
```

The scaling factor value 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..
```

Execution of shapes using scaling factors

Shape 1: Scale 1 from the programme zero.

Shape 2: Scale 0.5 with origin offset (G59) on X-30.

Shape 3: Scale 1.5 with origin offset (G59) on X45 and angular offset (ED) of 15 degrees.

REMARK *As the origin offset is affected by the scaling factor, values assigned to G59 must be proportional to the programmed scaling factor.*

Main programme

```
%300
N10 G00 G52 Z0
N20 T01 D01 M06 (BALL END CUTTER,
DIAMETER=8)
N30 S2000 M03 M40
N40 G59 X0 Y0                          Zero origin offset
N50 G74 E69000 = 1000                  Scaling factor 1
N60 G77 H3                             Subroutine call for shape 1
N70 G59 X-60                           Origin offset
N80 G74 E69000 = 500                   Scaling factor 0.5
N90 G77 H3                             Subroutine call for shape 2
N100 G59 X30 ED-15                     Origin offset
N110 G74 E69000 = 1500                 Scaling factor 1.5
N120 G77 H3                            Subroutine call for shape 3
N130 G00 G52 Z0 M05 M09
N140 M02
```

Subroutine

```
%3
$0 SHAPE C
N10 X10 Y5 Z3                          Point a
N20 G01 Z-2 F100 M08                   Approach on Z
N30 G03 X-10 Y5 R10 F300               Point b
N40 G01 Y-5                            Point c
N50 G03 X10 Y-5 R10                    Point d
N60 G73 G00 Z3                         Scaling factor cancel
```

4.14.15 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- A- B- C-

G51	Mirror function.
X- Y- Z- A- B- C-	The minus sign (-) enables the mirror function on the X, Y, Z axes or A- B- C-.

Properties of the Function

Function G51 is nonmodal. The axis arguments (X, Y, Z, A, B, C) 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+, Z+, A+, B+ or C+ 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 a 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 programmed axis X, Y, Z, A, B or C which is reversed. 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),
- table off-centering (DAT3),
- the tool dimensions (L, 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..
```

Influence of the mirror function on the A, B or C rotary axes

The function reverses the direction of rotation of the rotary axes and takes the complement to 360 degrees.

If B30 is programmed: the axis moves to 330 degrees in the negative direction.

If B-300 is programmed: the axis moves to 60 degrees in the positive direction.

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 interpolations

The function reverses the clockwise (G02) and counterclockwise (G03) rotations.

Example

Execution of a shape using mirror function G51 in the XY plane (G17)

```
%30
N10 G00 G52 Z0
N20 T05 D05 M06 (CUTTER DIAMETER=5)
N30 S1500 M40 M03
N40 G00 X30 Y10 Z2                    Point a, approach on XYZ
N50 G01 Z-2 F50 M08                   Penetration on Z
N60 X20                               Point b
N70 G02 X10 Y10 R5                    Point c
N80 G00 Z2                            Retraction on Z
N90 G51 Y-                            Mirror on Y
N100 G77 N40 N80                      Execution of shape 2
N110 G51 X- Y-                        Mirror on X and Y
N120 G77 N40 N80                      Execution of shape 3
N130 G51 X- Y+                        Mirror on X, cancellation of mirror on Y
N140 G77 N40 N80                      Execution of shape 4
N150 G51 X+ Y+                        Cancellation of mirror on X and Y
N160 G00 G52 Z0 M05 M09
N170 M02
```

4.14.16 Overall Dimensions of the Part in 3D Graphic Display

EM-/+	Overall dimensions of the part in 3D graphic display.

Programming of the part dimensions allows display of machining in 3D (see operator's manual).

Syntax

N.. **EM**- X.. Y.. Z.. **EM**+ X.. Y.. Z..

EM- X.. Y.. Z..	Minimum dimensions of the part blank.
EM+ X.. Y.. Z..	Maximum dimensions of the part blank.

Notes

For 3D graphic display of the part, it is necessary to:
- declare functions EM+ and EM- in the same ISO block,
- declare the tool dimensions on the «TOOL OFFSETS» pages.

3D graphic display is used for machining performed with cylindrical, toroidal and spherical cutters.

Machining operations depending on the shape of a tool with a special profile are not displayed.

Example

Dimensions of the part blank to be programmed for 3D display

```
%50
N10 ...
N..
N40 ...
EM- X0 Y0 Z0 EM+ X100 Y50 Z20
N50 ...
N..
N..
N..
N450 M02
```

4.14.17 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 blocks including movements and functions processed in state G999 are enabled and executed with the exception of certain functions that 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 Y50 G04 F5
N.. G01 Z10
N.. ...
N.. M05
N.. G998
N.. ...
N.. ...
N.. X200 T02 M06
N.. ...
N.. G997
```

Blocks concatenated by G999

Enabling of S2500 M03 («pre» M function) and movements on the axes to X100, Y50 and Z10

Movement to X200 and enabling of T02 M06

Enabling of G04 F4 (dwell) and M05 («post» M function)

4.14.18 3D Curve Smoothing

G104 3D curve smoothing.

This function executes 3D curves of any shape. It automatically performs polynomial interpolation between a sequence of points defined by their X, Y and Z coordinates. Curve smoothing must be programmed by a minimum of three blocks.

General Syntax

N.. X.. Y.. Z..

N.. [G01] **G104** X.. Y.. Z.. [F..]

[Intermediate points]

N.. G80 X.. Y.. Z..

X.. Y.. Z..	Coordinates of the curve start point.
G104 X.. Y.. Z..	Curve smoothing function and coordinates of the second point on the curve.
Intermediate points	Sequence of blocks each containing the coordinates of the intermediate points on the curve (optional).
G80 X.. Y.. Z..	Cancellation function G80 followed by the curve end point coordinates.

Property of the Function

Function G104 is modal.

Cancellation

Function G104 is cancelled by function G80.

Notes

The block containing the curve smoothing start point must always be programmed before calling function G104.

The block containing the end point must always include function G80.

As the function is interpreted directly by the system without prior computations, there is no limit to the number of points in the curve and no stop at the end.

The blocks defining curve smoothing may contain:
- Miscellaneous M functions,
- Programme variables L,
- External parameters E.

Curve smoothing can be executed in all the CNC Modes: AUTO, SINGLE, etc. The SEARCH mode can be used for partial execution of a curve.

The location of each point on the curve can be measured by teach-in (see Operator Manual).

Example

Programming of curve smoothing to execute polynomial interpolation between points a and f.

REMARK *The tangents to the end points (points a and f) are the tangents to the circles going through the first three points (a-b-c) and the last three points (d-e-f) of the curve.*

```
%123
N.. ...
N.. ...
N100 Xa Ya Za                      Curve start point
N110 G01 G104 Xb Yb Zb F200        Smoothing function and second point
N120 Xc Yc Zc
N130 Xd Yd Zd                      Intermediate points
N140 Xe Ye Ze
N150 G80 Xf Yf Zf                  Cancellation and end point
N.. ...
M02
```

4.15 Special Programming for Multi-axis Groups

4.15.1 Programme Declaration

A machining programme for multi-axis groups is an association of the programmes for each axis group.

The programme names have a common radical followed by an index specifying the group, e.g.:

%61.1 Programme number for group 1

%61.2 Programme number for group 2

Programme number format: %05.1 (indices .1 to .8).

The programmes for each of the groups must be loaded in the NC memory. Otherwise, the machining programme cannot be called (unless one or more groups are inhibited).

4.15.2 Programming Notes

It is essential not to use the programme start symbol «%» in comments in the programme body.

Programmable Axes

The NC can control 32 axes 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: Machine with 9 axes in 3 groups

Group 1: axes X Y Z A

Group 2: axes U V W B

Group 3: axis C

Axes in different groups can have the same address. In the above example, axes U V and W of group 2 can have addresses X, Y and Z.

Each axis group is enabled by the PLC interface. One or more groups can be inhibited by a switch on the machine panel (see manufacturer's technical data).

T Function

The machine manufacturer has been given two registers for the T address.

The same tool number T.. can be used for different tools in the programme with index .1 and in one other programme, e.g.:

```
%10.1                    %10.2
N.. ...                  N.. ...
N.. T05 M06 (DRILL)      N.. T05 M06 (CUTTER)
N..                      N..
```

Different tool numbers must be used for the programmes with indexes .2 to .8.

For more details on processing of the T function, refer to the Automatic Control Function Programming Manual.

Tool Corrections

The tool correction tables are common to all the programmes.

Machining Cycles

Function G46 (pocket or facing cycle with any contour) is only active on one axis group at a time. If two cycles are programmed simultaneously on two different axis groups, the system returns error message 260 (working memory busy).

Programme Variables

L variables can be used in each programme.

L0 to L19, L100 to L199 and L900 to L959: these variables can be used for each axis group (no interaction between programmes).

External Parameters

External parameters E are common to all the programmes (except for parameters E50000, E51000, E6x000 and E7x000 which relate to their individual axis group).

4.15.3 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).

4.15.3.1 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.

4.15.3.2 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).

4.15.3.3 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).

4.15.3.4 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. 4.11.4) 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.

4.15.4 Programming the Spindles

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

A spindle can be:
- controlled by an axis group,
- independent.

Spindle control by an axis group

A spindle can only be controlled by a single axis group at a time, and an axis group can only control one spindle at a time.

In an axis group, selection of another spindle by one of functions M62 to M65 is ignored unless the spindle is not controlled by any other group (otherwise, the system returns error message 38).

Independent spindle

An independent spindle is a spindle which has not yet been controlled by any axis group or which has been released by the group that was controlling it.

A spindle is released from a group by function M61 (see Sec. 4.15.5).

Only an independent spindle can be selected by a group for control by this group (see Sec. 4.3.5).

Spindles not assigned to a group are independent.

Spindle Control

As described (see Sec. 4.3.5).

Spindle Measurement

As described (see Sec. 4.3.6).

Reminder

On a machine equipped with a milling spindle and a turning 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.

4.15.5 Releasing the Current Spindle in the Axis Group

M61	Release of current spindle in the axis group.

This function releases the current spindle in the group to allow it to be controlled by another group.

Syntax

N.. **M61**

M61 Release of current spindle in the 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 a spindle by function M61, the spindle can be controlled by one of functions M62 to M65 in the programme of another group.

When the spindle to be controlled has not been released, the system returns error message 38.

Example

Programme for group 1 Programme for group 2

```
%40.1                          %40.2
N.. ...                        N.. ...
N..                            N.. G97 S500 M03 M65 M40
N..                            N..
N..                            N.. M61                 Release of spindle 2
N..                            N..
N.. G97 S1000 M03 M41 M65      N..
N..                            N..
```

4.15.6 Synchronising the Axis Groups

G78	Axis group synchronisation.

This function identifies and manages the 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 is defined by two digits separated a decimal point: - j is the marker number tested, - i is the index of the group whose marker is 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 has a default value of zero (Q0).

During the programme, the markers can be reset by programming G78 Q0.

Notes

Function G78 can be followed by several arguments, but at least one is mandatory.

The declaration of a marker and the programme continuation conditions can be programmed in the same block, e.g.:

```
N.. G78 Q3 P5.2 P6.3
```

The axis group selected and the associated step transition conditions can be enabled selectively by the automatic control function (see Automatic Control Function Programming Manual).

The markers included in a a programme must be numbered in increasing order, but not necessarily sequentially, e.g.:

REMARK *A marker is crossed when the tested marker is reached.*

```
N.. ...
N90 G78 Q1 ...
N..
N200 G78 Q6 ...
N..
```

Example

Programme synchronisation with markers

```
%50.1                      %50.2                  %50.3
N..                        N..                    N..
N..                        N..                    N.. G78 P2.1 P1.2 Wait f
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.. G78 Q2 P1.2 Wait for Q1 N..                   N..
N..             of %50.2   N..                    N..
N..                        N..G78 Q1              N..G78 Q6
N..                        N..                    N..
N.. G78 P6.3    Wait for Q6 N..
N..             of %50.3   N..G78 P5.1Wait for 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 by function G78 or initiated by programming M functions.

Programming by 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 programme command (M02) cancels the conditions on this programme and is equivalent to inhibiting the programme.

If all the programmes are on wait, the system returns error message 33 (synchronisation error). If this error is detected 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 continue together when markers Q6, Q4 and Q5 are reached.

Programming with M functions

The following functions are used as meeting points:
- programmed feed stop (M12),
- programme stop (M00),
- optional stop (M01).

When M00 or M01 (enabled) are programmed in only one of the programmes, the movements of the axes in the group are stopped and the system waits for the end of the other programmes (M02) before executing M00. When the cycle restarts, only this programme continues.

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 was not enabled, each of the M01 sequences is a meeting point, but the cycle continues automatically when all the programmes are on their respective M01 commands.

4.16 Special Programming of PLC Axes

4.16.1 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 programme memory zone with a number strictly above zero. The search for the programme is first made in zone 1 and ends where applicable in zone 3.

When the main programme is stored after creation or replaced after editing, the PLC axis group must be reset so that its presence is taken into account.

Structure of a PLC axis group programme

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
..
```

4.16.2 Programming of the PLC Axes

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),
- pocket or facing cycle with any contour (G46...).

4.16.2.1 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).*

4.16.3 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 a 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.

4.16.4 Exchanging Axes between Groups

The NC and PLC axis groups can exchange axes via the part program. 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 authorize start of the cycle).*

4.16.5 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).

4.17 Special Features of Mixed Machines (MX)

Machines combining standard milling functions with turning functions are called mixed machines.

Such mixed machines involve:
- Special features related to the machine axes,
- Special programming of certain ISO functions.

Below, the descriptions are limited to the following types of machines:
- reamers,
- mixed machines.

