CNC 8070

FAGOR

REF. 0504

(SOFT V02.0x)

PROGRAMMING MANUAL

(Soft V02.0x)

Ref. 0504

PRELIMINARY WARNINGS

MACHINE SAFETY

It is up to the machine manufacturer to make sure that the safety of the machine is enabled in order to prevent personal injury and damage to the CNC or to the products connected to it.

On start-up and while validating CNC parameters, it checks the status of the following safety elements:

- *Feedback alarm for analog axes.*
- *Software limits for analog and sercos linear axes.*
- *Following error monitoring for analog and sercos axes (except the spindle) both at the CNC and at the drives.*
- *Tendency test on analog axes.*

If any of them is disabled, the CNC shows a warning message and it must be enabled to guarantee a safe working environment.

FAGOR AUTOMATION shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC resulting from any of the safety elements being disabled.

HARDWARE EXPANSIONS

FAGOR AUTOMATION shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC resulting from any hardware manipulation by personnel unauthorized by Fagor Automation.

If the CNC hardware is modified by personnel unauthorized by Fagor Automation, it will no longer be under warranty.

COMPUTER VIRUSES

FAGOR AUTOMATION guarantees that the software installed contains no computer viruses. It is up to the user to keep the unit virus free in order to guarantee its proper operation.

Computer viruses at the CNC may cause it to malfunction. An antivirus software is highly recommended if the CNC is connected directly to another PC, it is part of a computer network or floppy disks or other computer media is used to transmit data.

FAGOR AUTOMATION shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC due a computer virus in the system.

If a computer virus is found in the system, the unit will no longer be under warranty.

INDEX

10. Canned cycles

11. Multiple machining

12. Cycle editor

FAGOR

CNC 8070

(SOFT V02.0x)

13. Coordinate transformation

14. CNC variables

15. Statements and instructions

16. Probing canned cycles.

ABOUT THIS MANUAL

Title

Programming Manual.

Type of documentation

It describes functions and instructions of the CNC language.

Internal code

It is part of the manual directed to the end user. The code of the manual depends on the software version –standard– or –advanced–.

CNC 8070 USER (IN) STAN	Code 03753611
CNC 8070 USER (IN) AVANZ	Code 03753631

Version

It corresponds to the software version: (Soft V02.0x).

Start-up

Verify that the machine that integrates this CNC meets the 89/392/CEE Directive.

Before starting up the CNC, read the instructions of chapter 1 in the Installation Manual.

Warning

The information described in this manual may be changed due to technical modifications.

FAGOR AUTOMATION, S. Coop. reserves the right to make any changes to the contents of this manual without prior notice.

ABOUT THE PRODUCT

Software options

Bear in mind that some of the features described in this manual depend on the software options that are installed.

	"M" model	"GP" model
Number of execution channels	1 to 4	1 to 4
Number of axes	4 to 28	4 to 28
Number of spindles	1 to 4	1 to 4
Number of tool magazines	1 to 4	1 to 4
COCOM version	Option	Option
Sercos digital drive system	Option	Option
Tool radius compensation	Standard	Option
"C" axis	Standard	Not available
RTCP transformation	Option	Not available
High speed machining (HSC).	Option	Option
Probing canned cycles	Option	Not available
Tandem axes	Option	Not available
Synchronism and cams	Option	Not available

VERSION HISTORY

Here is a list of the features added in each software version and the manuals that describe them.

The version history uses the following abreviations:

INST	Installation manual
PRG	Programming manual
OPT	Operation manual

Software V01.0x **February of 2002**

First version.

Software 1.1x **September of 2002**

Feature	
Probe management through a digital input. It is not possible to manage it through the "Counter" module connector.	INST
Set the numbering of the digital I/O.	INST
Kinetics for rotary table.	INST
Possibility to park and unpark SERCOS axes from the PLC.	INST
Keyboard simulation from the PLC.	INST
New treatment of the JOG panel (Key + Direction).	INST / OPT
New machine parameters. • Probe setting. • Numbering of the digital I/Os. • Kinetics for rotary table. • Repositioning feedrate after a tool inspection.	INST
New variables. • Probe setting. • Numbering of the digital I/Os. • Key simulation. • Repositioning feedrate after a tool inspection. • General scaling factor. • Kinetics dimensions.	INST PRG
General scaling factor (#SCALE).	PRG
Pobring canned cycles (#PROBE).	PRG
Probe selection (#SELECT PROBE).	PRG
Programming of warnings (#WARNING).	PRG
Block repetition ($RPT).	PRG
Improved programming of high speed machining (#HSC).	PRG
Improved programming of axis swapping (#SET AX, #CALL AX, #FREE AX, #RENAME).	PRG
Macros: The number of macros in a program is now limited to 50.	PRG
Improved tool table.	OPT
Protection passwords.	OPT
Manual mode (jog). Tool calibration with or without probe.	OPT
Manual mode (jog). Automatic loading of zero offsets table.	OPT

Feature	
Manual mode (jog). Programming of feedrate “F” and spindle speed “S”.	OPT
Axis selection/deselection for handwheel jog.	OPT
Theoretical path simulation.	OPT
Definition of the first block of a block search.	OPT
Confirm that the CNC is not in automatic mode when executing a program.	OPT
Syntax check in MDI.	OPT

January of 2005 **Software: 2.0x**

Feature	
Operation under Windows XP	INST
Emergency shutdown with battery (Central unit PC104)	OPT
New languages (Basque and Portuguese)	INST
Multi-channel system, up to 4 channels. • Spindle swapping • Axis swapping • Communication and synchronization between channels. • Common arithmetic parameters. • Access to variables per channel.	INST PRG OPT
Multi-spindle system, up to 4 spindles	PRG / INST
Tool management with up to 4 magazines.	INST
Tandem axis.	INST
New kinematics table-spindle (TYPE13 to TYPE16).	INST
New kinematics for C axis (TYPE 41 to TYPE 43)	INST
Parameter matching between the CNC and the Sercos drive	INST
New machine parameters. • Warning level on Gantry axes (WARNCOUPE) • Placing the vertical softkeys on the right or on the left (VMENU). • Apply cross compensation to either theoretical or real coordinates (TYPCROSS). • Apply leadscrew compensation to either theoretical or real coordinates (TYPLSCRW). • Defining the default compensation mode (IRCOMP). • Defining the type of reference pulse (REFPULSE). • Memory sharing between applications (PLCDATASIZE). • Generic OEM machine parameters (MTBPAR). • Reading Sercos variables from the CNC (DRIVEVAR). • Backlash peak compensation (BAKANOUT, BAKTIME, ACTBAKAN).	INST
The behavior of rotary axes has been changed. Machine parameters AXISMODE, UNIDIR, SHORTESTWAY.	INST
Possibility of Sercos transmission at 8 Mhz and 16 Mhz. Parameter SERBRATE.	INST
Define the anticipation time for the axes to be considered to be in position. Machine parameter ANTIME and the PLC mark ADVINPOS.	INST
The "(V.).TM.MZWAIT " variable is not necessary in the subroutine associated with M06.	INST
Filters to eliminate the resonance of the spindle when it works as C axis or in rigid tapping.	INST
PLC. The TMOPERATION may take the values 13 and 14.	INST
PLC. New mark MMCWDG to detect that the lockup of the operating system.	INST
PLC. Accessing arithmetic parameters and OEM parameters with CNCRD returns the value multiplied by 10000 (reading in float mode).	INST
PLC. The CNCEX command and the FREE mark to execute a CNC block.	
New commands at the PLC. • New mark to disable the cross compensation tables (DISCROSS). • New mark to correct the parallelism on Gantry axes (DIFFCOMP). • Definition of external symbols (PDEF).	INST
New variables. • Software version. • Variables to be set via PLC. • Variables for adjusting the position. • Fine adjustment variables. • Feedback inputs.	INST / PRG

Feature	
Electronic-cam editor.	INST
Optimize the reading and writing of variables from the PLC. Only the following will be asynchronous. • The tool variables will be read asynchronously when the tool is neither the active one nor in the magazine. • The tool variables will be written asynchronously whether the tool is the active one or not. • The variables referred to local arithmetic parameters of the active levels will be read and written asynchronously.	INST / PRG
Spindle parking and unparking.	INST
The RESETIN mark is not necessary to park/unpark axes or spindles from the PLC.	INST
Sercos control in velocity.	INST
Behavior of the beginning and end of tool radius compensation when not programming a movement.	PRG
Changing the type of radius compensation while machining.	PRG
Via program, loading a tool in a specific magazine position.	PRG
Programming of modal subroutines (#MCALL).	PRG
Executing a block in a channel (#EXBLK).	PRG
Programming the number of repetitions in the block (NR).	PRG
Direct resolution of 2D and 3D pockets without requiring a softkey.	PRG
Simulating a canned cycle of the editor separately.	PRG
New method to jog the axes using the JOG keyboard. Axis keys and independent directions.	INST / OPT
Importing DXF files from the program editor or from the profile editor.	OPT
Importing programs of the 8055/8055i CNC from the program editor.	OPT
Use a softkey to select the repositioning of the spindle after tool inspection.	OPT
Backup-restore utility.	OPT
Improved profile editor.	OPT
Assistance in the program editor. Contextual programming assistance. • When programming "#", it shows the list of instructions. • When programming "$", it shows the list of instructions. • When programming "V.", it shows the list of variables.	OPT
Specific password for the machine parameters for kinematics.	OPT
Save the CAN configuration for testing it when starting up the system.	OPT
The diagnosis mode shows detailed information on the Sercos connection (Type and version of the drive and motor connected to it).	OPT
It is possible to print all the information on the configuration from any section of the diagnosis mode.	OPT
It is possible to simulate a cycle separately from the cycle editor.	OPT
Setup assistance. • Oscilloscope. • Bode diagram. • Circularity (roundness) test.	OPT

Feature	
New values of machine parameter SERPOWSE for the "Sercos II" board.	INST
Independent-axis programming commands.	INST
Electronic-cams programming commands.	INST
New signals that may be consulted and changed for the independent interpolator (electronic cam and independent axis)	INST
The simulated axes are ignored regarding the validation code.	
When unifying parameters, G00FEED and MAXVOLT are not sent out to the drive.	INST
Electronic-cam programming instructions (#CAM ON / #CAM OFF).	PRG
Independent-axis programming instructions (#MOVE ABS / #MOVE ADD / #MOVE INF / #FOLLOW ON / #FOLLOW OFF).	PRG
G112. Change the drive's parameter set .	PRG
DDSSETUP mode	OPT
G31. Temporary polar origin shift to the center of interpolation.	PRG

DECLARATION OF CONFORMITY

Manufacturer:

Fagor Automation, S. Coop.

Barrio de San Andrés 19, C.P. 20500, Mondragón -Guipúzcoa- (SPAIN).

We declare:

under our responsibility that the product:

Fagor CNC8070 Numerical Control

meets the following directives:

Safety:

EN 60204-1	Machine safety. Electrical equipment of the machines.

Electromagnetic compatibility:

EN 50081-2	Emission.
EN 55011	Radiated. Class A, Group 1.
EN 55011	Conducted. Class A, Group 1.
EN 61000-3-2	Current armonics.
EN 61000-3-3	Flickers and Voltage fluctuations.
EN 50082-2	Immunity.
EN 61000-4-2	Electrostatic discharges.
EN 61000-4-4	Bursts and Fast transients.
EN 61000-4-5	High Voltage conducted pulses (Surges).
EN 61000-4-11	Voltage fluctuations and Outages.
EN 61000-4-3	Radiofrequency radiated electromagnetic fields.
EN 61000-4-6	Conducted disturbance induced by radio frequency fields.

As instructed by the European Community Directives 73/23/EEC, modification 93/68/ECC on Low Voltage and 89/336/CEE on Electromagnetic Compatibility.

In Mondragón on February 1st 2002.

Fagor Automation, S. Coop. Ltda.
Director Gerente

Fdo.: Julen Busturia

SAFETY CONDITIONS

Read the following safety measures in order to prevent harming people or damage to this product and those products connected to it.

This unit may only be repaired by authorized personnel at Fagor Automation.

Fagor Automation shall not be held responsible of any physical damage or defective unit resulting from not complying with these basic safety regulations.

PRECAUTIONS AGAINST PERSONAL DAMAGE

- Interconnection of modules.

  Use the connection cables provided with the unit.

- Use proper cables.

  To prevent risks, use the proper cables for mains, Sercos and Bus Can recomended for this unit.

- Avoid electrical overloads.

  In order to avoid electrical discharges and fire hazards, do not apply electrical voltage outside the range selected on the rear panel of the Central Unit.

- Ground connection.

  In order to avoid electrical discharges, connect the ground terminals of all the modules to the main ground terminal. Before connecting the inputs and outputs of this unit, make sure that all the grounding connections are properly made.

- Make sure that it is connected to ground.

  In order to avoid electrical shock, before turning the unit on verify that the ground connection is properly made.

- Do not work in humid environments.

  In order to avoid electrical discharges, always work under 90% of relative humidity (non-condensing) and 45ºC (113ºF).

- Do not work in explosive environments.

  In order to avoid risks or damages, do no work in explosive environments.

PRECAUTIONS AGAINST PRODUCT DAMAGE

- Working environment.

  This unit is ready to be used in Industrial Environments complying with the directives and regulations effective in the European Community.

  Fagor Automation shall not be held responsible for any damage suffered or caused when installed in other environments (residential or homes).

- Install the unit in the right place.

It is recommended, whenever possible, to install the CNC away from coolants, chemical product, blows, etc. that could damage it.

This unit complies with the European directives on electromagnetic compatibility. Nevertheless, it is recommended to keep it away from sources of electromagnetic disturbance such as:

- Powerful loads connected to the same AC power line as this equipment.
- Nearby portable transmitters (Radio-telephones, Ham radio transmitters).
- Nearby radio/TV transmitters.
- Nearby arc welding machines.
- Nearby High Voltage power lines.
- Etc.

☐ Enclosures.

The manufacturer is responsible of assuring that the enclosure involving the equipment meets all the currently effective directives of the European Community.

☐ Avoid disturbances coming from the machine tool.

The machine-tool must have all the interference generating elements (relay coils, contactors, motors, etc.) uncoupled.

☐ Use the proper power supply.

Use an external regulated 24Vdc power supply for the keyboard and the remote modules.

☐ Grounding of the power supply.

The zero volt point of the external power supply must be connected to the main ground point of the machine.

☐ Analog inputs and outputs connection.

It is recommended to connect them using shielded cables and connecting their shields (mesh) to the corresponding pin (see chapter 1 in the Installation Manual).

☐ Ambient conditions.

The working temperature must be between +5ºC and +45ºC (41ºF and 113ºF)

The storage temperature must be between -25ºC y 70ºC (-13ºF y 158ºF)

☐ Monitor enclosure.

Make sure that the gaps between the Central Unit and each wall of the enclosure are respected as indicated in chapter 1 of the Installation Manual.

Use a DC fan to improve enclosure ventilation.

☐ Main AC power switch.

This switch must be easy to access and at a distance between 0.7 and 1.7 m (2.3 and 5.6 ft) off the floor.

PROTECTIONS OF THE UNIT ITSELF

☐ Remote modules.

All the digital inputs and outputs have galvanic isolation via optocouplers between the CNC circuitry and the outside.

PRECAUTIONS DURING REPAIR

- Do not get into the inside of the unit.

  Only personnel authorized by Fagor Automation may manipulate the inside of this unit.

- Do not handle the connectors with the unit connected to AC power.

  Before manipulating the connectors (inputs/outputs, feedback, etc.) make sure that the unit is not connected to AC power.

SAFETY SYMBOLS

- Symbols that may appear on the manual.

Symbol of danger or prohibition.

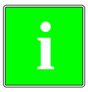
It indicates actions or operations that may hurt people or damage products.

Warning symbol.

It indicates situations that certain operations could cause and the suggested actions to prevent them.

Obligation symbol.

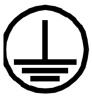
It indicates actions and operations that must be carried out.



Information symbol.

It indicates notes, warnings and advises.

- Symbols that the product may carry.



Ground protection symbol.

It indicates that that point must be under voltage.

WARRANTY TERMS

All products manufactured or marketed by Fagor Automation has a warranty period of 12 months from the day they are shipped out of our warehouses.

The mentioned warranty covers repair material and labor costs, at Fagor Automation facilities, incurred in the repair of the products.

Within the warranty period, Fagor Automation will repair or replace the products verified as being defective.

Fagor Automation is committed to repairing or replacing their products from the time they launch them up to 8 years after they disappear from the product catalog.

It is entirely up to Fagor Automation to determine whether a repair is to be considered under warranty.

Excluding clauses

The repair will take place at our facilities; therefore, all shipping expenses as well as travelling expenses incurred by technical personnel are NOT under warranty even when the unit is under warranty.

This warranty will be applied so long as the equipment has been installed according to the instructions, it has not been mistreated or damaged by accident or negligence and has been manipulated by personnel authorized by Fagor Automation.

If once the service call or repair has been completed, the cause of the failure is not to be blamed ON the FAGOR product, the customer must cover all generated expenses according to current fees.

No other implicit or explicit warranty is covered and FAGOR AUTOMATION shall not be held responsible, under any circumstances, of the damage which could be originated.

Service contracts

Service and Maintenance Contracts are available for the customer within the warranty period as well as outside of it.

MATERIAL RETURNING TERMS

When sending the Central Unit or the Remote Modules, pack them in its original package and packaging material. If the original packaging material is not available, pack it as follows:

1. Get a cardboard box whose three inside dimensions are at least 15cm (6 inches) larger than those of the unit. The cardboard being used to make the box must have a resistance of 170Kg (375 lb.).
2. Attach a label indicating the owner of the unit, person to contact, type of unit and serial number. In case of malfunction also indicate symptom and a brief description of the problem.
3. Wrap the unit in a polyethylene roll or similar material to protect it.

   When sending the Central Unit, protect especially the screen.
4. Pad the unit inside the cardboard box with poly-utherane foam on all sides.
5. Seal the cardboard box with packing tape or industrial staples.

ADDITIONAL REMARKS

Mount the CNC away from coolants, chemical products, blows, etc. which could damage it.

Before turning the unit on, verify that the ground connections have been properly made.

In order to avoid electrical shock at the Central Unit, use the proper power (mains) cable. Use 3-wire power cables (one for ground connection).

In case of a malfunction or failure, disconnect it and call the technical service. Do not get into the inside of the unit.

RELATED DOCUMENTATION

Manuals directed to the machine manufacturer or to the person in charge of doing the installation and start-up of the CNC.

Hardware manual.

It describes the hardware configuration and the technical data of each element.

Installation Manual.

It describes how to install and start up the CNC.

Manuals directed to the end user; that is, to the CNC operator.

Operating Manual.

Describes how to operate the CNC.

Programming Manual.

It describes how to program the CNC.

Examples manual.

It contains programming examples.

Other manuals, directed to the machine manufacturer and to the end user.

Manual for New Features.

It is optional. It describes the new features and modifications implemented since the version of the installation, operating and programming manuals.

Error solving manual.

It offers a description of the error messages that may appear on the CNC indicating the probable causes that originate them and how to solve them.

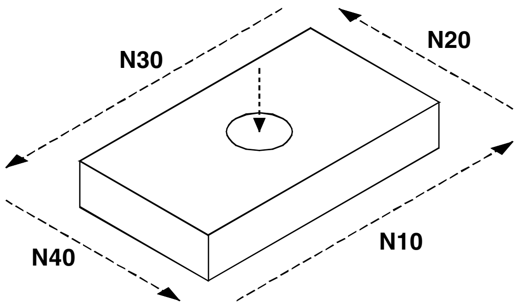
CREATING A PROGRAM

1

1.1 Program structure

A CNC program consists of a set of blocks or instructions that properly organized, in subroutines or in the program body, provide the CNC with the necessary data to machine the desired part.

Each block contains all the functions or command necessary to execute an operation that may be machining, preparing the cutting conditions, controlling the elements of the machine, etc.




```
%example
```
(Name of the program)

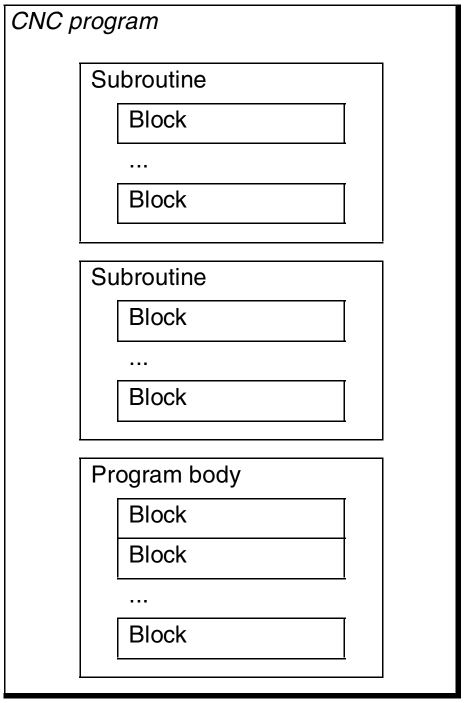
```
N5 F550 S1000 M3 M8 T1 D1
```
(Sets the machining conditions)

```
N6 G0 X0 Y0
```
(Positioning)

```
N10 G1 G90 X100
N20 Y50
N30 X0
N40 Y0
```
(Machining)

```
N50 M30
```
(End of program)

The CNC program may consist of several subroutines and the body of the program.



Local subroutines

A subroutine is a set of blocks that, once properly identified, may be called upon several times from another subroutine or from the program body.

Programming subroutines is optional and they must be defined before the program body. Subroutines are normally used for defining a bunch of operations or movements that are repeated several times throughout the program.

The beginning of a subroutine is defined by "%L<name>", where <name> may be up to 14 characters long and consist of uppercase and lowercase letters as well as numbers (no blank spaces are allowed). Subroutine call is case sensitive, the name must be written exactly as it has been defined. . The end of the subroutine is defined with M17, M29 or #RET.

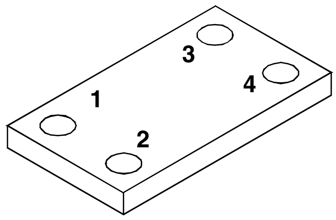
```
%L sub_name1      (Subroutine definition)
 N10...
 N20...
 N30...
M17               (End of subroutine)

%L sub_name2      (Subroutine definition)
 N10...
 N20...
 N30...
M17               (End of subroutine)
```

Program body

The beginning of the program body is defined by "%<name>", where <name> may be up to 14 characters long and consist of uppercase and lowercase letters as well as numbers (no blank spaces are allowed). It needs not be programmed when no subroutines are defined.

The end of the program body is defined by "M02" or "M30" and IT MUST be programmed.

```
%L POINTS
 G01 X2 Y2
 G01 X3 Y3
 G01 X4 Y4
M17
```




```
%PROGRAM
 G81 X1 Y1 ···     (Center punching definition)
 LL POINTS         (Call the subroutine)
 G81 X1 Y1 ···     (Drilling definition)
 LL POINTS         (Call the subroutine)
 G84 X1 Y1 ···     (Tapping definition)
 LL POINTS         (Call the subroutine)
 G80
M30
```

1.2 Block structure

The blocks or instructions comprising the subroutines or the program body may be defined by commands in ISO code or in high-level language. Each block must be written in either language but not mixed; a program may combine blocks written in both languages. Empty blocks (empty lines) are also allowed.

In either language, it is also possible to use any type of arithmetic, relational or logic expression.

ISO coded language

It is especially designed to control the movement of the axes because it provides movement data and conditions as well as feedrate and speed.

This language has:

- Preparatory functions for movement establishing the geometry and work conditions such as linear and circular interpolations, threading, canned cycles, etc.
- Functions to control cutting conditions such as feedrate of the axes, spindle speed and accelerations.
- Functions to control the tools.
- Additional functions containing technological instructions.
- Definition of position values.

High level language

This language provides the user with a set of control commands with a terminology similar to the one used by other languages, such as $IF, $GOTO, #MSG, #HSC, etc.

Its types of commands are:

- Programming instructions.
- Flow controlling instructions to make loops and jumps within the program.
- To define and call upon subroutines with local parameters where a local variable is the one only known to the subroutine where it has been defined.

It is also possible to use any type of arithmetic, relational or logic expression.

Parameters, constants and expressions

Constants, parameters, variables and arithmetic expressions may be used from ISO blocks as well as from special commands $ and #.

1.3 Programming in ISO code

ISO-coded functions consist of letters and numbers.

- The letters are "N", "G", "F", "S", "T", "D", "M", "H", "NR" plus those identifying the axes.
- The numbers include digits "0" through "9", the "+" and "-" signs and the decimal point ".". On the other hand, the numbers may be replaced by a parameter, variable or arithmetic expression whose result is a number as explained in the section on *"1.5 Parameters, constants and expressions"* later in this chapter.

Programming allows blank spaces between letters, numbers and a sign as well as not using the sign with positive values.

Block structure

A block may have the following data, but needs not contain all of them.

/	N—	G— G—	X— Y— Z—	F— S—	T—	M— H—	NR—	(—)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

These data don't have a preestablished order, except the block skip and block label which must always be programmed at the beginning of the block.

1. Conditional block skip "/"

If the block-skip mark is active, the CNC will skip the blocks having this character (not executing them) and will go on to the next block.

The CNC reads several blocks ahead of the one in execution, in order to calculate in advance the path to travel. The block-skip condition is examined at the time when the block is read.

2. Block identification "N"

They must be programmed when the block is used as the destination of references or jumps. In this case, it is recommended to program it alone in the block. It may be represented in two ways:

- The letter "N" followed by the block number (0-4294967295) and the ":" character (only when the label is used as the destination of a block jump); they need not follow a particular order or be consecutive.

  If the label is not a jump target and is programmed without ":", it may go in any position of the block, not necessarily at the beginning.

- "[<name>]" type labels, where <name> may be up to 14 characters long and may consist of uppercase and lowercase characters as well as numbers (no blank spaces are allowed).

Both types of data may be programmed in the same block.

3. Preparatory "G" functions

They determine the geometry and work conditions such as linear and circular interpolations, chamfers, canned cycles, etc.

The section on ***"1.3.1 List of preparatory "G" functions"*** in this chapter shows the available G functions.

4. Point coordinates "X, Y, Z..."

They determine the movement of the axes.

The axis name is defined by 1 or 2 characters. The first character must be one of the letters X - Y - Z - U - V - W - A - B - C. The second character is optional and will be a numerical suffix between 1 and 9. This way, the name of the axes may be any in the "X, X1...X9,...C, C1...C9" range.

The movements are represented by the axis letter followed by the target position for the axis.

X100 Y34.54 X2 = 123,4 A5=78.532

Depending on the units, the programming format will be:

- In millimeters, format ±5.4 (5 integers and 4 decimals).
- In inches, format ±4.5 (4 integers and 5 decimals).

5. Technological functions "F" and "S"

They determine the feedrate of the axes and the spindle speed.

The feedrate is represented by the letter "F" followed by the desired feedrate value.

6. The spindle speed is represented by the letter "S" followed by the desired speed value.

7. Tool number "T" and tool offset "D"

It selects the tool and tool offset to be used to carry out the programmed machining operation. The tool is represented by the letter "T" followed by the tool number (0-4294967295). The tool offset is represented by the letter "D" followed by the tool offset number.

8. Auxiliary functions "M" and "H"

With the auxiliary functions, it is possible to control machine elements such as spindle turning direction, coolant, etc.

They are represented by the letters "M" or "H" followed by the function number (0-65535)

9. Number of block repetitions "NR"

It indicates the number of times the block will be executed.

It can only be programmed in blocks containing a movement.

If the block is under the influence of a modal canned cycle, the latter will be repeated as many times as the block repetition has been programmed. When programming NR0, the movements will be executed, but the modal canned cycle is not executed at the end of each one.

10. Block comment "(...)"

To associate a comment with the block. When executing the program, the CNC ignores this information.

The information to be considered as comment must go between parentheses "(" and ")". It needs not go at the end of the block; it may go in the middle and there may be more than one comment in the same block.

1.3.1 List of preparatory "G" functions

The following tables shows a list of preparatory functions available at the CNC. The meaning of the "M", "D" and "V" fields of the table is the following:

M Means that the function is modal; in other words, once programmed, it will remain active until an incompatible "G" code is programmed or an M02 or an M30 or until an EMERGENCY or a RESET is carried out or the CNC is turned off and back on.

Those cases indicated with "!", mean the function remains active even after an M02, M30 or a RESET and after the CNC is powered off and back on.

D Means that the function is active by default; in other words, that the CNC will assume it on power-up, after executing an M02 or M30 and after an EMERGENCY or a RESET.

Those cases indicated with "?" mean that the default quality of the function depends on the settings of the general CNC machine parameters.

V Means that the function is displayed in automatic and jog modes next to the current machining conditions.

Function	M	D	V	Meaning	Section
G00	*	?	*	Rapid traverse	6.1
G01	*	?	*	Linear interpolation	6.2
G02	*		*	Clockwise circular (helical) interpolation	6.3 / 6.6
G03	*		*	Counterclockwise circular (helical) interpolation	6.3 / 6.6
G04			*	Dwell	8.1
G05	*	?	*	Controlled corner rounding (modal)	7.3
G06			*	Arc center in absolute coordinates (not modal)	6.3.5
G07	*	?	*	Square corner (modal)	7.1
G08			*	Arc tangent to previous path	6.4
G09			*	Arc defined by three points	6.5
G10	*	*		Mirror image cancellation	7.8
G11	*		*	Mirror image on X	7.8
G12	*		*	Mirror image on Y	7.8
G13	*		*	Mirror image on Z	7.8
G14	*		*	Mirror image in the programmed directions	7.8
G17	*	?	*	Main plane X-Y, and longitudinal axis Z	3.1
G18	*	?	*	Main plane Z-X, and longitudinal axis Y	3.1
G19	*		*	Main plane Y-Z, and longitudinal axis X	3.1
G20	*		*	Main plane by two directions and longitudinal axis	3.1.1
G30				Polar origin preset	4.6
G31			*	Temporary polar origin shift to the center of arc	6.3.4
G33	*		*	Electronic threading with constant pitch	6.7
G36			*	Automatic radius blend	7.4
G37			*	Tangential entry	7.6
G38			*	Tangential exit	7.7
G39			*	Automatic chamfer blend	7.5
G40	*	*		Cancellation of tool radius compensation	9.1
G41	*		*	Left-hand tool radius compensation	9.1
G42	*		*	Right-hand tool radius compensation	9.1
G50	*	?		Semi-rounded corner	7.2
G53	*			Zero offset cancellation	4.5
G54	!		*	Absolute zero offset 1	4.4
G55	!		*	Absolute zero offset 2	4.4
G56	!		*	Absolute zero offset 3	4.4
G57	!		*	Absolute zero offset 4	4.4

Function	M	D	V	Meaning	Section
G58	!		*	Absolute zero offset 5	4.4
G59	!		*	Absolute zero offset 6	4.4
G60			*	Square corner (not modal)	7.1
G61			*	Controlled corner rounding (not modal)	7.3
G63	*		*	Rigid tapping	6.8
G70	*	?	*	Programming in inches	3.2
G71	*	?		Programming in millimeters	3.2
G72			*	Scaling factor	7.10
G73	*		*	Coordinate system rotation (pattern rotation)	7.9
G74			*	Home search	2.4.2
G80	*	*		Canned cycle cancellation	10.1.3
G81	*		*	Drilling canned cycle	10.2
G82	*		*	Drilling canned cycle with variable peck	10.3
G83	*		*	Deep-hole drilling canned cycle with constant peck	10.4
G84	*		*	Tapping canned cycle	10.5
G85	*		*	Reaming canned cycle	10.6
G86	*		*	Boring canned cycle	10.7
G87	*		*	Rectangular pocket canned cycle.	10.8
G88	*		*	Circular pocket canned cycle	10.9
G90	*	?		Programming in absolute coordinates	3.3
G91	*	?	*	Programming in incremental coordinates	3.3
G92	!		*	Coordinate preset	4.3
G93	*		*	Machining time in seconds	5.2.1
G94	*	?		Feedrate in millimeters/minute (inches/minute)	5.2.1
G95	*	?	*	Feedrate in millimeters/revolution (inches/revolution)	5.2.1
G96	*		*	Constant surface speed	5.3.1
G97	*	*		Constant turning speed	5.3.1
G98	*	*		Withdrawal to the starting plane	10.1.4
G99	*		*	Withdrawal to the reference plane	10.1.4
G100			*	Probing	8.6
G101	*			Include probe offset	8.6.1
G102	*			Exclude probe offset	8.6.1
G108	*	*		Feedrate blending at the beginning of the block	5.2.2
G109			*	Feedrate blending at the end of the block	5.2.2
G112	*			Changing of parameter range of an axis	8.5
G130	*		*	Percentage of acceleration to be applied per axis	5.2.5
G131	*		*	Percentage of acceleration to be applied to all the axes	5.2.5
G132	*		*	Percentage of jerk to be applied per axis	5.2.6
G133	*		*	Percentage of jerk to be applied to all the axes	5.2.6
G134	*		*	Percentage of Feed-Forward to be applied	5.2.7
G135	*		*	Percentage of AC-Forward to be applied	5.2.8
G136	*		*	Circular transition between blocks	9.1.1
G137	*	*		Linear transition between blocks	9.1.1
G138	*		*	Direct activation/cancellation of tool compensation	9.1.1
G139	*	*		Indirect activation/cancellation of tool compensation	9.1.1
G151	*	*	*	Programming in diameters	3.4
G152	*			Programming in radius	3.4
G157	*		*	Excluding axes in the zero offset	4.4.2
G158	*		*	Incremental zero offset	4.4.1
G159	!		*	Additional absolute zero offsets	4.4
G160			*	Multiple machining in straight line	11.1
G161			*	Multiple machining in rectangular pattern	11.2
G162			*	Multiple machining in grid pattern	11.3
G163			*	Multiple machining in a full circle	11.4
G164			*	Multiple machining in arc pattern	11.5
G165			*	Machining programmed with an arc-chord	11.6
G170	*			Hirth axes OFF	8.3
G171	*	*		Hirth axes ON	8.3
G180			*	OEM Subroutine	8.4
G181			*	OEM Subroutine	8.4
G182			*	OEM Subroutine	8.4

Function	M	D	V	Meaning	Section
G183			*	OEM Subroutine	8.4
G184			*	OEM Subroutine	8.4
G185			*	OEM Subroutine	8.4
G186			*	OEM Subroutine	8.4
G187			*	OEM Subroutine	8.4
G188			*	OEM Subroutine	8.4
G189			*	OEM Subroutine	8.4
G192	*		*	Turning speed limit	5.2.2
G193			*	Interpolating the feedrate	5.2.2
G196	*		*	Constant cutting point feedrate	5.2.3
G197	*	*		Constant tool center feedrate	5.2.3
G198	*			Setting of lower software travel limits	8.2
G199	*			Setting of upper software travel limits	8.2
G200				Exclusive manual intervention	6.9.2
G201	*			Activation of additive manual intervention	6.9.1
G202	*	*		Cancellation of additive manual intervention	6.9.1
G261	*		*	Arc center in absolute coordinates (modal)	6.3.5
G262	*	*		Arc center referred to starting point	6.3.5
G263	*		*	Arc radius programming	6.3.2
G264	*		*	Cancellation of arc center correction	6.3.6
G265	*	*		Activation of arc center correction	6.3.6
G266			*	Feedrate override at 100%	5.2.4
G281			*	Conversational center-punching cycle	12.2
G282			*	Conversational drilling cycle 1	12.3
G283			*	Conversational drilling cycle 2	12.4
G284			*	Conversational tapping cycle	12.5
G285			*	Conversational reaming cycle	12.6
G286			*	Conversational boring cycle 1	12.7
G287			*	Conversational rectangular pocket cycle	12.10
G288			*	Conversational circular pocket cycle	12.11
G289			*	Conversational simple pocket cycle	12.9
G290			*	Conversational surface milling cycle	12.17
G291			*	Conversational rectangular boss cycle	12.15
G292			*	Conversational circular boss cycle	12.16
G293			*	Conversational point-to-point profiling cycle	12.18
G294			*	Conversational profiling cycle	12.19
G295			*	Conversational slot milling cycle	12.20
G296			*	Conversational pre-emptied pocket cycle	12.12
G297			*	Conversational boring cycle 2	12.8

1.4 High-level language programming

The commands of high level language are made up of control instructions "#" and flow control instructions "$".