4.17.1 Features Related to the Machine Axes

Programmable axes on MX:
- Primary axes X, Y and Z,
- Secondary axes U, V and W,
- Rotary axes A, B and C.

For general information on the axes, refer to the milling and turning programming manuals.

4.17.1.1 Axis Declaration

Reamer type machines

For such machines, radial axis U must be declared independent.

Reamer or mixed machines

On such machines, the Z and W axes can be declared as carried or independent.

REMARK *When Z and W are declared carried, the tool length correction is applied to only one of these two axes.*

4.17.1.2 X and U Axes with Reference to Diameter or Radius

The X and U axes can be declared programmed with reference to the diameter or radius (see the turning programming manual).

Programming the X and U axes with reference to diameter is used only when programming function G20 defining the ZX interpolation plane for turning.

4.17.1.3 Axis Orientation and Definition of the Programme Origin

Reamer Type Machine

The programme origin (OP) for the U axis is defined in the spindle axis. Homing on the U axis is carried out in the same way as on the other axes (with DAT1 and DAT2).

Mixed Type Machine

The programme origin (OP) for the X axis is defined in the table axis.

4.17.1.4 Assignment of Tool Dimensions to the Axes

Types of Tool Dimensions Used on Mixed Machines

Type 0: Milling tool (dimensions D.. L.. R.. @..)
Type 1: Turning tool (dimensions D.. X.. Z.. C..)
Type 2: Boring tool (dimensions D.. U.. Z.. C..)

Tool Type Selection

The tool type (0, 1 or 2) is determined when entering (or loading) the tool dimensions from the keyboard, a peripheral or by programming (for additional information, refer to the turning and milling operator manuals).

The selection of the correction type requires declaring two mandatory addresses for each tool dimension type:
- L.. and R.. for type 0 (milling)
- X.. and Z.. for type 1 (turning)
- U.. and Z.. for type 2 (boring).

Notes on Tool Dimension Assignment

Milling

The tool length L is assigned to the tool axis specified by tool orientation function G16. The tool axis must be a primary axis or a carried secondary axis. In no case can it be an independent secondary axis.

The tool radius R for radius offset (G41 or G42) is assigned to the two axes of the interpolation plane (G17, G18 or G19), be they primary, secondary, carried or independent.

Turning

The X and Z dimensions are assigned to axes X and Z or U and W if they are declared as carried (see machine parameter P64).

The tool radius R for radius offset (G41 or G42) is assigned to the two axes of the interpolation plane (G20), be they primary, secondary, carried or independent.

Boring

The U dimension is assigned only to the U axis.

The Z dimension is assigned to the Z axis or to the W axis if it is declared as carried.

The tool radius R for radius offset (G41 or G42) is assigned to the two axes of the interpolation plane (G20), be they primary, secondary, carried or independent.

REMARK *The axes are declared as carrier/carried in machine parameter P64 (see Parameter Manual).*

Requirements for Taking Into Account the Tool Dimensions

Type 0 (milling)

The tool L, R and @ dimensions are taken into account when the type 0 correction number Dxxx and the axes concerned are programmed. The interpolation plane is G17, G18 or G19.

Type 1 (turning)

The tool X, Z and R dimensions are taken into account when the type 1 correction number Dxxx and the axes concerned are programmed. The tool nose orientation C is taken into account on the X and Z axes. The interpolation plane is G20.

Type 2 (boring)

The tool U, Z and R dimensions are taken into account when the type 2 correction number Dxxx and the axes concerned are programmed. The tool nose orientation C is taken into account on the U and Z axes. The interpolation plane is G20.

REMARK *For type 1 tools (turning) or type 2 tools (boring), the tool length correction is not taken into account on the Y and V axes.*

Preselection of the Programme Origin (with function G92)

Function G92 includes an extension dedicated to boring and mixed type machines.

The axes and values programmed after the function define the current the position of the tool with respect to the new programme origin. DAT1 is computed according to the current tool length L; taking into account of L is therefore defined by the tool type: 0, 1 or 2.

Type 0 (milling)

L is taken into account according to the tool axis orientation function (G16 P, G16 Q or G16 R) on:
- the main axis if it is independent
- one of the axes if they are carrier/carried axis pairs.

Type 1 (turning)

L is taken into account on the X axis (or U if X and U are carried) and the Z axis (or W if Z and W are carried).

Type 2 (boring)

L is taken into account on axes U and Z (or W if Z and W are carried).

Error Messages

If the tool type is incompatible with the interpolation plane, the system returns error message 77:
- Tool type 0 (milling) with turning plane G20, G21 or G22,
- Tool type 1 (turning) or type 2 (boring) with milling plane G17, G18 or G19.

4.17.2 Notes on Programming the ISO Functions

Certain ISO functions used for milling and turning have special features, generally related to the machine axes declared. The other functions are used in the standard way.

For general information on the functions, refer to the turning and milling programming manuals.

Milling Function

G17, G18, G19: Interpolation plane selection

Possibility of various machining and contouring operations in the three main planes involving all the primary axes (X, Y, Z), secondary axes (U, V, W) and rotary axes (A, B, C).

Milling cycles

All the standard milling cycles can be used in the three interpolation planes.

Turning Functions

G20, G21, G22: Interpolation plane selection

Possibility of transforming the ZX coordinate system (G20) by programming functions G21 and G22 with the rotary C axis (do not confuse the spindle with the rotary C axis). Programming of G20, G21 or G22 sets function G18 (ZX plane for milling).

Turning cycles

The following cycles can only be used in the ZX plane (G20)
G63: Roughing cycle with groove.
G64: Turn/face roughing cycle.
G65: Groove roughing cycle.
G66: Plunging cycle.
G33: Constant lead thread cutting. Possible on X or U or Z or W axes.
G38: Sequenced thread cutting. Possible on X or U or Z or W axes.

Common Turning and Milling Functions

M62, M63, M64, M65: Spindle control

Possibility of controlling the spindles simultaneously or not by choosing the spindle number to be controlled (1 to 4).

M66, M67, M68, M69: Spindle measurement

Possibility of measuring the spindles by selecting the spindle number to be measured (1 to 4).

G94 or G95: Feed rate

Possible feed rate in mm/min (G94) or mm/rev (G95) on all the linear axes (X, Y, Z, U, V, W) and selection of the measured spindle in G95 (see M66 to M69).

G96: Constant surface speed in m/min

Possibility of defining the speed of one of the spindles (see M66 to M69) according to the position of the X or U axis. This function is only active in planes G18 (milling) or G20 (turning). The choice of the X or U axis is related to the type of correction selected (1 or 2).

4.17.3 Interactive Programming on Mixed Machines

Interactive programming on mixes machines is similar to the specific use for turning or milling:
- All the possibilities of turning or milling part programming are available
- For turning, the use of technological data files is possible (tool and cutting/material conditions files)
- For milling, there are no technological data files, which are therefore not available in the milling function on a mixed machine.

For use of interactive programming, refer to the following manuals:
- PROCAM TURN Interactive Programming
- PROCAM MILL Interactive Programming
- PROCAM TURN Technological Data.

4.18 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.

4.18.1 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..
```

4.18.2 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 message error 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 =
```

5 Profile Geometry Programming

5.1	**Profile Geometry Programming (PGP)**		5 - 3
	5.1.1	General	5 - 3
	5.1.2	Definition of the Geometric Elements	5 - 3
	5.1.3	Definition of the Addresses Characterising PGP	5 - 5
	5.1.3.1	Addresses with Values	5 - 5
	5.1.3.2	Addresses without Values	5 - 7
	5.1.4	Programming of Geometric Elements	5 - 9
	5.1.4.1	Programming of Fully Defined Geometric Elements	5 - 9
	5.1.4.2	Programming of Geometric Elements not Completely Defined	5 - 10
	5.1.4.3	Programming Chamfers and Fillets Between Two Elements	5 - 17
	5.1.4.4	Examples of Programming in PGP	5 - 18
5.2	**PROFIL Function**		5 - 24
	5.2.1	Accessing PROFIL	5 - 24
	5.2.2	Calling a Contour Created by PROFIL	5 - 25
	5.2.2.1	Calling a Contour by Function G77	5 - 25

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 to the plane selected (XY, ZX, YZ) by function G17, G18 or G19 (the change of plane must be programmed on a fully defined point),
- allows programming of the tool axis (Z in the XY plane) in a block.

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 and 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 (excluding fillets and/or 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 following PGP addresses are processed in the XY plane (G17). For the other planes (ZX and YZ), use the axes of the plane selected.

5.1.3.1 Addresses with Values

X../Y.. or **X.. Y..**

X../Y.. or X.. Y..:

Coordinates of the end point of a line.

EA..

EA..: Angle element of a line.

X.. Y..

X.. Y..: Coordinates of the end point of a circle.

I.. J..

I.. J..: 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 and circle/circle intersecting elements, the system allows two possible solutions. Programming of the discriminant with ES (ES+ or ES-) is therefore mandatory (see Sec. 5.3.2, figures 3a and 3b for instance).

Programming the discriminant with tangent elements

The system allows only two possible solutions (only continuous tangencies without backtracking are allowed by the system).

When there are two possible solutions, these solutions result in:
- the creation of an arc less than 180 degrees, and
- 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 Y..
or N.. G01 X.. Y..
```

```
   N.. G01 EA.. X..
or N.. G01 EA.. Y..
```

Fully defined circular geometric element (point «a» defined)

```
N.. G02/G03 X.. Y.. I.. J..
```

```
N.. G02/G03 X.. Y.. R.. E+/E-
```

5

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).

```
N.. G01 EA.. ES
N.. EA X.. Y..
```

```
N.. G01 EA.. ES
N.. EA..
N.. G02/G03 I.. J.. R../X.. Y..
```

```
N.. G01 EA.. ES-
N.. G02/G03 I.. J.. R../X.. Y..
```

```
N.. G01 EA.. ES+
N.. G02/G03 I.. J.. R../X.. Y..
```

```
N.. G01 EA.. ES-
N.. G02/G03 I..J..
N.. G01 EA.. X.. Y..
```

```
N.. G01 EA.. ES+
N.. G02/G03 I..J..
N.. G01 EA.. X.. Y..
```

N.. G01 EA.. ES-
N.. G02/G03 I.. J..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G01 EA.. ES-
N.. G02/G03 I.. J.. ET-
N.. G02/G03 I.. J.. R../X.. Y..

N.. G01 EA..
N.. G02/G03 I.. J..

N.. G01 EA..
N.. G02/G03 R..
N.. G01 EA.. X.. Y..

N.. G01 EA..
N.. G02/G03 R.. X.. Y..

N.. G01 EA.. ET+
N.. G02/G03 R.. X.. Y..

N.. G01 EA..
N.. G02/G03 R..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G01 EA.. ET+
N.. G02/G03 R..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G01 ET
N.. G02/G03 I.. J.. R../X.. Y..

N.. G01 ET
N.. G02/G03 I.. J..
N.. G01 EA.. X.. Y..

N.. G01 ET
N.. G02/G03 I.. J..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G01 ET
N.. G02/G03 I.. J.. ET-
N.. G02/G03 I.. J.. R../X.. Y..

The first element is a circle (start point «a» is fully defined).

```
N.. G02/G03 I.. J..
N.. G01 EA..
```

```
N.. G02/G03 I.. J..
N.. G01 X.. Y..
```

```
N.. G02/G03 I.. J..
N.. G01 ET
N.. G02/G03 I.. J.. R../X.. Y..
```

```
N.. G02/G03 I.. J..
N.. G01 ET
N.. G02/G03 I.. J.. R../X.. Y..
```

```
N.. G02/G03 I.. J..
N.. G02/G03 I.. J..
```

```
N.. G02/G03 I.. J..
N.. G02/G03 I.. J..
```

N.. G02/G03 I.. J..
N.. G02/G03 R.. X.. Y..

N.. G02/G03 I.. J..
N.. G02/G03 R..
N.. G01 EA.. X.. Y..

N.. G02/G03 I.. J..
N.. G02/G03 R..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G02/G03 I.. J..
N.. G02/G03 R..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G02/G03 I.. J.. ES+
N.. G01 EA.. X.. Y..

N.. G02/G03 I.. J.. ES-
N.. G01 EA..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G02/G03 I.. J.. ES+
N.. G02/G03 I.. J.. R../X.. Y..

N.. G02/G03 I.. J.. ES-
N.. G02/G03 I.. J..
N.. G01 EA X.. Y..

N.. G02/G03 I.. J.. ES+
N.. G02/G03 I.. J..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G02/G03 I.. J.. ES-
N.. G02/G03 I.. J.. ET+
N.. G02/G03 I.. J.. R../X.. Y..

N.. G02/G03 R..
N.. G01 EA.. X.. Y..

N.. G02/G03 R.. ET-
N.. G01 EA.. X.. Y..

N.. G02/G03 R..
N.. G01 EA..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G02/G03 R..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G02/G03 R..
N.. G02/G03 I.. J..
N.. G01 EA.. X.. Y..

N.. G02/G03 R.. ET+
N.. G02/G03 I.. J..
N.. G01 EA.. X.. Y..

N.. G02/G03 R..
N.. G02/G03 I.. J..
N.. G02/G03 I.. J.. R../X.. Y..

N.. G02/G03 R..
N.. G02/G03 I.. J..
N.. G02/G03 I.. J.. R../X.. Y..

5.1.4.3 Programming Chamfers and Fillets Between Two Elements

```
N.. G01 EA.. ES EB-..
N.. G01 EA.. X.. Y..
```

```
N.. G01 EA.. ES- EB+..
N.. G02/G03 I.. J.. X.. Y..
```

5

5.1.4.4 Examples of Programming in PGP

Examples

In PGP, definition of the part profile in the XY plane (G17)

Machining paths

```
%61
N10 G00 G52 Z0
N20 T03 D03 M06 (CUTTER DIAMETER=10)
N30 S600 M40 M03
N35 G92 R10
N40 G00 G41 X-60 Y-25                    Point a, approach
N50 Z-6                                  Position on Z
N60 G01 Y0 EB-5 F150 M08                 Point b
N70 EA0 ES EB15                          Point c
N80 EA60 X-25 Y25                        Point d
N90 Y40 EB10                             Point e
N100 X25 EB10                            Point f
N110 Y25                                 Point g
N120 EA-60 Y0 EB13                       Point h
N130 X60 EB-5                            Point i
N140 Y-20                                Point j
N150 G00 G40 G52 Z0 M05 M09              Retraction
N160 M02
```

In PGP, definition of the half-profile of the part

Use of the mirror function to machine the other half-profile in the XY plane (G17).

Machining paths

```
%3
N10 G00 G52 Z0
N20 T01 D01 M06 (CUTTER DIAMETER=20)
N30 S280 M40 M03
N40 G00 G41 X60 Y0                    Point a, approach
N50 Z12 M08                           Position on Z
N55 G92 R10                           Limit value of tangential feed rate
N60 G03 X28.3 Y0 I44.15 J0 F200       Point b
N70 G02 I14.3 J0 R14 ET               Machining of the first half-profile
N80 G03 R16 ET
N90 G02 X0 Y-24 I0 J0
N100 G02 I0 J0 R24 ET
N110 G03 R16 ET
N120 G02 X-28.3 Y0 I-14.3 J0
N130 G03 X-60 Y0 I-44.15 J0
N140 G51 X- Y-                        Mirror
N150 G77 N60 N120                     Machining of the second half-profile
N160 G51 X+ Y+
N170 G00 G40 G52 Z0 M05 M09
N180 M02
```

In PGP, definition of the part profile in the XY plane (G17)

Machining paths

```
%18
N10 G00 G52 Z0
N20 T02 D02 M06 (CUTTER DIAMETER=6)
N30 S1200 M40 M03
N40 G92 R4                          Limiting of the tangential feed rate
N50 G00 G41 X61 Y20 Z3              Start point
N60 G01 Z-5 F50 M08                 Penetration on Z
N70 Y10 F120
N80 G02 I55 J10
N90 G01 ET
N100 G03 I37 J-3 R7
N110 G01 EA-90 ES+
N120 G02 I0 J0 R50
N130 I-22.5 J-38.971
N140 G03 I0 J0 ES-
N150 G01 EA94
N160 G02 I0 J0 R15
N170 G01 EA79 Y40 EB10
N180 EA0 ES+
N190 G02 I4 J19.2 R32 ES-
N200 G01 EA180
N210 G03 I25 J21 R5
N220 G01 ET
N230 G02 I55 J10 X61 Y10
N240 G01 Y0 F500
N250 G00 G40 G52 Z0 M05 M09
N260 M02
```

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 Accès à 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. It can be called by the subroutine call function G77.

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..
```

REMARK *If limits N.. N.. are used, the start and end block numbers N.. N.. must be created by the user.*

6 Parametric Programming

6.1	**Programme L Variables**		6 - 3
	6.1.1	Definition	6 - 3
	6.1.2	List of L Variables	6 - 3
	6.1.3	Variable Assignments	6 - 3
	6.1.4	Initialisation	6 - 3
	6.1.5	Application	6 - 3
	6.1.6	Use	6 - 4
	6.1.7	Operations Possible with L Variables	6 - 4
	6.1.8	Comparison Symbols Used with L Variables	6 - 5
	6.1.9	Conversion of the Internal Unit	6 - 5
	6.1.10	Programming Syntax for L Variables	6 - 6
	6.1.10.1	Assignment of a Variable to an NC function	6 - 6
	6.1.10.2	Declaration of a Variable in the Programme	6 - 7
	6.1.10.3	Test of a Variable for a Conditional Branch	6 - 8
	6.1.11	Notes on Programming Variables L100 to L199 and L900 to L959	6 - 9
	6.1.12	Equivalence of Variables L900 to L925	6 - 10
	6.1.13	Symbolic Addressing of Variables L900 to L925 and L926 to L951	6 - 10
	6.1.14	Examples of Programming L Variables	6 - 11
6.2	**External E Parameters**		6 - 20
	6.2.1	Definition	6 - 20
	6.2.2	Application	6 - 20
	6.2.3	Assignment	6 - 20
	6.2.4	Initialisation	6 - 20
	6.2.5	Use	6 - 20
	6.2.6	Operations Possible with External E Parameters	6 - 21
	6.2.7	Comparison Symbols Used with External E Parameters	6 - 22
	6.2.8	Conversion of the Internal Unit	6 - 22
	6.2.9	List of External E Parameters	6 - 23
	6.2.9.1	Exchange Parameters with the Automatic Control Function	6 - 23
	6.2.9.2	Programme Analysis Access Parameters	6 - 24
	6.2.9.3	Machine State Access Parameters	6 - 28
	6.2.9.4	Machining Parameters	6 - 31
	6.2.9.5	Group Axis Access Parameters	6 - 36
	6.2.9.6	Spindle Access Parameters	6 - 40
	6.2.9.7	Common NC Parameters	6 - 42
	6.2.9.8	Machine Axis Access Parameters	6 - 43

6.2.10 External E Parameter Programming Syntax 6 - 50
6.2.10.1 Assignment of an External Parameter to an NC Function 6 - 50
6.2.10.2 Declaration of an External Parameter in the Programme 6 - 51
6.2.10.3 Test on an External Parameter for a Conditional Branch 6 - 52
6.2.10.4 Examples of Use of External E Parameters 6 - 53

6.3 Address Equivalences 6 - 58

6.4 Transfer of the Current Values of L Variables and E Parameters into the Part Programme 6 - 59

6.5 Message Display with Wait for an Operator Response 6 - 61

6.6 Display of Messages with Parametric Value 6 - 63

6.7 Reading the Programme Status Access Symbols 6 - 64
6.7.1 Symbols for Accessing the Data of the Current or Previous Block 6 - 64
6.7.1.1 Symbols Addressing Boolean Values 6 - 65
6.7.1.2 Symbols Addressing Numerical Values 6 - 66

6.8 General Diagrams of Parametric Programming 6 - 68
6.8.1 Load of a Parametric Expression 6 - 68
6.8.2 Comparison for Conditional Branch 6 - 69

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.

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

Use of function 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).

Use of function M

Case of a system with 0.0001 degree as internal unit for rotary axes.

External parameter E74000 defines the reference position of axis 4 (B axis).

Conversion from 0.0001 degree (internal unit) to the programming unit in degrees. For any position on the B axis, the programming is:

L0=ME74000 then L0=[.IRX(8)]

Variable L0 returns the position on the B 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.

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 Sections 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..
```

6.1.11 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 be preceded by a block whose execution requires knowledge of the following block(s), e.g.:
- Profile Geometry Programming (PGP), or
- 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).

Writing of variables and transfer of current values into the part programme are suspended until execution of the previous blocks is completed (M999 allows anticipated execution of these operations).

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

6.1.14 Examples of Programming L Variables

Use of the variables with a conditional test on the number of holes

Execution of five holes in the XY plane (G17) with programming of a single drilling cycle.

```
%40
L1=20 (START ON X)
L2=25 (START ON Y)
L3=15 (INCREMENT ON X BETWEEN HOLES)
N10 G00 G52 Z-50
N20 T01 D01 M06 (DRILL)
N30 S600 M40 M03
N40 XL1 YL2
N50 L4=L4+1
N60 G81 Z-10 ER2 F100
N70 G00 G80 G91 XL3
N80 G90 G79 L4 < 5 N50

N90 G00 G80 Z200 M05
N100 M02
```

Positioning in the axis of hole 1

Drilling in cycle
Movement by the relative increment
Condition: if L4 < 5, branch to block N50, else continue

Use of variables assigned to rotation speeds, feed rates, dimensions and continuous test on the number of passes.

Facing in the XY plane (G17) by consecutive roughing and finishing passes in Z.

The diameter of the cutter is greater than the width of the part.

Programme origin:
- X OP = Cutter radius plus clearance,
- Y OP = Cutter axis in the part axis,
- Z OP = Unmachined top surface of the part.

```
%25
$0 FACING PROGRAMME
L1= 3 (Z VALUE OF ROUGHING PASS)
L2= 0.2 (Z VALUE OF FINISHING PASS)
L3= 3 (NUMBER OF ROUGHING PASSES)
L4= 200 (PASS LENGTH ON X)
L5= 100 (RETRACTION ON Y FOR RETURN)
L6= 150 (ROUGHING ROTATION SPEED)
L7= 120 (ROUGHING FEED RATE)
L8= 150 (FINISHING ROTATION SPEED)
L9= 80 (FINISHING FEED RATE)
N10 G00 G52 Z-50
N20 T01 D01 M06 (FACING CUTTER
DIAMETER=50)
N30 SL6 M40 M03
N40 X0 Y0 Z30                    Point a
N50 Z0                           Approach on Z0
N60 G91 Z-L1                     Pass setting on Z
N70 G01 X-L4 FL7 M08             Point b
N80 G00 YL5                      Point c
N90 XL4                          Point d
N100 Y-L5                        Point e
N110 L3 = L3 - 1                 Decrement
N120 G79 L3 < 0 N140             Condition: if L3 < 0, branch to block
                                 N40, else continue
N130 G79 N60
$0 FINISHING
N140 G00 Z-L2                    Finishing pass setting on Z
N150 G01 X-L4 SL8 FL9            Finishing
N160 G00 G90 Z0 M05 M09
N170 M02
```

6

Use of variables to compute positions and conditional test on angular values.

Finishing of the profile of a gear wheel with use of PGP in the XY plane (G17)

```
%21
N10 G00 G52 Z0
N20 T12 D12 M06 (CUTTER DIAMETER=8)
N30 S1000 M40 M03
L0 = 75 + 10 * C30                    Calculation of start point a on X
L1 = 75 + 10 * S30                    Calculation of start point a on Y
N35 G92 R1                            Limiting of tangential feed rate
N40 XL0 YL1 Z3 M08                    Point a, approach
N50 G01 Z-4 F50                       Approach on Z
N60 G41 EA180 ES+ F200
L3 = 75 * C30                         Calculation of point b on X
L4 = 75 * S30                         Calculation of point b on Y
N70 G03 XL3 YL4 IL0 JL1
N80 G02 I0 J0 ES- EB5
N90 G01 EA 210
N100 G03 I50 J0 R5
N110 G01 EA0 ES+ EB7.5
N120 G02 XL3 Y-L4 I0 J0               Point c
L10 = L10 - 60                        Decrement of the angular offset
N130 EDL10                            Angular offset
N140 G79 L10 > -360 N80               Test on 360°
N150 G03 IL0 JL1 R10 ES-
N160 G00 G40 EA90 XL0 YL1
N170 G40 G52 Z0 M05 M09
N180 M02
```

Drilling cycle with 12 holes angularly offset in the XY plane (G17)

Movement by circular interpolation between holes.

```
%90
N10 G00 G52 Z0
N20 T01 D01 M06 (DRILL)
N30 S500 M03 M40
N40 L1=100 (RADIUS)
N50 L2=5 (APPROACH Z)
N55 L3=-5 (DRILLING Z)
N60 L4=30 (ED)
N65 L5=12 (NUMBER OF HOLES)
N70 X0 YL1                          Point a, approach on XY
N80 ZL2
N90 G79 N150                        Jump to block N150
N100 G81 ZL3 F100                   Cycle
N110 G80 G91 EDL4                   Angular offset by 30 degrees
N120 L10=L10+1                      Increment
N130 G79 L10=L5 N160                Condition: if L10 = 5, jump to block
                                    N160
N140 G90 G03 X0 YL1 I0 J0 F5000     Circular movement
N150 G77 N100 N140 SL5              Repeat the cycle
N160 G00 G52 Z-50
N170 G52 X-100 Y-50 M05
N180 M02
```

Use of variables to compute the position, programme origin offset and double angular offset in the XY plane (G17)

Drilling cycle with 8 circular patterns each including eight 6 mm diameter holes.

Main Programme

```
%92
N10 G00 G52 Z0
N20 G52 X0 Y0
N30 T11 D11 M06 (DRILL DIA. 6)
N40 S4500 M03 M40
N50 L10=100 (RADIUS 1) L11=20 (RADIUS 2)
N60 L0=0 L1=L10*CL3 L2=L10*SL3 G59 XL1 YL2
N70 G77 H93                                Subroutine branch
N80 L3=L3+45 G79 L3<360 N60                Test on 360 degrees
N90 G00 G52 Z0 M05
N100 M02
```

Subroutine