Block structure

A block programmed in high-level language may have the following data, but need not contain all of them.

/ (1)	N— (2)	High-level language commands (3)	(—) (4)

The block-skip condition and the block identification must always be programmed at the beginning of the block.

1. Conditional block skip "/"

If the block-skip mark is active, the CNC will skip the blocks having this character (not executing them) and will go on to the next block.

The CNC reads several blocks ahead of the one in execution, in order to calculate in advance the path to travel. The block-skip condition is examined at the time when the block is read.

2. Block identification "N"

They must be programmed when the block is used as the destination of references or jumps. In this case, it is recommended to program it alone in the block. It may be represented in two ways:

- The letter "N" followed by the block number (0-4294967295) and the ":" character (only when the label is used as the destination of a block jump); they need not follow a particular order or be consecutive.

  If the label is not a jump target and is programmed without ":", it may go in any position of the block, not necessarily at the beginning.

- "[<name>]" type labels, where <name> may be up to 14 characters long and may consist of uppercase and lowercase characters as well as numbers (no blank spaces are allowed).

Both types of data may be programmed in the same block.

3. High-level commands "#—" and "$—"

The high-level commands comprise the instructions and flow control instructions.

- Instructions are programmed preceded by the "#" sign and they can only be programmed one per block. They are used to carry out various functions.
- Flow control instructions are programmed preceded by the "$" sign and can only be programmed one per block. They are used to make loops and program jumps.

Assigning values to parameters and variables can also be considered as high-level commands.

In the chapter on *"15 Statements and instructions"* of this manual describes all the available instructions and instructions.

4. Block comment "(...)"

To associate a comment with the block. When executing the program, the CNC ignores this information.

The information to be considered as comment must go between parentheses "(" and ")". It needs not go at the end of the block; it may go in the middle and there may be more than one comment in the same block.

When programming in high-level language, a comment may also be defined using the instructions "#COMMENT BEGIN" and "#COMMENT END".

1.5 Parameters, constants and expressions

Constants, parameters, variables and arithmetic expressions may be used from ISO blocks as well as from special commands $ and #.

Constants

They are fixed values that cannot be modified by program; constants are numbers in decimal system and read-only tables and variables because their value cannot be changed within a program.

Variables

The CNC has a number of internal variables that may be accessed from the user program, from the PLC or from the interface. Ver el capítulo *"14 CNC variables"*.

The user may create his own variables, as follows.

V.P.name - User variable local to the program.

V.P.name - User variable global to the program.

Arithmetic parameters

Parameters are general purpose variables that the user may utilize to create his/her own programs. The CNC has global parameters (accessible from the program or any subroutine) and local parameters (accessible only from the program or subroutine where they have been programmed) and common parameters (accessible from all the channels).

The section on *"1.5.1 Arithmetic parameters"* in this chapter shows how to work with parameters.

Operators

An operator is a symbol that indicates the mathematical or logic operations to carry out.

The section on *"1.5.2 Operators and functions"* in this manual shows a description of the various types of operators and functions available.

Expressions

An expression is any valid combination of constants, parameters, variables and operators.

The section on *"1.5.3 Expressions"* in this chapter shows how to work with expressions

1.5.1 Arithmetic parameters

The CNC has three types of arithmetic parameters. The range of available parameters of each type is defined in the machine parameters.

- Local parameters may only be accessed from the program or subroutine where they have been programmed. There are seven groups of local parameters in each channel.

  The maximum range of local parameters is P0 to P99, the typical range being P0 to P25.

  When the parameters are used in the block calling a subroutine may also be referred to by the letters A-Z (except Ñ) so "A" is the same as P0 and "Z" the same as P25.

- Global parameters may be accessed from any program and subroutine called upon from the program. There is a group of global parameters in each channel.

  The maximum range of global parameters is P100 to P9999, the typical range being P100 to P299.

- The common parameters may be accessed from any channel. The value of these parameters is shared by all the channels.

  The maximum range of common parameters is P10000 to P19999, the typical range being P10000 to P10999.

The user may use the parameters when editing its own programs. During execution, the CNC will replace these parameters with the values assigned to them at the time.

```
P0=0 P1=1 P2=20 P3=50 P4=3
P10=1500 P100=800 P101=30
...
GP0 XP0 YP0 SP10 MP4          ==>   G0 X0 Y0 S1500 M3
GP1 XP2 YP3 FP100             ==>   G1 X20 Y50 F800
MP101                         ==>   M30
```

Programming

In blocks programmed in ISO code, it is possible to define the values of all the fields "N", "G", "F", "S", "T", "D", "M", "H", "NR" and axis coordinates using parameters. Using indirect addressing, it is also possible to define the number of a parameter with another parameter; "P[P1]", "P[P2+3]".

In blocks having "#" instructions, the values of any expression may be defined with parameters.

Parameters in the subroutines

The defined subroutines may be called upon from the main program or from another subroutine; they may in turn call a second one, the second one may call a third one, and so on. The CNC limits this calls to a maximum of 20 nesting levels.

Local parameters

The CNC has global parameters (accessible from the program or any subroutine) and local parameters (accessible only from the program or subroutine where they have been programmed).

Local parameters may be assigned to more than one subroutine up to 7 parameter nesting levels within the 20 subroutine nesting levels. Not all the subroutine call types change the nesting level; only the `#PCALL`, `#CALL`, calls and functions `G180` to `G189`.

Global parameters

Global parameters will be shared by the program and the subroutines of the channel. They may be used in any block of the program and of the subroutine regardless of the nesting level they may be at.

Common parameters

Common parameters will be shared by the program and the subroutines of any channel. They may be used in any block of the program and of the subroutine regardless of the nesting level they may be at.

1.5.2 Operators and functions

An operator is a symbol that indicates the mathematical or logic operations to carry out. The CNC offers the following types of operators.

Arithmetic

To perform arithmetic operations.

+	Add	P1 = 3+4	P1=7
-	Subtract Change sign	P2 = 5-2 P2 = -[3+4]	P2=3 P2 = -7
*	Multiply	P3 = 2*3	P3=6
/	Divide	P4 = 9/2	P4=4.5
MOD	Module or remainder of a division	P5 = 5 MOD 2	P5=1
**	Exponent	P6 = 2**3	P6=8

In the operation, when using the parameter or variable storing the result, the add, subtract, multiply and divide operators may be used as follows:

+=	Compounded addition	P1 += 3	P1=P1+3
-=	Compounded subtraction	P2 -= 5	P2=P2-5
*=	Compounded multiplication	P3 *= 2	P3=P3*2
/=	Compounded division	P4 /= 9	P4=P4/9

Relational

Used for doing comparisons.

==	Equal to	P1 == 4
!=	Different from, other than	P2 != 5
>=	Greater than or equal to	P3 >= 10
<=	Smaller than or equal to	P4 <= 7
>	Greater than	P5 > 5
<	Smaller than	P6 < 5

Binary

Used for doing binary comparisons between constants and/or arithmetic expressions.

&	Binary AND	P1 = P11 & P12
|	Binary OR	P2 = P21 | P22
^	Exclusive OR (XOR)	P3 = P31 ^ P32
INV[...]	Inverse	P4 = INV[P41]

If the constant or the result of the arithmetic expression is a decimal number, the decimal portion will be ignored.

Logic

Used for doing logic comparisons between conditions.

*	Logic AND	$IF [P11 == 1] * [P12 >=5]
+	Logic OR	$IF [P21 != 0] + [P22 == 8]

Each condition should go between brackets, otherwise, an undesired comparison may be done due to the priority between operators.

Boolean constants

TRUE	True	$IF V.S.VAR == TRUE
FALSE	Not true	$IF V.S.VAR == FALSE

Trigonometric

SIN[...]	Sine	P1 = SIN[30]	P1 = 0.5
COS[...]	Cosine	P2 = COS[30]	P2 = 0.866
TAN[...]	Tangent	P3 = TAN[30]	P3 = 0.5773
ASIN[...]	Arc-sine	P4 = ASIN[1]	P4 = 90
ACOS[...]	Arc-cosine	P5 = ACOS[1]	P5 = 0
ATAN[...]	Arc-tangent	P6 = ATAN[1]	P6 = 45
ARG[...]	Arctangent y/x	P7=ARG[-1,1]	P7=225

In these type of functions the following must be borne in mind:

- In the "TAN" function, the argument cannot take the values ...-90º, 90º, 270º...
- In the "ASIN" and "ACOS" functions, the argument must always be within ±1.
- There are two functions to calculate the arctangent:

  "ATAN" It returns the result between ±90º.

  "ARG" It returns the result between 0º and 360º.

Mathematical

ABS[...]	Absolute value	P1 = ABS[-10]	P1 = 10
SQR[...]	Square function	P2 = SQR[4]	P2 = 16
SQRT[...]	Square root	P3 = SQRT[16]	P3 = 4
LOG[...]	Decimal logarithm	P4 = LOG[100]	P4 = 2
LN[...]	Neperiam logarithm	P5 = LN[100]	P5 = 4.6051
EXP[...]	"e" function	P6 = EXP[1]	P6 = 2.7182
DEXP[...]	Decimal exponent	P6 = DEXP[2]	P7 = 100

In these type of functions the following must be borne in mind:

- In the "LN" and "LOG" functions, the argument must be grater than zero.
- In the "SQRT" function, the argument must be positive.

Other functions

INT[...]	Returns the integer	P1 = INT[4.92]	P1 = 4
FRACT[...]	Returns decimal portion	P2 = FRACT[1.56]	P2 = 0.56
ROUND[...]	Rounds up or down to the nearest integer	P3 = ROUND[3.12] P4 = ROUND[4.89]	P3 = 3 P4 = 5
FUP[...]	Returns the integer plus one. (If the number is an integer, it returns it)	P5 = FUP[3.12] P6 = FUP[9]	P5 = 4 P6 = 9
EXIST[...]	It checks whether the selected variable or parameter exists or not	$IF EXIST[P1] $IF EXIST[P3] == FALSE	

In the "EXIST" function, programming "$IF EXIST[P1] == TRUE" is the same as programming "$IF EXIST[P1]".

1.5.3 Expressions

An expression is any valid combination of operators, constants, parameters and variables.

The priorities of the operators and the way they can be associated determine how these expressions are calculated:

Priority from highest to lowest	They are associated
Functions, - (change sign)	from right to left.
** (exponent), MOD (remainder)	from left to right.
* (multiplication, logic AND), / (division)	from left to right.
+ (suma, OR lógico), - (resta)	from left to right.
Relational operators	from left to right.
& (AND),^ (XOR)	from left to right.
| (OR)	from left to right.

Brackets should be used in order to clarify the order in which the expression is to be evaluated. Using redundant or additional brackets will neither cause errors nor slow down the execution.

P3 = P4/P5 - P6 * P7 - P8/P9

P3 = [P4/P5] - [P6 * P7] - [P8/P9]

Arithmetic

Their result is a numerical value. They consist of a combination of arithmetic and binary operators with constants, parameters and variables.

This type of expressions may also be used to assign values to parameters and variables:

P100 = P9   P101 = P[P7]   P102 = P[P8 + SIN[P8*20]]

P103 = V.G.TOOL

V.G.FIXT[1].X=20   V.G.FIXT[1].Y=40   V.G.FIXT[1].Z=35

Relational

Their result is a TRUE or a FALSE. They combine relational and logic operators with arithmetic expressions, constants, parameters and variables.

... [P8==12.6] ...

It compares if the value of P8 is equal to 12.6.

... ABS[SIN[P4]] > 0.8 ...

It compares if the absolute value of the sine of P4 is greater than 0.8.

... [[P8<=12] + [ABS[SIN[P4]] >=0.8] * [V.G.TOOL==1]] ...

2 MACHINE OVERVIEW

2.1 Axis nomenclature

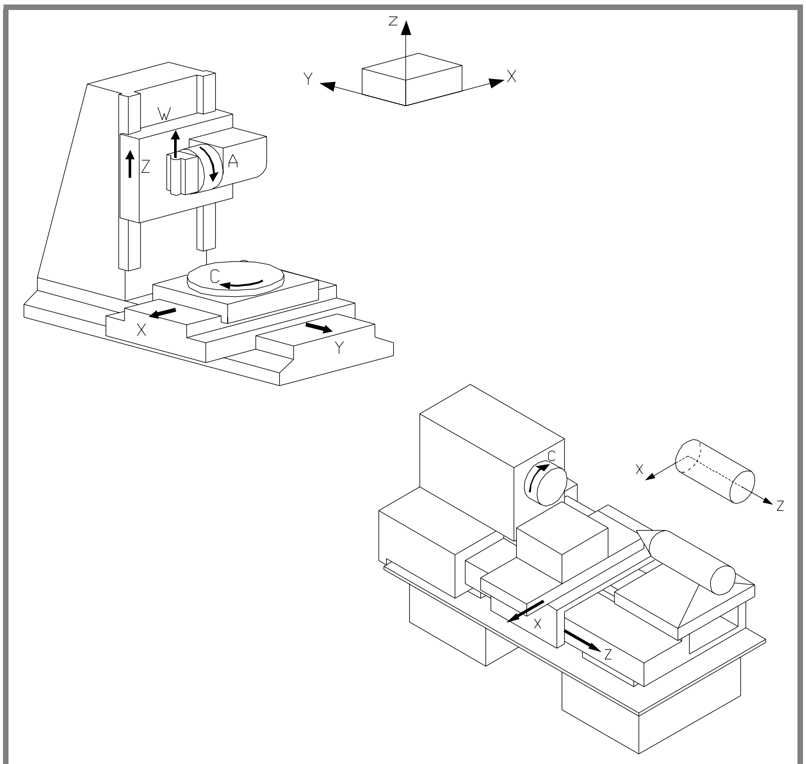
With this CNC, the manufacturer may select up to 28 axes (that must be properly defined as linear, rotary, etc. by setting machine parameters), without no limitation as how to program them and they may all be interpolated at the same time.

The DIN 66217 standard denomination for the axes is:

X-Y-Z	Main axes of the machine. The X-Y axes form the main work plane whereas the Z axis is parallel to the main axis of the machine and perpendicular to the XY plane.
U-V-W	Auxiliary axes, parallel to X-Y-Z respectively.
A-B-C	Rotary axes, on X-Y-Z respectively.

However, the machine manufacturer may call the axes differently.

As an option, the name of the axes may be followed by a number between 1 and 9 (X1, X3, Y5, A8...).

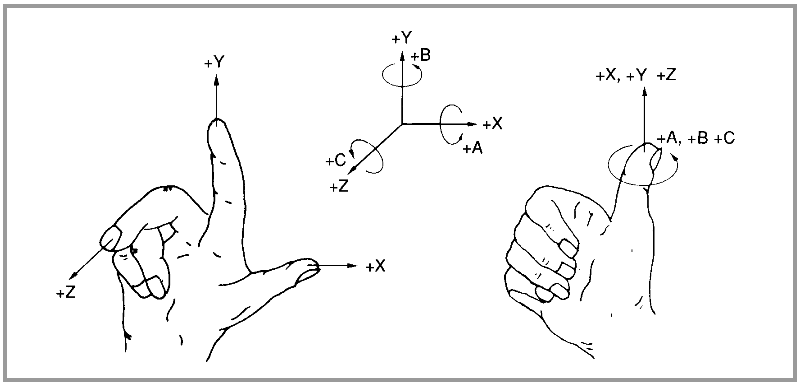


Axis nomenclature on different machines.

Right-hand rule

The direction of the X-Y-Z axes can easily be remembered using the right-hand rule (see the drawing below).

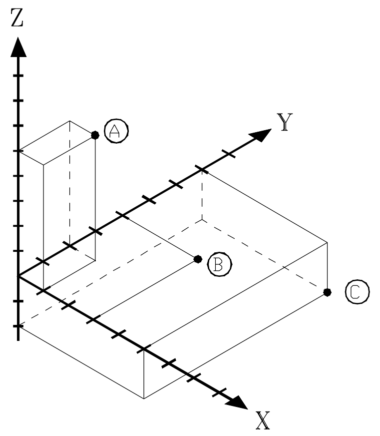
On rotary axes, the positive turning direction is determined by the direction pointed by your fingers when holding the rotary axis with your hand while your thumb points in the positive direction of the linear axis.



2.2 Coordinate system

Since one of the CNC's purposes is to control the movement and positioning of the axes, a coordinate system is required that permits defining the position of the various target (destination) points in the plane (2D) or in space (3D).

The main coordinate system is formed by the X-Y-Z axes. These axes are perpendicular to each other and they meet at the origin point used as reference for the various points.



P	(X,Y,Z)
A	(1,2,5)
B	(3,4,0)
C	(5,7,-2)

The position of a point "P" in the plane or in space is defined by its coordinates on the various axes.

Other types of axes such as auxiliary and rotary axes may also be part of the coordinate system.

2.3 Reference systems

A machine may use the following reference systems.

- Machine reference system.

  It is the coordinate system of the machine and it is set by the manufacturer of the machine.

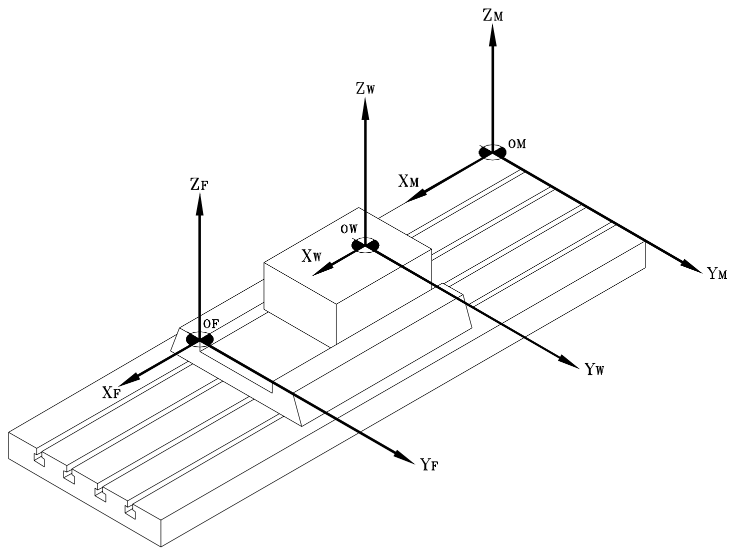
- Fixture reference system.

  It establishes a coordinate system associated with the fixtures being used. It is activated by program and may be set by the operator in any position of the machine.

  When the machine has several fixtures, each one may have its own reference system associated with it.

- Part reference system (datum point).

  It establishes a coordinate system associated with the part being machined. It is activated by program and may be set by the operator anywhere on the part.



X_M Y_M Z_M	*Machine reference system.*
X_F Y_F Z_F	*Fixture reference system.*
X_W Y_W Z_W	*Part reference system (datum point).*

2.3.1 Origins of the reference systems

The position of the different reference systems is determined by their respective origin points.

O_M **Machine zero**

It is the origin point of the machine reference system, set by the machine manufacturer.

O_F **Fixture zero**

It is the origin point of the fixture reference system being used. Its position is defined by the operator by using the "fixture offset" and is referred to machine zero.

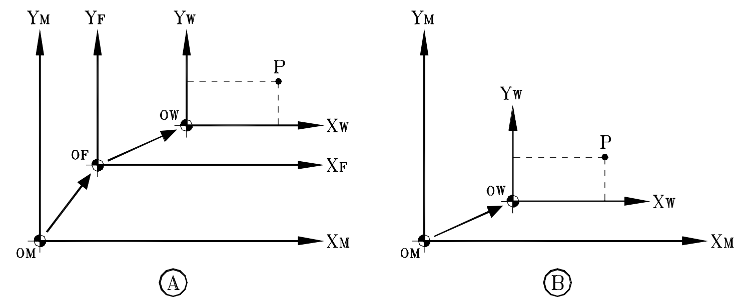
The "fixture offset" may be set by program or from the CNC's front panel, as described in the Operating Manual.

O_W **Part zero**

It is the origin point of the reference system of the part (workpiece). Its position is set by the operator using the "zero offset" and is referred:

- To the fixture offset, if the fixture reference system is active. When changing the fixture reference system, the CNC updates the part zero position by referring to the new fixture zero point.
- To the machine zero point (home), if the fixture reference system is NOT active. When activating the fixture reference system, the CNC updates the part zero position by referring it to the fixture zero point.

The "zero offset" may be set from the program or from the CNC front panel as described in the Operating Manual.



Zero offset when:
***(A)** The fixture reference system is deactivated.*
***(B)** The fixture reference system is activated.*

2.4 Home search

2.4.1 Definition of "Home search"

It is the operation used to synchronize the system. This operation must be carried out when the CNC loses the position of the origin point (e.g. by turning the machine off).

In order to perform the "Home search", the machine manufacturer has set particular points of the machine; the machine zero and the machine reference point.

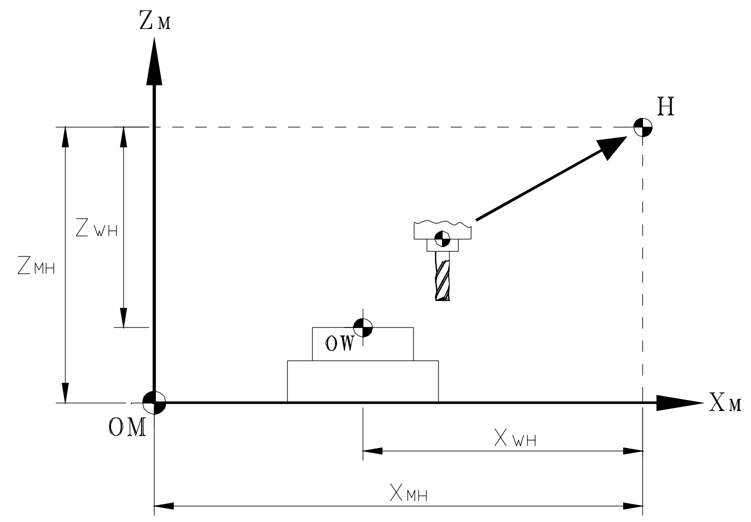
- Machine zero (home).

  It is the origin point of the machine reference system.

- Machine reference point.

  It is the physical point where the system is synchronized (except when the machine uses I_0 distance-coded reference marks or absolute feedback). It may be located anywhere on the machine.

When "searching home", the axes move to the machine reference point and the CNC assumes the coordinate values assigned to that point by the machine manufacturer, referred to machine zero. When using I_0 distance-coded reference marks or absolute feedback, the axes will only move the distance necessary to verify their position.



O_M	Machine zero (home).
O_W	Part zero.
H	Machine reference point.
X_{MH} Y_{MH}	Coordinates referred to machine reference system.
X_{WH} Y_{WH}	Coordinates referred to the part reference system.

When programming a "Home search", neither the fixture offsets nor the zero offsets are canceled; therefore, the coordinates are displayed in the active reference system.

On the other hand, if "Home search" is carried out one axis at a time in JOG mode (not in MDI), the active offsets are canceled and the coordinates being displayed are referred to machine zero.

2.4.2 "Home search" programming

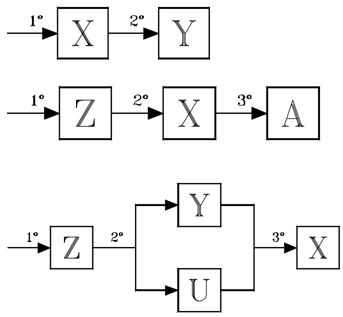
When programming a "Home search", the axes are homed sequentially in the order set by the operator. All the axes need not be included in the "Home search", only those being homed.

The "Home search" is programmed using the G74 function followed by the axes to be homed and the number indicating their homing order. If the same order number is assigned to several axes, those axes start homing at the same time and the CNC waits for all of them to end before homing the next one.

G74 X1 Y2

G74 X2 Z1 A3

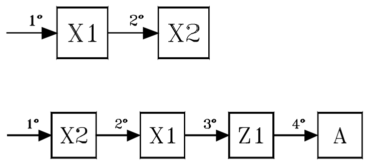
G74 Z1 Y2 X3 U2



When having numbered axes, they may be defined together with the other ones by assigning them the order number as follows.

G74 X1=1 X2=2

G74 X1=2 X2=1 A4 Z1=3



Spindle home search

When using a position controlled spindle, it may be included in the "Home search" like any other axis. In this case, the spindle home search is always carried out together with the first axis regardless of the order in which it has been defined.

Using an associated subroutine

If the machine manufacturer has associated a home-search subroutine to the G74 function, this function may be programmed alone in the block and the CNC will automatically execute the associated subroutine [G.M.P. "REFPSUB (G74)"].

When using a subroutine, the "Home search" is carried out exactly as described earlier.

COORDINATE SYSTEM

3

3.1 Plane selection (G17/G18/G19/G20)

Selecting the planes determines which axes form the work plane/trihedron and which axis will correspond to the tool's longitudinal axis. Plane selection is required to execute operations like:

- Circular and helical interpolations.
- Corner chamfering and rounding.
- Tangential entries and exits.
- Machining canned cycles.
- Tool radius and length compensation.

These operations, except tool length compensation, can only be executed in the active work plane. Tool length compensation, on the other hand, can only be applied on the longitudinal axis.

Programming

The work planes may be selected by program using these functions:

G17	Main plane X-Y; longitudinal axis and perpendicular Z.
G18	Main plane Z-X, longitudinal axis and perpendicular Y.
G19	Main plane Y-Z, longitudinal axis and perpendicular X.
G20	Work plane/trihedron and longitudinal axis.

And using the instruction:

#TOOL AXLongitudinal axis selection.

Considerations about functions G17, G18 and G19 and the channels

When in these functions we mention the X, Y and Z axes, it does not mean that the axes must have these names; it is a convention to refer to the first three axes of the channel.

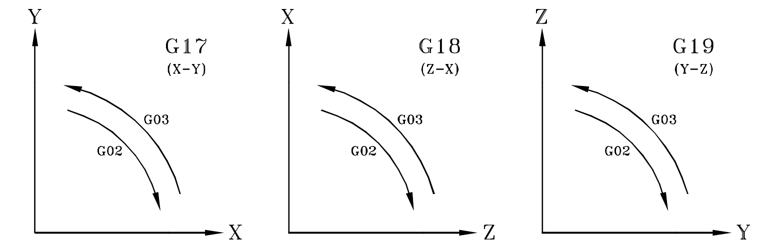
Therefore, when selecting G17, G18 or G19, it means the following.

G17	Main plane formed by the first and second axes of the channel. The perpendicular axis (helical) or longitudinal axis corresponds to the third axis of the channel.
G18	Main plane formed by the third and first axes of the channel. The perpendicular axis (helical) or longitudinal axis corresponds to the second axis of the channel.
G19	Main plane formed by the second and third axes of the channel. The perpendicular axis (helical) or longitudinal axis corresponds to the first axis of the channel.

The perpendicular (helical) axis is the one onto which the helical interpolations are carried out. Longitudinal axis is the one onto which the tool length compensation is applied. When programming G17, G18 and G19 the perpendicular and longitudinal axes are the same.

Main planes and axes

The main planes may be selected by program using functions G17, G18 and G19. The main planes are defined by two of the first axes of the channel. The third axis corresponds to the longitudinal axis which, for functions G17, G18 and G19, coincides with the perpendicular axis.



Circular interpolation in the main work planes.

These functions may be programmed anywhere in the program and they don't have to go alone in the block.

Properties of the functions

Functions G17, G18, G19 and G20 are modal and incompatible with each other.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G17 or G18 as set by the machine manufacturer [G.M.P. "IPLANE"].

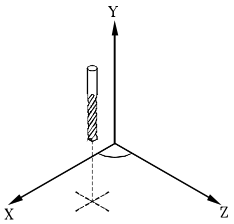
3.1.1 Work plane programming by two directions (G20)

Besides the main planes, any other work plane/trihedron formed by the first three axes of the channel may be defined using function G20.

Programming

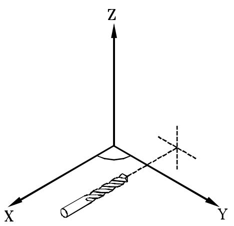
The work plane is defined by selecting the abscissa and ordinate axes, the perpendicular axis and the longitudinal axis of the tool. It is selected by assigning one of the following parameters to the axes programmed with G20.

"1"	To the 1st axis of the work plane (abscissa axis).
"2"	To the 2nd axis of the work plane (ordinate axis).
"3"	To the longitudinal axis of the tool and also perpendicular (helical) axis of the plane if parameter 5 is not defined.
"4"	Reserved.
"5"	To the axis perpendicular to the work plane; if not defined, it is the same as the longitudinal axis. Only when the longitudinal tool axis is the same as the abscissa or ordinate axis.



G20 X1 Z2 Y3

- X It is the first main axis.
- Z It is the second main axis.
- Y It is the longitudinal axis of the tool and the perpendicular axis.



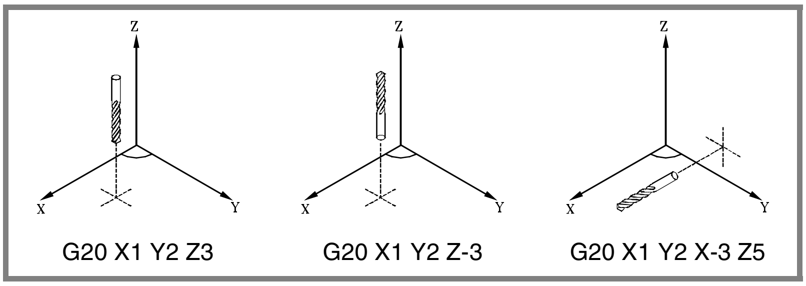
G20 X1 Y2 X3 Z5

- X It is the first main axis and the longitudinal axis.
- Y It is the second main axis.
- Z It is the third main axis or perpendicular axis.

Considerations

When selecting the longitudinal axis with G20, tool orientation may be established according to the programmed sign.

- If the parameter to select the longitudinal axis is positive, the tool is positioned in the positive direction of the axis.
- If the parameter to select the longitudinal axis is negative, the tool is positioned in the negative direction of the axis.



3.1.2 Longitudinal tool axis selection

The longitudinal axis of the tool may be selected using the instruction "#TOOL AX". This instruction allows to select any machine axis as the new longitudinal axis.

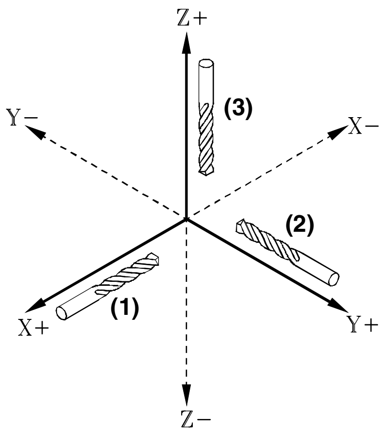
Programming

The longitudinal axis of the tool is defined using the instruction "#TOOL AX [<axis><sign>]", where:

- The <axis> parameter sets the new longitudinal axis of the tool.
- The <sign> parameter indicates tool orientation.