```
%93
N10 EDL0 G81 XL11 Y0 ER2 Z-10 F500         Cycle
N20 L0=L0+45 G79 L0<360 N10                Test on 360 degrees
N30 G80 G00 Z5
```

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 cannot be «written» by the programmer as they are read-only parameters,
- when used for tests (with G79), parameters E20000 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), or
- 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

Use of function 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).

Use of function M

Case of a system with 0.0001 degree as internal unit for rotary axes.

External parameter E74000 defines the reference position of axis 4 (B axis).

Conversion from 0.0001 degree (internal unit) to the programming unit in degrees. For any position on the B axis, the programming is:

L0=ME74000 then L0=[.IRX(8)]

Variable L0 returns the position on the B 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 give additional information on the machine status during 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 0 or 1 as value 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.

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». A reset sets this parameter to «1».

E11015: Angle crossing management enabled

When this parameter is set to «0», the angles are not analysed when calculating the end of block speed.

E11017: Inclined plane function enabled

Read-only parameter set to indicate that the inclined plane function (G24+) is enabled.

E11018: RTCP function enabled

Read-only parameter set to indicate that the RTCP function (G26+) is enabled .

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

Read/write parameters

E32000: Minimum interpolation block execution time

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

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:
- 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 numbers declared in operations 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. It returns a number corresponding to the current mode:
- 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 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 active,
- 1: turning simulation with material removal,
- 2: dynamic simulation for turning or milling.

E41004: Job reference image

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

REMARK *Parameters E33xyz and E43xyz are not accessible unless Ladder variable %Qrc3B.1 = 1 (see Automatic Control Function Programming Manual in Ladder Language).*

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).

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 distance is expressed in the internal system unit by sampling.

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(G16...).

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: E51000=2 (positive orientation on the Z axis), E51000=102 (negative orientation on the Z axis).

E5y001 and E5y255 - Tool Corrections

Read-only parameters.

The units are in microns (except for E56001 to E56255 and E57001 to E57255).

E50xxx: Tool length L of correction xxx

E51xxx: Cutter tip radius @ of correction xxx

E52xxx: Tool radius R of correction xxx

E53xxx: Dynamic length correction L of correction xxx

E54xxx: Dynamic radius correction R 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 - E6x004 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

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). Parameters E60002 and E60003 are 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 maximum or minimum).

The following programming is used to modify the limits:

```
%25
N..
E60002 = Xb
```
Declaration of the minimum dynamic travel on X

```
E60003 = Xa
```
Declaration of the maximum dynamic travel on X

```
E62002 = Za
```
Declaration of the minimum dynamic travel on Z

```
N..
N..
```

REMARK *During machining, the system applies the tool corrections to remain within the limits defined. For instance, for maximum travel, the cutter axis cannot exceed dimension Xa-R.*

E6x004: Eccentricity offsets on axis x (DAT3) (x=axis address from 0 to 5)

E6x005: Offset programmed by G59 on axis x

E69000 - Scaling factor

Read/write parameter.

The units are in thousandths (see use in 4.14.14).

E69003 - Axis assignment

Axis assignment positions the part programming three-axis reference system Xp Yp Zp with respect to the physical reference system of the machine, Xm, Ym, Zm to modify the programming reference system according to the workpiece position on the machine. Axis assignment applies only to three-axis milling groups and can be forward (positive) or reverse (negative).

It should be noted that axis assignment:
- Repositions the programmed dimensions, the interpolation plane (possibly reversing the direction of rotation of circles), the PQR material vectors, the coefficients of the polynomial functions, the 2D tool correction (possibly reversing G41/G42) and the tool direction (G16...)
- Is applied after programmed geometric transformations such as angular offset (ED..), programmed offset (G59..), mirroring (G51..) and scaling factor (G74).

Axis assignment does not apply to dimensions programmed with respect to the measurement origin (G52..).
Datum shifts DAT1, DAT2 and DAT3 are not affected by axis assignment.
When programming G16.., the programmed tool direction is repositioned according to the axis assignment and, after a system reset, assumes the assignment of the programme Z.

Axis assignment declaration

Axis assignment must be declared in the AXIS ASSIGN field of the SHIFT mode. The X, Y and Z axes are declared on a single line. After the X, Y or Z programme axis, function P, Q or R addresses the corresponding machine axis (X, Y or Z) and the + or - sign indicates whether the axis assignment is forward or reverse.

Example:
ASSIGN AXIS: XR- YQ+ ZP+

XR- means: Y programme axis assigned to Z machine axis with sign reversal
YQ+ means: Y programme axis unchanged
ZP+ means: Z programme axis assigned to X machine axis without sign reversal.

In case of a format error in the command or if the group to which it is applied is not in reset state, axis assignment is refused.

If the command is accepted, the tool direction (G16..) takes the B programme axis assignment. When an axis is reassigned, its assignment is indicated on the SHIFT display page.

Axis assignment by parameter E69003

Parameter E69003 is used to read the current assignments or declare new ones.

This parameter includes three words:
- The units and tens digits contain the X axis assignment,
- The hundreds and thousands digits contain the Y axis assignment,
- The ten thousands and hundred thousands digits contain the Z axis assignment.

In each word, the value 0 addresses the X machine axis, the value 1 addresses the Y machine axis and the value 2 addresses the Z machine axis to which is added the value 80 if the assignment is reverse.

Example:
Axis assignment XR+ YQ- ZP- is declared by E69003 = 808102.

REMARK *The tool direction (G16..) in the programme reference system must be declared again after changing the axis assignments by programming parameter E69003.*

A format error in programming of E69003 returns error message 94.

Programming of turning planes G20, G21 and G22 with reassigned axes returns error message 2.

Programming of E69003 with a turning plane not disabled returns error message 95.

6.2.9.5 Group Axis Access Parameters

E7x000 to E7x006, E7x100 and E7x101

Parameters where x represents the axis number in the group (0 to 8).

For parameters E7x000 to E7x004, 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 defined 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: Memory of the reference of an axis in a group for on-the-fly dimension measurement

Read/write parameters.

These parameters are used to directly read the tool position during movement after an NC priority interrupt (PLC IT inputs) programmed by function G10 (see Sec. 4.11.5).

The measurement is stored in the memory until:
- the next interrupt is received
- a manual or programmed (M02) reset is carried out.

When function G10 is read in the part programme, these parameters can be initialised with 99999999 or written by programming to make a test when acquiring the dimension (if there is 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 the linear axis of 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 able to be 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

Anew 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.

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 an axis is modified, the parameter related to the other axis in the primary/secondary axis pair is automatically modified.

A reset reconfigures the axis pairs as specified by machine parameter P54.

Example:

E70006 = 0 Axes X and U are independent
E72006 = 1 Axes Z and W are coupled.

E7x007: Axes programmed by diameter

Read-only parameters.

Only parameters E70007 (X axis) and E73007 (U axis) are programmable by diameter.

These parameters are reset for programming by radius and set for programming by 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.

6.2.9.6 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 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 all the axis groups to access all the spindles.

E9010x: Spindle x position reference

(Same format as parameter E79000).

E9011: 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/s2 (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 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 to E912xx and E920xx

Read/write parameters.

E900xx: Axis measurement

Read on measured or servo axes.

Write is possible only for non-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 no homing for the spindle. Since homing is automatic on the spindles, this makes it possible to reset the spindle position reference.

E912xx: N/M AUTO axes

These parameters define the axes concerned by the N/M AUTO function.

Use of the list of E912xx parameters concerns only the servo-controlled axes (error message 99 is generated by an attempt to assign an axis that is not servo-controlled). Parameters E912xx cannot use the PLC axes.

The number of parameters that can be used is limited to five (error message 99 is generated if this number is exceeded).

When the N/M AUTO function is enabled, the axes concerned by this function are no longer controlled by the interpolators. However, the interpolator computation is continued in parallel to allow the path to be resumed after inhibiting of function N/M AUTO.

Only a parameter E912xx belonging to the axis group or another axis group can be written (error message 99 is generated by an attempt to write E912xx for a group different from the one of the part programme in which it is programmed).

After a reset, parameters E912xx are reset to zero. When a parameter E912xx is reset, the part programme in which it is programmed is stopped if the corresponding N/M AUTO axis is being moved by an axis job or handwheel. Execution of the programme is not resumed until the axis is stopped.

For additional information on the N/M AUTO function, refer to the Supplementary Programming Manual.

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 inhibiting of the switches.

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 a «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 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 (C not slaved)
...
...
(USE AS C AXIS)
E94224=04
(by precaution, programme:)
E91004=1 (C slaved)
E91104=0 (HOMING completed)...
...
```

E950xx to E952xx

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/s2 or deg/s2.

E972xx: Axis acceleration at high speed

Acceleration expressed in mm/s2 or deg/s2.

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 B.

```
N..
E80000 = 18000
```
Declaration of the value of the parameter

```
G00 XE80000
```
The value of E80000 is equivalent to 18 millimetres (internal unit = µm)

```
G0 BE80000
```
The value of E80000 is equivalent to 1.8 degrees (format B034, 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

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
G79 E56003 = > 10 N50

N..
```

Increment of the parameter
Condition: if E56003 ≥ 10, branch to N50, else continue

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 X0 Y0 Z0
N20 G00 X50 Y50                          Point a, no trace (stylus raised)
N30 E31001 = 1 G01 X100                  Point b, dotted line
N40 Y100                                 Point c
N50 E31001 = 0 G03 I100 J50 R50 ES+ EB20 Point d, solid line
N60 G01 EA180 X25 Y75                    Point e
N70 E31000 = 2 G00 X0 Y0                 Point f, dashed line
N80...
```

Use of the external parameters related to DAT1 (E6x000), DAT2 (E6x001) and tool radius offset (E50xxx) (E52xxx)

Execution of spot facing on paths a, b, c, a with radius correction (G41) in the XY plane (G17).

```
%25
N10 G00 G52 Z0
E60000 = - 356232 (DAT1 ON X)
E61000 = - 225536 (DAT1 ON Y)
E62000 = - 260206 (DAT1 ON Z)
E62001 = 10000 (DAT2 ON Z)
E50001 = 125000 (TOOL LENGTH)
E52001 = 9000 (TOOL RADIUS)
L1 = E52001/1000 (CUTTER RADIUS)
L2 = 25 (BORE RADIUS)
L3 = L2 + L1/2 (ENGAGEMENT AND RETRACTION RADIUS)
L4 = 5 (SPOT FACING DEPTH)
N20 T01 D01 M06 (CUTTER DIA=18)
N30 S300 M40 M03
N40 G00 X0 Y0                    Point OP, centre of spot facing
N50 Z2                           Approach on Z
N60 G01 Z-L4 F50 M08             Penetration on Z
N70 G41 X-L1 F100                Point a, engagement on diameter L3
N80 G03 XL2 Y0 RL3               Point b, engagement on diameter L2
N90 G03 XL2 Y0 I0 J0             Point c, execution of the diameter
N100 G03 X-L1 Y0 RL3             Point OP, retraction in the centre
N..
```

Use of the external parameters associated with axis servos (E910xx) and assignment of an axis to another axis (E7x005)

Machine equipped with two rotary axes, B1 and B2, used alternately for machining.

Axis B1, physical address 7, assigned in word 2 of machine parameter P9, declared modulo (P1), measured (P2) and servo (P3) (see parameter manual).

Axis B2, physical address 6, declared measured only in machine parameter P2.

```
% 10
N10
N.. X.. Y.. B..
N.. Z..                    Machining with
N..                        axis B1
N..
E91007 = 0                 Declaration of B1 as non-servo
E77005 = -1                Release programme axis B
E77005 = 6                 Assignment of B to B2
E91006 = 1                 Declaration of B2 as servo
N.. X.. Y.. B..
N.. Z..                    Machining with
N..                        axis B2
N..
N.. M02                    B1 servo, B2 non-servo and assignment
                           of B to B1
```

6

Use of the external parameters related to the machine axes (E90xxx and E93xxx)

Automatic HOME finding programme on four axes: X (axis 0), Y (axis 1), Z (axis 2), B (axis 7, modulo 360°), 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+L0	Homing status
EL6=1	Homing not performed
EL2=0 EL1=-1000 EL2=1 L5=EL1/1000	Initialisation of the measurement with -1 (or 359° for modulo axis)
G79 L5>0 N40	Branch to N40 for modulo axis
$0LINEAR AXIS	
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 N60	Movement to the measurement origin until reaching a value of zero (if ORPOM is not zero, take this into account in the comparison)
$MODULO AXIS	
N40 G79 EL4=0 N50	Branch to N50 if the axis is not on the switch
L5=L5-1 G00 G52 @X-L5 G79 N40	Back off the axis by 1 degree, sign depending on the HOME direction
N50 EL6=1 EL2=0 EL3=1 EL1=100000 EL2=1	Initialisation of the measurement
G01 G52 G93 F0.1 @XL5 G10 L5=EL1/10000+180 @L0<50 N60	Movement in V/D on measurement origin by 180° until the value > 0°

```
G79 N50

N60 G94 F1000

N100
@X=X L0=0 G77 N10 N60
@X=Y L0=1 G77 N10 N60
@X=Z L0=2 G77 N10 N60
@X=B L0=7 G77 N10 N60
N110 M02
```

If the switch is not encountered, branch to N50 for new movement by 180°

Return to programming in mm/min after HOME on the modulo axis

Definition of address equivalences:
@X: axis name, L0: axis number

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:

@ [A " " Z] = [A " " Z]

The assignment of a value to an equivalent function is programmed by:

@ [A " " Z] followed by the value

Example :

@ X = U @ 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 (@ [A " " Z] = [A " " Z]) 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, 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

Transfer of the values of L and E into the current programme

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..
```

Transfer of the values of L and E into a subroutine

```
%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..