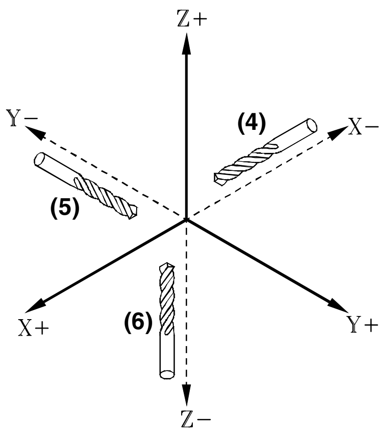
+	Positive if the tool positions in the positive direction of the axis.
-	Negative if the tool positions in the negative direction of the axis.

Both parameters MUST be programmed.



Positive orientation

(1)	#TOOL AX [X+]
(2)	#TOOL AX [Y+]
(3)	#TOOL AX [Z+]



Negative orientation

(4)	#TOOL AX [X-]
(5)	#TOOL AX [Y-]
(6)	#TOOL AX [Z-]

3.2 Programming in millimeters (G71) or in inches (G70)

The displacements and feedrates of the axes may be defined in millimeters or in inches. The unit system may be selected by program using the following functions:

G70	Programming in inches.
G71	Programming in millimeters.

Both functions may be programmed anywhere in the program; they do not have to go alone in the block.

Operation

After executing one of these functions, the CNC assumes that unit system for the following blocks. If none of these functions is programmed, the CNC uses the unit system set by machine manufacturer [G.M.P. "INCHES"].

When changing the unit system, the CNC converts the currently active feedrate into the new unit system.

```
...
G01 G71 X100 Y100 F508      (Programming in millimeters.
                             Feedrate: 508 mm/minute)
...
G70                         (Change unit system.
                             Feedrate: 20 inches/minute)
...
```

Properties of the functions

Functions G70 and G71 are modal and incompatible with each other.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G70 or G71 as set by the machine manufacturer [G.M.P. "INCHES"].

3.3 Absolute (G90) or incremental (G91) coordinates

The coordinates of the various points may be defined in absolute coordinates (referred to the active origin point) or incremental coordinates (referred to the current position). The type of coordinates may be selected by program using the following functions:

G90	Programming in absolute coordinates.
G91	Programming in incremental coordinates.

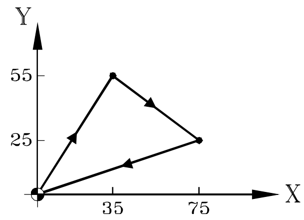
Both functions may be programmed anywhere in the program; they do not have to go alone in the block.

Operation

After executing one of these functions, the CNC assumes that programming mode for the following blocks. If none of these functions is programmed, the CNC uses the work mode selected by machine manufacturer [G.M.P. "ISYSTEM"].

Depending on the active work mode (G90/G91), the coordinates of the points are defined as follows:

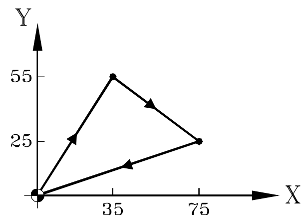
- When programming in absolute coordinates (G90), the coordinates of the point are referred to the current origin of the coordinate system, usually the part zero.




```
N10 G00 G71 G90 X0 Y0
N20 G01 X35 Y55 F450
N30 X75 Y25
N40 X0 Y0
N50 M30
```

Programming in absolute coordinates.

- When programming in incremental coordinates (G91), the coordinates of the point are referred to the current tool position. The preceding sign indicates the direction of the movement.




```
N10 G00 G71 G90 X0 Y0
N20 G01 G91 X35 Y55 F450
N30 X40 Y-30
N40 X-75 Y-25
N50 M30
```

Programming in incremental coordinates.

Properties of the function

Functions G90 and G91 are modal and incompatible with each other.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G90 or G91 as set by the machine manufacturer [G.M.P. "ISYSTEM"].

3.4 Programming in radius (G152) or in diameters (G151)

The following functions are oriented to lathe type machines. Programming in diameters is only available on the axes allowed by the machine manufacturer (DIAMPROG=YES).

Programming in radius or diameters may be selected by program with these functions:

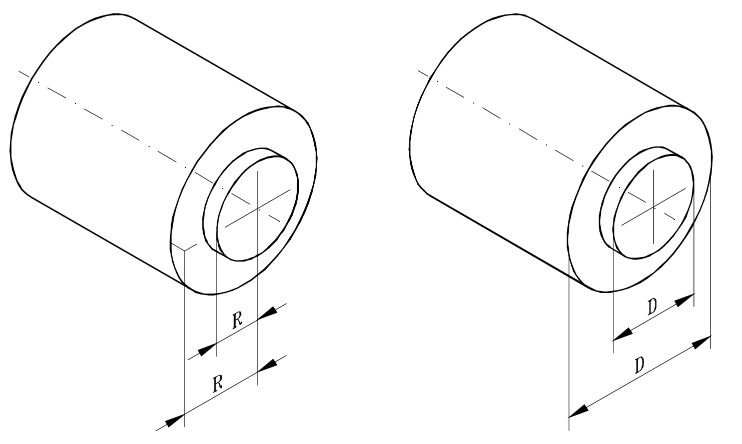
G151 Programming in diameters.

G152 Programming in radius.

These functions may be programmed anywhere in the program and they don't have to go alone in the block.

Operation

After executing one of these functions, the CNC assumes that programming mode for the following blocks.



Programming in radius. Programming in diameters.

When switching programming modes, the CNC changes the way it displays the coordinates of the corresponding axes.

Properties of the function

Functions G151 and G152 are modal and incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G151 if machine parameter DIAMPROG of any of the axes is set to YES.

3.5 Coordinate programming

3.5.1 Cartesian coordinates

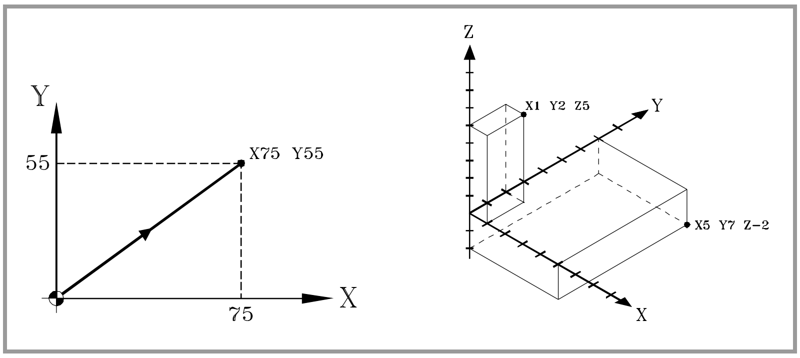
Coordinates are programmed according to a Cartesian coordinate system. This system consists of two axes in the plane and three or more in space.

Definition of position values

The position of a point in this system is given by its coordinates in the different axes. The coordinates are programmed in absolute or incremental coordinates and in millimeters or inches.

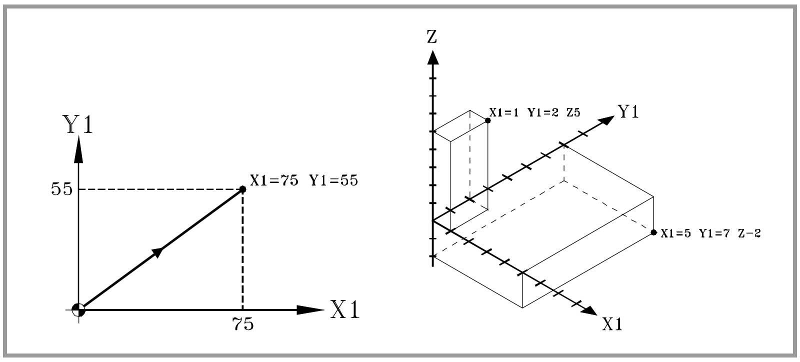
Standard axes (X...C)

The coordinates are programmed with the axis name followed by the coordinate value.



Numbered axes (X1...C9)

If the axis name is like X1, Y2... the "=" sign must be included between the axis name and the coordinate.



3.5.2 Polar coordinates

When having circular elements or angular dimensions, polar coordinates may be more convenient to express the coordinates of the various points in the plane.

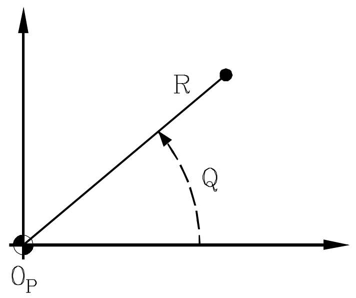
This type of coordinates requires a reference point referred to as "polar origin" that will be the origin of the polar coordinate system.

Definition of position values

The position of the various points is given by defining the radius "R" and the angle "Q" as follows:

Radius It will be the distance between the polar origin and the point.

Angle It will be the one formed by the abscissa axis and the line joining the polar origin with the point.



R	Radius
Q	Angle
O_P	Polar origin

The radius may be given in mm or in inches whereas the angle is given in degrees.

Both values may be given in either absolute (G90) or incremental (G91) coordinates.

- When working in G90, the "R" and "Q" values will be absolute. The value assigned to the radius must always be positive or zero.
- When working in G91, the "R" and "Q" values will be incremental. Although negative "R" values may be programmed, when programming in incremental coordinates, the resulting value assigned to the radius must always be positive or zero.

When programming a "Q" value greater than 360º, the module will be assumed after dividing it by 360. Thus, Q420 is the same as Q60 and Q-420 is the same as Q-60.

Polar origin preset

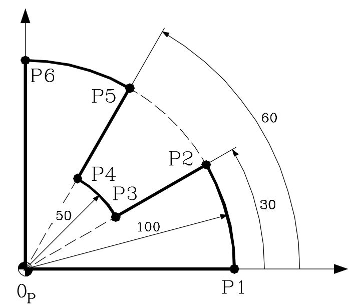
The "polar origin" may be selected from the program using function G30. If not selected, it assumes as "polar origin" the origin of the active reference system (part zero). Ver el capítulo *"4 Origin selection"*.

The selected "polar origin" is modified in the following instances:

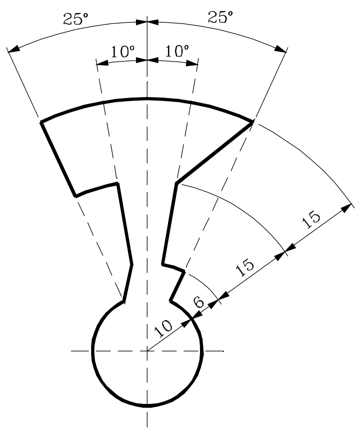
- When changing the work plane, the CNC assumes the part zero as the new "polar origin".
- On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes the part zero as the new polar origin.

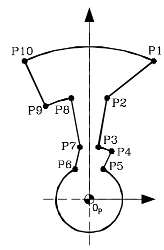
Examples

Point definition in polar coordinates.



	R	Q
P1	100	0
P2	100	30
P3	50	30
P4	50	60
P5	100	60
P6	100	90





	R	Q
P1	46	65
P2	31	80
P3	16	80
P4	16	65
P5	10	65

	R	Q
P6	10	115
P7	16	100
P8	31	100
P9	31	115
P10	46	115

ORIGIN SELECTION

4

With this CNC, it is possible to program movements in the machine reference system or apply offsets in order to use reference systems referred to the fixtures or the part without having to change the coordinates of the different points of the part in the program.

There are three types of offsets:

- Fixture offset.
- Zero offset.
- PLC offset.

Several offsets may be active at the same time. In this case, the coordinate system being used will be defined by the sum of the active offsets.

Fixture offset

A fixture offset is defined as the distance between the machine reference zero and the fixture zero.

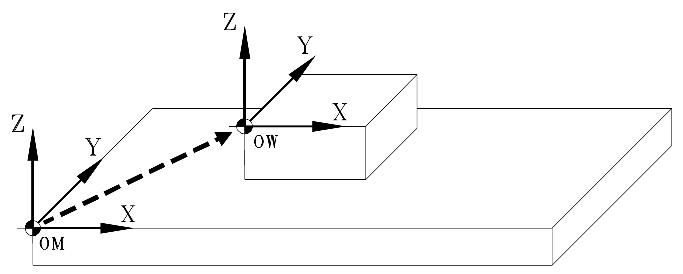
On machines using several fixtures, this offsets allows selecting the particular fixture to be used.

Zero offset

A zero offset is defined as the distance between the fixture zero and the part zero. If the fixture zero is not active (no fixture offset), the zero offset is measured from machine zero.

The zero offset may be set in two ways:

- By presetting a coordinate, the CNC assumes the programmed coordinates as the current position of the axes.
- By using absolute or incremental zero offsets, the CNC assumes the new part zero set by the selected offset.



Zero offset when the fixture offset is zero.

O_M : Machine zero (home).

O_W : Part zero.

PLC offset

Special offset handled by the PLC that is used to correct the deviations due to dilatations, etc.

This offset is always applied, even when programming with respect to machine zero.

4.1 Programming with respect to machine zero

Machine zero is the origin of the machine reference system. When executing a movement referred to machine zero, the CNC ignores the active offsets (except the PLC offset), the kinematics and cartesian transformations; therefore, the movement is carried out in the machine reference system. Once the movement has ended, the CNC restores the offsets, kinematics and cartesian transformations that were active.

Tool radius and length compensation is also canceled during the movements referred to machine zero.

When moving with respect to machine reference zero, function G70 or G71 programmed by the user is ignored. The movements are carried out in the units (millimeters or inches) set by the OEM (units assumed by the CNC on power-up).

Movements referred to machine zero are programmed using the instructions #MCS and #MCS ON/OFF.

The programmed movements do not admit polar coordinates, nor other kinds of transformations such as mirror image, coordinate (pattern) rotation or scaling factor. While the #MCS function is active, functions for setting a new origin such as G92, G54-G59, G158, G30, etc. are not admitted either.

#MCS instruction

This instruction may be added to any block containing a movement so it is executed in the machine reference system.

G00 X30 Y30	
G92 X0 Y0	(Coordinate preset)
G01 X20 Y20	
#MCS X30 Y30	(Movement referred to machine zero. Offsets canceled)
G01 X40 Y40	(Offsets restored)
G01 X60 Y60	
M30	

#MCS ON and #MCS OFF instructions

The #MCS ON and #MCS OFF instructions activate and deactivate the machine reference system; therefore, the movements programmed between them are executed in the machine reference system.

G92 X0 Y0	(Coordinate preset)
G01 X50 Y50	
#MCS ON	(Beginning of programming referred to machine zero)
G01 ...	
G02 ...	
G00 ...	
#MCS OFF	(End of programming referred to machine zero. Offsets restored)

Both instructions must be programmed alone in the block.

4.2 Fixture offset

With fixture offsets, it is possible to select the fixture system to be used (when having more than one fixture). When applying a new fixture offset, the CNC assumes the point set by the new selected fixture as the new fixture zero.

Defintion

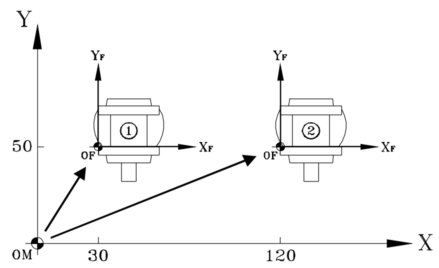
In order to apply a fixture offset, it must have been previously set. To do that, the CNC has a table where the operator may define up to 10 different fixture offsets. The table data may be defined:

- Manually from the CNC's front panel (as described in the Operating Manual).
- By program, assigning the corresponding value (of the "n" offset and of the "Xn" axis) to the "V.A.FIXT[n].Xn" variable.

Activation

Once the fixture offsets have been defined in the table, they may be activated via program by assigning to the "V.G.FIX" variable, the offset number to be applied.

Only one fixture offset may be active at a time; therefore, when applying a fixture offset, it will cancel the previous one. Assigning a value of "V.G.FIX=0" will cancel the active fixture offset.



	X	Y
V.G.FIX=1	30	50
V.G.FIX=2	120	50

```
...
N100 V.A.FIXT[1].X=30    V.A.FIXT[1].Y=50
N110 V.A.FIXT[2].X=120    V.A.FIXT[2].Y=50
...
N200 V.G.FIX=1          (It applies the 1st fixture offset)
N210 ...                (Programming at fixture 1)
N300 V.G.FIX=2          (It applies the 2nd fixture offset)
N310 ...                (Programming at fixture 2)
N400 V.G.FIX=0          (Cancel fixture offset. No fixture system is active)
...
```

Considerations

A fixture offset, by itself, does not cause any axis movement.

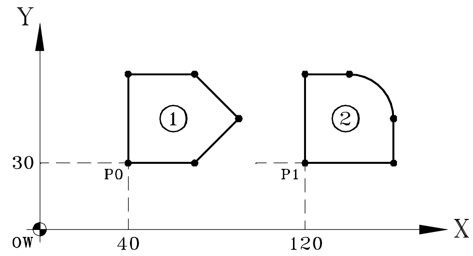
Properties

On power-up, the CNC assumes the fixture offset that was active when the CNC was turned off. On the other hand, the fixture offset is neither affected by functions M02 and M30 nor by RESETTING the CNC.

4.3 Coordinate preset (G92)

Coordinate presetting is done with function G92 and it may be applied onto any axis of the machine.

When presetting a coordinate, the CNC interprets that the axis coordinates programmed after the G92 set the current position of the axes. The rest of the axes that have not been defined with G92 are not affected by the preset.



...	
N100 G90 G01 X40 Y30	(Positioning at P0)
N110 G92 X0 Y0	(Presetting P0 as part zero)
...	(Machining of profile 1)
N200 G90 G01 X80 Y0	(Positioning at P1)
N210 G92 X0 Y0	(Presetting P1 as part zero)
...	(Machining of profile 2)
N300 G92 X120 Y30	(Recover O_W as part zero)
...	

Considerations

A coordinate preset, by itself, does not cause any axis movement.

When homing an axis in JOG mode, the preset for that axis is canceled.

Properties of the function

G92 is modal, the preset values remain active until the preset is canceled (with another preset, a zero offset or with G53).

On power-up, the CNC assumes the coordinate preset that was active when the CNC was turned off. On the other hand, the coordinate preset is neither affected by functions M02 and M30 nor by RESETTING the CNC.

4.4 Zero offsets (G54-G59/G159)

Using zero offsets, it is possible to place the part zero in different positions of the machine. When applying a zero offset, the CNC assumes as the new part zero the point defined by the selected zero offset.

Defintion

In order to apply a zero offset, it must have been previously defined. To do that, the CNC has a table where the operator may define up to 20 different zero offsets. The table data may be defined:

- Manually from the CNC's front panel (as described in the Operating Manual).
- By program, assigning the corresponding value (of the "n" offset and of the "Xn" axis) to the "V.A.ORGT[n].Xn" variable.

Activation

Once the zero offsets have been defined in the table, they may be activated by program using the following functions:

G54 to G59 - Absolute zero offset

To apply the first six zero offsets in the table. They are the same as programming G159=1 through G159=6.

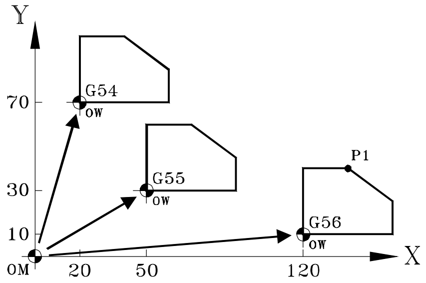
G54	applies the 1st zero offset (G159=1).
G59	applies the 6th zero offset (G159=6).

G159 - Additional absolute zero offsets

To apply any zero offset defined in the table.

The first six zero offsets are the same as programming G54 through G59.

G159=2	applies the 2th zero offset.
G159=11	applies the 11th zero offset.



	X	Y
G54 (G159=1)	20	70
G55 (G159=2)	50	30
G56 (G159=3)	120	10

```
...
N100 V.A.ORGT[1].X=20    V.A.ORGT[1].Y=70
N110 V.A.ORGT[2].X=50    V.A.ORGT[2].Y=30
N100 V.A.ORGT[3].X=120   V.A.ORGT[3].Y=10
...
N100 G54            (It applies the 1st zero offset)
...
N200 G159=2         (It applies the 2nd zero offset)
...
N300 G56 X20 Y30    (It applies the 3rd zero offset. The axes move to
                    point X20 Y30 (point P1) referred to the 3rd origin)
...
```

Only one zero offset may be active at a time; therefore, when applying a zero offset, the previous one will be canceled. When programming G53, the zero offset currently active will be canceled.

The function corresponding to the selected zero offset may be programmed in any block of the program. When added to a block with path information, the zero offset will be applied before executing the programmed movement.

Considerations

A zero offset, by itself, does not cause any axis movement.

When homing an axis in JOG mode, the absolute zero offset for that axis is canceled.

Properties of the functions

Functions G54, G55, G56, G57, G58, G59 and G159 are modal and incompatible with each other and with G53 and G92.

On power-up, the CNC assumes the zero offset that was active when the CNC was turned off. On the other hand, the zero offset is neither affected by functions M02 and M30 nor by RESETTING the CNC.

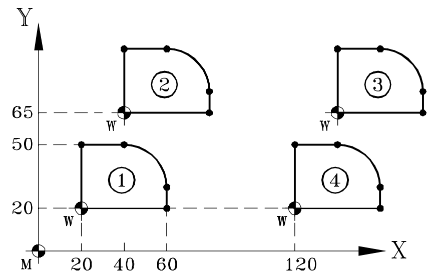
4.4.1 Incremental zero offset (G158)

When applying an incremental zero offset, the CNC adds it to the absolute zero offset active at a time.

Programming

Incremental zero offset are defined by program using function G158 followed by the values of the zero offset to be applied on each axis.

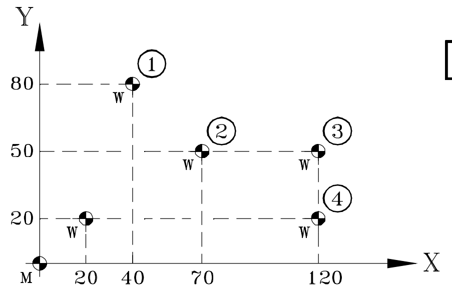
To cancel the incremental zero offset, program function G158 without axes in the block. To only cancel the incremental zero offset on particular axes, program an incremental offset of "0" on each of those axes.



	X	Y
G54 (G159=1)	20	20
G55 (G159=2)	120	20

...	
N100 G54	(It applies the 1st zero offset)
...	(Machining of profile 1)
N200 G158 X20 Y45	(Apply incremental zero offset)
...	(Machining of profile 2)
N300 G55	(It applies the 2nd zero offset. G158 stays active)
...	(Machining of profile 3)
N400 G158	(Cancel incremental zero offset. G55 stays active)
...	(Machining of profile 4)

Only one incremental zero may be active at a time for each axis; therefore, applying an incremental zero offset on an axis cancels the one that was active on that axis. The offsets on the rest of the axes are not affected.



	X	Y
G54 (G159=1)	20	20

```
...
N100 G54            (Apply absolute zero offset)
N200 G158 X20 Y60   (It applies the 1st incremental zero offset)
N300 G158 X50 Y30   (It applies the 2nd incremental zero offset)
N400 G158 X100      (It applies the 3st incremental zero offset)
N500 G158 Y0        (It applies the 4nd incremental zero offset)
N600 G158 X0        (Cancel incremental zero offset)
...
```

The incremental zero offset is not canceled after applying a new absolute zero offset (G54-G59 or G159).

Considerations

An incremental zero offset, by itself, does not cause any axis movement.

When homing an axis in JOG mode, the incremental zero offset for that axis is canceled.

Properties of the function

Function G158 is modal.

On power-up, the CNC assumes the incremental zero offset that was active when the CNC was turned off. On the other hand, the incremental zero offset is neither affected by functions M02 and M30 nor by RESETTING the CNC.

4.4.2 Excluding axes in the zero offset (G157)

Excluding axes allows to select on to which axes the next absolute zero offset will not be applied. After applying the zero offset, the programmed axis exclusion is canceled and it has to be programmed again in order to apply it again.

Activation

Axis exclusion must be programmed using function G157 followed by the axes and the value indicating whether that axis is excluded (<axis>=1) or not (<axis>=0).

The exclusion may also be activated by programming only the axes affected by the exclusion after function G157.

The exclusion and the zero offset may be programmed in the same block. In that case, the exclusion will be activated before applying the zero offset.

...	
G55	(It applies the 2nd zero offset on all the axes)
...	
G157 X Z	(Activation of the exclusion on the X-Z axes)
G57	(It applies the 4th zero offset, except on the X-Z axes. Those axes keep the previous zero offset)
...	
G159=8	(It applies the 8nd zero offset on all the axes)
...	
G59 G157 Y	(It applies the 6th zero offset, except on the Y axis. That axis keeps the previous zero offset)
...	
G54	(It applies the 1st zero offset on all the axes)
...	

Excluding axes does not affect the active zero offsets. If an axis is excluded, when applying a new zero offset, the CNC maintains the one that was active for that axis.

Considerations

Excluding axes does not affect the coordinate preset or the incremental zero offsets which are always applied on to all the axes. Likewise, neither fixture offsets nor PLC offsets are affected.

Properties of the function

Function G157 is modal and it remains active until an absolute zero offset is applied.

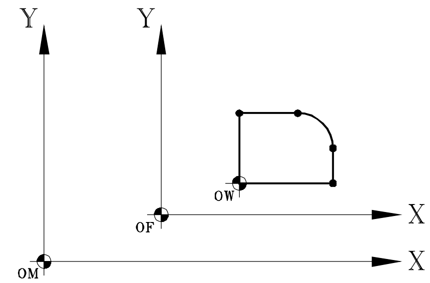
On power-up or after an EMERGENCY, the CNC does not assume any axis exclusion.

4.5 Zero offset cancellation (G53)

Executing function G53 cancels the active zero offset resulting either from a preset (G92) or from a zero offset, including the incremental offset and the defined axis exclusion. It also cancels the zero offset due to a probing operation.

Fixture offsets and PLC offsets are not affected by this function.

Contrary to the #MCS and #MCS ON/OFF instructions that always execute movements referred to machine zero, function G53 allows to execute movements referred to the fixture zero (if it is active).



N10 V.G.FIX=1	(Activate fixture offset. Program with respect to O_F)
N20 G54	(Apply the zero offset. Program with respect to O_W)
N30 #MCS X20 Y20	(Activate machine coordinate system. Program with respect to O_M)
N40 G01 X60 Y0	(Program with respect to O_W)
N50 G53	(Cancel zero offset G54. Program with respect to O_F)

Function G53 may be programmed in any block of the program. When added to a block with path information, the offset or preset is canceled before executing the programmed movement.

Considerations

Function G53, by itself, does not cause any axis movement.

Properties of the function

Function G53 is modal and incompatible with function G92, zero offsets and probing.

4.6 Polar origin preset (G30)

Function G30 may be used to preset any point of the work plane as the new polar origin. If not selected, it assumes as polar origin the origin of the active reference system (part zero).

Programming

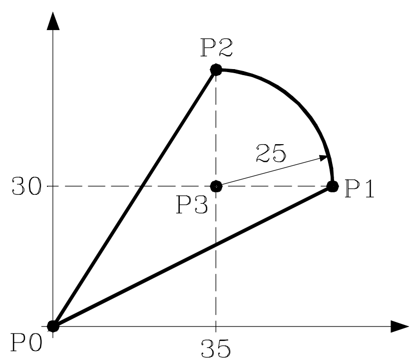
The polar origin preset must be programmed alone in the block. The programming format is "G30 I J", where:

I, J They define the abscissa and ordinate of the new polar origin. They must be defined in absolute coordinates referred to part zero.

When programmed, both parameters must be programmed.

If not programmed, it will assume the current tool position as the polar origin.

Assuming the initial point is X0 Y0:



G30 I35 J30	(Preset P3 as the polar origin)
G90 G01 R25 Q0	(Point P1)
G03 Q90	(Point P2)
G01 X0 Y0	(Point P0)
M30	

Therefore, function G30 may be programmed as follows:

G30 I J It assumes as the new polar origin the point whose abscissa is "I" and ordinate "J" referred to part zero.

G30 The current tool position is assumed as the new polar origin.

Properties of the function

Function G30 is modal. The polar origin stays active until another value is preset or the work plane is changed. When changing the work plane, it assumes the part zero of that plane as the new polar origin.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes the currently selected part zero as the new polar origin.

TECHNOLOGICAL FUNCTIONS 5

5.1 Machining feedrate (F)

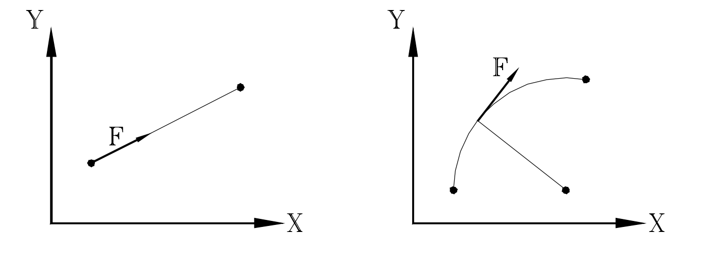
The machining feedrate may be selected by programmed using the "F" code which remains active until another value is programmed. The programming units depend on the active work mode (G93, G94 or G95) and the type of axis being moved (linear or rotary).

It is possible to program using parameters or arithmetic expressions.

Operation

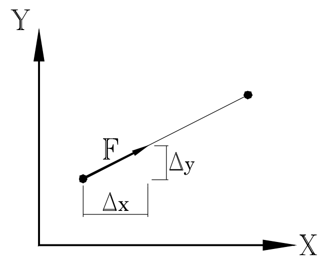
The programmed "F" is effective in movements of linear (G01) or circular interpolations (G02, G03). Movements in G00 (rapid traverse) are executed at the feedrate set by machine manufacturer [A.M.P. "G00FEED"] regardless of the programmed "F" value.

The feedrate is measured along the tool path, either along the straight line (linear interpolations) or along the tangent of the indicated arc (circular interpolations).



Feedrate direction on linear and circular interpolations.

When only the main axes (X-Y-Z) are involved in the interpolation, the relationship between the components of the feedrate on each axis and the programmed "F" is the same as between the displacement of each axis and the resulting programmed displacement.



$$Fx = \frac{F \cdot \Delta x}{\sqrt{((\Delta x)^2 + (\Delta y)^2)}}$$

$$Fy = \frac{F \cdot \Delta y}{\sqrt{((\Delta x)^2 + (\Delta y)^2)}}$$

Feedrate components.

FAGOR

CNC 8070

(SOFT V02.0x)

When rotary axes are involved in the interpolations, the feedrate of these axes is calculated so the beginning and the end of their movement coincides with the beginning and the end of the main axes. If the feedrate calculated for the rotary axis is greater than the maximum allowed, the CNC will adapt the programmed "F" so the rotary axis can turn at its maximum speed.

Feedrate override

The programmed feedrate "F" may be varied between 0% and 200% using the selector switch on the CNC's operator panel or it may be selected by program or by PLC. However, the maximum override is limited by the machine manufacturer [G.M.P. "MAXOVR"].

When making movements in G00 (rapid traverse), the feedrate override percentage will be fixed at 100% or it may be varied between 0% and 100% depending on how the machine manufacturer has set [G.M.P. "RAPIDOVR"].

When carrying out threading operations, the feedrate percentage will be fixed at 100% of the programmed feedrate.

5.2 Feedrate related functions

5.2.1 Feedrate programming units (G93/G94/G95)

The functions related to programming units permit selecting whether mm/minute (inches/minute) or mm/revolution (inches/rev.) are programmed or, instead, the time the axes will take to reach their target position.

Programming

The functions related to programming units are:

G94 Feedrate in millimeters/minute (inches/minute).

G95 Feedrate in millimeters/revolution (inches/revolution).

G93 Machining time in seconds.

These functions may be programmed anywhere in the program and they don't have to go alone in the block.

If the moving axis is rotary, the programming units will be in degrees instead of millimeters or inches as follows:

	Linear axes	Rotary axes
G94	millimeters (inches)/minute	degrees/minute
G95	millimeters (inches)/revolution	degrees/revolution
G93	seconds	seconds

G94 **Feedrate in millimeters/minute (inches/minute).**

After executing G94, the CNC interprets that the feedrates programmed with the "F" code are in millimeters/minute (inches/minute). If the moving axis is rotary, the CNC interprets that the programmed feedrate is in degrees/minute.

G95 **Feedrate in millimeters/revolution (inches/revolution)**

After executing G95, the CNC interprets that the feedrates programmed with the "F" code are in mm/rev (inches/rev) of the master spindle of the channel. If the moving axis is rotary, the CNC interprets that the programmed feedrate is in degrees/revolution.

This function does not affect the movements in G00 which are always executed in millimeters/minute (inches/minute).

G93 **Machining time in seconds**

After executing G93, the CNC interprets that the movements must be carried out in the time period (seconds) indicated by the "F" code.

This function does not affect the movements in G00 which are always executed in millimeters/minute (inches/minute).

Properties of the functions

Functions G93,G94 and G95 are modal and incompatible with each other.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G94 or G95 as set by the machine manufacturer [G.M.P. "IFEED"].