Notes

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=L0+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 give visibility into the functions programmed. They can also 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]
[•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]	[•BM62]	[•BM63]	[•BM64]	[•BM65]	[•BM66]	[•BM67]	[•BM68]
[•BM69]	[•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 Z0 G71
N..
N40 G97 S1000 M03 M41
N50 M60 G77 H9000                          Tool check subroutine call
N..
N350 M02

%9000
VAR
[GPLANE] [MRANGE] [MDIRROT] [GINCH] [GABS] Symbolic variables (see Chapter 7)
[XRETURN] [YRETURN] [ZRETURN]
ENDV

$0 CONTEXT SAVE
N10 [GPLANE]=17*[•BG17]
    [GPLANE]=18*[•BG18]+[GPLANE]           Interpolation plane
    [GPLANE]=19*[•BG19]+[GPLANE]
N20 [MRANGE]=40*[•BM40]                    Speed range
    [MRANGE]=41*[•BM41]+[MRANGE]
N30 [MDIRROT]=03*[•BM03]
    [MDIRROT]=04*[•BM04]+[MDIRROT]         Direction of rotation
    [MDIRROT]=05*[•BM05]+[MDIRROT]
N40 [GINCH]=70*[•BG70]                     Inches
    [GINCH]=71*[•BG71]+[GINCH]
N50 [GABS]=90*[•BG90]                      Absolute
    [GABS]=91*[•BG91]+[GABS]
[XRETURN]=E70000
[YRETURN]=E71000
[ZRETURN]=E72000
N60 G90 G70 G00 G52 Z0
N70 G52 X100 Y100 M05                      Tool check position
N80 G52 G10 Z-500
N90 G52 Z0
N100 G52 Z [ZRETURN]                       Return to Z position
N110 G52 X [XRETURN]
N120 G52 Y [YRETURN]
G[GPLANE] M[MRANGE]
M[MDIRROT] 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 + 1/32 + 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

7.1	**Programme Stack**		7 - 3
	7.1.1	Use of the Stack	7 - 3
	7.1.2	Stack Allocation	7 - 3
7.2	**Saving and Restoring L Variables**		7 - 3
	7.2.1	PUSH Function	7 - 3
	7.2.2	PULL Function	7 - 4
7.3	**Symbolic Variables**		7 - 6
	7.3.1	Declaring Symbolic Variables - VAR and ENDV Functions	7 - 6
	7.3.2	Using Symbolic Variables in ISO Programmes	7 - 7
	7.3.3	Acquisition of Variables in the Stack for Another Axis Group: Function VAR H.. N.. N..	7 - 10
	7.3.4	Deleting Symbolic Variables from the Stack	7 - 12
	7.3.4.1	Automatic Deletion of Symbolic Variables	7 - 12
	7.3.4.2	Programmed Deletion of Variables: DELETE Function	7 - 12

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 in 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).

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 current main programme or 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.

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 programme can access only the symbolic variables declared in it or, if a subroutine, those from lower nesting levels.

Programming axes by variable L or parameter E defined by symbolic variable

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 unit for the value given by the symbolic variable or variable L is mm or degree if the axis is rotary.*
If XE80000 or XE[symb1] is programmed, the value given by parameter E (set or not) is defined in the internal system unit (see Sec. 2.1), by default in 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 an ellipse

```
%35
M999
VAR
[SPSPD] = 1000  [DIRN] = 3  [SPDRNG] = 40
[FDRT] = 200  [CLRNC] = 120  [DPTHZ] = 15
[MAJOR RD] = 100  [MINOR RD] =  50 [INCR] = 10
[STRT ANG] = 0  [END ANG] = 360
[COS]  [SIN]
ENDV
$ MACHINING
N10 G0 G52 Z0
N20 T1 D1 M06 (CUTTER R=5)
N30 G97  S [SPSPD]  M [DIRN]  M [SPDRNG]
N40 G0 X [CLRNC] Y0
N50 Z-[DPTHZ]
N60 G01 X [MAJOR RD]  F [FDRT]
N70 $ MACHINING AN ELLIPSE
[COS] = [MAJOR RD] * C [STRT ANG]
[SIN] = [MINOR RD] * S [STRT ANG]
X [COS]  Y [SIN]
[STRT ANG] = [STRT ANG] + [ANCR]
G79 [STRT ANG] = < [END ANG] N70 + 1
N80 X [CLRNC]
$ END OF MACHINING
N90 G0 G52 Z0 M05
N100 M02
```

Variable declaration

View of machining

7.3.3 Acquisition of Variables in the Stack for Another Axis Group: Function VAR H.. N.. N..

VAR H.. N.. N..	Acquisition of variables from the stack for another axis group.

Function VAR H.. N.. N.. is used to read the symbolic variables declared in another programme from the programme stack of the other 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 where the symbolic variables are to be fetched.
:s	Number «s» of the memory zone where programme H.. is searched (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 with the same name as the variable declared by VAR N.. N.. has already been declared in the same programme or subroutine, the previous variable is destroyed.

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
N10 ...
N..
VAR H2 N50 N60 L0=[I] L1=[T(1,3)]
N..
```

```
%2
N10
N..
VAR
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

The return to the programme deletes all the variables declared in a subroutine and releases the space they occupied in 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.

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

8.1 General			8 - 3
	8.1.1	Error Numbers	8 - 3
	8.1.2	Error Messages	8 - 3
8.2 Creating Error Messages			8 - 3

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.1	**G Function Summary Table**	A - 3
A.2	**M Function Summary Table**	A - 18
A.3	**Additional Function Summary Table**	A - 23

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±] X.. Y.. Z..

Cancellation:
G01/G02/G03.

G01*: Linear interpolation at programmed feed rate (see Sec. 4.5.1)

Syntax:
N.. [G90/G91] **G01** [R±] X.. Y.. Z.. [F..]

Cancellation:
G00/G02/G03.

G02: Clockwise circular interpolation at programmed feed rate (see Sec. 4.5.2)

Syntax (XY plane):
N.. [G17] [G90/G91] **G02** X.. Y.. I..J.. or R.. [F..]

Cancellation:
G00/G01/G03.

G03: Counterclockwise circular interpolation at programmed feed rate (see Sec. 4.5.2)

Syntax (XY plane):
N.. [G17] [G90/G91] **G03** X.. Y.. I..J.. or R.. [F..]

Cancellation:
G00/G01/G02.

A

G04: Programmable dwell (see Sec. 4.14.1)

Syntax:
N.. **G04** F..

Cancellation:
End of block.

G06: Spline curve execution command (see Sec. 4.13.2.2)

Syntax:
N.. **G06** NC..

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.. Y.. Z.. [F..]

Cancellation:
End of block.

G10: Interruptible block (see Sec. 4.11.5)

Syntax:
N.. [G40] [G04] [G00/G01/G02/G03] X.. Y.. Z.. **G10** [:n] [+X.. or F..] [@n < > Value] N.. [+ Number] [EF..]

Cancellation:
End of block.

G12: Overspeed by handwheel (see Sec. 4.14.3)

Syntax:
N.. [G01/G02/G03] **G12** X.. Y.. Z.. [F..] [$0 ...]

Cancellation:
End of block.

G16*: Definition of the tool axis orientation with addresses P, Q, R (see Sec. 4.8.2)

Syntax:
N.. **G16** P±/Q±/R±

Cancellation:
G16 P±/Q±/R±.

G17*: XY plane selection (see Sec. 4.2)

Syntax:
N.. **G17**

Cancellation:
G18/G19.

G18: ZX plane selection (see Sec. 4.2)

Syntax:
N.. **G18**

Cancellation:
G17/G19.

G19: YZ plane selection (see Sec. 4.2)

Syntax:
N.. **G19**

Cancellation:
G17/G18.

G23: Circular interpolation defined by three points (see Sec. 4.5.4)

Syntax: (XY plane)
N.. [G17] [G90/G91] **G23** X.. Y.. I.. J.. [F..]

Cancellation:
End of block.

A

G29: 3D tool correction with 3 or 5 axes (see Sec. 4.8.5.1)

Syntax:
N.. [D..] [G01] **G29** X.. Y.. Z.. P.. Q.. R.. [I.. J.. K..] [A.. /B.. /C..]

Cancellation:
G40/G41/G42.

G31: Thread chasing cycle (see Sec. 4.9.12)

Syntax (XY plane):
N.. [G17] [M03/M04] [S..] **G31** [X.. Y..] Z.. [ER..] [EH..] K.. P.. [F..] [EF..] [EC..]

Cancellation:
G80 to G89.

G40*: Cutter compensation cancel (see Sec. 4.8.4)

Syntax:
N.. [G00/G01] **G40** X.. Y.. Z..

Cancellation:
G29/G41/G42.

G41: Cutter compensation, left (see Sec. 4.8.4)

Syntax (XY plane):
N.. [G17] [D..] [G00/G01/G02/G03] **G41** X.. Y..

Cancellation:
G29/G40/G42.

G42: Cutter compensation, right (see Sec. 4.8.4)

Syntax (XY plane):
N.. [G17] [D..] [G00/G01/G02/G03] **G42** X.. Y..

Cancellation:
G29/G40/G41.

G43: 3D tool correction with cylindrical tool (see Sec. 4.8.5.2)

Syntax:
N.. [D..] [G01] **G43** X.. Y.. Z.. P.. Q.. R..[I.. J.. K..] [A.. / B.. / C..]

Cancellation:
G40.

G45: Simple pocket cycle (see Sec. 4.10.1)

Syntax:
N.. **G45** X.. Y.. Z.. [ER..] EX.. EY.. [EB..] P.. Q.. [I..] [J..] [EG2/EG3] EP.. EQ.. EI.. EJ..

Cancellation:
End of block.

G46 NU..: Pocket and facing cycles with any contours (see Sec. 4.10.2)

The syntax specific to each cycle is described below (syntax defined in the XY plane).

Cancellation of G46:
End of block.

G46 NU0: Geometric definition header block (see Sec. 4.10.2.5)

N.. **G46 NU0** NP.. ED.. Q.. [J..] [NR±] [R03/R04] [LX.. LY..] [EX.. EY..]

G46 NU1: Pocket introduction block (see Sec. 4.10.2.6)

N.. **G46 NU1** [LX.. LY..]
N.. Pocket contour definition

G46 NU2: Island introduction block (see Sec. 4.10.2.6)

N.. **G46 NU2** [LX.. LY..]
N.. Island contour definition

G46 NU3: Facing introduction block (see Sec. 4.10.2.7)

N.. **G46 NU3**
N.. Facing contour definition

A

G46 NU4: Cavity introduction block (see Sec. 4.10.2.7)

N.. **G46 NU4**
N.. Cavity contour definition

G46 NU5 and G46 NU6: Facing and wall introduction blocks (see Sec. 4.10.2.8)

N.. **G46 NU5**
N.. Facing contour definition

N.. **G46 NU6**
N.. Wall contour definition

G46 NU9: Geometric definition end block (see Sec. 4.10.2.9)

N.. **G46 NU9**

G46 NU10: Initial drilling orders (see Sec. 4.10.2.10)

N.. **G46 NU10** NP.. G81 Z.. [ER..] [F..]

N.. **G46 NU10** NP.. G83 Z.. P.. [Q..] [ER..] [EF..] [F..]

N.. **G46 NU10** NP.. G87 Z.. P.. [Q..] [ER..] [EF..] [F..]

G46 NU15: Roughing order (see Sec. 4.10.2.11)

N.. **G46 NU15** NP.. Z.. P.. [ER..] [EH..] [EP..] [EQ..]

G46 NU20: Finishing and semi-finishing orders (see Sec. 4.10.2.12)

N.. **G46 NU20** NP.. Z.. P.. [ER..] [EH..] [EI..] [EJ..] [J..]

G48: Spline curve interpolation (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. 4.14.15)

Syntax:
N.. **G51** X- Y- Z-

Cancellation:
G51 X- Y- Z- cancelled by G51 X+ Y+ Z+

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.. Y.. Z.. A.. B.. 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.

A

G59: Programme origin offset (see Sec. 4.12.4)

Syntax:
N.. [G90/G91] **G59** X.. Y.. Z.. U.. V.. W.. A.. B.. C.. [I.. J.. K..]

Cancellation:
Cancelled by different G59 X.. Y.. Z..

G70: Inch data input (see Sec. 4.14.4)

Syntax:
N.. **G70**

Cancellation:
G71.

G71*: Metric data input (see Sec. 4.14.4)

Syntax:
N.. **G71**

Cancellation:
G70.

G73*: Scaling factor cancel (see Sec. 4.14.14)

Syntax:
N.. [G40] **G73**

Cancellation:
G74.

G74: Scaling factor enable (see Sec. 4.14.14)

Syntax:
N.. [G40] **G74**

Cancellation:
G73.

G75: Emergency retraction subroutine declaration (see Sec. 4.11.7) **Syntax:** N.. **G75** N.. **Cancellation:** Cancelled by G75 N0 or different G75 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:** 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+** [H..] N.. [+number] ISO block
G76-: ISO block deletion (see Sec. 4.11.12.5) **Syntax:** N.. **G76-** [H..] N.. [+number]

A

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. 4.15.6)

Syntax:
N.. **G78** Q../Pj.i

Cancellation:
Cancelled by G78 Q0 or by different G78 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 (XY plane)
N.. [G00/G01/G02/G03] X.. Y.. Z.. **G79 +/-** X.. / F..

Cancellation:
End of block.

G80*: Machining cycle cancel (see Sec. 4.9.2)

Syntax:
N**.. G80**

Cancellation:
G31/G81-G89.

G81: Drilling/centre drilling cycle (see Sec. 4.9.3)

Syntax (XY plane):
N.. [G17] **G81** [X.. Y..] Z.. [ER..] [EH..] [F..]

Cancellation:
G31/G80/G82-G89.

G82: Counterboring cycle (see Sec. 4.9.4)

Syntax (XY plane):
N.. [G17] **G82** [X.. Y..] Z.. [ER..] [EH..] EF.. [F..]

Cancellation:
G31/G80/G81/G83-G89.

G83: Peck drilling cycle (see Sec. 4.9.5)

Syntax (XY plane):
N.. [G17] **G83** [X.. Y..] Z.. [ER..] [EH..] [P..]/[ES..] [Q..] [EP..] [F..] [EF..]

Cancellation:
G31/G80-G82, G84-G89.

G84: Tapping cycle (see Sec. 4.9.6.1)

Syntax (XY plane):
N.. [G17] **G84** [X.. Y..] Z.. [ER..] [EH..] EF.. [F..]

Cancellation:
G31/G80-G83, G85-G89.

A

G84: Rigid tapping cycle (see Sec. 4.9.6.2)

Syntax (XY Plane):
N.. [G17] [M03/M04] [M04-M05] **G84** [X.. Y..] Z.. [ER..] [EH..] K.. [EK..]

Cancellation
G31/G80-G83, G85-G89

G85: Boring cycle (see Sec. 4.9.7)

Syntax (XY plane):
N.. [G17] **G85** [X.. Y..] Z.. [ER..] [EH..] [F..] [EF..]

Cancellation:
G31/G80/G81-G84, G86-G89.

G86: Boring cycle with indexed stop and clearance at hole bottom (see Sec. 4.9.8)

Syntax (XY plane):
N.. [G17] **G86** [X.. Y..] Z.. [ER..] [EH..] [EC..] [EA..] [EP..] [F..]

Cancellation:
G31/G80-G85, G87-G89.

G87: Drilling cycle with chip breaking (see Sec. 4.9.9)

Syntax (XY plane):
N.. [G17] **G87** [X.. Y..] Z.. [ER..] [EH..] [P..]/[ES..] [Q..] [EP..] [EF..] [F..]

Cancellation:
G31/G80-G86, G88/G89.

G88: Boring and facing cycle (see Sec. 4.9.10)

Syntax (XY plane):
N.. [G17] **G88** [X.. Y..] Z.. [ER..] [EH..] [F..]

Cancellation:
G31/G80-G87/G89.

G89: Boring cycle with dwell at the bottom of the hole (see Sec. 4.9.11)

Syntax (XY plane):
N.. [G17] **G89** [X.. Y..] Z.. [ER..] [EH..] [EF..] [F..]

Cancellation:
G31/G80-G88.

G90*: Programming in absolute dimensions with respect to the programme origin (see Sec. 4.1.1)

Syntax:
N.. **G90** X.. Y.. Z.. A.. B.. 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.. Y.. Z.. A.. B.. C..

Cancellation:
G90.

G92 X.. Y.. Z..: Programme origin preset (see Sec. 4.12.3)

Syntax:
N.. **G92** X.. Y.. Z..

Cancellation:
End of block.

G92 R..: Programming of the tangential feed rate (see Sec. 4.7.4)

Syntax:
N.. **G92** R..

Cancellation:
Cancelled by G92 R0 or by different G92 R..

A

G93: Feed rate expressed as inverse time (V/L) (see Sec. 4.7.2)

Syntax:
N.. **G93** F.. G01 X.. Y.. Z.. A.. B.. C..

Cancellation:
G94/G95.

G94*: Feed rate expressed in millimetres, inches or degrees per minute (see Sec. 4.7.1)

Syntax:
N.. **G94** F.. G01/G02/G03 X.. Y.. Z.. A.. B.. C..

Cancellation:
G93/G95.

G95: Feed rate expressed in millimetres or inches per revolution (see Sec. 4.7.3)

Syntax:
N.. **G95 F**.. G01/G02/G03 X.. Y.. Z..

Cancellation:
G93/G94.

G97*: Spindle speed expressed in revolutions per minute (see Sec. 4.3.2)

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

Cancellation:
G96 (combined machine).

G104: 3D curve smoothing (see Sec. 4.14.18)

General syntax:
N.. X. . Y.. Z.. (first point on the curve)
N.. [G01] **G104** X..Y.. Z.. [F..]
[Intermediate points on the curve]
N..G80 X.. Y.. Z.. (last point on the curve)

Cancellation:
G80.

G997: Enabling and execution of all functions stored in state G999 (see Sec. 4.14.17) **Syntax:** N.. **G997** **Cancellation:** G998/G999.
G998: Enabling of execution of the blocks and part of the functions processed in state G999 (see Sec. 4.14.17) **Syntax:** N.. **G998** **Cancellation:** G997/G999.
G999: Suspension of execution and forcing of block concatenation (see Sec. 4.14.17) **Syntax:** N.. **G999** **Cancellation:** G997/G998.

A

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.

Several decoded M functions can be programmed in the same block, e.g. N.. S100 M03 M40 M08.

M00: Programme stop (see Sec. 4.14.7)

Syntax:
N.. [G40] **M00** [$0 ...]

Cancellation:
Action on the machine panel CYCLE key.

M01: Optional stop (see Sec. 4.14.8)

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.. [D..] **M06** [$0.. / (...)]

Cancellation:
M function report (CRM).

M07: Coolant 2 on (see Sec. 4.14.6)

Syntax:
N.. **M07**

Cancellation:
M09.

M08: Coolant 1 on (see Sec. 4.14.6)

Syntax:
N.. **M08**

Cancellation:
M09.

A

M09*: Coolant off (see Sec. 4.14.6)

Syntax:
N.. **M09**

Cancellation:
M07/M08.

M10: Clamp (see Sec. 4.14.5)

Syntax:
N.. [G00/G01/ G02/ G03] **M10** X.. Y.. Z.. A.. B.. C..

Cancellation:
M11.

M11: Unclamp (see Sec. 4.14.5)

Syntax:
N.. M11

Cancellation:
M10.

M12: Programmed feed stop (see Sec. 4.14.2)

Syntax:
N.. **M12** [$0...]

Cancellation:
Press CYCLE on the machine panel.

M19: Spindle index (see Sec. 4.3.4)

Syntax:
N.. [S..] [M03/M04] EC±.. **M19**

Cancellation:
M03/M04/M05.

M40-M45: Spindle speed ranges (see Sec. 4.3.3)

Syntax:
N.. [S..] [M03/M04] **M40 to M45**

Cancellation:
Cancel one another.

M48*: Enable overrides (see Sec. 4.4.11)

Syntax:
N.. **M48**

Cancellation:
M49.

M49: Disable overrides (see Sec. 4.14.11)

Syntax:
N.. **M49**

Cancellation:
M48.

M61: Release the current spindle in the axis group (see Sec. 4.15.5)

Syntax:
N.. **M61**

Cancellation:
M62-M65.

M64*, M65, M62, M63: Spindle 1 to 4 control (see Sec. 4.3.5)

Syntax:
N.. [S..] **M62**/**M63**/**M64**/**M65** [M40-M45] M03/M04

Cancellation:
Cancel one another.

A

M66*, M67, M68, M69: Spindle 1 to 4 measurement (see Sec. 4.3.6)

Syntax:
N.. [S..] **M66/M67/M68/M69**

Cancellation:
Cancel one another.

M997: Forced block sequencing (see Sec. 4.14.10)

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. 4.14.9)

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. 4.14.9)

Syntax:
N.. **M999**

Cancellation:
M997/M998/M02.

A.3 Additional Function Summary Table

ED: Programmed angular offset (see Sec. 4.12.5)

Syntax:
N.. **ED**..

Cancellation:
ED0 or different ED..

EG: Acceleration modulation (see Sec. 4.14.13)

Syntax:
N.. **EG**..

Cancellation:
Different EG..

EM+/-: Outside dimensions of the part in 3D graphic display (see Sec. 4.14.16)

Syntax:
N.. **EM+** X.. Y.. Z.. **EM-** X.. Y.. Z..

F: Feed rate, dwell, number of thread starts

Syntax:
N.. G93 **F..** (Feed rate in V/L, see Sec. 4.7.2)
N.. G94 **F..** (Feed rate in mm/min, degrees/min, inches/min see Sec. 4.7.1)
N.. G95 **F..** (Feed rate in mm/rev, inches/rev see Sec. 4.7.3)
N.. G04 **F..** (Dwell in seconds, see Sec. 4.14.1)
N.. G31 **F..** (Number of thread starts, see Sec. 4.9.12)

Cancellation:
For G93, G94, G95: different F.. In G04, G31: end of block.

S: Spindle rpm, number of subroutine repetitions

Syntax:
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)

Cancellation:
S0 or different S..

A

T: Tool number (see Sec. 4.8.1) **Syntax**: N.. **T..** M06 (Tool change) **Cancellation**: T0 or different T..
D: 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.1 Parameters in the PLC Memory B - 3

B.2 Parameters in the NC Memory B - 3

B

Parameters E are accessible for read only or for read/write by the part programme.

The values or units 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 turning are included in the tables and are identified by the word «Turning».

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 or 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	6.2.0.2	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 by diameter	0 or 1	Read/write (+ graphic)
E11006	Turning	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	Angle crossing management enabled	0 or 1	Read/write (+ graphic)
E11016	Turning	Front or rear turret	0 or 1	Read/write
E11017	6.2.9.2	Inclined plane function enabled	0 or 1	Read only
E11018	6.2.9.2	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
E32003	6.2.9.2	Angle crossing speed analysis angle	1/ 10000 degree	Read/write

Parameters	See Sec.	Description	Value or units	Access by the part programme
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 9	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 the 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 length	IU	Read/write
E51000	6.2.9.4	Current tool orientation	0 to 2 or 100 to 102	Read only

Parameters	See Sec.	Description	Value or units	Access by the part programme
E51001 to E51255	6.2.9.4	Cutter tip radius	IU	Read/write
E52001 à E52255	6.2.9.4	Tool radius	IU	Read/write
E53001 to E53255	6.2.9.4	Dynamic tool length correction	IU	Read/write
E54001 to E54255	6.2.9.4	Dynamic tool radius correction	IU	Read/write
E55001 to E55255	Turning	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)	6.2.9.4	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	Turning	Grinder axis inclination	1/10000 degree	Read/write (+ graphic)
E69002	Turning	Inclined wheel on grinder	1/10000 degree	Read/write (+ graphic)

Parameters	See Sec.	Description	Value or units	Access by the part programme
E69003	6.2.9.4	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 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	6.2.9.5	Axes programmed by diameter	0 or 1	Read/write
E7x100 x = axis number (0-8)	6.2.9.5	Position reference from the interpolator	IU	Read/write
E7x101 x = axis number (0-8)	6.2.9.5	Interpolation speed limitation	0 or 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 80050	6.2.9.7	Local data	- 99999999 to 99999999	Read/write

B

Parameters	See Sec.	Description	Value or units	Access by the part programme
E81000 to 81999	6.2.9.7	Master axis reference positions	IU	Read/write
E82000 to 82999	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	rpm	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	rpm	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	6.2.9.8	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 to 4	Read only
E94000 to E94031	6.2.9.8	Assignment of a master axis to a with a master axis (or spindle)	-1 to 31	Read/write
E94100 to E94131	6.2.9.8	Association of a slave axis (or spindle) master axis	0 to 4	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

Parameters	See Sec.	Description	Value or units	Access by the part programme
E97100 to E97131	6.2.9.8	Axis acceleration at the work rate	mm/s^2 or deg/s^2	Read only
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 µ (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 B axis, the format is B+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 (.. plane)
Y+053	Movement on the Y axis. In a cycle, end point on the machining axis (.. plane)
Z+053	Movement on the Z axis. In a cycle, end point on the machining axis (.. plane)
I+053 J+053 K+053 I053 J053 J053 K053	For circular/helical interpolation (G02, G03), centre of the circle and pitch of the helix With programmed origin offset (G59), centre of rotation and angular offset (ED) For thread chasing cycle (G31), thread pitch = I, J or K depending on the interpolation plane For pocket cycle (G45), axial finishing pass For pocket cycle (G45), lateral finishing pass In cycle G46 NU0/NU20, lateral finishing or semi-finishing allowance For leadscrew tapping cycle (G84), tap 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	Movement on the C axis
E+/E-	For profile geometry programming (PGP), discriminant
EA+033 EA+033 EA+033	For profile geometry programming (PGP), line angle In a boring cycle with indexed spindle stop (G86), angle between EC.. and tool cutting edge In polar programming, angle of the line
EB+053 EB-053 EB053	For contour definition and PGP, radius or fillet between two interpolations For contour definition and PGP, chamfer between two linear interpolations For profile Geometry Programming (PGP), chamfer between 2 lines
EC+033 EC+033 EC+033	For spindle indexing (M19), indexing angle For thread chasing cycle (G31), tool orientation in the hole bottom For boring cycles (G86), indexing position
ED+034 ED053	Programmed angular offset In a G46 NU0 cycle, roughing cutter diameter
EF022 EF022 EF.. EF.. EF..	For cycles (G82, G84, G87, G89 or G31), dwell time In a G46 NU15 cycle, dwell at the end of infeed (initial drilling orders G83 or G87) In a reaming cycle (G85), retraction rate Feed rate specific to fillets (EB+) or chamfers (EB-) Maximum feed rate after an interrupt (G10)
EG01	Pocket cutting direction (EG02: clockwise; EG03: counterclockwise)
EG03	For interpolation, acceleration modulation
EH+053 EH+053	In (G8x) cycles, dimension of the impact plane on the machining axis In a G46 NU15/NU20 cycle, dimension of the material impact plane on the machining axis
EI+053 EI.. EI..	In polar programming, length of the line (start/circle centre) In a pocket cycle (G45), axial finishing feed rate In a G46 NU20 cycle, axial finishing feed rate
EJ.. EJ..	In a pocket cycle (G45), lateral finishing feed rate In a G46 NU20 cycle, lateral finishing feed rate
EK01	For rigid tapping cycle (G84), retraction/penetration speed ratio
ES+/ES-	For profile geometry programming (PGP), secant element
ET+/ET-	For profile geometry programming (PGP), tangent element