5.2.2 Feedrate blend (G108/G109/G193)

With these functions, it is possible to blend the feedrate between consecutive blocks programmed with different feedrates.

Programming

The functions related to feedrate blending are:

G108 Feedrate blending at the beginning of the block.

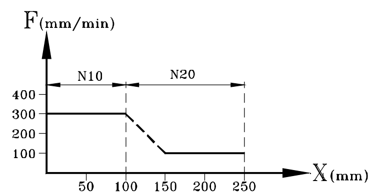
G109 Feedrate blending at the end of the block.

G193 Interpolating the feedrate.

These functions may be programmed anywhere in the program and they don't have to go alone in the block.

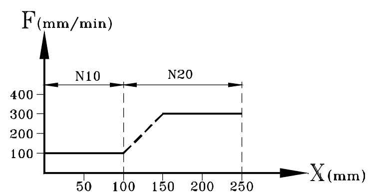
G108 **Feedrate blending at the beginning of the block**

When G108 is active, the adaptation to the new feedrate (by accelerating or decelerating) takes place at the beginning of the next block and the current block ends at the programmed feedrate "F".



N10 G01 G108 X100 F300

N20 X250 F100

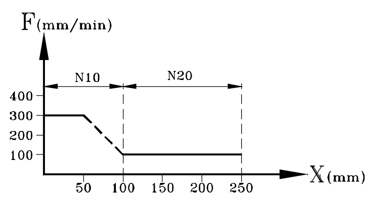


N10 G01 G108 X100 F100

N20 X250 F300

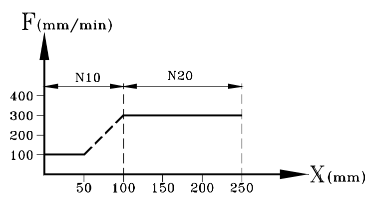
G109 **Feedrate blending at the end of the block**

When programming G109 the adaptation to the new feedrate (by accelerating or decelerating) takes place at the end of the current block so the next block starts executing at its programmed feedrate "F".



N10 G01 G109 X100 F300

N20 X250 F100

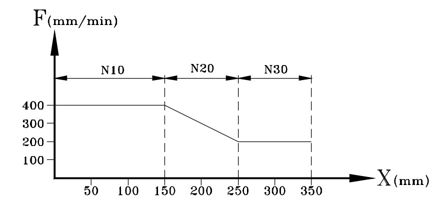


N10 G01 G109 X100 F100

N20 X250 F300

G193 **Interpolating the feedrate**

When programming G193, the adaptation to the new feedrate is interpolated linearly during the movement programmed in the block.




```
N10 G01 X150 F400
N20 G193 X250 F200
N30 X350
```

Considerations

Although the default function is G108 (feedrate blending at the beginning of the block); during a transition from G00 to G01, G02 or G03, the feedrate blend always takes place at the end of the block (G109) where G00 has been programmed.

Feedrate interpolation is only applied when the manufacturer has set the machine to work with linear acceleration [G.M.P. "SLOPETYPE"]. In the rest of the cases, the feedrate is adapted at the beginning of the block (G108).

Function G109 is only applied when the manufacturer has set the machine to operate with either trapezoidal or square-sine (bell shaped) acceleration.

Properties of the functions

Functions G109 and G193 are NOT modal and are incompatible with each other and with G108 (modal).

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G108.

5.2.3 Constant feedrate mode (G197/G196)

With these functions, it is possible to choose whether the feedrate at the tool center is maintained constant while machining or the feedrate at the cutting edge so when working with tool radius compensation, the programmed "F" corresponds to the contact point between the part and the tool.

Programming

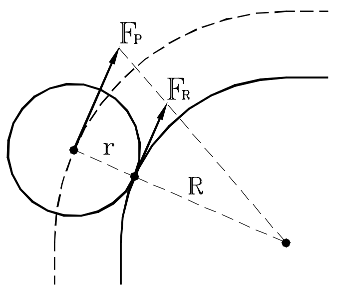
The functions related to the feedrate type are:

G197	Constant tool center feedrate.
G196	Constant cutting point feedrate.

These functions may be programmed anywhere in the program and they don't have to go alone in the block.

G197 **Constant tool center feedrate**

After executing G197, the CNC interprets that the programmed "F" corresponds to the tool center. This means that the feedrate at the cutting point increases on inside arcs and decreases on outside arcs.



Constant cutting point feedrate:

$$F_R = \frac{R}{R + r} \cdot F_P$$

Where:

F_P	Programmed feedrate.
R	Path radius.
r	Tool radius.

G196 **Constant cutting point feedrate**

After executing G196, the CNC interprets that the programmed "F" corresponds to the contact point between the tool and the part. This results in a uniform part surface even on arcs.

Minimum radius for applying constant feedrate

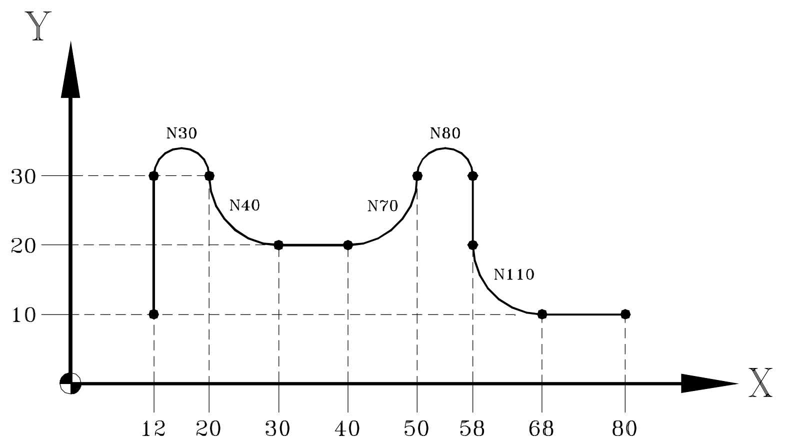
Using the instruction "#TANGFEED RMIN [<radius>]" a minimum radius may be set so that constant tangential feedrate is only applied on those arcs whose radius is bigger than the minimum set. If it is not programmed or it is set to zero, the CNC will apply constant tangential feedrate on all the arcs.

The minimum radius is applied from the next motion block on and it keeps its value after executing G197.

Properties of the functions

Functions G197 and G196 are modal and incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G197.



N10 G01 G196 G41 X12 Y10 F600	(Tool radius compensation and constant tangential feedrate)
N20 G01 X12 Y30	
N30 G02 X20 Y30 R4	(Constant tangential feedrate)
N40 G03 X30 Y20 R10	(Constant tangential feedrate)
N50 #TANGFEED RMIN [5]	(Minimum radius = 5)
N60 G01 X40 Y20	
N70 G03 X50 Y30 R10	(Constant tangential feedrate)
N80 G02 X58 Y30 R4	(No constant tangential feedrate. $R_{PROGRAMMED} < R_{MINIMUM}$)
N90 G01 X58 Y20	
N100 #TANGFEED RMIN [15]	(Minimum radius = 15)
N110 G03 X68 Y10 R10	(No constant tangential feedrate. $R_{PROGRAMMED} < R_{MINIMUM}$)
N120 G01 X80 Y10	
N130 G01 G40 X100	
N140 M30	

5.2.4 Cancellation of the % of feedrate override (G266)

G266 **Feedrate override at 100%**

This function sets the feedrate override at 100%, which can neither be changed by selector switch on the operator panel nor via PLC.

Function G266 only affects the block where it has been programmed, therefore, it only makes sense to add it to a block that defines a movement (motion block).

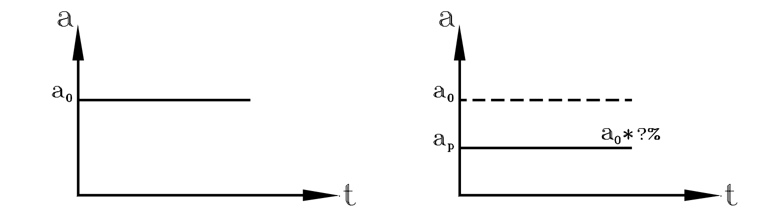
5.2.5 Acceleration control (G130/G131)

These functions allow to change the acceleration and deceleration of the axes.

Programming

The functions related to acceleration control are:

G130	Percentage of acceleration to be applied per axis.
G131	Percentage of acceleration to be applied, global.



a_0 : Nominal acceleration, set by the machine manufacturer.
a_P : Acceleration to be applied, set by the operator.

G130 **Percentage of acceleration to be applied per axis**

The percentaje of aceleration to be applied per axis is set by G130 followed by the axes and the percentage to be applied to each axis.

The acceleration values to be applied must be integers (not decimals).

```
...
G00 X0 Y0
G01 X100 Y100 F600
G130 X50 Y20            (Acceleration on the X axis = 50%)
                        (Acceleration on the Y axis = 20%)
G01 X0
G01 Y0
G131 100 X50 Y80        (Restore 100% of acceleration on all the axes)
                        (Movement to point X=50 Y=80)
...
```

G131 **Percentage of acceleration to be applied, global**

The percentaje of aceleration to be applied to all the axes is set by G131 followed by the new acceleration value to be applied to all the axes.

The acceleration values to be applied must be integers (not decimals).

When added to a motion block, the new values will be assumed before executing the move.

Considerations

The #SLOPE instruction determines the influence of the values defined with these values.

- In rapid positioning (G00)
- In the acceleration or deceleration stage.
- In the jerk of the acceleration or deceleration stages.

The programmed percentages are absolute, in other words, programming a 50% twice means that 50% will be applied, not 25%.

Properties of the functions

Functions G130 and G131 are modal and incompatible with each other.

On power-up, after an M02, M30, EMERGENCY or a RESET, the CNC restores 100% of acceleration for all the axes.

5.2.6 Jerk control (G132/G133)

The axis jerk may be modified with these functions.

Programming

The functions associated with jerk control are:

G132	Percentage of jerk to be applied per axis.
G133	Percentage of jerk to be applied, global.

G132 Percentage of jerk to be applied per axis

The percentaje of jerk to be applied per axis is set by G132 followed by the axes and the new jerk to be applied to each axis.

The jerk values to be applied must be integers (not decimals).

G00 X0 Y0	
G01 X100 Y100 F600	
G132 X20 Y50	(Jerk on the X axis = 20%)
	(Jerk on the Y axis = 50%)
G01 X0	
G01 Y0	
G133,100 X50 Y80	(Restore 100% of jerk on all the axes. Movement to point X=50 Y=80)

G133 Percentage of jerk to be applied, global

The percentaje of jerk to be applied to all the axes is set by G133 followed by the new jerk value to be applied to all the axes.

The jerk values to be applied must be integers (not decimals).

When added to a motion block, the new jerk values will be assumed before executing the move.

Considerations

The #SLOPE instruction determines whether the new percentages are to be applied or not on to rapid traverse movements (G00).

The programmed percentages are absolute, in other words, programming a 50% twice means that 50% will be applied, not 25%.

Properties of the functions

Functions G132 and G133 are modal and incompatible with each other.

On power-up, after an M02, M30, EMERGENCY or a RESET, the CNC restores 100% of jerk for all the axes.

5.2.7 Feed-Forward control (G134)

Feed-Forward control may be used to reduce the amount of following error (axis lag).

Feed-forward may be applied via machine parameters and via PLC as well as by program. The value defined by PLC will be the one with the highest priority and the one defined by the machine parameters will have the lowest priority.

Programming

G134 **Percentage of Feed-Forward to be applied**

The Feed-Forward percentaje is set by G134 followed by the axes and the new percentage of Feed-Forward to be applied to each axis.

The Feed-forward values to be applied may be defined with up to two decimals.

G134 X50.75 Y80 Z10	(Percentage of Feed-Forward to be applied:)
	(On the X axis = 50.75%)
	(On the Y axis = 80%)
	(On the Z axis = 10%)

Considerations

The maximum Feed-Forward value to be applied is 120%.

The programmed percentages are absolute, in other words, programming a 50% twice means that 50% will be applied, not 25%.

The value defined with G134 prevails over those defined in the machine parameters, but not over the one defined by PLC.

Properties of the functions

Function G134 is modal.

On power-up, after an M02 or M30, EMERGENCY or a RESET, the CNC restores the Feed-Forward set by the machine manufacturer for each axis.

Variable to define the feed-forward from the PLC

The write variable `(V.)A.PLCFFGAIN.Xn` may be used to set the feed-forward for each axis from the PLC. The value defined by this variable prevails over the ones defined by machine parameters or by program.

Setting this variable with a negative value cancels its effect (a zero value is also valid). This variable is initialized neither by a reset nor when validating the parameters.

5.2.8 AC-Forward control (G135)

AC-Forward control may be used to improve system response in acceleration changes and reduce the amount of following error (axis lag) on the acceleration and deceleration stages.

AC-forward may be applied via machine parameters and via PLC as well as by program. The value defined by PLC will be the one with the highest priority and the one defined by the machine parameters will have the lowest priority.

Programming

G135 **Percentage of AC-Forward to be applied**

The AC-Forward percentaje is set by G135 followed by the axes and the new percentage of AC-Forward to be applied to each axis.

The AC-forward values to be applied may be defined with up to one decimal.

G135 X55.8 Y75 Z110	(Percentage of AC-Forward to be applied:)
	(On the X axis = 55.8%)
	(On the Y axis = 75%)
	(On the Z axis = 110%)

Considerations

The maximum AC-Forward value to be applied is 120%.

The programmed percentages are absolute, in other words, programming a 50% twice means that 50% will be applied, not 25%.

The value defined with G135 prevails over those defined in the machine parameters, but not over the one defined by PLC.

Properties of the functions

Function G135 is modal.

On power-up, after an M02 or M30, EMERGENCY or a RESET, the CNC restores the AC-Forward set by the machine manufacturer for each axis.

Variable to define the AC-forward from the PLC

The write variable `(V.)A.PLCACGAIN.Xn` may be used to set the AC-forward for each axis from the PLC. The value defined by this variable prevails over the ones defined by machine parameters or by program.

Setting this variable with a negative value cancels its effect (a zero value is also valid). This variable is initialized neither by a reset nor when validating the parameters.

5.3 Spindle speed (S)

The spindle speed is selected by program using the spindle name followed by the desired speed. The speeds of all the spindles of the channel may be programmed in the same block.

```
S1000
S1=500
S1100 S1=2000 S4=2345
```

The programmed speed stays active until another value is programmed. The programming units will be RPM unless selected otherwise.

It is possible to program using parameters or arithmetic expressions.

Spindle start and stop

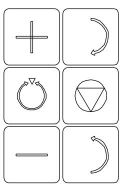
Defining a speed does not imply starting the spindle. The startup is defined using the following auxiliary functions. Ver ***"Spindle control (M03/M04/M05/M19)"*** en la página 83.

M03	- Starts the spindle clockwise.
M04	- Starts the spindle counterclockwise.
M05	- Stops the spindle.

Maximum speed

The maximum turning speed in each range (gear) is limited by the machine manufacturer. When programming a higher turning speed, the CNC limits its value to the maximum allowed by the active range (gear). The same thing occurs when trying to exceed the maximum limits using the "+" and "-" keys of the operator panel or doing it via PLC or by program.

Speed override



The programmed "S" speed may be varied between 50% and 120% using the "+" and "-" of the operator panel or via PLC. However the maximum and minimum variation may be different depending on how the machine manufacturer has set [A.M.P. "MINOVR" and "MAXOVR"].

Likewise, the incremental step associated with the "+" and "-" keys of the Operator Panel to change the programmed spindle speed "S" will be 10; but this value may be different depending on the setting of axis machine parameter ["STEPOVR"]

During threading operations, the programmed speed cannot be overridden and it will be set at 100% of the programmed "S" speed.

5.3.1 Spindle speed programming

The following functions are oriented to lathe type machines. In order for Constant Surface Speed mode to be available, the machine manufacturer must have set one of the axis -face axis- (usually axis perpendicular to the shaft of the part).

The functions related to spindle speed programming may be used to select either Constant Surface Speed mode or Constant turning speed mode. Constant Surface Speed is only available at the master spindle of the channel.

At constant surface speed, the CNC changes the spindle speed as the perpendicular axis moves in order to maintain the cutting speed constant between the tool and the part, thus optimizing the machining conditions.

Programming

The functions related to spindle speed programming are:

G96 Constant surface speed.

G97 Constant turning speed.

These functions may be programmed anywhere in the program and they don't have to go alone in the block.

G96 **Constant surface speed**

The G96 function only affects the master spindle of the channel.

After executing G96, the CNC interprets that the spindle speeds programmed for the master spindle of the channel are in meters/ minute (feet/minute). This work mode is activated when programming a new speed while G96 is active.

It is recommended to program the speed in the same block as the G96 function. The spindle gear (range) (M41, M42, M43, M44) must be selected in the same block or in a previous one.

G97 **Constant turning speed**

The G97 function affects all the spindles of the channel.

After executing G97, the CNC interprets that the spindle speeds programmed are in rpm and starts working at constant turning speed.

It is recommended to program the speed in the same block as the G97 function; if not programmed, the CNC assumes as programmed speed the one the spindle is currently turning at. The spindle gear (range) (M41, M42, M43, M44) may be selected at any time.

Properties of the functions

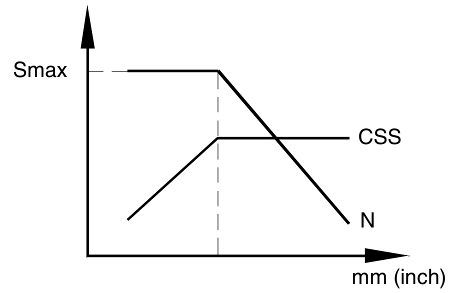
Functions G96 and G97 are modal and incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G97.

5.3.2 Turning speed limit

The following function is oriented to lathe type machines.

When working at constant surface speed and because the turning speed changes with the movement of the perpendicular axis, the maximum turning speed must be programmed. When the spindle reaches that speed, it keeps working at constant turning speed.



CSS	Constant surface speed.
N	Constant turning speed.
Smax	Maximum turning speed.

This limitation is only valid for the master spindle of the channel when it is working at constant cutting speed. It will be ignored when working at constant turning speed and the maximum speed allowed will be the one set for the active gear (range).

G192 **Turning speed limit in constant cutting speed mode**

The turning speed limit is set by programming function G192 and then the maximum turning speed for constant surface speed. The maximum turning speed is always set in RPM.

When executing G192, the CNC limits the maximum turning speed to the value set with "S". This means that the spindle will not exceed this speed in G96 even when programming higher speeds. The maximum speed cannot be exceeded either using "+" and "-" keys of the Operator Panel.

G192 S2500	Maximum turning speed = 2500 rpm
G96 S180	Constant surface speed. =180m/min.
...	
G97 S1000 M3	Constant turning speed. = 1000RPM
...	
G96	
...	
S230	It activates constant surface speed mode.
	The turning speed limit stays active at 2500RPM.

5.4 Tool number (T)

The "T" code identifies the tool to be selected. The tools may be in a magazine managed by the CNC or in a manual magazine (referred to as ground tools).

The programming format is T<0-4294967294> and it can be programmed using parameters and arithmetic expressions. In these cases, by default, the value calculated is rounded up to an integer. If the result is negative, the CNC will issue the pertinent error message.

Defintion

To load a tool in the spindle, it must be previously defined. To do that, the CNC offers a table where the user may define the data for each tool.

On the other hand, when having a magazine managed by the CNC, one must define the magazine position occupied by each tool. To do that, the CNC offers a table where the user may define the position of each tool.

The table data may be defined:

- Manually from the CNC's front panel (as described in the Operating Manual).
- Via program, using the associated variables (as described in the relevant chapter of this manual).

Load a tool in the spindle.

The tool required for machining may be selected by program using the "T<n>" code where <n> is the number of the tool to be loaded in the spindle.

The "T" code only selects the tool. After selecting a tool, function M06 must be programmed to load it into the spindle. Loading/unloading is carried out depending on the subroutine associated with the M06 (if so defined by the machine manufacturer).

```
N10 G00 X0 Y0 F500 S1000 M03
N20 T1          (Select tool T1)
N30 M06         (Load tool T1 into the spindle)
N40 ...
N50 T2          (Select tool T2)
N60 ...
N70 ...
N80 ...
N90 M06         (Load tool T2 into the spindle)
N100 ...
N110 M30
```

Loading and unloading a tool in the magazine

To load the tools into the magazine, the magazine must be in load mode. To unload the tools from the magazine, the magazine must be in unload mode. The tools are loaded from ground to the magazine going through the spindle and are unloaded to ground going through the spindle.

The magazine's work mode is set by variable `V.[n].TM.MZMODE` where n is the channel number. Depending on the value of the variable, the manager will assume one of the following work modes.

Value	Meaning
0	Normal mode (by default and after Reset).
1	Magazine loading mode.
2	Magazine unloading mode.

Once the magazine is load or unload mode, the operation is carried out from the program using the code T*n*where *n* is the tool number. Once the tools have been loaded or unloaded, the magazine must be set to normal mode (value of ·0·).

```
V.[1].TM.MZMODE = 1
T1 M6
T2 M6
...
V.[1].TM.MZMODE = 0
```

Loading a tool in a specific magazine position

Some tools, due to their characteristics (size, weight, etc.) must be placed in a specific magazine position.

The command POS*n* defines the magazine position for the tool. It must always be programmed in the same block as T*n*.

```
V.[1].TM.MZMODE = 1
T3 M6 POS24
    (Places tool 3 in magazine position 24)
...
V.[1].TM.MZMODE = 0
```

The magazine position can only selected when the magazine is in load mode. Otherwise, it issues the relevant error message.

Loading a tool in a system with several magazines

When using more than one tool magazine, one must indicate in which one of them the tool is to be loaded using the code `MZ`*n*, where *n* indicates the magazine number. It must always be programmed in the same block as `T`*n*.

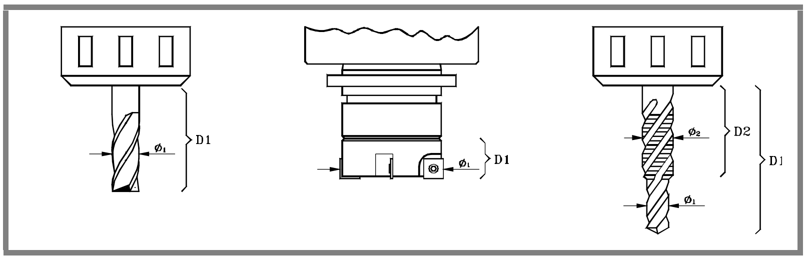
```
T1 MZ1 M6
    (Place tool 1 in the first magazine)
T8 MZ2 POS17 M6
    (Place tool 8 in position 17 of the second magazine)
```

Considerations

The machine manufacturer may have associated a subroutine with the "T" code, that will be executed automatically when selecting the tool. If the M06 has been included in this subroutine, the tool will be loaded into the spindle when executing the "T" code.

5.5 Tool offset number (D)

The tool offset contains the tool dimensions. Each tool may have several offsets associated with it in such a way that when using combined tools having parts with different dimensions, a different offset number will be used for each of those parts.



When activating a tool offset, the CNC assumes the tool dimensions defined for that offset, therefore when working with tool radius or length compensation, the CNC will apply those dimensions for compensating the path.

Defintion

To activate an offset, it must be previously defined. To do that, the CNC offers a portion of the tool table where the user may define several offsets. The table data may be defined:

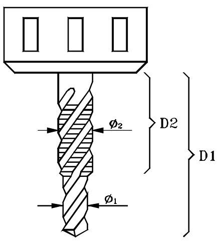
- Manually from the CNC's front panel (as described in the Operating Manual).
- Via program, using the associated variables (as described in the relevant chapter of this manual).

The offsets are only associated with the tool for which they have been defined. This means that when activating a tool offset, the offset of the corresponding active tool will be activated.

Activation

Once the tool offsets have been defined in the table, they may be selected by program using the "D<n>" code where <n> is the number of the offset to be applied. The offset number may also be defined using a parameter or an arithmetic expression.

If no tool offset is programmed, the CNC assumes tool offset D1.




```
N10 ...
N20 T7 D1              (Select tool T7 and tool offset D1)
N30 M06                (Load tool T7 into the spindle)
N40 F500 S1000 M03
N50 ...                (Operation 1)
N60 D2                 (Select tool offset D2 of T7)
N70 F300 S800
N80 ...                (Operation 2)
N90 ...
```

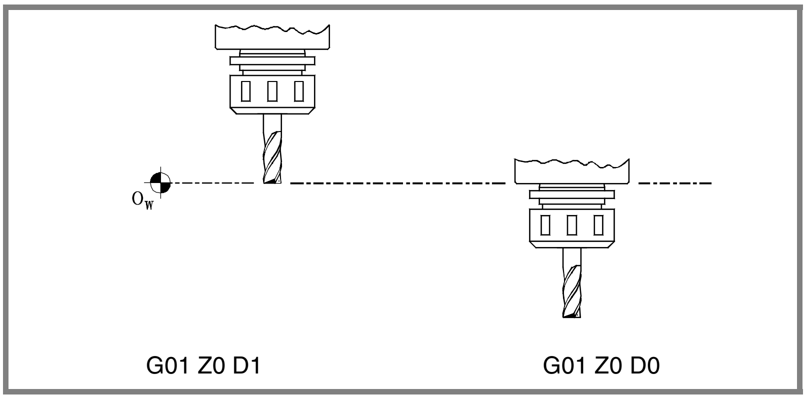
Only one tool offset may be active at a time; therefore, activating a tool offset will cancel the previous one. Programming "D0" will cancel the active offset.

N10 ...	
N20 T1 M06	(Select and load tool T1. Offset D1 is activated by default)
N30 F500 S1000 M03	
N40 ...	(Operation 1)
N50 T2	(Prepare tool T2)
N60 D2	(Select tool offset D2 for tool T1)
N70 F300 S800	
N80 ...	(Operation 2)
N90 M6	(loading tool T2 with its offset D1)
N100 F800 S1200 M03	
N110 ...	(Operation 3)
N120 ...	

Considerations

Activating the tool offset also activates tool length compensation. This compensation is also activated after a tool change because it "D1" is assumed after the change (if another one has not been programmed).

Canceling the tool offset with "D0" also cancels tool length and radius compensation.



5.6 Auxiliary (miscellaneous) functions (M)

Auxiliary "M" functions are related to the overall CNC program execution and the control of the various devices of the machine such as spindle gear change, coolant, tool changes and so on.

Programming

Up to 7 "M" functions may be programmed in a block. The programming format is M<0 - 65535>, and it can be programmed using parameters and arithmetic expressions. In these cases, by default, the value calculated is rounded up to an integer. If the result is negative, the CNC will issue the pertinent error message.

Execution

Depending on how they have been set by the machine manufacturer ("M" function table):

- The "M" functions will be executed before or after the movement of the block where they were programmed.

  When setting an "M" to be executed after the move of the block, depending on the active function G05 or G07:

  G05 The "M" function is executed with the theoretical end of the movement (when the axes have not reached position).

  G07 The "M" function is executed with the real end of the movement (when the axes are already in position).

- The CNC will wait or not for the confirmation that the "M" function has been executed before resuming program execution. If it has to wait for confirmation, it will have to be received before or after executing the movement of the block where it has been programmed.
- The "M" functions that have not been set in the table will be executed before the movement of the block where they have been programmed and the CNC will wait for the "M-done" confirmation before executing the movement of the block.

Certain "M" functions have a particular internal meaning in the CNC. The section on *"5.6.1 List of "M" functions"* in this chapter shows a list of these functions with their internal meaning for the CNC.

Associated subroutine

The "M" functions may have an associated subroutine that will be executed instead of the function.

If, within a subroutine associated with an "M" function, the same "M" function is programmed, this function will be executed, but not its associated subroutine.

5.6.1 List of "M" functions

Program Interruption (M00/M01)

M00 **Program stop.**

Function M00 interrupts the execution of the program. It does not stop the spindle or initialize the cutting conditions.

The [CYCLE START] key of the operator panel must be pressed again in order to resume program execution.

This function should be set in the "M" function table so it is executed at the end of the block where it is programmed.

M01 **Conditional program stop.**

When the external conditional stop switch is active (PLC signal "M01 STOP"), it interrupts program execution. It does not stop the spindle or initialize the cutting conditions.

The [CYCLE START] key of the operator panel must be pressed again in order to resume program execution.

This function should be set in the "M" function table so it is executed at the end of the block where it is programmed.

End of program (M02/M30)

M02/M30 **End of program.**

Both functions indicate the end of the program. Executing it sets the channel to its initial conditions and selects the first block of the program. It also stops the spindle (if so defined by parameter SPDLSTOP) and initializes the cutting conditions.

This function should be set in the "M" function table so it is executed at the end of the block where it is programmed.

End of subroutine (M17/M29)

M17/M29 **End of subroutine.**

Both functions indicate the end of a subroutine.

Tool change (M06)

M06 **Tool change.**

The M06 function executes the tool change. The CNC will manage the tool change and update the table for the tool magazine.

This function should be set in the "M" table so it executes the subroutine corresponding to the tool changer installed on the machine.

Spindle control (M03/M04/M05/M19)

M03 **Starts the spindle clockwise.**

M04 **Starts the spindle counterclockwise.**

Function M03 starts the spindle clockwise and function M04 counterclockwise. These functions remain active until another spindle controlling function is programmed (M03/M04/M05/M19).

These functions should be set in the "M functions" table so they are executed at the end of the block where it is programmed.

These functions may be defined together with the programmed speed or in a separate block. If the block where they are programmed does not mention any spindle, they will be applied to the master spindle of the channel.

```
S1000 M3
    (The spindle "S" starts clockwise at 1000 rpm)
S1=500 M4
    (The spindle "S1" starts counterclockwise at 500 rpm)
M4
    (The master spindle starts counterclockwise)
```

If several spindles are programmed in a single block, functions M3 and M4 apply to all of them. To start the spindles in different directions, define next to each M function the spindle it is associated with, as follows.

M3.S / M4.S	M3 or M4 associated with the spindle S.

```
S1000 S2=456 M3
    (Spindle "S" turning at 1000 rpm and S2 at 456 rpm, both clockwise)
M3.S S1000 S2=456 M4.S2
    (The spindle "S" turns clockwise at 1000 rpm)
    (The spindle "S2" turns counterclockwise at 456 rpm)
```

M05 **Spindle stop.**

Function M05 stops the spindle. This function remains active until another spindle controlling function is programmed (M04/M03/M19).

To stop a spindle, define next to the M5 the spindle it is associated with, as follows. If it does not mention any spindle, it applies to the master spindle.

M5.S	Function M5 associated with the spindle S.

```
S1000 S2=456 M5
```
(Stops the master spindle)

```
M5.S M5.S2 S1=1000 M3.S1
```
(Stops the spindles "S" and "S2")
(Spindle "S1" turns clockwise)

M19 **Spindle orientation.**

Function M19 orients the spindle. This function stays active until a speed controlling function is programmed (M03/M04/M05).

This work mode is only available on machines having a rotary encoder installed on the spindle.

When executing function M19, the CNC interprets that the value entered with the "Sn" code indicates the angular position of the spindle. If several spindles are programmed in a single block, function M19 applies to all of them.

This angular position is programmed in degrees and it is always assumed in absolute coordinates, thus not being affected by functions G90/G91.

```
M19 S0
```
(Positioning of spindle S at 0º)

```
M19 S2=120
```
(Positioning of spindle S2 at 120º)

```
M19 S1=10 S2=34
```
(Positioning of spindle S1 at 10º and S2 at 34º)

To orient the spindle to the ·0· position, it may also be programmed by defining, next to the M19, the spindle to be oriented.

M19.S1	Positioning of spindle S1 at 0º.

```
M19.S4
```
(Positioning of spindle S4 at 0º)

```
M19
```
(Positioning of the master spindle at 0º)

Every positioning move requires an M19. An "S" code without an M19 is interpreted as a new turning speed for the next time the spindle is turned on in speed mode using functions M03/M04.

How is positioning carried out

When executing the M19 function for the first time, it homes the spindle. The M19 functions programmed afterwards only orient the spindle. To home the spindle again, use function G74.

When executing function M19, the positioning is carried out as follows.

1. The spindle stops (if it was turning).
2. The CNC no longer works in speed mode and it switches to positioning mode.
3. If it is the first time the M19 is executed, the CNC homes the spindle (home search).
4. It positions the spindle at 0º or at the angular position defined by the "S" code (if it has been programmed). To do that, it will calculate the module (between 0 and 360º) of the programmed value and the spindle will reach that position.

Setting the turning direction for spindle orientation

If when executing function M19, there was an M3 or M4 active, even if the speed is zero, this function will set the spindle orienting direction.

If no M3 or M4 is active, the turning direction is set depending on machine parameter `SHORTESTWAY`.

- If it is a `SHORTESTWAY` spindle, it positions via the shortest way.
- If it is not a `SHORTESTWAY` type spindle; by default, it positions in the same direction as the last spindle movement. It is also possible to define the M19 with the positioning direction as follows.

  M19.POSPositioning in the positive direction.

  M19.NEGPositioning in the negative direction.

  To set a particular spindle turning direction, it must be programmed as follows.

```
M19.POS S120 S1=50
    (The positive direction is applied to spindle "S" and "1")
M19.NEG.S1 S1=100 S34.75
    (The negative direction is applied to spindle "1")
```

When programming the orienting direction for a `SHORTESTWAY` type spindle, the programmed direction will be ignored.

Positioning speed.