Format	Description
EM-/EM+	For 3D graphic display, outside dimensions of the part
EP053 EP053 EP053 EP.. EP..	In a boring cycle with indexed spindle stop (G86), lateral backoff in hole bottom 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 pocket cycle (G45), axial roughing feed rate In a G46 NU15 cycle, axial roughing feed rate
EQ053 EP..	For pocket cycle (G45), lateral roughing feed rate In a G46 NU15 cycle, lateral roughing feed rate
ER+053 ER+053	In cycles, infeed or retraction dimension on the machining axis In a G46 NU0/NU10/NU15/NU20 cycle, retraction plane on the tool axis
ES02	In cycles G83 and G87, number of infeeds at constant value
EX+053 EX+053 EX053	In polar programming, angle of the line (start/end) In a G46 NU0 cycle, roughing contour end coordinate For pocket cycle (G45), dimension along X (or U)
EY+053 EY053	In a G46 NU0 cycle, roughing contour end coordinate For pocket cycle (G45), dimension along Y (or V)
EZ+053 EZ053	In a G46 NU0 cycle, roughing contour end coordinate For pocket cycle (G45), dimension along Z (or W)
LX+053	In cycle G46 NU0/NU1/NU2, contour start coordinate
LY+053	In cycle G46 NU0/NU1/NU2, contour start coordinate
LZ+053	In cycle G46 NU0/NU1/NU2, contour start coordinate
NP05	In cycle G46 NU0/NU10/NU15/NU20, pocket (or facing) number
NR	In cycle G46 NU0, work type (NR+: swallowing, NR-: against feed direction)
NU02	In cycle G46, type of definition block or machining order (mandatory after G46)
P+053 P+043 P053 P053 P053 P043 P041 P+ / P-	For thread chasing cycles (G31), withdrawal of tool For 3D correction (G29), X component of the normal vector For cycle (G83 or G87), value of the first penetration In a G46 NU10 cycle, first infeed (initial drilling orders G83 or G87) In a G46 NU15/NU20 cycle, axial pass when machining by consecutive passes For pocket cycles (G45), axial roughing path With axis group synchronisation (G78), wait for a marker Tool axis orientation (G16), along X (or U)

Format	Description
Q+053	For cycles (G83 or G87), value of the last penetration
Q+043	For correction in space (G29), Y component of the normal vector
Q053	In a G46 NU0 cycle, lateral roughing pass
Q053	In a G46 NU10 cycle, last infeed (initial drilling orders G83 or G87)
Q043	For pocket cycle (G45), lateral roughing pass
Q04	With axis group synchronisation (G78), declaration of a marker
Q+ / Q-	Tool axis orientation (G16), along Y or V
R01	In a G46 NU0 cycle, tool rotation direction (R3 or R4)
R+053	For circular/helical interpolation (G02, G03), radius of the circle
R053	For tangential feed rate programming (G92), value of the curve radius
R+043	For space correction (G29), Z component of the normal vector
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 V/L (G93)
F..	Feed rate in inches/revolution (G95)
F022	Dwell in seconds (G04)
F01	For thread chasing cycles (G31), number of threads (1 to 9)
M02	Miscellaneous functions 0 to 99
M03	Miscellaneous functions 100 to 899
NC04	For spline curve interpolation (with G06, G48, G49), curve number (1 to 9999)
S05 or S032	Spindle rotation speed in rpm (G97)
S02	In a subroutine branch (G77), number of subroutine iterations (1 to 99)
T08	Tool number (0 to 99999999)
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.1 Miscellaneous Errors and Machine Errors		D - 3
D.2 Parametric Programming Errors		D - 5
D.3 Profile Geometry Programming (PGP) Errors		D - 6
D.3.1	The end point is not determined or cannot be computed from the elements in the blocks	D - 6
D.3.2	The point of tangency or intersection cannot be computed from the data in two blocks	D - 6
D.3.3	The points of tangency or intersection cannot be computed from the data in three blocks	D - 6
D.3.4	Fillet or chamfer definition errors	D - 7
D.3.5	Miscellaneous errors in PGP	D - 7
D.4 Miscellaneous Errors		D - 7
D.5 Request for Movements Outside the Machine Travel Limits		D - 8
D.6 Structured Programming Errors		D - 8
D.7 Axis Errors		D - 8
D.8 Errors in Pocket Cycles		D - 9
D.9 Axes Not Identified on the Bus		D - 10
D.10 Dynamic Operators in C		D - 10
D.11 Spline Curve Interpolation Errors		D - 10
D.12 Errors in Numaform		D - 11
D.13 Cycle Programming Errors		D - 12

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 byanother 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.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.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

! 6 - 5, 6 - 21
$ 6 - 63
$0 4 - 314
$1 to $6 4 - 316
% 2 - 9
& 6 - 5, 6 - 21
* 6 - 4, 6 - 21
+ 6 - 4, 6 - 21
- 6 - 4, 6 - 21
/ 4 - 275, 6 - 4, 6 - 21
< 6 - 5, 6 - 22
< = 6 - 5, 6 - 22
< > 6 - 5, 6 - 22
= 6 - 5, 6 - 22
> 6 - 5, 6 - 22
> = 6 - 5, 6 - 22
@ 1 - 14
[•BGxx] 6 - 65
[•BMxx] 6 - 65
[•Rxx] 6 - 66

3D curve 4-292
3D curve smoothing 4-292
3D graphic display 4-287
3D tool correction
    3-axis or 5-axis 4-99
    with toroid or spherical tool 4-99
    with cylindrical tool 4- 107

A

A 6-4, 6-21
Absolute encoder 6-45
Acceleration
    of spindle x with indexing 6-41
    override 4-277
Access to Profil 5-24
Accurate stop at the end of a block 4-60
Acquisition of variables from the stack of another axis group 7-10
Addition 6-4, 6-21
Address equivalences 6-58
Addresses
    characterising PGP 5-5
    with assigned values 5-5
    without assigned values 5-7
Addressing
    of 8I8O analogue card analogue inputs 6-30
    of 8I8O analogue card analogue outputs 6-30
    of a value 6-66
    of G functions 6-65
    of M functions 6-65
    of the deceleration function on several blocks 6-25
    of the PLC input terminals 6-30
    of the PLC output terminals 6-30
Amplitude of the antistick pulse on reversal 6-49
Angle crossing speed analysis angle 6-26
Angle element of a line 5-5
Angular offset 4-241, 6-24
Arc tangent 6-4, 6-21
Arguments 2-19
    programmed alone 2-20
Arithmetic functions 6-4, 6-21
Arithmetic operations 6-4, 6-21
Assigning
    a slave axis to a master axis 6-46
    a variable to a CNC function 6-6
    an external parameter to a function 6-50
    axes 6-35
    tool dimensions to axes 4-310
    variables 6-3
Associating a slave axis with a master axis 6-46
Automatic deletion of symbolic variables 7-12
Automatic homing 4-219, 4-296
Automatic homing subroutine branch 4-219, 4-296
Available parameters 6-32
Axes 1-5
    programmed by diameter 6-38
Axis
    acceleration anticipation time constant 6-48
    acceleration at high speed 6-48
    acceleration at the work rate 6-48
    clamp 4-264
    clamp enable status 6-44
    coupling 6-36
    declaration 4-308
    declared rotary modulo 360 degrees 6-45
    duplicated in “automatic” modes 6-47
    duplicated in manual mode (Jog) 6-47
    group position servo-loop time constant 6-29
    group synchronisation 4-300
    measurement 6-43
    measurement correction (or spindle) 6-47
    orientation 1-5, 4-309
    origin switch 6-44
    origin switch status 6-44
    programming 3-1
    reference memory 6-36
    reference offset 6-47
    retraction in the plane 4-141
    servo-control coefficient 6-48
    servo or non-servo status 6-43
    unclamp 4-264
Axis (or spindle) in-position window 6-45
Axis/spindle switching 6-46

B

Backoff between two infeeds 4-141
Basic cycles 4-109
Block
- deletion by G76- 4-228
- execution enable 4-289
- format 2-7
- insertion by G76+ 4-226
- processing 4-289
- sequencing without stopping movement 4-212
- skip 4-275

Blocks 2-7
Boolean values 6-65
Boring 4-126, 4-141
- and facing 4-133, 4-141
- and facing cycle 4-133
- cycle 4-126
- cycle with dwell in hole bottom 4-135
- cycle with indexed spindle stop in hole bottom 4-128
- with dwell in hole bottom 4-135
- with indexed spindle stop in hole bottom 4-128

BS 2-10

C

C 6-4, 6-21
Call to subroutine return block 4-222
Calling
- a contour by function G77 5-25
- a sequence of blocks with return 4-193

Cancel
- a canned cycle 4-112
- radius offset 4-86

Cancellation
- of EDIT mode 4-271
- of MDI mode 4-271

Canned cycles in the tool axis 4-109
Cavity 4-166
Cavity introduction block 4-166
Centre drilling 4-113, 4-141
Centre drilling cycle 4-113
Centre of rotation 1-12
Chamfer
- between two intersecting lines 5-6
- between two linear interpolations 4-59

Change of direction 6-25
Characteristics of ISO and EIA codes 2-13
Characters
- used in EIA code 2-16
- used in ISO code 2-15

Circular interpolation 4-31
- defined by three points 4-45
- plane selection and radius correction 4-10

Classification
- of miscellaneous M functions 2-21
- of preparatory G functions 2-18C

Clearance after peck 4-141
Clockwise circular interpolation 4-31
Clockwise spindle rotation 4-12
CNC part programme 2-3
CNC processor 1-7
Combined encoder 6-45
Common CNC parameters 6-42
Common turning-milling functions 4-313
Comparison for conditional branch 6-69
Comparison symbols for use with external parameters 6-22
Comparison symbols for use with variables 6-5
Concatenating blocks 4-289
Contour by Profil 5-25
Control of direction of rotation 4-12
Control of spindles 1 to 4 4-19
Conversion of the internal unit 6-5, 6-22
Coolant 4-266
- 1 4-266
- 2 4-266

Coolant off 4-266
Coordinate system 1-5
Coordinates
- of the centre of a circle 5-5
- of the end point of a circle 5-5
- of the end point of a line 5-5

Corrected length 1-15
Corrected radius 1-15
Correction of slave axes with respect to master axes 6-42
Cosine 6-4, 6-21
Counter boring 4-115, 4-141
Counter boring cycle 4-115
Counterclockwise circular interpolation 4-31
Counterclockwise spindle rotation 4-12
Creating error messages 8-3
Creation of a programme 2-3
Creation of a programme by G76+ 4-224
Current axis group number 6-28
Current block data access symbols 6-62
Current mode 6-27
Current mode number 6-28
Current speed on path programmed in block 6-31
Current spindle release in axis group 4-299
Current tool axis orientation 6-31
Current tool correction number 6-31
Cutter tip radius 1-14
Cutter tip radius “@” of correction xxx 6-32
Cycles 4-141
- G81 to G89 4-141
- with any contours 4-178

D

D.. 4-81
Dashed line 6-25
DAT1 1-9, 1-12
DAT1 on X axis 6-32
DAT2 1-9
DAT2 on the X axis 6-32

DAT3 1-12, 4-245
Datum shift DAT1 and DAT2 enable 4-232
Datum shift DAT1 and DAT2 inhibit 4-232
Deceleration function on several blocks 6-25
Declaration of a variable in the programme 6-7
Decoded M functions 2-22
Definition
of a circle by its arc angle 4-52
of a spline curve 4-247
of an entity 5-4
of geometric elements 5-3
of offsets 1-9
of origins 1-7
of the addresses characterising the PGP 5-5
of tool dimensions 1-14
of travels 1-7
DELETE 7-12
DELETE function 7-12
Deletion of a spline curve 4-240, 4-248
Deletion of symbolic variables in the stack 7-12
Different from 6-5, 6-22
Direction of axis movement during interpolation 6-36
Discriminant 5-7
Displaying
a message with wait for an answer 6-61
messages with a parametric value 6-63
Distance remaining to be travelled in the current block 6-31
Division 6-4, 6-21
DL 1-15
Dotted line 6-26
Double twist head 4-103
DR 1-15
Drawing of a complete circle 6-25
Drilling cycle with chip breaking 4-130
Drilling with chip breaking 4-130, 4-141, 4-171
Drilling with chip breaking order 4-171
DRIPFEED mode 4-221
Dwell 4-141, 4-257
Dwell on each infeed 4-141
Dynamic correction length L correction 6-32
Dynamic correction xxx radius R correction 6-32
Dynamic tool correction definition 1-15
Dynamic tool length correction 1-15
Dynamic tool radius correction 1-15

E

E+/E- 5-7
EA 5-5
EB+ 4-58
EB+.. 5-6
EB- 4-59
EB-.. 5-6
ED.. 4-241
EDIT mode 4-271
Editing programmes 4-306
EF 4-74
EG.. 4-277
EIA codes and standards 2-3
EIA programme structure 2-10
EM-/+ 4-287
Emergency retract subroutine 4-215
Emergency retraction 4-215
Emergency retraction on a PLC axis group 4-305
Enabling and execution of functions stored in state G999 4-289
Encoded M functions 2-22
Encoder 6-45
End
dimensions of the part in 3D 4-287
of geometric definition 4-170
of hole 4-141
of programme 2-9, 2-10
of programme load 2-9
of retraction 4-141
of symbolic variable declaration 7-6
ENDV 7-6
ENVD functions 7-6
EOR 2-10
Equal to 6-5, 6-22
Equivalence of variables L900 to L925 6-10
Error messages 4-312, 8-1, 8-3
Error numbers 8-3
ES 5-7
ET 5-7, 6-5, 6-21
Examples of use of external parameters E 6-53
Exchanging
axes between groups 4-306
spindles between groups 4-307
External parameter declaration in the programme 6-51
External parameter E programming syntax 6-50
External parameters 4-295
External parameters E 6-20

F

Facing 4-166, 4-168
and cavity introduction blocks 4-166
and wall introduction blocks 4-166, 4-168
cycles with any contours 4-155
introduction block 4-166, 4-168
Feed rate
in degrees per minute 4-62
in inches per minute 4-62
in inches per revolution 4-70
in inverse time (V/D) 4-66
in millimetres per minute 4-62
in millimetres per revolution 4-70
limiting after interrupt 4-212
specific to chamfers EB- 4-74
specific to fillets EB+ 4-74
Fillet
between two interpolations 4-58
between two intersecting elements 5-6
Finishing 4-173
Finishing orders 4-175
Following error cancellation 6-25