The positioning speed of the spindle Sn is defined using the command `Sn.POS` as follows:

Sn.POS Positioning speed of spindle Sn.

```
M19 S.POS=120 S1.POS=50
    (Positioning of spindle "S" at 120 rpm and S2 at 50 rpm)
```

The positioning speed is given in rpm.

If no positioning speed is programmed, the CNC assumes the one set by machine parameter `REFEED1` as the positioning speed.

```
N10 G97 S2500 M03
    (The spindle turns at 2500 RPM)
N20 M19 S50
    (Spindle controlled in position. Home search and positioning at 50º)
N30 M19 S150
    (Positioning at 150º)
N40 S1000
    (New spindle speed. The spindle stays in positioning mode)
N50 M19 S-100
    (Positioning at -100º)
N60 M03
    (Spindle controlled in speed. The spindle turns at 1000 RPM)
N70 M30
```

Gear change (M41-M44)

M41-M44 **Spindle gear change.**

The spindle gear (range) desired for the programmed speed is selected with functions M41, M42, M43 and M44. The CNC may have up to 4 different spindle gears.

These functions may be defined together with the programmed spindles or in a separate block. If the block where they are programmed does not mention any spindle, they will be applied to the master spindle of the channel.

```
S1000 M41
S1=500 M42
M44
```

When using Sercos axes, functions M41-M44 also involve changing the drive's velocity gear.

If several spindles are programmed in a single block, the functions apply to all of them. To apply different gears to the spindles, define next to each M function the spindle it is associated with, as follows.

M41.S	Function M41 associated with the spindle S.

```
S1000 S2=456 M41
    (Gear 1 with spindle "S" and with S2)
M41.S M42.S3
    (Gear 1 with spindle "S")
    (Gear 2 with spindle "S3")
```

The maximum speed for each gear is limited by the machine manufacturer. Likewise, if the machine manufacturer has set the spindle gear change so it is executed automatically [S.M.P. "AUTOGEAR"] the CNC will manage functions M41, M42, M43 and M44 and will change the gears according to the programmed S speed.

5.7 Auxiliary functions (H)

Auxiliary "H" functions are used to send information out to the PLC. They differ from the "M" functions in that the "H" functions do not wait for confirmation that the function has been executed in order to go on executing the program.

Programming

Up to 7 "H" functions may be programmed in the same block. The programming format is H<0 - 65535>, and it can be programmed using parameters and arithmetic expressions. In these cases, by default, the value calculated is rounded up to an integer. If the result is negative, the CNC will issue the pertinent error message.

Execution

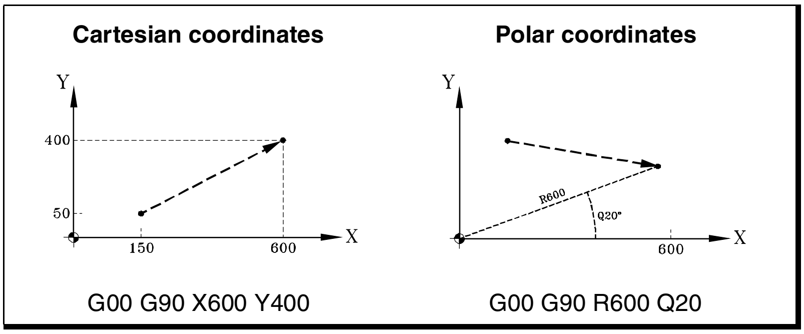
The auxiliary "H" functions are executed at the beginning of the block where they have been programmed.

TOOL PATH CONTROL

6

6.1 Rapid traverse (G00)

Movements programmed after G00 are executed in a straight line and at the rapid feedrate set by the machine manufacturer from the current position to the destination or target point. Regardless of the number of axes involved, the resulting path is always a straight line.



When auxiliary or rotary axes are involved in rapid positioning, the movement is carried out so it begins and ends when the main axes begin and end their movement.

Programming

The movements may be defined as follows:

- In Cartesian coordinates ("X","X1"..."C9")

  Defining the coordinates of the end point on the various axes.

  All the axes need not be programmed, only the ones to move.

- In polar coordinates ("R", "Q")

  Defining the radius and the angle of the end point referred to the polar origin.

  The "R" radius will be the distance between the polar origin and the point. The "Q" angle will be formed by the abscissa axis and the line joining the polar origin with the point.

  If the angle or the radius is not programmed, it keeps the value programmed for the last move.

Feedrate behavior

A G00 movement temporarily cancels the programmed "F" and the rapid traverse movement is carried out at the value set by the machine manufacturer [A.M.P. "G00FEED"]. The "F" value is restored when programming a G01, G02 or G03 type function.

When several axes are involved, the resulting feedrate is calculated so at least one of the axis moves at its maximum speed.

When defining an "F" value and G00 in the same block, the CNC will store the value assigned to "F" and it will apply it next time a G01, G02 or G03 type function is programmed.

The override percentage is set at 100% or may be varied between 0% and 100% with the switch at the operator panel depending on how the machine manufacturer has set [G.M.P. "RAPIDOVR"].

Properties of the function

Function G00 is modal and incompatible with G01, G02, G03, G33 and G63.

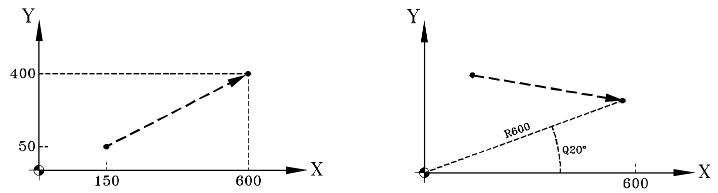
Function G00 may be programmed as G0.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G00 or G01 as set by the machine manufacturer [G.M.P. "IMOVE"].

6.2 Linear interpolation (G01)

Movements programmed after G01 are executed in a straight line and at the programmed feedrate "F" from the current position to the indicated target point. Regardless of the number of axes involved, the resulting path is always a straight line.

Cartesian coordinates | **Polar coordinates**



G01 G90 X600 Y400 F150 | G01 G90 R600 Q20 F185

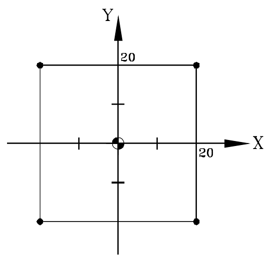
Auxiliary and rotary axes may also be programmed in the linear interpolation block. In those cases, the CNC will calculate the feedrate for those axes so their movement begins and ends simultaneously with the main axes.

Programming

- In Cartesian coordinates ("X","X1"..."C9")

  Defining the coordinates of the end point on the various axes.

  All the axes need not be programmed, only the ones to move.



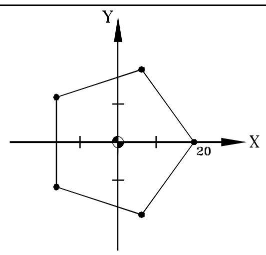

```
G00 X20 Y0
G01 Y20 F350
G01 X-20
G01 Y-20
G01 X20
G01 Y0
M30
```

- In polar coordinates ("R", "Q")

  Defining the radius and the angle of the end point referred to the polar origin.

  The "R" radius will be the distance between the polar origin and the point. The "Q" angle will be formed by the abscissa axis and the line joining the polar origin with the point.

  If the angle or the radius is not programmed, it keeps the value programmed for the last move.




```
G00 X20 Y0
G01 R20 Q72 F350
G01 Q144
G01 Q216
G01 Q288
G01 Q360
M30
```

Feedrate behavior

The programmed feedrate "F" stays active until a new value is programmed, thus not being necessary to program it in every block.

When several axes are involved, the CNC calculates the feedrate for each axis so the resulting path is executed at the programmed feedrate "F" .

The programmed feedrate "F" may be varied between 0% and 200% using the selector switch on the CNC's operator panel or it may be selected by program or by PLC. However, the maximum override is limited by the machine manufacturer [G.M.P. "MAXOVR"].

The feedrate of the auxiliary axes

The behavior of the auxiliary axes is determined by general machine parameter FEEDND.

- If its value is TRUE, none of the axes will exceed the programmed feedrate.
- If its value is FALSE, the feedrate is applied to the main axes whereas the auxiliary axes may exceed it, but without ever exceeding their MAXFEED. If an axis were to exceed the MAXFEED, the programmed feedrate of the main axes would be limited.

Properties of the function

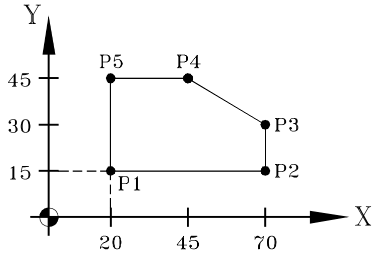
Function G01 is modal and incompatible with G00, G02, G03, G33 and G63.

Function G01 may also be programmed as G1.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G00 or G01 as set by the machine manufacturer [G.M.P. "IMOVE"].

Programming examples

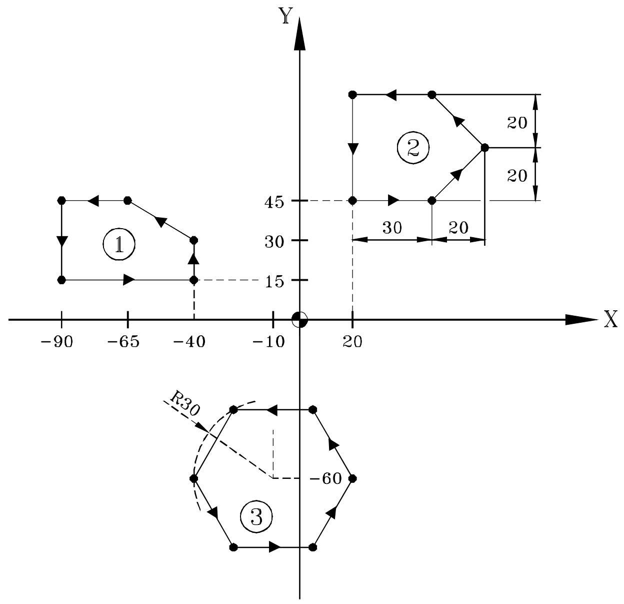
Programming in cartesian coordinates.



	X	Y
P1	20	15
P2	70	15
P3	70	30
P4	45	45
P5	20	45

Absolute coordinates	**Incremental coordinates**
N10 G00 G90 X20 Y15	N10 G00 G90 X20 Y15
N20 G01 X70 Y15 F450	N20 G01 G91 X50 Y0 F450
N30 Y30	N30 Y15
N40 X45 Y45	N40 X-25 Y15
N50 X20	N50 X-25
N60 Y15	N60 Y-30
N70 G00 X0 Y0	N70 G00 G90 X0 Y0
N80 M30	N80 M30

Programming in Cartesian and polar coordinates.



N10 T1 D1	
N20 M06	
N30 G71 G90 F450 S1500 M03	(Initial conditions)
N40 G00 G90 X-40 Y15 Z10	(Approach to profile 1)
N50 G01 Z-5	
N60 X-40 Y30	(Machining of profile 1)
N70 X-65 Y45	
N80 X-90	
N90 Y15	
N100 X-40	(End of profile 1)
N110 Z10	
N120 G00 X20 Y45 F300 S1200	(Approach to profile 2)

N130 G92 X0 Y0	(Preset new part zero)
N140 G01 Z-5	
N150 G91 X30	(Machining of profile 2)
N160 X20 Y20	
N170 X-20 Y20	
N180 X-30	
N190 Y-40	(End of profile 2)
N200 G90 Z10	
N210 G92 X20 Y45	(Restore previous part zero)
N220 G30 I-10 J-60	(Polar origin preset)
N230 G00 R30 Q60 F350 S1200	(Approach to profile 3)
N240 G01 Z-5	
N250 Q120	(Machining of profile 3)
N260 Q180	
N270 Q240	
N280 Q300	
N290 Q360	
N300 Q60	(End of profile 3)
N310 Z10	
N320 G00 X0 Y0	
N330 M30	

6.3 Circular interpolation (G02/G03)

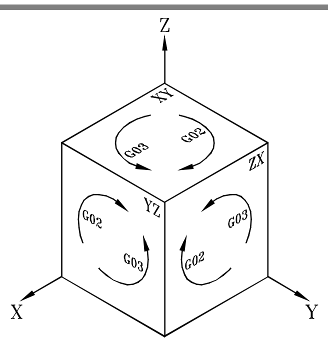
Movements programmed after G02 and G03 are executed along a circular path at the programmed feedrate "F" from the current position to the indicated target point.

A circular interpolation can only be executed in the active plane. There are two types of circular interpolations:

G02 Clockwise circular interpolation.

G03 Counterclockwise circular interpolation.

Clockwise (G02) and counterclockwise (G03) moving directions have been established according to the following coordinate system.



The coordinate system is referred to the movement of the tool on the part.

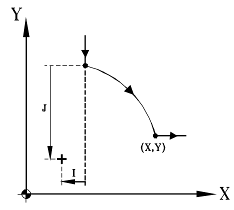
Programming

A circular interpolation may be defined as follows:

- In cartesian coordinates, by defining the coordinates of the target point and the center of the arc.
- In cartesian coordinates, by defining the coordinates of the target point and the arc radius.
- In polar coordinates, defining the radius and the angle of the end point as well as the arc center coordinates.

Cartesian coordinates

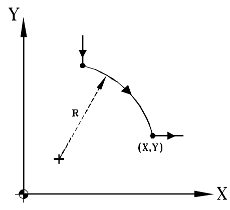
(arc center)



G02/G03 X Y I J

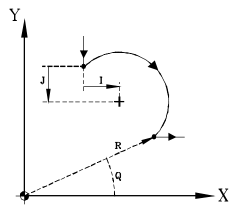
Cartesian coordinates

(arc radius)



G02/G03 X Y R

Polar coordinates



G02/G03 R Q I J

Feedrate behavior

The programmed feedrate "F" stays active until a new value is programmed, thus not being necessary to program it in every block.

The programmed feedrate "F" may be varied between 0% and 200% using the selector switch on the CNC's operator panel or it may be selected by program or by PLC. However, the maximum override is limited by the machine manufacturer [G.M.P. "MAXOVR"].

Properties of the function

Functions G02 and G03 are modal and incompatible with each other and with G00, G01, G33 and G63.

Function G74 (Home search) also cancels functions G02 and G03.

Functions G02 and G03 may also be programmed as G2 and G3.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G00 or G01 as set by the machine manufacturer [G.M.P. "IMOVE"].

6.3.1 Cartesian coordinates (Arc center programming)

The arc is defined by programming function G02 or G03 followed by the coordinates of the arc's end point and those of its center (referred to the starting point of the arc) according to the axes of the active work plane.

Coordinates of the arc's final point

It is defined with its coordinates along the axes of the active work plane and may be given in either absolute or incremental coordinates.

If they are not programmed or are the same as the starting point, a full circle will be executed.

Arc center coordinates

The arc center coordinates are defined by the letters "I", "J" or "K" depending on the active plane.

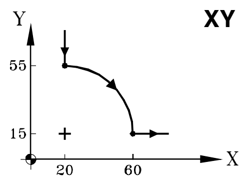
G17 G18 G19	Letters "I", "J" and "K" are associated with the first, second and third axis of the channel respectively.
G20	Letters "I", "J" and "K" are associated with the abscissa, ordinate and perpendicular axes of the defined plane.

When the center coordinate on an axis is "0", it does not have to be programmed. These coordinates are not affected by functions G90 and G91.

Depending on the active work plane, the programming format is:

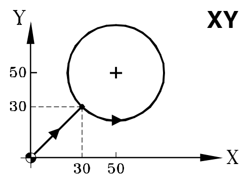
XY plane (G17)	G02/G03	X...	Y...	I...	J...
ZX plane (G18)	G02/G03	X...	Z...	I...	K...
YZ plane (G19)	G02/G03	Y...	Z...	J...	K...

Circular interpolation programming by defining the center.



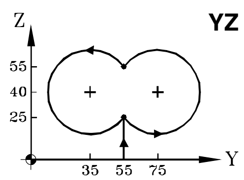
XY

```
...
G02 X60 Y15 I0 J-40
...
```



XY

```
N10 G17 G71 G94
N20 G01 X30 Y30 F400
N30 G03 X30 Y30 I20 J20
N40 M30
```



YZ

```
N10 G19 G71 G94
N20 G00 Y55 Z0
N30 G01 Y55 Z25 F400
N40 G03 Z55 J20 K15
N50 Z25 J-20 K-15
N60 M30
```

6.3.2 Cartesian coordinates (Radius programming)

The arc is defined by programming function G02 or G03 followed by the coordinates of the arc's end point and its radius.

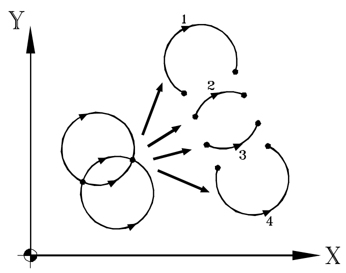
Coordinates of the arc's final point

It is defined with its coordinates along the axes of the active work plane and may be given in either absolute or incremental coordinates.

Arc radius

The arc radius is defined with the letter "R" or using assignments "R1=<radius>" or "G263=<radius>". The radius value stays active until a new value is assigned or an arc is programmed using the center coordinates or a movement is programmed in polar coordinates.

If the arc is smaller than 180º, the radius will be programmed with a positive sign and with a negative sign if it is greater than 180º. This way and depending on the selected circular interpolation (G02 or G03), the desired arc will be defined.

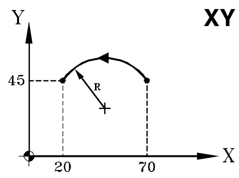


Arc 1	G02 X... Y... R-...
Arc 2	G02 X... Y... R+...
Arc 3	G03 X... Y... R+...
Arc 4	G03 X... Y... R-...

Depending on the active work plane, the programming format is:

XY plane (G17)	G02/G03	X...	Y...	R+/-
ZX plane (G18)	G02/G03	X...	Z...	R+/-
YZ plane (G19)	G02/G03	Y...	Z...	R+/-

Different formats to define the same arc.

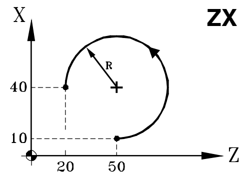


XY

Nxx G03 G17 X20 Y45 R30

Nxx G03 G17 X20 Y45 G263=30

Nxx G03 G17 X20 Y45 R1=30

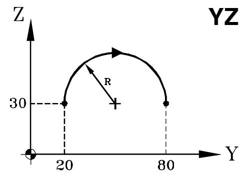


ZX

Nyy G03 G18 Z20 X40 R-30

Nyy G03 G18 Z20 X40 G263=-30

Nyy G03 G18 Z20 X40 R1=-30



YZ

Nzz G02 G19 Y80 Z30 R30

Nzz G02 G19 Y80 Z30 G263=30

Nzz G02 G19 Y80 Z30 R1=30

The radius may also be programmed in a block prior to the one defining the circular interpolation. In this case, the radius is defined using the assignments "R1=<radius>" or "G263=<radius>".

```
N10 G01 G90 X0 Y0 F500      N10 G01 G90 X0 Y0 F450
N20 G263=50                 N20 G01 G263=50
N30 G02 X100                N30 G02 X100
```

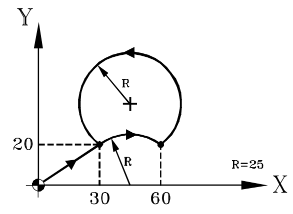
```
N10 G01 G90 X0 Y0
N20 G02 G263=50
N30 X100
```

The previous examples make semicircles of a 50 mm radius. Although the examples use function "G263=<radius>", they're also valid if they are programmed using "R1=<radius>".

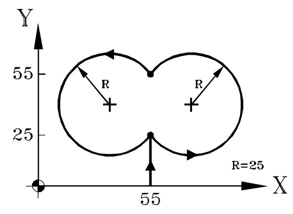
The CNC keeps the radius value until a circular interpolation is programmed by defining the center coordinates or a movement is programmed in polar coordinates.

When programming an arc using the radius, it is not possible to program full circles because there are infinite solutions.

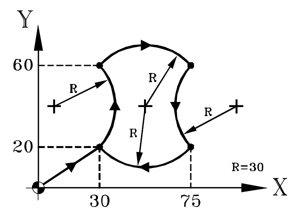
Circular interpolation programming by defining the radius.




```
N10 G01 G90 G94 X30 Y20 F350
N20 G263=25
N30 G02 X60
N40 G263=-25
N50 G03 X30
N60 M30
```




```
N10 G17 G71 G94
N20 G00 X55 Y0
N30 G01 X55 Y25 F400
N40 G263=-25
N50 G03 Y55
N60 Y25
N70 M30
```




```
N10 G17 G71 G94
N20 G01 X30 Y20 F400
N30 R1=30
N40 G03 Y60
N50 G02 X75
N60 G03 Y20
N70 G02 X30
N80 M30
```

6.3.3 Polar coordinates

The arc is defined by programming function G02 or G03 followed by the coordinates of the arc's end point and those of its center (referred to the starting point of the arc) according to the axes of the active work plane.

Coordinates of the end point

The position of the end point is given by defining the radius "R" and the angle "Q" as follows:

Radius	Distance between the polar origin and the point.
Angle	Angled formed by the line joining the polar origin with the point and the horizontal going through the polar origin.

If the angle or the radius is not programmed, it keeps the value programmed for the last move. The radius and the angle may be defined both in absolute (G90) and incremental coordinates (G91).

When programming the angle in G91, it is incremented with respect to the polar origin of the previous point; if programmed in G90, It indicates the angle formed with the horizontal going through the polar origin.

Programming a 360º angle in G91 means programming a whole circle. Programming a 360º angle in G90 means programming an arc where the target point forms a 360º angle with the horizontal going through the polar origin.

Center coordinates

The arc center coordinates are defined by the letters "I", "J" or "K" depending on the active plane.

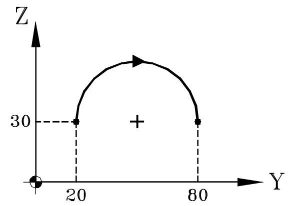
G17 G18 G19	Letters "I", "J" and "K" are associated with the first, second and third axis of the channel respectively.
G20	Letters "I", "J" and "K" are associated with the abscissa, ordinate and perpendicular axes of the defined plane.

When the center coordinate on an axis is zero, it does not have to be programmed; if neither of them are programmed, it will assume the polar origin as the arc center. These coordinates are not affected by functions G90 and G91.

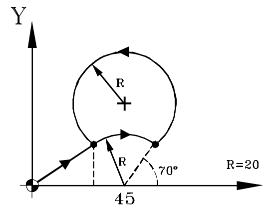
Depending on the active work plane, the programming format is:

XY plane (G17)	G02/G03	R...	Q...	I...	J...
ZX plane (G18)	G02/G03	R...	Q...	I...	K...
YZ plane (G19)	G02/G03	R...	Q...	J...	K...

Circular interpolation programming in polar coordinates.

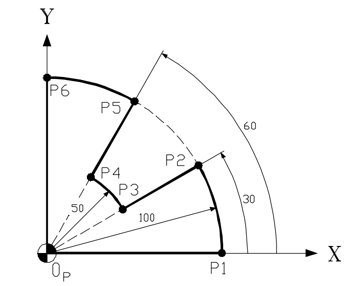



```
N10 G0 G90 X20 Y30 F350
N20 G30
N30 G02 R60 Q0 I30
N40 M30
```



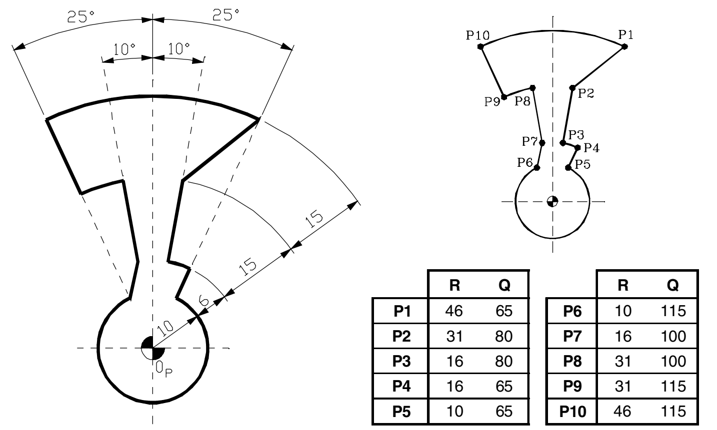

```
N10 G0 G90 X0 Y0 F350
N20 G30 I45 J0
N30 G01 R20 Q110
N40 G02 Q70
N50 G03 Q110 I-6.8404 J18.7938
N60 M30
```

Programming examples



	R	Q
P1	100	0
P2	100	30
P3	50	30
P4	50	60
P5	100	60
P6	100	90

Absolute coordinates	Incremental coordinates	
G00 G90 X0 Y0 F350	G00 G90 X0 Y0 F350	(Point P0)
G01 R100 Q0	G91 G01 R100 Q0	(Point P1)
G03 Q30	G03 Q30	(Point P2)
G01 R50 Q30	G01 R-50	(Point P3)
G03 Q60	G03 Q30	(Point P4)
G01 R100 Q60	G01 R50	(Point P5)
G03 Q90	G03 Q30	(Point P6)
G01 R0 Q90	G01 R-100	(Point P0)
M30	M30	



	R	Q
P1	46	65
P2	31	80
P3	16	80
P4	16	65
P5	10	65

	R	Q
P6	10	115
P7	16	100
P8	31	100
P9	31	115
P10	46	115

Absolute coordinates	Incremental coordinates	
G90 R46 Q65 F350	G90 R46 Q65 F350	(Point P1)
G01 R31 Q80	G91 G01 R-15 Q15	(Point P2)
G01 R16	G01 R-15	(Point P3)
G02 Q65	G02 Q-15	(Point P4)
G01 R10	G01 R-6	(Point P5)
G02 Q115	G02 Q-310	(Point P6)
G01 R16 Q100	G01 R6 Q-15	(Point P7)
G01 R31	G01 R15	(Point P8)
G03 Q115	G03 Q15	(Point P9)
G01 R46	G01 R15	(Point P10)
G02 Q65	G02 Q-50	(Point P0)
M30	M30	

6.3.4 Temporary polar origin shift to the center of arc (G31)

When defining an arc in polar coordinates, the polar origin may be shifted temporarily to the center of the arc.

G31 **Temporary polar origin shift to the center of arc**

Function G31 shifts temporarily the polar origin to the center of the programmed arc. This function only acts in the block that contains it; once the block has been executed, it restores the previous polar.

This function is added to the programmed circular interpolation G2/G3. In this case, at least one of the center coordinates must be programmed.

6.3.5 Arc center in absolute coordinates (G06/G261/G262)

When defining an arc, one may select whether the center position is referred to the starting point of the arc or it is defined in absolute coordinates.

Programming

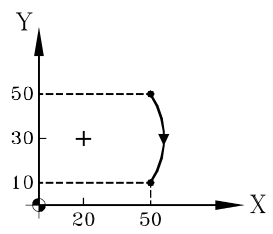
This selection is made using the following functions:

G06	Arc center in absolute coordinates (not modal).
G261	Arc center in absolute coordinates (modal).
G262	Arc center referred to starting point.

G06-G261 Arc center in absolute coordinates

While one of these functions is active, the CNC interprets that the arc center coordinates are referred to the active reference system origin (part zero, polar origin, etc).

Function G261 stays active throughout the program whereas G06 only acts in the block where it has been programmed, therefore it can only be added to a block where a circular interpolation has been defined.



G261
G90 G02 X50 Y10 I20 J30

G261
G91 G02 X0 Y-40 I20 J30

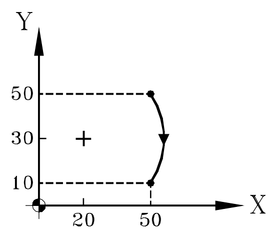
G90 G06 G02 X50 Y10 I20 J30

G91 G06 G02 X0 Y-40 I20 J30

The example shows 4 different ways to define an arc using absolute center coordinates.

G262 Arc center referred to starting point

When this function is active, the CNC interprets that the coordinates of the arc center are referred to the starting point of the arc.



G262
G90 G02 X50 Y10 I-30 J-20

G262
G91 G02 X0 Y-40 I-30 J-20

The example shows two different ways to define an arc by indicating its center with respect to the starting point of the arc.

Properties of the functions

Functions G261 and G262 are modal and incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G262.

6.3.6 Arc center correction (G264/G265)

In order to execute the programmed arc, the CNC calculates the radii of the initial and final points which must be the same. When this is not the case, using center correction it is possible to execute the programmed arc by correcting its its center.

The tolerance allowed for the difference between both radii or for locating the corrected arc center is set by the machine manufacturer [G.M.P. "CIRINERR" and "CIRINFACT"].

Programming

Arc center correction may be turned on and off using the following functions:

G264 Cancellation of arc center correction.

G265 Activation of arc center correction.

G264 **Cancellation of arc center correction**

When the difference between the initial and final radii is within the allowed tolerance, it executes the arc with the radius calculated using the initial point. The center position stays the same.

If the difference between both radii exceeds the allowed tolerance, the relevant error message will be issued.

G265 **Activation of arc center correction.**

If the initial and final arc radii are not the same, the CNC tries to calculate a new center within the set tolerance so as to be able to execute and arc between the programmed points as close as possible to the defined arc.

To calculate whether the error margin is within tolerance or not, the CNC considers two values:

- The absolute error (radius difference).
- The relative error (% over the radius).

If any of these values is within the tolerance set by the OEM, the CNC corrects the center position.

If the CNC cannot the center within those limits, it will issue the pertinent error message.

Properties of the functions

Functions G264 and G265 are modal and incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G265.

6.4 Arc tangent to previous path (G08)

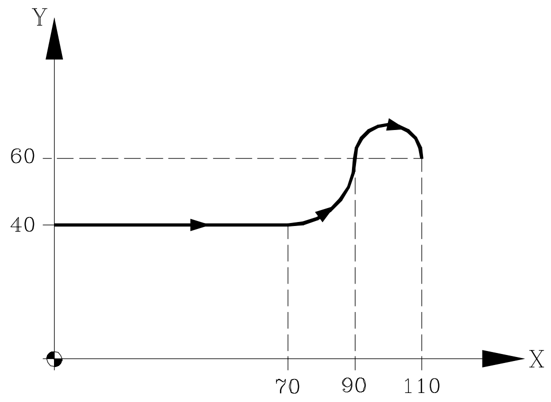
Function G08 may be used to program a circular path tangent to the previous path without having to program the center coordinates (I, J or K).

Programming

Only the coordinates of the arc's final (end) point must be programmed either in polar or Cartesian coordinates along the axes of the work plane.

The previous path may be either linear or circular.

Assuming the starting point is X0 Y40, we would like to program a straight line, then an arc tangent to it and finally an arc tangent to the previous one.



G90 G01 X70

G08 X90 Y60 (Arc tangent to previous path)

G08 X110 (Arc tangent to previous path)

Properties of the function

Function G08 is not modal, consequently, it must be programmed every time when programming an arc tangent to the previous path. After executing it, the CNC restores the G01, G02 or G03 function that was active before.

Function G08 may also be programmed as G8.

Function G08 cannot be used to program full circles because there are infinite solutions.

6.5 Arc defined by three points (G09)

G09 may be used to define an arc by programming the end point and an intermediate point (the initial point of the arc is the starting point of the move). In other words, instead of programming the center coordinates, any intermediate point is programmed.

Coordinates of the end point

It may defined in cartesian or polar coordinates both absolute and incremental.

Coordinates of the intermediate point

It must be defined in cartesian coordinates by the letters "I", "J" or "K" depending on the active plane.

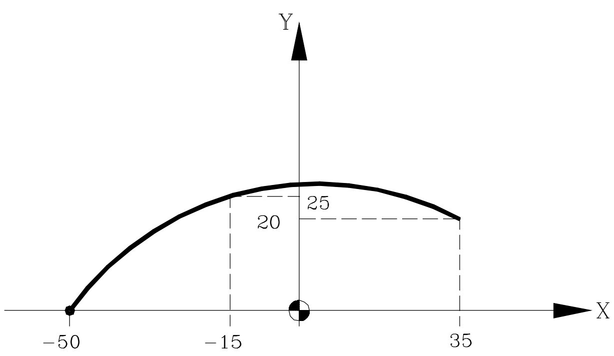
G17 G18 G19	Letters "I", "J" and "K" are associated with the X, Y and Z axes respectively.
G20	Letters "I" and "J" are associated with the abscissa and ordinate axes of the defined plane.

These coordinates are affected by functions G90 and G91.

The programming format depends on the active work plane. In the XY plane is:

XY plane (G17)	G02/G03	X...	Y...	I...	J...
	G02/G03	R...	Q...	I...	J...

Where X-50 Y0 is the starting point.



G09 X35 Y20 I-15 J25

Programming G09 does not require programming the direction of the movement (G02 or G03).

Properties of the function

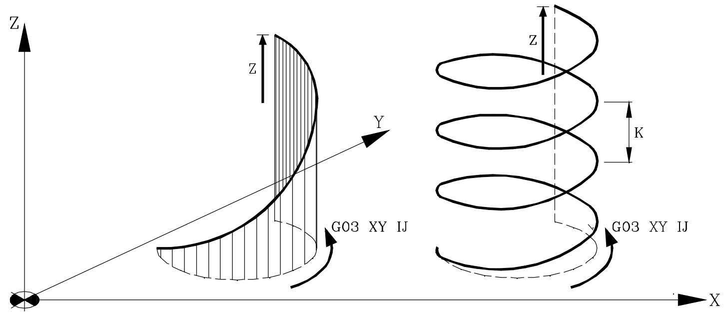
Function G09 is not modal, consequently, it must be programmed every time when programming an arc defined by three points. After executing it, the CNC restores the G01, G02 or G03 function that was active before.

Function G09 may be programmed as G9.

Function G09 may not be used to programmed a full circle because all three points must be different.

6.6 Helical interpolation (G02/G03)

Helical interpolation consists of a circular interpolation in the work plane and a linear movement of the rest of the axes programmed.



(A) **(B)**

(A) Simple helical interpolation.

(B) Multi-turn helical interpolation.

Helical interpolation is programmed in a block whose circular interpolation must be programmed using function G02, G03, G08 or G09.

Programming

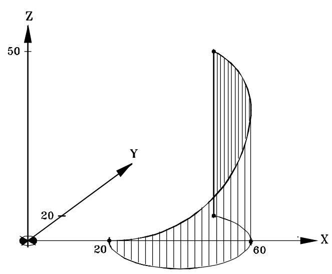
Simple helical interpolation

The helical interpolation is defined by programming the circular interpolation in the active plane and then the linear movement of the other axes.

The programming format depends on the active work plane. In the XY plane is:

XY plane (G17)	G02/G03	X...	Y...	I...	J...	<axes>
	G02/G03	X...	Y...	R...	<axes>	
	G02/G03	R...	Q...	I...	J...	<axes>
	G08	X...	Y...	<axes>		
	G09	X...	Y...	I...	J...	<axes>

Different ways to program a helical interpolation.



G03 X40 Y20 I20 J0 Z50

G03 X40 Y20 R-20 Z50

G03 R44.7213 Q26.565 I20 J0 Z50

G09 X40 Y20 I60 J0 Z50

Starting point: X20 Y0 Z0
End point: X40 Y20 Z50

Programming

Multi-turn helical interpolation

If the helical interpolation is to make several turns, besides programming the circular interpolation in the active work plane and the linear movement of the other axes, the helical pitch must also be programmed.

When defining the center of the circular interpolation, it is not necessary to define the coordinates fo the end point in the work plane. This point will be calculated by the CNC depending on the height and pitch of the helix.

Pass definition

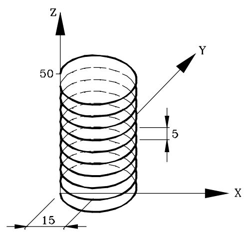
The helical pitch is defined using the letter "I", "J" or "K" associated with 3rd axis of the active work plane.

G17 G18 G19	The pitch is defined with the letter "K" (G17), "J" (G18) or "I" (G19).
G20	The pitch is defined with the letter "K".

The programming format depends on the active work plane. In the XY plane is:

XY plane (G17)	G02/G03	X...	Y...	I...	J...	<axes>	K...
	G02/G03	I...	J...	<axes>	K...		
	G02/G03	R...	Q...	I...	J...	<axes>	K...
	G08	X...	Y...	<axes>	K...		
	G09	X...	Y...	I...	J...	<axes>	K...

Programming a helical interpolation where the starting point is X0 Y0 Z0.




```
G03 X0 Y0 I15 J0 Z50 K5

G03 R0 Q0 I15 J0 Z50 K5
```

6.7 Electronic threading with constant pitch (G33)

For electronic threading, the machine must have a rotary encoder installed on the spindle.

When doing an electronic threading, the CNC does NOT interpolate the movement of the axes with the spindle. Although this type of threading are often carried out along an axis, the CNC allows doing it by interpolating more than one axis at time.

Programming

An electronic threading is programmed with G33 followed by the coordinates of the end point of the thread and the thread pitch.

Coordinates of the end point

It may defined in cartesian or polar coordinates both absolute and incremental.

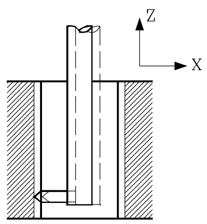
Pass definition

When one of the planes G17, G18 or G19 is active, the letters "I", "J" and "K" will be associated with the X, Y and Z axes respectively.

The threading feedrate depends on the programmed spindle speed and thread pitch (Feedrate = Spindle speed x Pitch).

To make the following thread in a single pass:

Position : X30 Y30 Z0
Depth : 30mm
Pitch : 1.5mm




```
...
S100 M03
G01 G90 X30 Y30 Z0
G33 Z-30 K1.5
M19 S0                  (Spindle orientation)
G91 X3                  (Tool withdrawal)
G90 Z10                 (Withdrawal. Exit the hole)
...
```

The machining feedrate will be: 100x1.5 = 150mm/min.

Considerations

The electronic threading is carried out at 100% of the feedrate "F" and spindle speed "S", and these values cannot be modified from the CNC's operator panel or via PLC.

Properties of the functions

Function G33 is modal and incompatible with G00, G01, G02, G03 and G63.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G00 or G01 as set by the machine manufacturer [G.M.P. "IMOVE"].

6.8 Rígid tapping (G63)

For rigid tapping, the machine must have a rotary encoder installed on the spindle.

When rigid tapping, the CNC interpolates the movement of the longitudinal axis with the spindle.

Programming

To define a rigid tapping, program function G63 and then the coordinates of the end point of the thread which may be defined in Cartesian or polar coordinates. The thread pitch will be calculated by the CNC depending on the active feedrate "F" and spindle speed "S" (Pitch = Feedrate / Spindle speed).

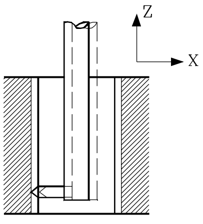
Function G63 starts the spindle in the direction indicated by the programmed speed "S" ignoring the active M3, M4, M5 or M19 functions. A negative turning speed can only be programmed if function G63 is active.

```
...
G94 F300
G01 G90 X30 Y30 Z50
G63 Z20 S200
...
```

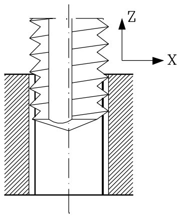
The thread pitch will be: $\frac{F}{S} = \frac{300}{200} = 1{,}5\,mm$

Since G63 does not withdraw the tool automatically after the tap, an inverted tap must be programmed in order to withdraw the tool by inverting the turning direction of the spindle (by changing the sign of the "S" speed). If the thread is made with a cutter tip, the tool may be also be withdrawn by orienting the spindle (M19) and separating the tool tip away from the thread.

To make a 4 mm pitch thread in X30 Y30 Z0 in a single pass with a depth of 30mm.




```
G94 F400
G01 G90 X30 Y30 Z0
G63 Z-30
M19 S0
G91 X3
G90 Z10
```




```
G94 F400
G01 G90 X30 Y30 Z0
G63 Z-30 S100
G63 Z0 S-100
G01 Z10
```

Multiple-entry threads

With this type of threading, it is possible to make threads with several entry points. The positioning for each entry must be defined before each threading operation.

```
...
G90 G01 X0 Y0 Z0 F150
M19 S0                      (First entry at 0º)
G63 Z-50 S150                   (Tapping)
G63 Z0 S-150                    (Withdrawal)
M19 S120                    (Second entry at 120º)
G63 Z-50 S150
G63 Z0 S-150
M19 S240                    (Third entry at 240º)
G63 Z-50 S150
G63 Z0 S-150
...
```

3-entry thread, 50mm deep and 1mm pitch.

Spindle speed behavior

Depending on where the turning speed is defined, the operation will be:

- If the threading speed is defined while G63 is active, the speed will remain active until G63 is canceled, and it will then restore the speed that was active before activating the threading operation.
- If no particular threading speed is defined, it will be executed at the speed active at the time.

The spindle turning direction is determined by the sign of the programmed "S" speed ignoring the active M3, M4, M5 or M19 functions. Programming any of these functions will cancel G63.

Considerations

While rigid tapping, the feedrate may be varied between 0% and 200% using the feedrate override switch on the CNC's operator panel of via PLC. The CNC will adapt the spindle speed in order to keep the interpolation between the axis and the spindle.

Properties of the functions

Function G63 is modal and incompatible with G00, G01, G02, G03 and G33.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G00 or G01 as set by the machine manufacturer [G.M.P. "IMOVE"].

6.9 Manual intervention (G200/G201/G202)

With these functions, it is possible to activate the JOG mode by program; in other words, the axes may be jogged even while executing a program. The movement may be made using handwheels or the JOG keys (incremental or continuous JOG).

Programming

The functions related to manual intervention are:

G200	Exclusive manual intervention.
G201	Activation of additive manual intervention.
G202	Cancellation of additive manual intervention.

The difference between exclusive and additive interventions is that the exclusive one (G200) interrupts the execution of the program to activate the jog mode whereas the additive one (G201) lets you jog an axis while executing the programmed movements.

Feedrate behavior

The feedrate of the jogging movements during manual intervention is independent from the active "F" and may be defined by the operator using instructions in high-level language (as described in the chapter on "*"15 Statements and instructions"*" of this manual); a different feedrate may be set for each work mode (incremental or continuous JOG). If not defined, the movements are carried out at the feedrate set by the machine manufacturer.

The feedrate may be varied between 0% and 200% using the feedrate override switch on the CNC's operator panel which affects the same way the programmed "F" and the feedrate of manual intervention.

Properties of the functions

Functions G201, G202 (modal) and G200 (not modal) are incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G202.

6.9.1 Additive manual intervention (G201/G202)

With additive manual intervention, it is possible to jog the axes using handwheels or the JOG keys (continuous or incremental) while executing the program.

It may be applied on any axis of the machine. It cannot be applied on the spindle even if it can work in positioning mode.

G201 **Activation of additive manual intervention.**

To activate the additive manual intervention, program G201 followed by the axes affected by it using the instruction "#AXIS[<axes>]".

Function G201 must always be followed by the "#AXIS" instruction defining at least one axis.

G202 **Cancellation of additive manual intervention**

To cancel the additive manual intervention, program G202 followed by the axes to be canceled using the instruction AXIS[<axes>].

Programming G202 alone cancels manual intervention on all the axes.

...	
N100 G71 G90 X0 Y0 F400	
N110 G201 #AXIS [X, Z]	(Activate additive manual intervention on the X-Z axes)
N120 G01 X100 Y50	(The X-Z axes may be jogged)
N130 G202 #AXIS [X]	(Cancel manual intervention on X)
N140 G01 X50 Y150	(The Z axis may be jogged)
N150 G202 #AXIS [Z]	(Cancel manual intervention on Z)
...	
N200 G201 #AXIS [X, Y, Z]	(Activate additive manual intervention on the X-Y-Z axes)
N220 G01 X100 Y50	(The X-Y-Z axes may be jogged)
N230 G202	(Cancel intervention on all axes)
...	

Considerations

Axis machine parameters MANFEEDP, IPOFEEDP, MANACCP, IPOACCP determine the feedrate and maximum acceleration permitted for each type of movement (jog or automatic). If the addition of the two exceeds 100%, it will be up to the user to ensure that the two movements are not simultaneous on the same axis because it may cause the dynamics to overshoot.

6.9.2 Exclusive manual intervention (G200)

With exclusive manual intervention, the axes may be jogged using handwheels or JOG keys (continuous or incremental) by interrupting the execution of the program.

(a)

To cancel manual intervention and resume program execution, press the [CYCLE START][(a)] key.

It may be applied on any axis of the machine. It cannot be applied on the spindle even if it can work in positioning mode.

G200 **Exclusive manual intervention**

To activate exclusive manual intervention, program G200 followed by the axes affected by it using the instruction "#AXIS[<axes>]".

Programming G200 alone selects manual intervention on all the axes.

```
...
N100 G71 G90 X0 Y0 F400
N110 G200 #AXIS [X, Z]       (Interrupts program execution. Activates
                             manual intervention on the X-Z axes)
                             (Press the [CYCLE-START])
N120 G01 X100 Y100
N130 G200                    (Interrupts program execution. Activates
                             manual intervention on all axes)
                             (Press the [CYCLE-START])
N140 G01 X50 Y150
N150 G01 X0 Y0
...
```

Considerations

If a manual intervention is executed before a circular interpolation and one of the axes involved in the circular interpolation is jogged, it could issue an error message indicating that a circle has been programmed wrong or it may execute a circle other than the one programmed.

GEOMETRY ASSISTANCE

7

7.1 Square corner (G07/G60)

When working in square corner mode, the CNC does not begin executing the next movement until the axis reaches the programmed position. The CNC considers that the programmed position has been reached when the axis is located within the "in position" zone set by the machine manufacturer (OEM) [A.M.P. "INPOSW"].

Programming

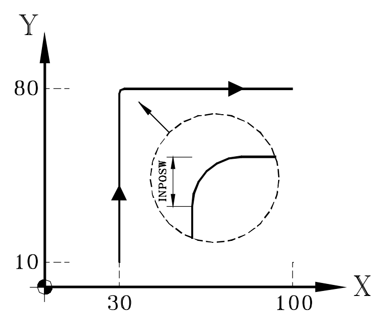
The square corner machining mode may be activated by program using two different functions:

G07	Square corner (modal).
G60	Square corner (not modal).

Function G07 remains active throughout the program whereas function G60 only affects the block that contains it; therefore, it can only be added to a block containing a movement.

```
...
G01 G91 G60 Y70 F500
G01 X70
...
```

```
...
G07
G01 G91 Y70 F500
G01 X70
...
```



The theoretical and real profiles are the same, thus resulting in square corners as shown in the figure.

Properties of the functions

Function G07 is modal and incompatible with G05, G50, G60, G61 and the HSC mode.

Function G60 is not modal. After it is executed, the CNC restores the function G05, G07, G50 or HSC that was previously active.

On power-up, after executing M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G05, G07 or G50 as set by the OEM [G.M.P. "ICORNER"].

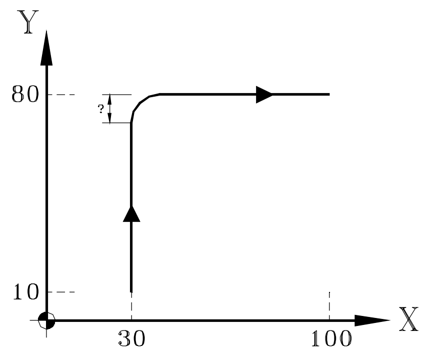
7.2 Semi-rounded corner (G50)

When working in semi-rounded corner, the CNC starts executing the next movement once the theoretical interpolation of the current move is completed without waiting for the axes to be in position. The distance from the programmed position to the position where the next move begins depends on the feedrate of the axis.

Programming

The semi-rounded corner machining mode may be activated by program using function G50.

```
...
G50
G01 G91 Y70 F500
G01 X70
...
```



This function provides rounded corners as shown in the figure.

Properties of the function

Function G50 is modal and incompatible with G05, G07, G60, G61 and the HSC mode.

On power-up, after executing M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G05, G07, G50 or HSC as set by the OEM [G.M.P. "ICORNER"].

7.3 Controlled corner rounding, radius blend, (G05/G61)

When working in round corner, it is possible to control the corners of the programmed profile. How this machining is carried out depends on the type of corner rounding selected.

Programming

The type of corner rounding is selected with the "#ROUNDPAR" instruction and stays active until a different one is selected. Section *"7.3.1 Types of corner rounding"* of this chapter shows a description of the different types of corner rounding available.

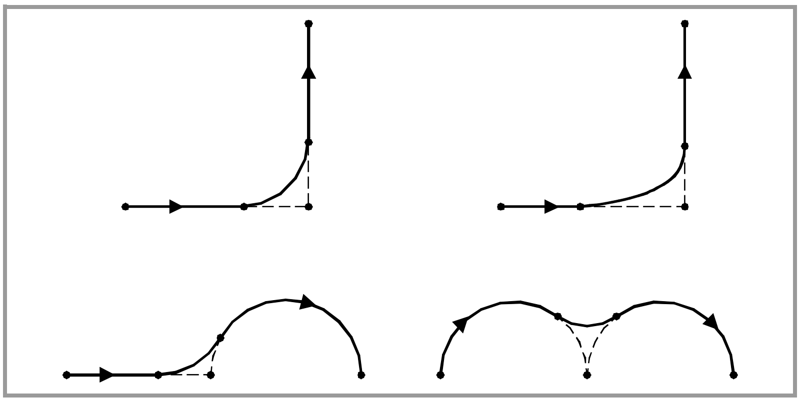
After selecting the type of corner rounding, it may be activated by program using functions:

G05	Control corner rounding, radius blend (modal).
G61	Control corner rounding, radius blend (not modal).

Function G05 remains active throughout the program whereas function G61 only affects the block that contains it; therefore, it can only be added to a block containing a movement.

Considerations

This operation may be applied to any corner, regardless of whether it is defined between straight and/or circular paths.



The corner is machined along a curved path, not with arcs. The shape of the curve depends on the type of corner rounding selected and on the dynamic conditions (feedrate and acceleration) of the axes involved.

Properties of the functions

Function G05 is modal and incompatible with G07, G50, G60, G61 and the HSC mode.

Function G61 is not modal. After it is executed, the CNC restores the function G05, G07, G50 or HSC that was previously active.

On power-up, after executing M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes function G05, G07 or G50 as set by the OEM [G.M.P. "ICORNER"].

7.3.1 Types of corner rounding

There are 5 different corner contouring types. The first 4 execute the different corner rounding types whereas the last one executes a square corner. The last one is aimed at special machines (Laser, water jet, etc.), that use it to avoid "burning" the corner, thus not being recommended for a milling machine.

Corner rounding is selected and defined through the parameters associated with the "#ROUNDPAR" instruction. This instruction may have associated up to 6 parameters whose meaning will depend on the type of corner rounding selected.

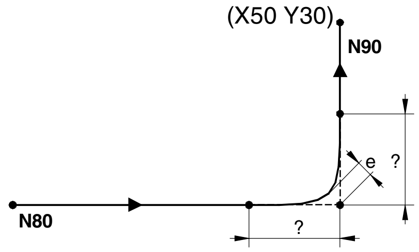
Type 1 **#ROUNDPAR [1,e]**

Set the maximum deviation allowed between the programmed point and the profile resulting from rounding the corner.

The corner is rounded by giving priority to the machining dynamic conditions (feedrate and acceleration). It executes the machining operation that is closer to the programmed point without exceeding the programmed deviation and that does not require decreasing the programmed feedrate "F".

```
...
N70 #ROUNDPAR [1,3]
N80 G01 G91 G61 X50 F850
N90 G01 Y30
...
```

```
...
N70 #ROUNDPAR [1,3]
N75 G05
N80 G01 G91 X50 F850
N90 G01 Y30
...
```



#ROUNDPAR [1,e]

e : Distance between programmed point and real profile.

The distances from the programmed point to the points where the corner rounding begins and ends are calculated automatically and they cannot be greater than half the path programmed in the block. Both distances will be the same, except when one of them is limited to half the programmed path.

For this type of corner rounding, only the values of the first two parameters of the "#ROUNDPAR" instructions are used, therefore, all parameters need not be included.

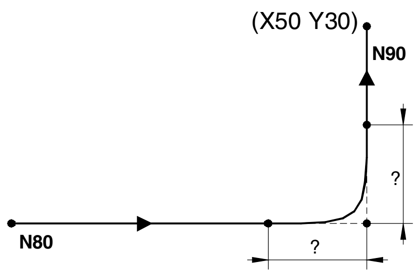
Type 2 **#ROUNDPAR [2,f]**

Set the percentage of the active feedrate "F" to be used to carry out the corner rounding.

It executes the corner rounding closer to the programmed point and that may be machined at the set feedrate percentage.

```
...
N70 #ROUNDPAR [2.40]
N80 G01 G91 G61 X50 F850
N90 G01 Y30
...
```

```
...
N70 #ROUNDPAR [2.40]
N75 G05
N80 G01 G91 X50 F850
N90 G01 Y30
...
```



#ROUNDPAR [2,f]

f : Porcentage of feedrate "F" for corner rounding.

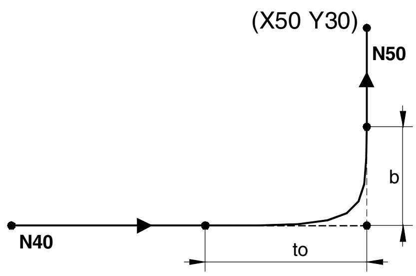
The distances from the programmed point to the points where the corner rounding begins and ends are calculated automatically and they cannot be greater than half the path programmed in the block. Both distances will be the same, except when one of them is limited to half the programmed path.

For this type of corner rounding, only the values of the first two parameters of the "#ROUNDPAR" instructions are used, therefore, all parameters need not be included.

Type 3 **#ROUNDPAR [3,a,b]**

It defines the distance from the programmed point to the points where corner rounding begins and ends.

```
...
N20 #ROUNDPAR [3,10,3]
N30 G00 G90 X0 Y0
N40 G01 X50 F850
N50 Y30
...
```



#ROUNDPAR [3,a,b]

a : Distance to the starting point of corner rounding.

b : Distance to the end point of the corner rounding.

Depending on parameters "a" and "b", a deviation may occur at the programmed profile (as shown in the example).

For this type of corner rounding, only the values of the first three parameters of the "#ROUNDPAR" instructions are used, therefore, all parameters need not be included.

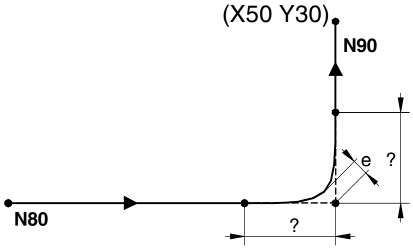
Type 4 **#ROUNDPAR [4,e]**

Set the maximum deviation allowed between the programmed point and the profile resulting from rounding the corner.

The corner is rounded by giving priority to the machining geometrical conditions. The programmed machining operation is executed by decreasing the programmed feedrate "F" if necessary.

```
...
N70 #ROUNDPAR [4.3]
N80 G01 G91 G61 X50 F850
N90 G01 Y30
...
```

```
...
N70 #ROUNDPAR [4.3]
N75 G05
N80 G01 G91 X50 F850
N90 G01 Y30
...
```



#ROUNDPAR [4,e]

e : Distance between programmed point and real profile.

The distances from the programmed point to the points where the corner rounding begins and ends are calculated automatically and they cannot be greater than half the path programmed in the block. Both distances will be the same, except when one of them is limited to half the programmed path.

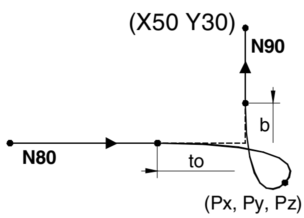
For this type of corner rounding, only the values of the first two parameters of the "#ROUNDPAR" instructions are used, therefore, all parameters need not be included.

Type 5 **#ROUNDPAR [5,a,b,Px,Py,Pz]**

It defines the distance from the programmed point to the points where corner rounding begins and ends. Also set the coordinates of an intermediate point of the corner rounding.

```
...
N70 #ROUNDPAR [5,7,4,55,-15,0]
N80 G01 G91 G61 X40 F850
N90 G01 Y20
...
```

```
...
N70 #ROUNDPAR [5,7,4,55,-15,0]
N75 G05
N80 G01 G91 X40 F850
N90 G01 Y20
...
```

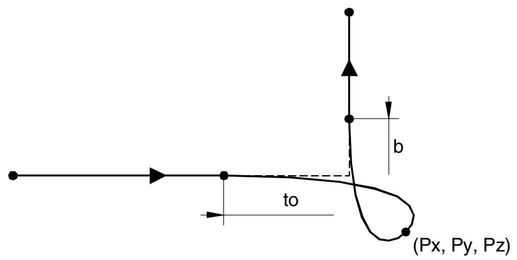


#ROUNDPAR [5,a,b,Px,Py,Pz]

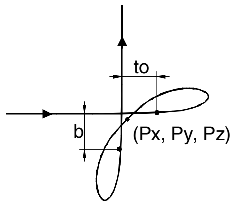
a : Distance to the starting point of corner rounding.
b : Distance to the end point of the corner rounding.
Px : X coordinate of the intermediate point.
Py : Y coordinate of the intermediate point.
Pz : Z coordinate of the intermediate point.

This type of corner rounding only uses the values of the first six parameters of the "#ROUNDPAR" instruction.

In this type of corner rounding, the shape of the curve depends on the position of the intermediate point and on the distance from the programmed point to the starting and ending points of the corner rounding.

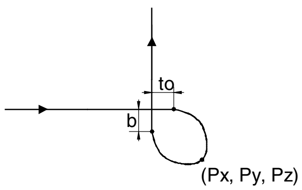



```
...
G92 X0 Y0
G71 G90
#ROUNDPAR [5,-30,-30,55,-5,0]
G01 G61 X50 F850
N90 G01 Y40
...
```



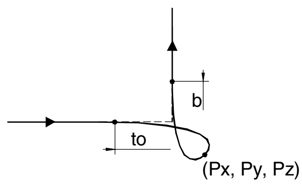
"a" and "b" distances negative and greater (in absolute value) than the distance from the programmed point to the intermediate point on each axis (about 4 times).

```
...
G92 X0 Y0
G71 G90
#ROUNDPAR [5,-5,-5.65,-15.0]
G01 G61 X50 F850
G01 Y40
...
```



"a" and "b" distances, negative and smaller (in absolute value) than the distance from the programmed point to the intermediate point on each axis.

```
...
G92 X0 Y0
G71 G90
#ROUNDPAR [5,5,5,65,-15,0]
G01 G61 X50 F850
G01 Y40
...
```



Positive "a" and "b" distances.

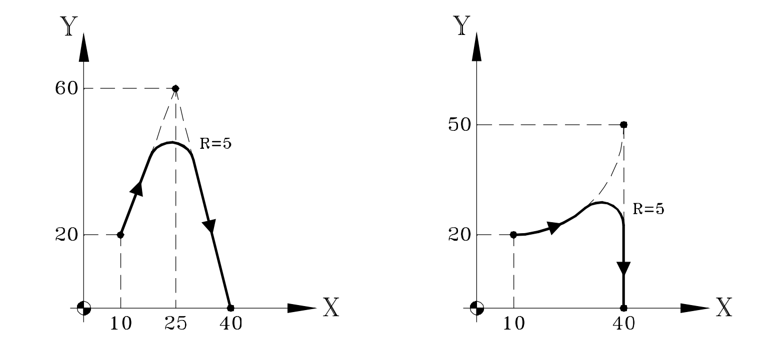
7.4 Corner rounding, radius blend, (G36)

G36 may be used to round a corner with a particular radius without having to calculate either the center or the starting and ending points of the arc.

Programming

The rounding definition must be programmed between the two paths that define the corner to be rounded. These paths may be linear and/ or circular.

The programming format is "G36 I<radius>", where the radius value is programmed in millimeters or in inches, depending on which are the active units.



G01 G90 X25 Y60	G03 G90 X40 Y50 I0 J30
G36 I5	G36 I5
G01 X40 Y0	G01 X40 Y0

Considerations

The "I" value of the rounding radius remains active until another value is programmed, therefore, it won't be necessary to program it in successive rounding operations with the same radius.

The "I" value of the rounding radius is also used by functions:

G37 (Tangential entry) as entry radius.

G38 (Tangential exit) as exit radius.

G39 (corner chamfering) as size of the chamfer.

This means that the rounding radius set in G36 will be the new value of the entry radius, exit radius or chamfer size when programming one of these functions or vice versa.

N10 G01 X10 Y10 F600	
N20 G01 X10 Y50	
N30 G36 I5	(Rounding. Radius=5)
N40 G01 X50 Y50	
N50 G36	(Rounding. Radius=5)
N60 G01 X50 Y10	
N70 G39	(Chamfer. Size=5)

N80 G01 X90 Y10	
N90 G39 I10	(Chamfer. Size=10)
N100 G01 X90 Y50	
N110 G36	(Rounding. Radius=10)
N120 G01 X70 Y50	
N130 M30	

The programmed rounding feedrate depends on the type of movement programmed afterwards:

- If the next movement is in G00, the rounding will be carried out in G00.
- If the next movement is in G01, G02 or G03, the rounding will be carried out at the feedrate programmed in rounding definition block. If no feedrate has been programmed, the rounding will be carried out at the active feedrate.

N10 G01 G94 X10 Y10 F600	
N20 G01 X10 Y50	
N30 G36 I5	(Chamfering in G00)
N40 G00 X50 Y50	
N50 G36	(Chamfer. F=600mm/min.)
N60 G01 X50 Y10	
N70 G36 F300	(Chamfer. F=300mm/min.)
N80 G01 X90 Y10 F600	
N90 M30	

When defining a plane change between the two paths that define a rounding, it is carried out in the plane where the second path is defined.

N10 G01 G17 X10 Y10 Z0 F600	
N20 X10 Y50	(X-Y plane)
N30 G36 I10	
N40 G18	(Z-X plane. The rounding is carried out in this plane)
N50 X10 Z30	
N60 M30	

Properties of the function

Function G36 is not modal, therefore, it must be programmed every time a corner is to be rounded.

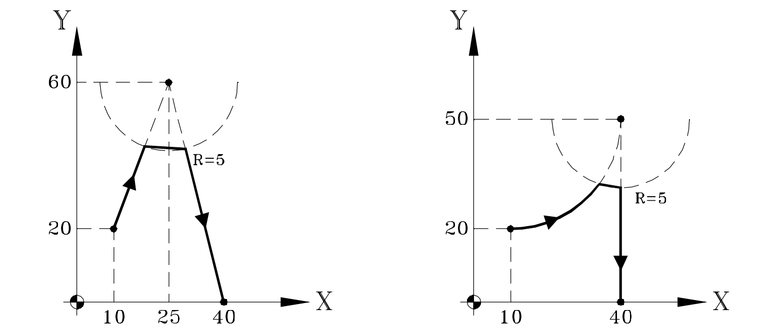
7.5 Corner chamfering, (G39)

Function G39 may be used to insert a chamfer of a particular size without having to calculate the intersection points.

Programming

The chamfer definition must be programmed between the two paths that define the corner to be chamfered. These paths may be linear and/or circular.

The programming format is "G39 I<size>", where the size value is programmed in millimeters or in inches, depending on which are the active units.



G01 G90 X25 Y60	G03 G90 X40 Y50 I0 J30
G39 I5	G39 I5
G01 X40 Y0	G01 X40 Y0

Considerations

The "I" value of the chamfer size remains active until another value is programmed, therefore, it won't be necessary to program it in successive chamfering operations of the same size.

The "I" value of the chamfer size is also used by functions:

G36 (Corner rounding) as rounding radius.

G37 (Tangential entry) as entry radius.

G38 (Tangential exit) as exit radius.

This means that the chamfer size set in G39 will be the new value of the entry radius, exit radius or rounding radius when programming one of these functions or vice versa.

N10 G01 X10 Y10 F600	
N20 G01 X10 Y50	
N30 G36 I5	(Rounding. Radius=5)
N40 G01 X50 Y50	
N50 G36	(Rounding. Radius=5)
N60 G01 X50 Y10	
N70 G39	(Chamfer. Size=5)
N80 G01 X90 Y10	

N90 G39 I10	(Chamfer. Size=10)
N100 G01 X90 Y50	
N110 G36	(Rounding. Radius=10)
N120 G01 X70 Y50	
N130 M30	

The programmed chamfering feedrate depends on the type of movement programmed afterwards:

- If the next movement is in G00, the chamfer will be carried out in G00.
- If the next movement is in G01, G02 or G03, the chamfer will be carried out at the feedrate programmed in chamfer definition block. If no feedrate has been programmed, the chamfer will be carried out at the active feedrate.

N10 G01 G94 X10 Y10 F600	
N20 G01 X10 Y50	
N30 G39 I5	(Chamfering in G00)
N40 G00 X50 Y50	
N50 G39	(Chamfer. F=600mm/min.)
N60 G01 X50 Y10	
N70 G39 F300	(Chamfer. F=300mm/min.)
N80 G01 X90 Y10 F600	
N90 M30	

When defining a plane change between the two paths that define a chamfer, it is carried out in the plane where the second path is defined.

N10 G01 G17 X10 Y10 Z0 F600	
N20 X10 Y50	(X-Y plane)
N30 G39 I10	
N40 G18	(Z-X plane. The chamfer is carried out in this plane)
N50 X10 Z30	
N60 M30	

Properties of the function

Function G39 is not modal, therefore, it must be programmed every time a corner is to be chamfered.

7.6 Tangential entry (G37)

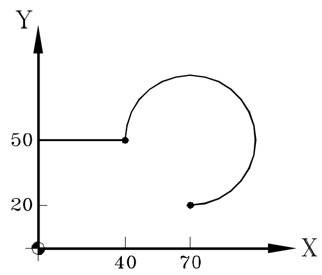
Function G37 may be used to begin machining with a tangential entry of the tool without having to calculate the intersection points.

Programming

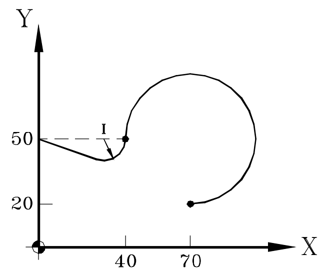
Tangential entry must be programmed alone in the block and after the block whose path is to be modified; this path must be a straight line (G00 or G01).

The programming format is "G37 I<radius>", where the radius value is programmed in millimeters or in inches, depending on which are the active units.

The linear path before the tangential entry must have a length equal to or greater than twice the entry radius. Likewise, the radius must be positive and when working with tool radius compensation, it must be greater than the tool radius.