Forced block sequencing 4-273
Functions 0 to 255 6-28

G

G function processing 4-289
G functions
incompatible with the programme state 2-18
with associated arguments 2-19
General diagrams of parametric programming 6-68
General information on modes 1-3
General information on the system 1-3
General programme structure 2-9
Geometric contour definition blocks 4-157
Geometric definition
end block 4-170
header 4-161
header block 4-161
Geometric elements 5-3
Geometric transformations
with 3-axis correction 4-104
with 5-axis correction 4-106
Gradual acceleration 6-25
Graphic machining simulation status 6-29
Greater than 6-5, 6-22
Greater than or equal to 6-5, 6-22
Group axis access parameters 6-36
Group axis position reference 6-36

H

Helical interpolation 4-39
High precision contouring 6-27
Homing 1-7
direction 6-45
status on an axis 6-43
status without switch wiring 6-45

I

I.. J.. 5-5
Image of system job reference 6-29
Inch data input 4-255
Inclined plane function 6-25
Incremental encoder 6-45
Infeed 4-141
Initial drilling 4-171
Initial drilling orders 4-171
Instruction 1-3, 2-4
Interactive programming on mixed machines 4-304
Internal measurement unit 3-3
Interpolation speed limiting 6-39
Interpolator position references 6-39
Interruptible block 4-208
Island 4-163, 4-164
Island introduction block 4-164
ISO and EIA codes 2-13
ISO codes and standards 2-3
ISO function programming notes 4-312
ISO functions 2-12
ISO programme
structure 2-9
tape 2-15
tape format 2-14
ISO programming 4-1

J

Job reference image 6-29
Jump to a sequence without return 4-203

L

L 1-14
L + DL 1-15
Left radius correction 4-85
Less than 6-5, 6-22
Less than or equal to 6-5, 6-22
Limit switches 1-7
Linear interpolation 4-26
at high speed 4-23
at programmed feed rate 4-26
Line type for G00 in graphic mode 6-26
Line type for G01, G02 and G03 in graphic mode 6-26
Linear X, Y and Z axes 1-5
List of
characters used in EIA code 2-16
characters used in ISO code 2-15
external parameters E 6-23
variables L 6-3
Load of a parametric expression 6-68
Logical operations 6-5, 6-21

M

M functions 2-21
M post-functions 2-21
M pre-functions 2-21
Machine 1-5
automatic control function exchange parameters 6-23
axes 4-208
description 1-6
origin (Om) 1-7
status access parameters 6-28
Machining orders 4-159
Machining parameters 6-31
Main programme 2-11
Mandatory arguments 2-19
Master axis position reference 6-42
Maximum authorised speed at angle crossings 6-48
Maximum axis speed 6-48
Maximum dynamic machine travel on the X axis 6-33
Maximum dynamic machine travel on the Y axis 6-33
Maximum spindle x indexing speed 6-41

Maximum static machine travel 6-36
MDI mode 4-271
Measured axis 6-45
Measurement dimensions 1-11, 1-13
Measurement encoder type 6-45
Measurement origin (OM) 1-7
Measurement origin offset 1-8
Meeting point 4-302
Meeting point programming 4-302
Message transmission to the display 4-314
Message with wait for answer 6-61
Messages 4-314
Messages with parametric value 6-63
Milling functions 4-312
Minimum interpolation block execution time 6-26
Minimum static machine travel 6-36
Mirror function 4-283
Mirror function processing 6-24
Miscellaneous M functions 2-21
Mixed line 6-26
Mixed machines 4-308
Mixed programming 4-8
Modal G functions 2-18
Modal M functions 2-21
Mode 1-3
Movement origin selection 4-229
Movements 4-141
Multiple axis groups 4-294
Multiplication 6-4, 6-21
MX 4-308

N

N/M AUTO axes 6-43
N/M-AUTO 6-43
No line (stylus raised) 6-26
Nonmodal G functions 2-18
Nonmodal M functions 2-21
Normal vector 4-101
Notes on EIA code 2-17
Notes on ISO code 2-17
Notes on mixed machines (MX) 4-308
Notes related to machine axes 4-308
Number
    of CNC axis groups 6-29
    of constant infeeds 4-141
    of machine axis groups 6-29
    of total speed anticipation filter terms 6-27
    programming 8-1
Numbers assigned to parameters 6-26
Numerical values 6-66

O

Off-centring
    of the table by DAT3 4-245
    of the workpiece 1-12
    on the X axis 6-34
Offset programmed by G59 on the X axis 6-34
Offsets 1-8, 1-9
    on the X axis 1-10
    on the Y axis 1-10
    on the Z axis 1-9
On-the-fly measurement 6-36
OP 1-9
Op 1-9
Operations executable with external parameters 6-21
Operations executable with variables L 6-4
Optional arguments 2-19
Optional stop 4-269
OR 6-5, 6-21
Origin 1-5, 1-7
Origin switch status 6-45
Origins 1-7
ORPOM 1-8
Other cycles 4-146
Other functions 4-256
Overall dimensions of the part in 3D 4-287
Overspeed 4-260
Overspeed by handwheel 4-260
Overspeed coefficient on the path in G12 6-26

P

Parameters E 4-295
Parametric geometric transformations 6-32
Parametric programming 6-1, 6-68
Part origin 1-9
Part origin offset 1-9
Pass settings 4-148
Path sequencing conditions 4-60
Peck drilling 4-117, 4-141, 4-171
Peck drilling cycle 4-117
Peck drilling order 4-171
PGP 5-3
PGP programming 5-18
Plane selection 4-10
PLC axes 4-304
Pocket 4-163
    and island introduction blocks 4-163
    cycles 4-155
    introduction block 4-163
Polar angle reference axis 4-47
Polar programming 4-47
    of a circle 4-50
    of a line 4-48
Position of origin switch with respect to machine origin 6-44
Preparatory G functions 2-18
Preselecting the programme origin 4-233, 4-311
Previous block data access symbols 6-64
Primary X, Y and Z axes 1-6, 3-3
Processing of mirror function 6-24
PROFIL 5-24
PROFIL function 5-24
Profile geometry programming 5-1
Programmable axes 4-294

Programme 1-3, 2-8
analysis access parameters 6-24
backup 4-304
backup media 1-3
creation by G76+ 4-224
declaration and backup 4-304
definition 1-3
deletion by G76- 4-225
dimensions 1-11, 1-13
load by peripheral 2-12
numbering 2-12
origin 1-9, 4-233, 4-309
origin definition 4-309
origin offset 1-9, 4-235, 6-24
stack 7-1, 7-3
start 2-9, 2-10
status access symbols 6-64
stop 4-267
structure 2-1
variables 4-295
variables L 6-3
Programmed cancellation
of edit (EDIT) mode 4-271
of manual data input (MDI) mode 4-271
of subroutine calls by the automatic control function 4-271
Programmed deletion of variables 7-12
Programmed feed stop 4-258
Programmed M function processing 4-289
Programming
by diameter enabled 6-24
in absolute dimensions with respect to the measurement origin 4-229
in absolute dimensions with respect to the programme origin 4-7
in absolute or incremental dimensions 4-7
in inches 4-262
in incremental dimensions relative to the block start point 4-7
in metric units 4-262
of A, B or C axes declared as non-rotary 3-7
of axes by variables L or parameters E defined by symbolic units 7-7
of chamfers 4-58
of chamfers and fillets located between two elements 5-17
of entirely defined geometric elements 5-9
of fillets 4-58
of geometric elements 5-9
of geometric elements not entirely defined 5-10
of independent secondary axes 3-4
of movements 4-26
of parallel carrier/carried axis pairs 3-5
of PLC axes 4-305
of rotary axes modulo 360 degrees 3-6
of servoed rotary axes with limited excursion 3-7
Programming examples
in PGP 5-18
of cycles G81-G89 4-142
of cycles with any contours 4-178
of variables L 6-11
Programming notes
on geometric contour definition blocks 4-157
on machining orders 4-159
on variables L100-L199 and L900-L959 6-9
PULL 7-4
PULL function 7-4
PUSH 7-3
PUSH function 7-3

R

R 1-14, 6-4, 6-21
R + DR 1-15
R.. 5-6
Radius of a circle 5-6
Rapid positioning 4-23
Reactivation
of edit (EDIT) modes 4-271
of manual data input (MDI) modes 4-271
of subroutine calls by the automatic control function 4-271
Reading programme status access symbols 6-64
Reaming 4-124, 4-139
Reaming cycle 4-124
Reset 4-220, 4-296
Restoring the values of variables L 7-4
Restoring variables L 7-3
Restrictions due to drip feed mode 4-221
Retraction 4-141
Retraction after penetration 4-141
Review 1-1
of axis orientation 1-5
of definitions 1-5
of machine 1-5
Right radius correction 4-85
Rigid tapping 4-122
Rigid tapping cycle 4-122
Rotary A, B and C axes 1-5, 3-3
Rotation centre offset 1-12
Rotation direction 4-12
Roughing 4-173
Roughing order 4-173
RTCP function 6-25
RTCP function enabled 6-25
Ruler with encoded reference marks 6-45

S

S 6-4, 6-21
Sagittal error 6-27
Sampling period 6-29
Saving the values of variables L in the stack 7-3
Saving variables L 7-3

Scaling factor 4-279, 6-34
enable 4-279
inhibit 4-279
Secant element 5-7
Secondary linear U, V and W axes 1-6
Secondary U, V and W axes 3-3
Selection of programming system 4-7
Semi-finishing 4-175
Semi-finishing order 4-175
Sending a message
to a PC 4-316
to a peripheral 4-316
to a remote server 4-316
to the display 4-314
to the PLC function 4-316
Sequence 2-7
Sequence breaks 4-193
Sequence interrupt 4-208
Sequence without return 4-203
Sequences 4-193
Servo-control error tolerated on a circle 6-26
Setting of the feed rate potentiometer assigned to the group 6-31
Simple drilling order 4-171
Simple pockets 4-146
Simple pocket cycle 4-146
Sine 6-4, 6-21
Solid line 6-26
Special cycle programming blocks 4-156
Special programming of multiple axis groups 4-294
Special programming of PLC axes 4-307
Speed demand for the spingle controlled by its group 6-40
Speed of movement 4-62
Speed threshold below which spindle x is considered stopped 6-42
Spindle 4-12, 4-141
access parameters 6-40
and feed rate potentiometer enable 4-274
and feed rate potentiometer inhibiting 4-274
control 4-12, 4-298
control selection 4-19
indexing 4-17
measurement 4-21, 4-283, 4-298
measurement selection 4-21
position reference used by group 6-40
potentiometer 6-25
programming 4-298
speed control 4-14
speed in rpm 4-14
speed range 4-16
stop 4-12
Spindle x in-position window 6-41
Spindle x indexing gain 6-41
Spindle x modulo 6-40
Spindle x position reference 6-40
Spindle x speed setting 6-40
Spline 4-247
Spline curve 4-247
deletion 4-247, 4-255
execution command 4-247, 4-251
interpolation 4-248
Spline interpolation 4-247
Square root 6-4, 6-21
Stack allocation 7-3
Subroutine branch
by M function 4-200
on reset 4-220, 4-296
with multiple axis groups 4-296
Subroutine branches and calls 2-11
Subroutine call by automatic control function 4-205
Subroutine display inhibited 2-12
Subroutine nesting 4-197
Subroutine return block 4-222
Subroutines 2-11
Subtraction 6-4, 6-21
Surface 4-101
Suspension of execution and forcing of block concatenation 4-289
Symbolic addressing
of variables L900 to L925 6-10
of variables L926 to L951 6-10
Symbolic variable declaration 7-6
Symbolic variables 7-1, 7-6
Symbols addressing Boolean values 6-65
Symbols addressing numerical values 6-66
Symmetrically driven axis 6-48
Synchronised axis 6-48
System 1-3

T

T 6-4
T function 4-295
Table summarising cycles G81 to G89 4-141
Tangent element 5-7
Tangential feed rate 4-72
Tapping 4-120, 4-141
Tapping cycle 4-120
Temporary suspension of next block preparation 4-213
Test on a variable for a conditional jump 6-8
Test on an external parameter for a conditional jump 6-52
Thread chasing 4-137
Time constant to absorb the antistick pulse 6-49
Tolerance interval coefficient 6-42
Tool
call 4-76
correction call 4-81
corrections 4-79, 6-32
dimension types 4-310
dimensions 1-14
infeeds 4-176
length 1-14
length L of correction xxx 6-32
position with respect to the part 4-85
programming 4-76
radius 1-14

radius “R” of correction xxx 6-32
retractions 4-176
type of correction xxx 6-32
type selection 4-310
Tool axis 4-79
orientation 4-79
orientation definition 4-79
Tools 4-76
Toroid and spherical tools 4-102
Transferring the current values of parameters E 6-59
Transferring the current values of variables L 6-59
Travels 1-7
Truncating 6-4
Turning functions 4-312

U

Unconditional subroutine branch 4-193
Use of symbolic variables for programming 7-7
Use of the stack 7-3

V

VAR 7-6
VAR function 7-6
VAR H.. N.. N.. 7-10
VAR H.. N.. N.. function 7-10
Variable L programming syntax 6-6
Variables
L 4-295, 7-1
L100 to L199 6-9
L900 to L925 6-10
L900 to L959 6-9
L926 to L951 6-10

W

Wall 4-168
Wall introduction block 4-168
Word 2-4
defining a dimension 2-6
format 2-4
not defining a dimension 2-6
Writing a block defining
a path 2-8
a tool change 2-7
the start of rotation 2-8
Writing a programme 1-4

X

X and U axes with reference to diameter or radius 4-308
X.. Y.. 5-5
XOFF 2-8
XY plane 4-10

Y

YZ plane 4-10

Z

Zero point 1-5, 1-7
ZX plane 4-10