G01 G90 X40 Y50 F800
G02 X70 Y20 I30 J0



G01 G90 X40 Y50 F800
G37 I10
G02 X70 Y20 I30 J0

Considerations

The "I" value of the tangential entry radius remains active until another value is programmed, therefore, it won't be necessary to program it in successive tangential entries with the same radius.

The "I" value of the entry radius is also used by functions:

G36 (Corner rounding) as rounding radius.

G38 (Tangential exit) as exit radius.

G39 (corner chamfering) as size of the chamfer.

This means that the entry radius set in G37 will be the new value of the exit radius, rounding radius or chamfer size when programming these functions or vice versa.

Properties of the function

Function G37 is not modal, therefore, it must be programmed every time a tangential entry is to be carried out.

7.7 Tangential exit (G38)

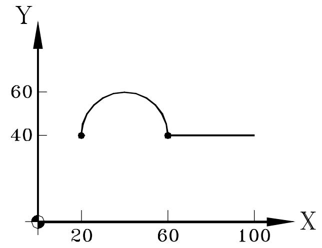
Function G38 may be used to end machining with a tangential exit of the tool without having to calculate the intersection points.

Programming

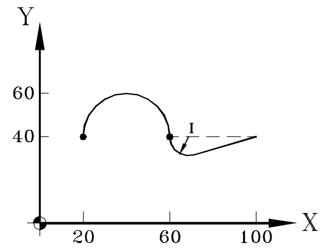
Tangential exit must be programmed alone in the block and before the block whose path is to be modified; this path must be a straight line (G00 or G01).

The programming format is "G38 I<radius>", where the radius value is programmed in millimeters or in inches, depending on which are the active units.

The linear path after the tangential exit must have a length equal to or greater than twice the exit radius. Likewise, the radius must be positive and when working with tool radius compensation, it must be greater than the tool radius.



G02 X60 Y40 I20 J0 F800
G01 X100



G02 X60 Y40 I20 J0 F800
G38 I10
G01 X100

Considerations

The "I" value of the tangential exit radius remains active until another value is programmed, therefore, it won't be necessary to program it in successive tangential exits with the same radius.

The "I" value of the exit radius is also used by functions:

G36 (Corner rounding) as rounding radius.

G37 (Tangential entry) as entry radius.

G39 (corner chamfering) as size of the chamfer.

This means that the exit radius set in G38 will be the new value of the entry radius, rounding radius or chamfer size when programming these functions or vice versa.

Properties of the function

Function G38 is not modal, therefore, it must be programmed every time a tangential exit is to be carried out.

7.8 Mirror image (G11, G12, G13, G10, G14)

Mirror image may be used to repeat the programmed machining operation in a symmetrical position with respect one or more axes. When using with mirror image, the movements of the axes where mirror image is applied are executed with the opposite sign.

Programming

Mirror image may be applied by program using these functions:

G10	Mirror image cancellation.
G11	Mirror image in X.
G12	Mirror image in Y.
G13	Mirror image in Z.
G14	Mirror image in the programmed directions.

G10 Mirror image cancellation

It cancels mirror image on all axes, including the mirror image activated with G14.

If it is added to a path defining block, the mirror image will be canceled before the movement.

G11 to G13 Mirror image on X, on Y or on Z

Functions G11, G12 and G13 activate mirror image on the X, Y and Z axis respectively. These functions do not cancel each other, thus being possible to keep mirror image active on several axes at the same time.

If they are added to a path defining block, the mirror image will be activated before the movement.

```
G11
    (Mirror image on the X axis)
G12
    (Mirror image on the Y axis. The one on the X axis remains active)
...
G10
    (Mirror image cancellation on all the axes)
```

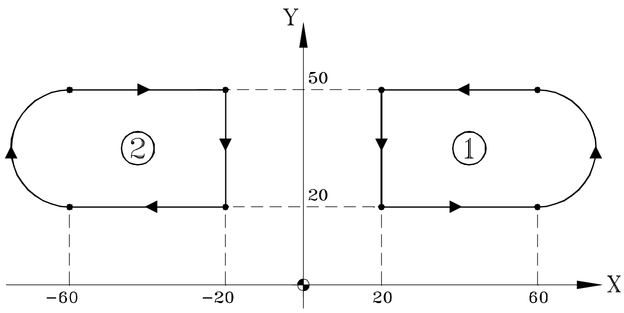
G14 Mirror image in the programmed directions

It may be used to activate or cancel mirror image on any axis. The activation or cancellation is defined by programming function G14 and then, the axes next to the value that determines whether to activate (<axis>=-1) or to cancel (<axis>=1) mirror image on that axis.

```
G14 X-1 V-1
    (Mirror image on the X and V axes)
G14 X1
    (Mirror image cancellation on the X axis. The one on the V axis remains
    active)
...
G14 V1
    (Mirror image cancellation on the V axis)
```

Considerations

When machining a profile with a mirror image, the machining direction is opposite to that of the programmed profile. If this profile has been defined with tool radius compensation, when activating the mirror image, the CNC will change the type of compensation (G41 or G42) to obtain the programmed profile.



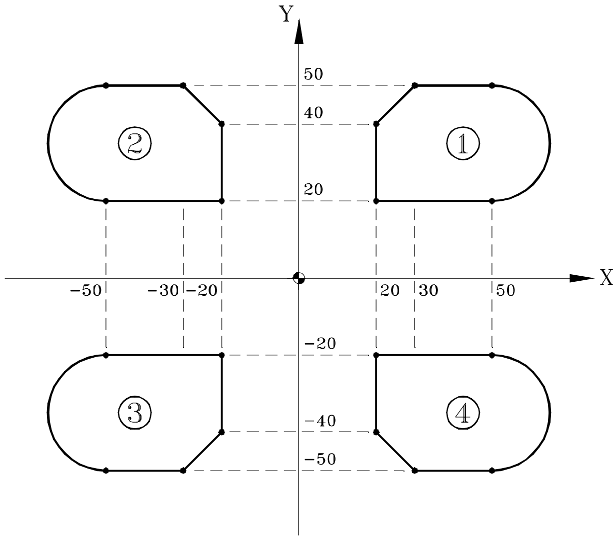
%PROGRAM	(Main program)
G00 G90 X0 Y0 Z20	
...	(Machining of profile 1)
G11	(Mirror image on X).
...	(Machining of profile 2)
G10	(Mirror image cancellation on all the axes)
M30	

Properties of the functions

Functions G11, G12, G13 and G14 are modal. Once mirror image is active on an axis, it stays active until canceled with G10 or G14.

Functions G10 and G14 are incompatible with each other as well as with G11, G12 and G13.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G10.




```
%L PROFILE              ("PROFILE" subroutine definition)
 N10 G00 X10 Y10
 N20 G01 Z0 F400
 N30 G01 X20 Y20 F850
 N40 X50
 N50 G03 X50 Y50 R15
 N60 G01 X30
 N70 X20 Y40
 N80 Y20
 N90 X10 Y10
 N100 Z10 F400
M29                     (End of subroutine)

%PROGRAM                (Main program)
 N10 G0 X0 Y0 Z10
 N20 LL PROFILE         (Call to a subroutine. Profile 1)
 N30 G11                (Mirror image on X).
 N40 LL PROFILE         (Call to a subroutine. Profile 2)
 N50 G12                (Mirror image on X and Y).
 N60 LL PROFILE         (Call to a subroutine. Profile 3)
 N70 G14 X1             (Mirror image cancellation on the X axis)
 N80 LL PROFILE         (Call to a subroutine. Profile 4)
 N90 G10                (Mirror image cancellation on all the axes)
 N100 G00 X0 Y0 Z50
M30
```

7.9 Coordinate system rotation, pattern rotation, (G73)

Function G73 may be used to rotate the coordinate system taking as rotation center the active reference system (part zero) or the programmed rotation center.

Programming

The coordinate system rotation must be programmed alone in the block. The programming format is "G73 Q I J", where:

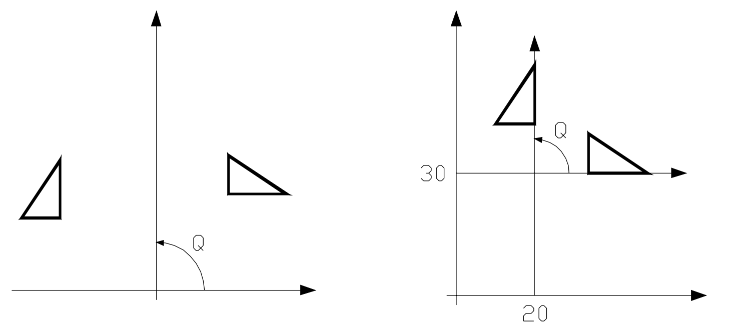
Q — Indicates the rotation angle in degrees.

I, J — They define the abscissa and ordinate of the rotation center. They must be defined in absolute coordinates referred to part zero.

When programmed, both parameters must be programmed.

If not programmed, the part zero will be assume as the rotation center.

To cancel the coordinate (pattern) rotation, program function G73 alone, with no additional data.



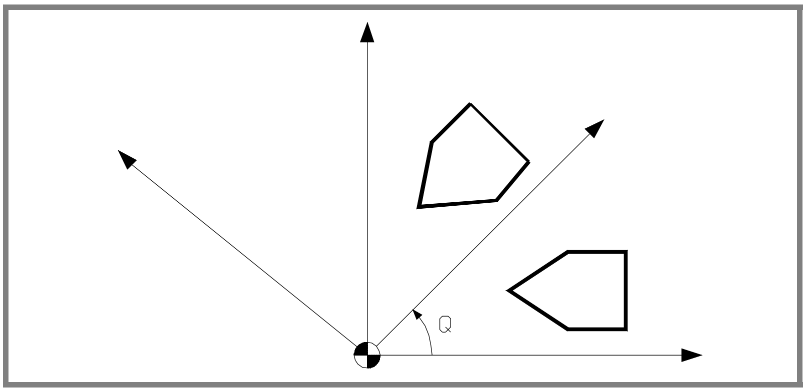
G73 Q90 | G73 Q90 I20 J30

Therefore, function G73 may be programmed as follows:

G73 Q I J	Rotate "Q" degrees with the center at abscissa "I" and ordinate "J" referred to part zero.
G73 Q	Rotate "Q" degrees with the center at part zero.
G73	Cancellation of coordinate (pattern) rotation.

Considerations

Function G73 is incremental; i.e. the various "Q" values programmed are added up.



The "I" and "J" values are affected by the active mirror images. If any mirror image function is active, the CNC applies first the mirror image and then the coordinate system rotation.

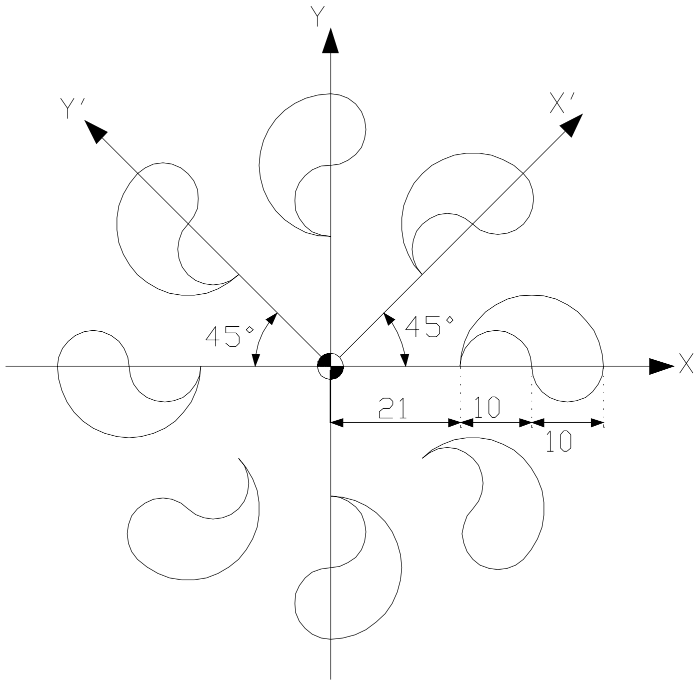
Properties of the function

Function G73 is modal. The coordinate rotation stays active until it is canceled by function G73 or until the work plane is changed.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC cancels the active coordinate system (pattern) rotation.

Programming example

Assuming the initial point is X0 Y0:




```
%L PROFILE          (Subroutine with the profile)
 G01 X21 Y0 F300
 G02 Q0 I5 J0
 G03 Q0 I5 J0
 G03 Q180 I-10 J0
M29                 (End of subroutine)

%PROGRAM            (Program)
$FOR P0=1, 8, 1     (Repeats the profile and the pattern rotation 8 times)
 LL PROFILE         (Machining of the profile)
 G73 Q45            (Coordinate rotation)
$ENDFOR
M30
```

7.10 General scaling factor

It may be used to enlarge or reduce the scale of the programmed paths and contours. This permits using a single program to make sets of similar profiles of different dimensions.

The general scaling factor is applied to all the axes of the channel. After activating the scaling factor, all the programmed coordinates will be multiplied by the defined scaling factor until it is canceled or a new scaling factor is programmed.

Activate the scaling factor

The general scaling factor may be activated using the commands G72 or #SCALE. Either command may be used.

Although there are two different commands, the scaling factor is the same; i.e. the scaling factor programmed with G72 modifies the one programmed with #SCALE and vice versa.

Programming with G72.

Program function G72 and then the scaling factor set by parameter S as follows.

```
G72 S<scale>
```

Programming function G72 alone or a scaling factor of ·1· cancels the active scaling factor.

Parameter "S" that sets the scaling factor must be programmed after function G72. If programmed before, it will be interpreted as spindle speed.

Programming with #SCALE.

Program the instruction #SCALE and then the scaling factor as follows. The brackets must be programmed.

```
#SCALE [<scale>]
```

Programming a scaling factor of ·1· cancels the active scaling factor.

```
#G72 S2
#SCALE [3]
#G72
#SCALE [1]
```

Cancel the scaling factor

The general scaling factor is canceled using the same commands G72 or #SCALE, setting a scaling factor of ·1·.

When using function G72, the scaling factor is also canceled by programming it alone in the block.

Considerations

Activating the machine coordinate system (#MCS ON) cancels the scaling factor temporarily until the machine coordinate system is canceled (#MCS OFF).

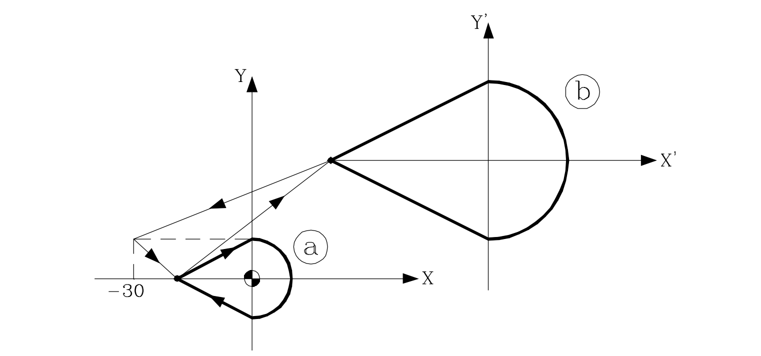
While the machine coordinate system is active, the scaling factor can neither be activated nor modified.

Properties

The scaling factor stays active until canceled with another scaling factor.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC cancels the active scaling factor.

Programming example




```
%L PROFILE                  (Profile to be machined)
G90 X-19 Y0
G01 X0 Y10 F150
G02 X0 Y-10 I0 J-10
G01 X-19 Y0
M29

%PROGRAM
G00 X-30 Y10
#CALL PROFILE               (Machining of profile "a")
G92 X-79 Y-30               (Coordinate preset)
#SCALE [2]                  (Applies a scaling factor of 2)
#CALL PROFILE               (Machining of profile "b")
#SCALE [1]                  (Cancels the scaling factor)
M30
```

ADDITIONAL PREPARATORY FUNCTIONS

8

8.1 Dwell (G04)

The dwell may be used to interrupt the execution of the program for the specified period of time.

Programming

The value of the dwell is given in seconds and may be programmed with the following expressions:

"G04 K<time>"	(or also "G04 < time> "when the time is programmed with a constant)
"#TIME [<time>]"	(or also "#TIME < time>" when the time is programmed with either a constant or a parameter)

Different ways to program a dwell using function G04 and the #TIME instruction.

G04 K0.5	(0.5 second dwel)
G04 5	(5 second dwel)
...	
P1=3	
G04 KP1	(3 second dwel)
G04 K[P1+7]	(10 second dwel)
...	
#TIME 1	(1 second dwel)
...	
P1=2	
#TIME P1	(2 second dwel)
#TIME [P1+7]	(9 second dwel)

Properties of the function

Function G04 is not modal, therefore, it must be programmed every time a dwell is desired.

Function G04 may also be programmed as G4.

8.2 Software limits by program (G198-G199)

The software limits for each axis may be changed by program using the following functions:

G198 Setting of lower software travel limits.

G199 Setting of upper software travel limits.

When programming G198 or G199, the CNC interprets that the coordinates programmed next set the new software limits.

```
G198 X-1000 Y-1000
    (New lower limits X=-1000 Y=-1000)
G199 X1000 Y1000
    (New upper limits X=1000 Y=1000)
```

Depending on the active work mode G90 or G91, the position of the new limits will be defined in absolute coordinates (G90) in the machine reference system or in incremental coordinates (G91) referred to the current active limits.

```
G90
G198 X-800
    (New lower limit X=-800)
G199 X500
    (New upper limit X=500)
G90 X-800
G91
G198 X-700
    (New incremental lower limit X=-1500)
```

Considerations

Both limits may be positive or negative; but the lower limits must always be smaller than the upper ones.

If after setting the new limits, an axis positions beyond them, it will be possible to move that axis towards the work zone (between those limits).

Properties of the functions

Functions G198 and G199 are modal and incompatible with each other.

On power-up or after validating the axis machine parameters the CNC assumes the software limits set by the manufacturer of the machine.

After an M02 or M30 and after an EMERGENCY or a RESET, the CNC maintains the software limits set by G198 and G199.

8.3 Hirth axes (G170-G171)

Hirth axes may be canceled and activated by program. When a Hirth axis is active, it can only reach concrete positions whereas when deactivated, it behaves like a normal rotary or linear axis and can reach any position.

Programming

Hirth axes may be canceled and activated using:

G170 Hirth axes OFF.

G171 Hirth axes ON.

To activate or cancel a Hirth axis, program its relevant function and then the axes to be activated or canceled and the number indicating the order in which those axes will be activated.

G171 B1 C2	(Activate B and C axes as Hirth axes)
G01 B50 C20	(Interpolate both axes)
...	
G170 B1	(B axis deactivation)
G01 X100 B33	

Assuming that the B and C axes have been set as rotary Hirth axes with a 10º pitch.

If when activating a Hirth axis, it is located in the wrong position, the CNC will issue a warning so the operator can turn it to a correct position.

Considerations

A Hirth axis must always be positioned at specific positions. When positioning, the active zero offset (preset or zero offset) is taken into consideration.

Both linear and rotary axes may be Hirth. Only those axes defined as Hirth axes by the machine manufacturer [A.M.P. "HIRTH"] may be activated as Hirth axes.

Properties of the functions

Functions G170 and G171 are modal and incompatible with each other.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC activates all the Hirth axes.

8.4 OEM subroutines (G180-G189)

The OEM subroutines are defined by the machine manufacturer. The machine manufacturer may define up to 10 subroutines and associate them with functions G180 through G189 in such a way that when executing one of these functions its associated subroutine will also be executed.

Executing the subroutine associated with one these functions generates a new nesting level for local parameters (up to 7 nesting levels)

Programming

Functions G180 through G189 allow initializing local parameters of the subroutine. The parameter values may be defined after the subroutine calling function and may be defined using the parameter numbers P0-P25 or their letters A-Z (except "Ñ"), "A" for P0 and "Z" for P25.

```
%PROGRAM
 F1000
 P0=10 P1=20 P2=30
 G1 XP0 YP1 ZP2
 G180 P0=100 P1=200 C300     (Initialize parameters)
M30

%SUB_180                     (Subroutine associated with G180)
 G1 XP0 YP1 ZP2
M29
```

In the main program, the axes move to X10 Y20 Z30. Executing the subroutine, they move to X100 Y200 Z300.

Besides initializing the parameters, any other type of additional information may be added to these functions, even movements. This information must be programmed before the subroutine calling function; otherwise, the data will considered as for initializing the parameters.

The associated subroutine is executed once the execution of the rest of the information programmed in the block has ended.

```
...
G01 X50 F450 G180 P0=15 P1=20
...
```

It executes the programmed movement and, then, the subroutine associated with G180 and setting parameters P0 and P1.

```
...
G180 P0=15 P1=20 G01 X50 F450
...
```

All the data is interpreted as parameter setting, where P6(G)=1, P23(X)=50 and P5(F)=450.

Considerations

Since a subroutine may be called upon from the main program (or a subroutine) and another subroutine from this one and so on, the CNC limits the number of these calls to a maximum of 20 nesting levels. When using local parameters in the subroutines, besides generating a new nesting level, it will also generate a new nesting level for the local parameters; there may be up to 7 nesting levels of local parameters within the 20 nesting levels of the subroutines.

Properties of the functions

Functions G180 through G189 are not modal.

8.5 Changing of parameter range of an axis (G112)

The CNC may have up to 4 sets of parameters for each axis to define different dynamic characteristics (acceleration, gains, etc.) for each of them.

The parameter set may be selected by program using function G112. This function does not carry out any physical change on the machine (gear change), it only assumes the parameters of the active set.

When using Sercos axes, function G112 also involves changing the drive's velocity gear.

Programming

Changing the parameter range of the axes.

To assume a different set of parameter, program G112 and then the axes and the new parameter set to be selected for each one of them.

```
...
G112 X2 Y3        (Selects the 2ndnd set of parameters for the X axis, and
                  the 3rd one for the Y axis)
....
```

Changing the parameter set for the spindle.

In this case, changing the parameter set will be used when working in positioning mode (M19). When working in speed mode (M03/M04), function G112 will only change the parameter set; it is NOT the same as functions M41 through M44 because it does not make a physical gear change.

```
...
G112 S2           (Selects the 2nd set of spindle parameters)
....
```

When making a gear change using M41 through M44, it is not necessary to program G112.

Properties of the function

Function G112 is modal.

After validating the machine parameters, every time a program is executed from the automatic mode, on power-up, after executing an M02 or M30, after an EMERGENCY or a RESET, the CNC acts as follows depending on the value assigned to machine parameter "DEFAULTSET".

If DEFAULTSET is other than 0, it maintains the range defined by G112. Otherwise, it assumes the range defined by machine parameter DEFAULTSET.

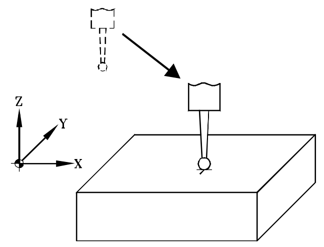
8.6 Probing (G100)

With function G100, it is possible to program movements that will end when the CNC receives the probe signal.

Operation

The probing movement is defined using function G100 followed by the coordinates of the probe's target point.

The probe will move along the programmed path until the CNC receives the signal from the probe or until the programmed position is reached. At that point, the block will be completed and the CNC will assume the current axis position as the theoretical position.




```
...
G100 X50 Y20 Z0 F150
...
```

If the CNC receives the probe signal before reaching the programmed target point, using G101 the CNC will assume as the theoretical position of the axes the programmed coordinate . Ver ***"8.6.1 Include/ exclude probe offset (G101/G102)"*** en la página 152.

Feedrate behavior

The probing feedrate will be the active "F" and this feedrate will be limited by machine parameter PROBEFEED of each probing axis. This value may also be limited by parameters PROBERANGE and PROBEDELAY so if the acceleration and jerk of the axis are active, it will always respect the maximum probing distance.

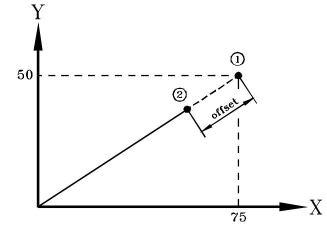
The programmed feedrate "F" may be varied between 0% and 200% using the selector switch on the CNC's operator panel or it may be selected by program or by PLC. However, the maximum override is limited by the machine manufacturer [G.M.P. "MAXOVR"].

Properties of the function

Function G100 is not modal, therefore it must be programmed whenever a new probing movement is desired.

8.6.1 Include/exclude probe offset (G101/G102)

The probe offset is the difference between the programmed coordinate and the coordinate reached by the probe.



...
G100 X75 Y50 F200
...

(1)	Programmed coordinate.
(2)	Probe signal (coordinate reached).
offset	Difference between the programmed coordinate and the one reached

Programming

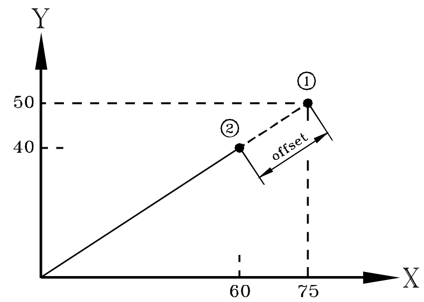
The functions associated with the probe offset are:

G101	Include probe offset.
G102	Exclude probe offset.

G101 - Include offset resulting from the measurement

With this function, the CNC will take into account the probe offset to set the theoretical axis positions; in other words, the CNC will assume as theoretical axis position the programmed coordinate (position reached by the probe + probe offset).

The offset inclusion is determined by programming G101 followed by the axes whose offset is to be included and the inclusion factor of each one. This factor indicates how many times the offset is included.



G100 X75 Y50 F200	
G101 X1 Y1	(It assumes X75 Y50)
	(X=60+offset*1) (Y=40+offset*1)
G100 X75 Y50 F200	
G101 X3 Y2	(It assumes X105 Y60)
	(X=60+offset*3) (Y=40+offset*2)

Function G101 can only be executed after probing.

G102 - Exclude offset resulting from the measurement

Using this function, the CNC will ignore the probe offset to set the theoretical axis position.

The exclusion of the offset is defined by programming function G102 followed by the axes whose offsets are to be excluded.

```
...
G102 X Y        (Exclude the offsets of the X and Y axes)
...
G102            (Exclude the offsets of all the axes)
...
```

Programming G102 alone will cancel the offsets of all the axes.

Properties of the functions

Functions G101 and G102 are modal and incompatible with each other.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC maintains the values programmed with G101.

9 TOOL COMPENSATION

Tool compensation allows programming the machining contour based on the dimensions of the part without taking into account the dimensions of the tool that will be used later on. This way, there is no need to calculate and redefine the tool path depending on the radius and length of each tool.

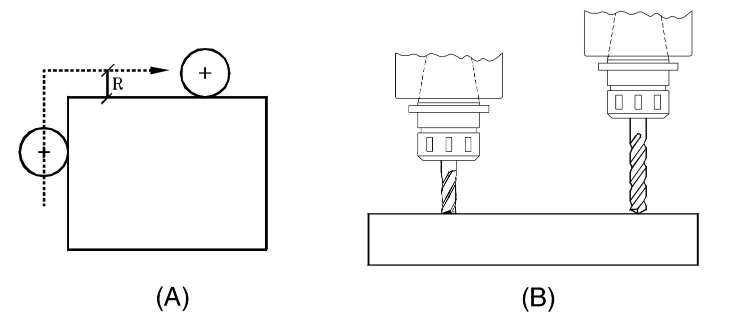
Types of compensation

Tool radius compensation.

When working with tool radius compensation, the tool center follows the programmed path at a distance equal to the tool radius. Thus obtaining the right dimensions of the programmed part.

Tool length compensation.

When working with tool length compensation, the CNC compensates for the length difference between the different programmed tools.



(A) *Tool radius compensation.*
(B) *Tool length compensation.*

Compensation values

The compensation values applied in each case is calculated from the tool dimensions.

- In tool radius compensation, the applied value is the sum of the radius and radius wear of the selected tool.
- In tool length compensation, the applied value is the sum of the length and length wear of the selected tool.

The tool "T" and the tool offset "D", containing the tool dimensions, may be selected anywhere in the program, even while tool compensation is active. If no tool offset is selected, the CNC assumes tool offset "D1".

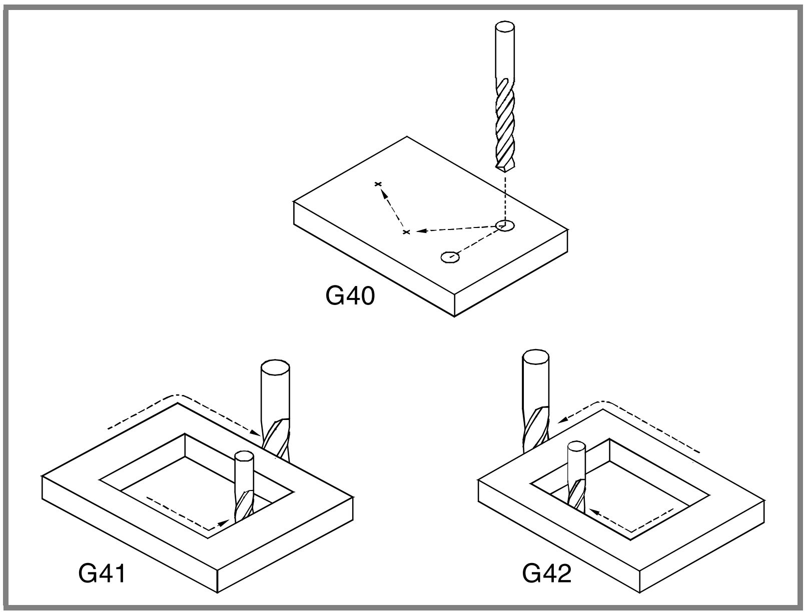
9.1 Tool radius compensation

Radius compensation is applied in the active work plane, previously selected using functions G17 (XY plane), G18 (ZX plane), G19 (YZ plane) or G20 (user defined plane).

Programming

The functions for selecting tool radius compensation are:

G41	Left-hand tool radius compensation.
G42	Right-hand tool radius compensation.
G40	Cancellation of tool radius compensation.



Depending on the type of compensation selected (G41/G42), the tool will be placed to the left or to the right of the programmed path along the machining direction and at a distance equal to the tool radius. If no tool compensation is selected (G40), the CNC will place the tool center right on the programmed tool path.

Being tool radius compensation active, the CNC analyzes in advance the blocks to be executed in order to detect compensation errors related to steps, null arcs, etc. When detected, the CNC will not execute the blocks that cause them and the screen will display a warning to let the operator know that the programmed profile has been modified. A warning will come up for every profile correction made.

Properties of the functions

Functions G40,G41 and G42 are modal and incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G40.

9.1.1 Functions associates with radius compensation

The functions associated with tool compensation may be programmed anywhere in the program, even while tool radius compensation is active.

Selecting the type of transition between blocks

The transition between blocks determines how the compensated paths are joined together.

Programming

The type of transition may be selected from the program by means of the following functions:

G136 Circular transition between blocks.

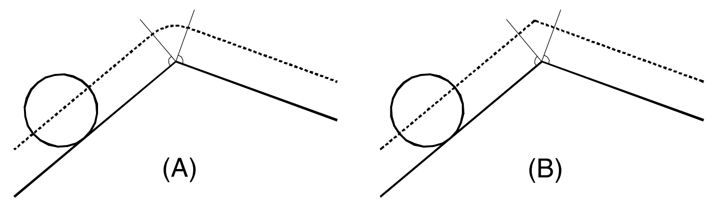
G137 Linear transition between blocks.

G136 **Circular transition between blocks.**

Being function G136 active, the CNC joins the compensated paths using circular paths.

G137 **Linear transition between blocks.**

Being function G137 active, the CNC joins the compensated paths using linear paths.



(A) *Circular transition between blocks (G136).*
(B) *Linear transition between blocks (G137).*

Remarks

Later sections of this chapter offer graphic descriptions of how different paths are joined, depending on the type of transition selected (G136/G137).

Properties of the functions

Functions G136 and G137 are modal and incompatible with each other.

On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G136 or G137 depending on the value of machine parameter IRCOMP.

How tool radius is activated and canceled

The functions associated with the strategy for activating and canceling establish how tool radius compensation starts and ends.

Programming

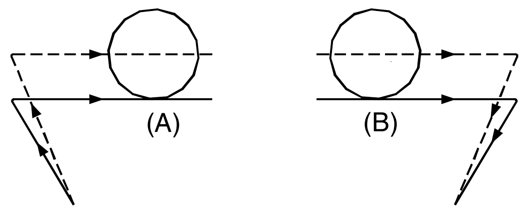
The type of strategy may be selected from the program by means of the following functions:

G138	Direct activation/cancellation of tool compensation.
G139	Indirect activation/cancellation of tool compensation.

G138 **Direct activation/cancellation of tool compensation.**

When compensation is turned on, the tool moves directly to the perpendicular of the next path (without contouring the corner).

When compensation is turned off, the tool moves directly to the programmed end point (without counting the corner).

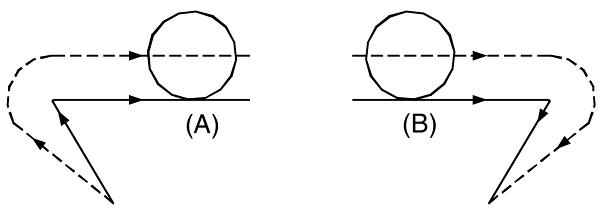


***(A)** Beginning of compensation.*
***(B)** End of compensation.*

G139 **Indirect activation/cancellation of tool compensation.**

When compensation is turned on, the tool moves to the perpendicular of the next path contouring the corner.

When compensation is turned off, the tool moves to the end point contouring the corner.



***(A)** Beginning of compensation.*
***(B)** End of compensation.*

The way the tool goes around the corner depends on the type of transition selected (G136/G37).

Remarks

Later sections of this chapter offer a graphic description of how tool radius compensation begins and ends depending on the selected type of compensation ON/OFF (G138/G139).

Properties of the functions

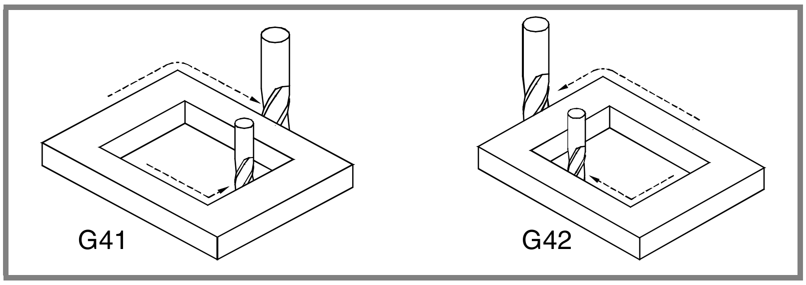
Functions G138 and G139 are modal and incompatible with each other.

On power-up, after an M02 or M30 and after an EMERGENCY or a RESET, the CNC assumes the function set by the machine manufacturer [G.M.P. "IRCOMP"].

9.1.2 Beginning of tool radius compensation

Tool radius compensation is selected with these functions:

G41	Left-hand tool radius compensation.
G42	Right-hand tool radius compensation.



After executing one of these functions, radius compensation will be active for the next movement in the work plane, that must be a linear movement.

The way radius compensation will begin depends on how it is activated G138/G139 and on the type of transition G136/G137 selected:

- G139/G136

  The tool moves to the perpendicular of the next path contouring the corner along a circular path.

- G139/G137

  The tool moves to the perpendicular of the next path contouring the corner along linear paths.

- G138

  The tool moves directly to the perpendicular of the next path. Regardless of the type of transition (G136/G137) programmed.

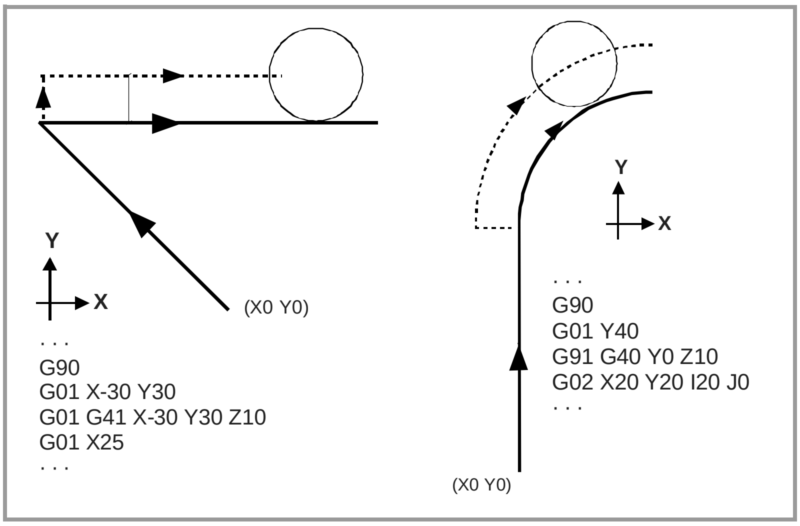
The following tables show the different ways tool compensation may begin, depending on the selected functions. The programmed path is shown with solid line and the compensated path with dashed line.

Beginning of the compensation without programmed movement

After activating the compensation, it may occur that the axes of the plane will not be involved in the first motion block. For example, because they have not been programmed, or the current tool position has been programmed or an incremental movement has been programmed.

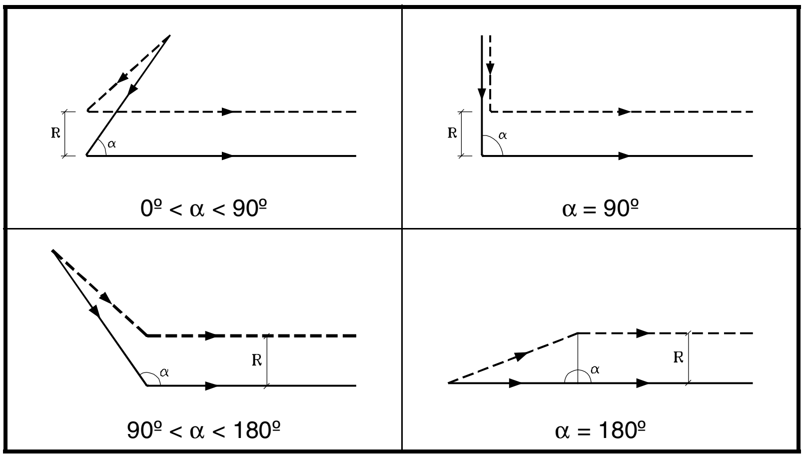
In this case, the compensation is applied at the same point where the tool is, as follows. Depending on the first movement programmed in the plane, the tool moves perpendicular to the path on its starting point.

The first movement programmed in the plane may be either linear or circular.

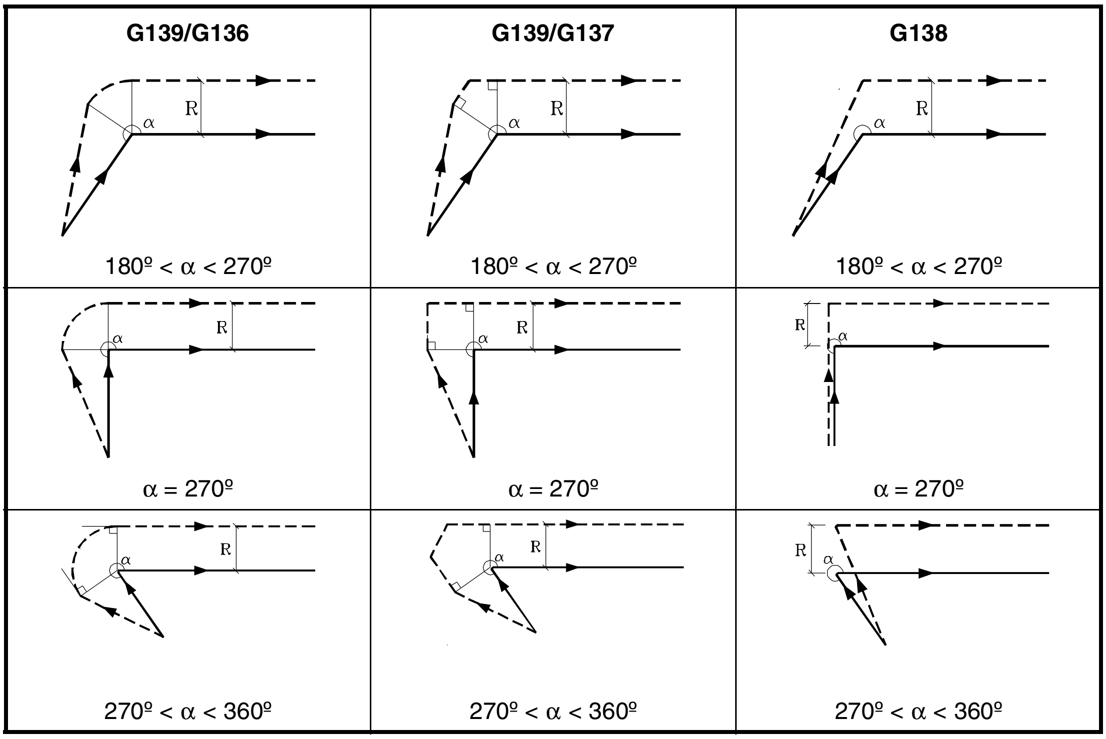


STRAIGHT-TO-STRAIGHT path

When the angle between paths is smaller than or equal to 180º, the way radius compensation is activated is independent from the functions G136/G137 or G138/G139 selected.

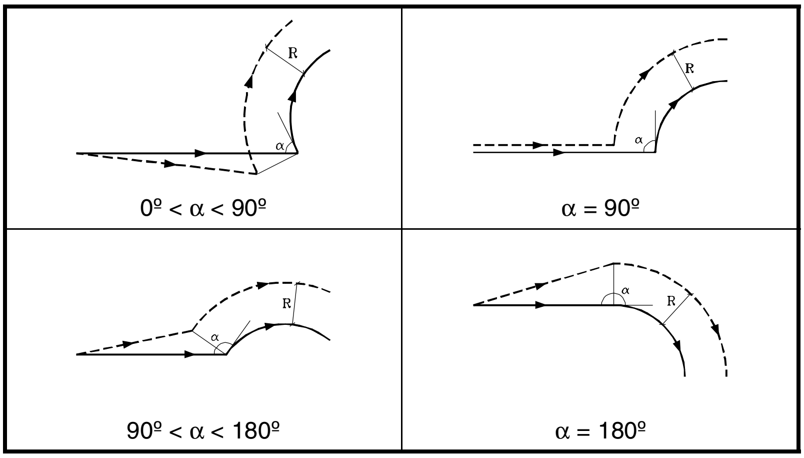


When the angle between paths is greater than 180º, the way radius compensation is activated depends on the functions selected for type of beginning (G138/G139) and type of transition (G136/G137).

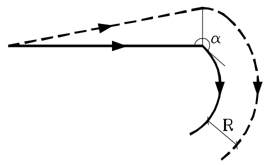
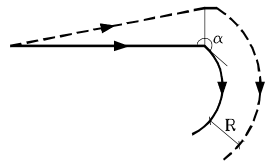
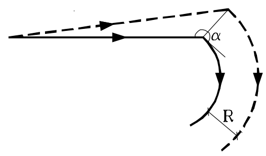
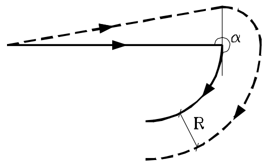
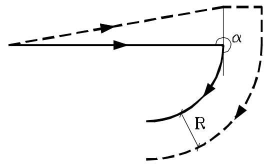
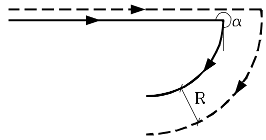
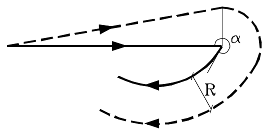
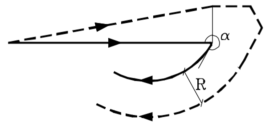
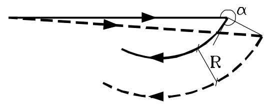


STRAIGHT-TO-ARC path

When the angle between the straight path and the tangent of the arc is smaller than or equal to 180º, the way radius compensation is activated is independent from the functions G136/G137 and G138/G139 selected.



When the angle between the straight path and the tangent of the arc is greater than 180º the way radius compensation is activated depends on the type of beginning (G138/G139) and type of transition (G136/G137) selected.

G139/G136	G139/G137	G138
 180º < α < 270º	 180º < α < 270º	 180º < α < 270º
 α = 270º	 α = 270º	 α = 270º
 270º < α < 360º	 270º < α < 360º	 270º < α < 360º

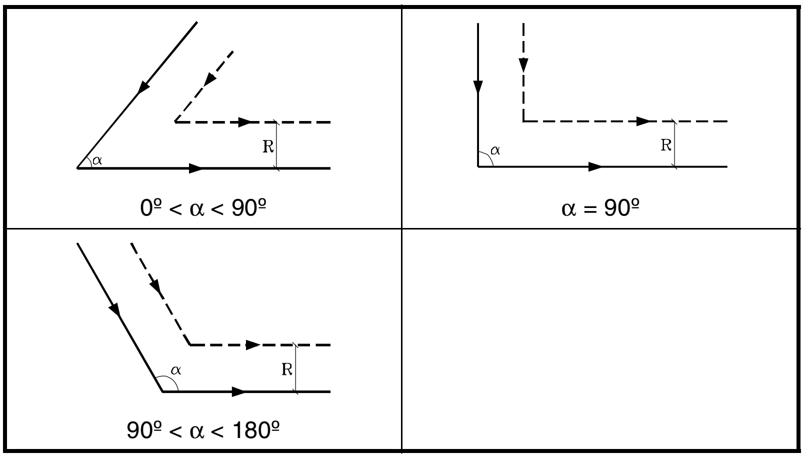
9.1.3 Sections of tool radius compensation

The way the compensated paths are joined depends on the type of transition selected (G136/G137).

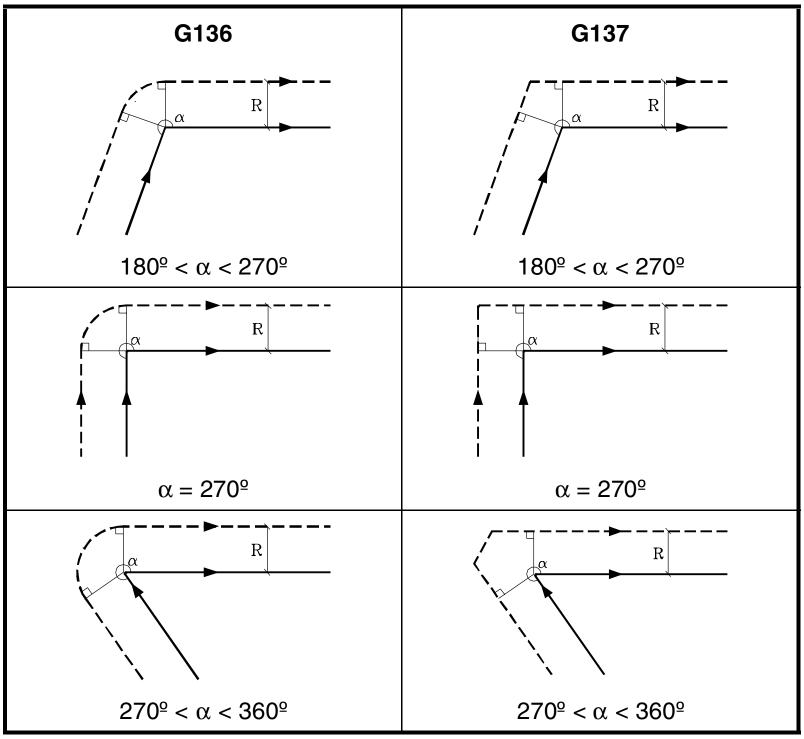
The following tables show the different transition possibilities between various paths depending on the selected function G136 or G137. The programmed path is shown with solid line and the compensated path with dashed line.

STRAIGHT-TO-STRAIGHT path

When the angle between paths is smaller than or equal to 180º, the transition between paths is independent from the G136/G137 function selected.

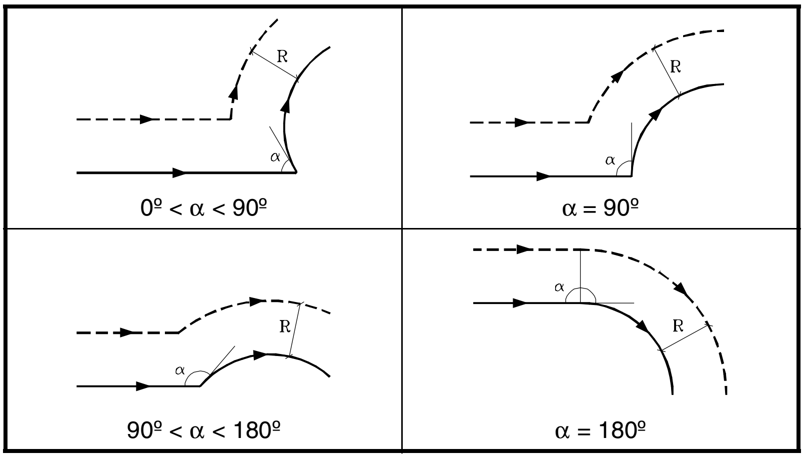


When the angle between paths is greater than 180º, the way the compensated paths are joined depends on the type of transition selected (G136/G137).

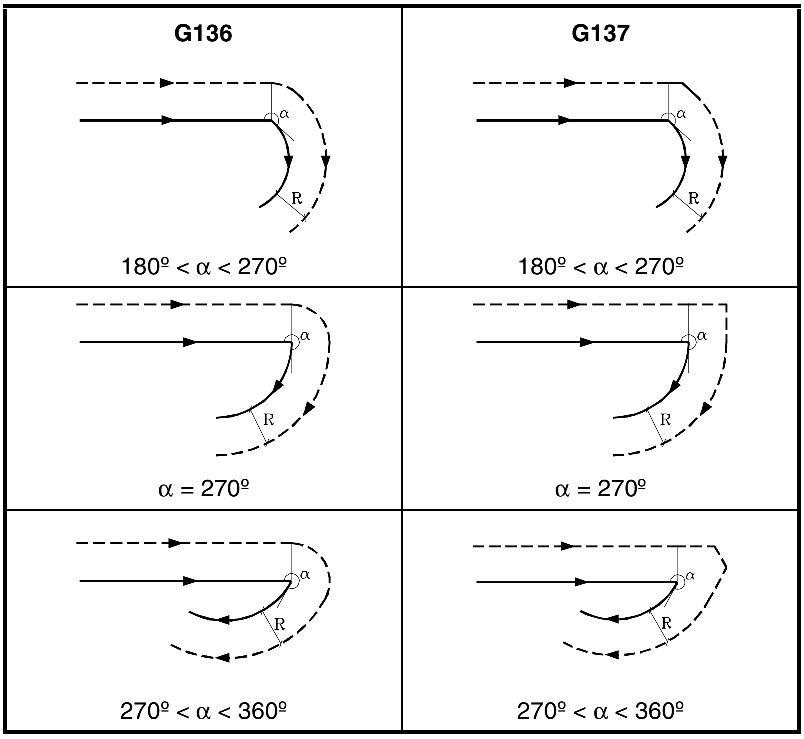


STRAIGHT-TO-ARC path

When the angle between the straight line and the tangent of the arc is smaller than or equal to 180º, the transition between the paths is independent from the selected G136/G137 function.

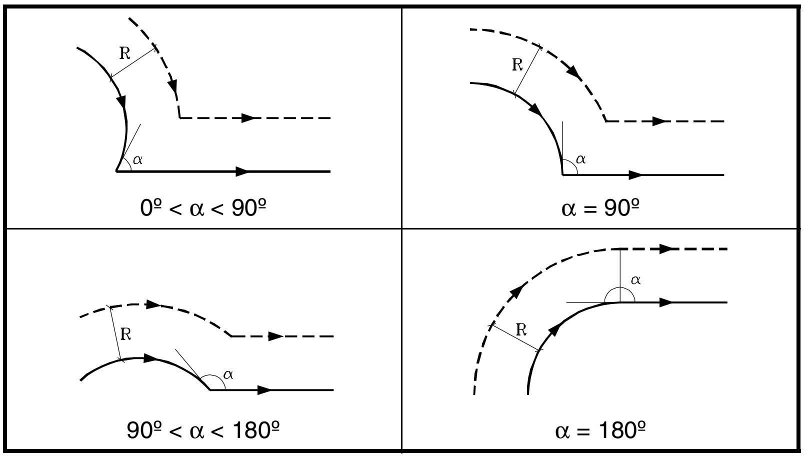


When the angle between the straight path and the tangent of the arc is greater than 180º, the way the compensated paths are joined depends on the type of transition selected (G136/G137).

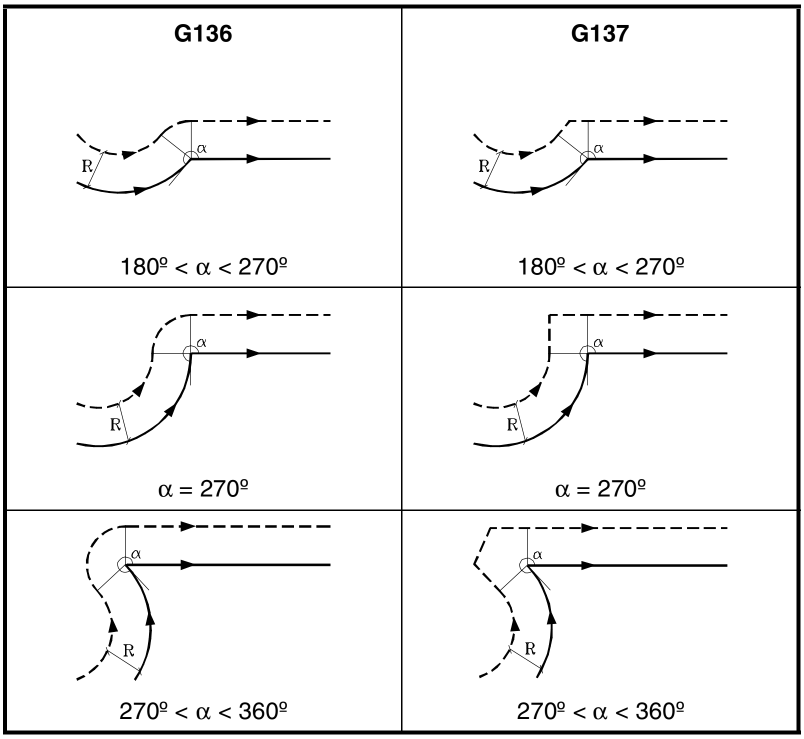


ARC-TO-STRAIGHT path

When the angle between the tangent of the arc and the straight line is smaller than or equal to 180º, the transition between the paths is independent from the selected G136/G137 function.

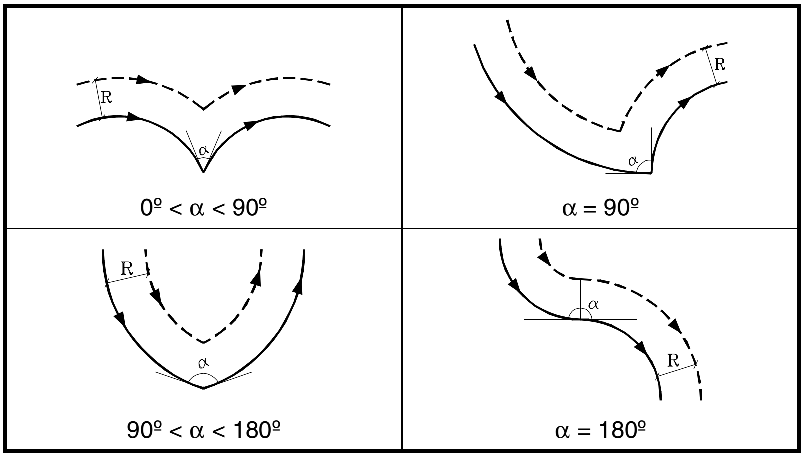


When the angle between the tangent of the arc and the straight line is greater than 180º, the way the compensated paths are joined depends on the type of transition selected (G136/G137).

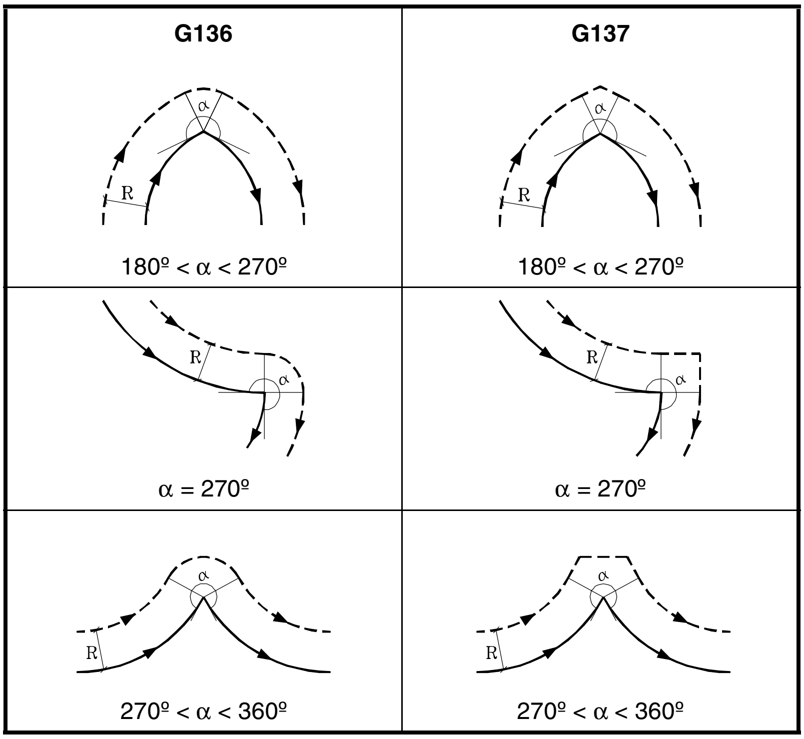


ARC-TO-ARC path

When the angle between the tangents of the arcs is smaller than or equal to 180º, the transition between the paths is independent from the selected G136/G137 function.



When the angle between the tangents of the arcs is greater than 180º, the way the compensated paths are joined depends on the type of transition selected (G136/G137).



9.1.4 Change of type of radius compensation while machining

The compensation may be changed from G41 to G42 or vice versa without having to cancel it with a G40. It may be changed in any motion block or even in a motionless one; i.e. without moving the axis of the plane or by programming the same point twice.

The last movement before the change and the first movement after the change are compensated independently. To change the type of compensation, the different cases are solved according to the following criteria:

A. The compensated paths intersect each other.

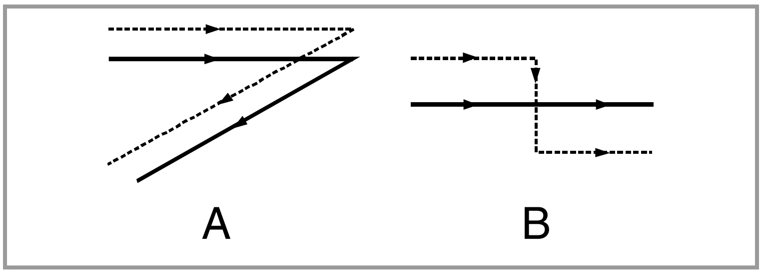
The programmed paths are compensated each on its corresponding side. The side change takes place in the intersection point between both paths.

B. The compensated paths do not intersect each other.

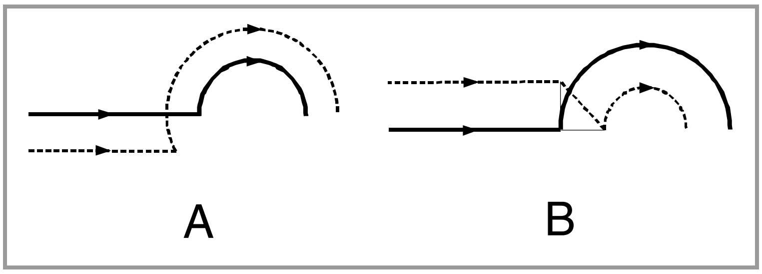
An additional section is inserted between the two paths. From the point perpendicular to the first path at the end point up to the point perpendicular to the second path at the starting point. Both points are located at a distance R from the programmed path.

Here is a summary of the different cases:

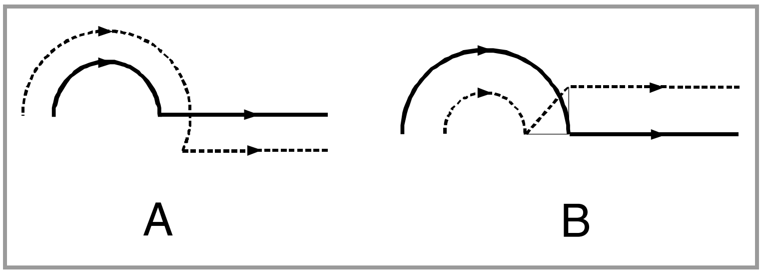
- Straight - straight path:



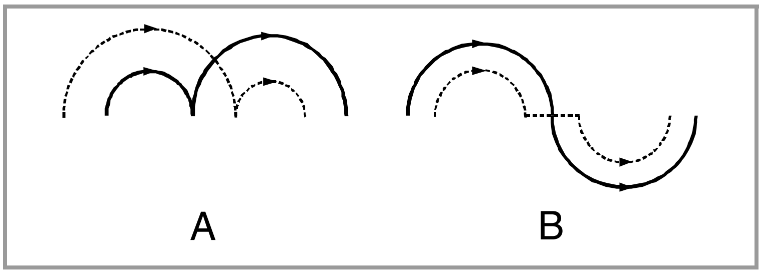
- Straight - circle path:



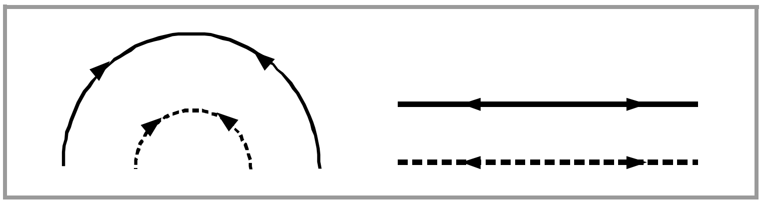
- Circle - straight path:



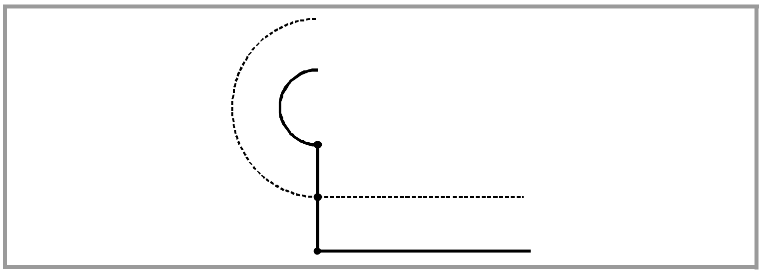
- Circle - circle path:



- Back-and-forth path along the same way.



- Intermediate path as long as the tool radius:



9.1.5 Cancellation of tool radius compensation

Tool radius compensation is canceled with function G40.

After executing one of this function, radius compensation will be cancelled during the next movement in the work plane, that must be a linear movement.

The way this compensation is canceled depends on the type of cancellation end (G138/G139) and the type of transition G136/G137 selected:

- G139/G136

  The tool goes to the endpoint, contouring the corner along a circular path.

- G139/G137

  The tool goes to the endpoint, contouring the corner along linear paths.

- G138

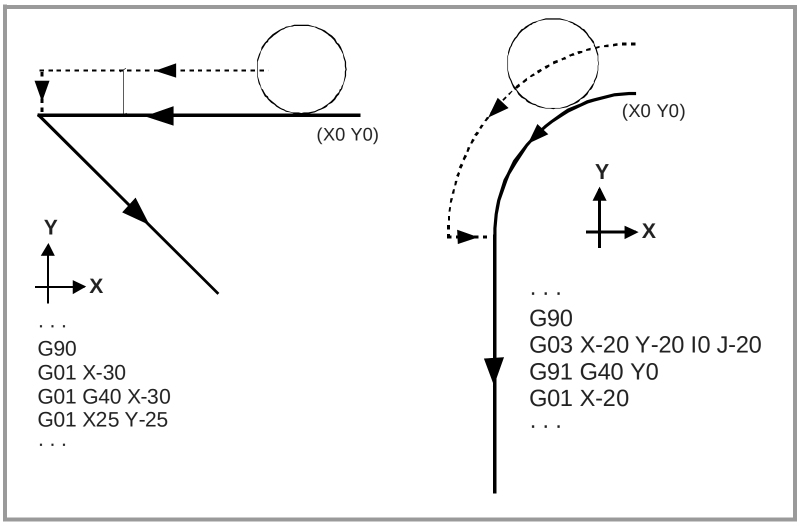
  The tool goes straight to the endpoint. Regardless of the type of transition (G136/G137) programmed.

The following tables show the different possibilities of canceling tool radius compensation depending on the selected functions. The programmed path is shown with solid line and the compensated path with dashed line.

End of the compensation without programmed movement

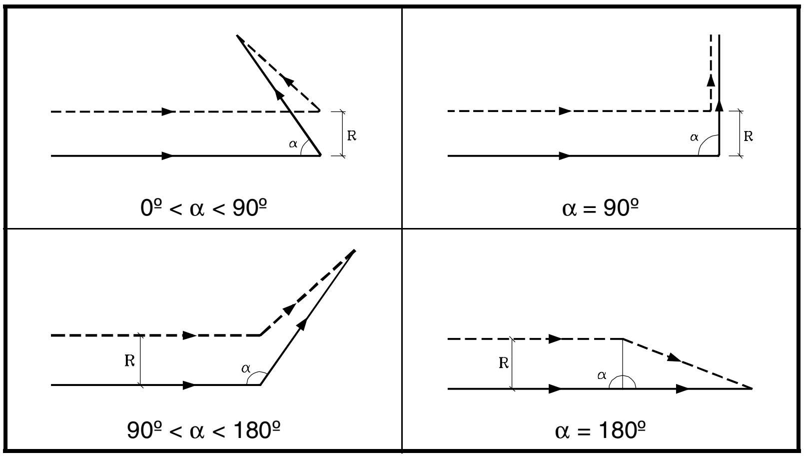
After canceling the compensation, it may occur that the axes of the plane will not be involved in the first motion block. For example, because they have not been programmed, or the current tool position has been programmed or an incremental movement has been programmed.

In this case, the compensation is canceled at the same point where the tool is, as follows. Depending on the last movement made in the plane, the tool moves to the end point (uncompensated) of the programmed path.

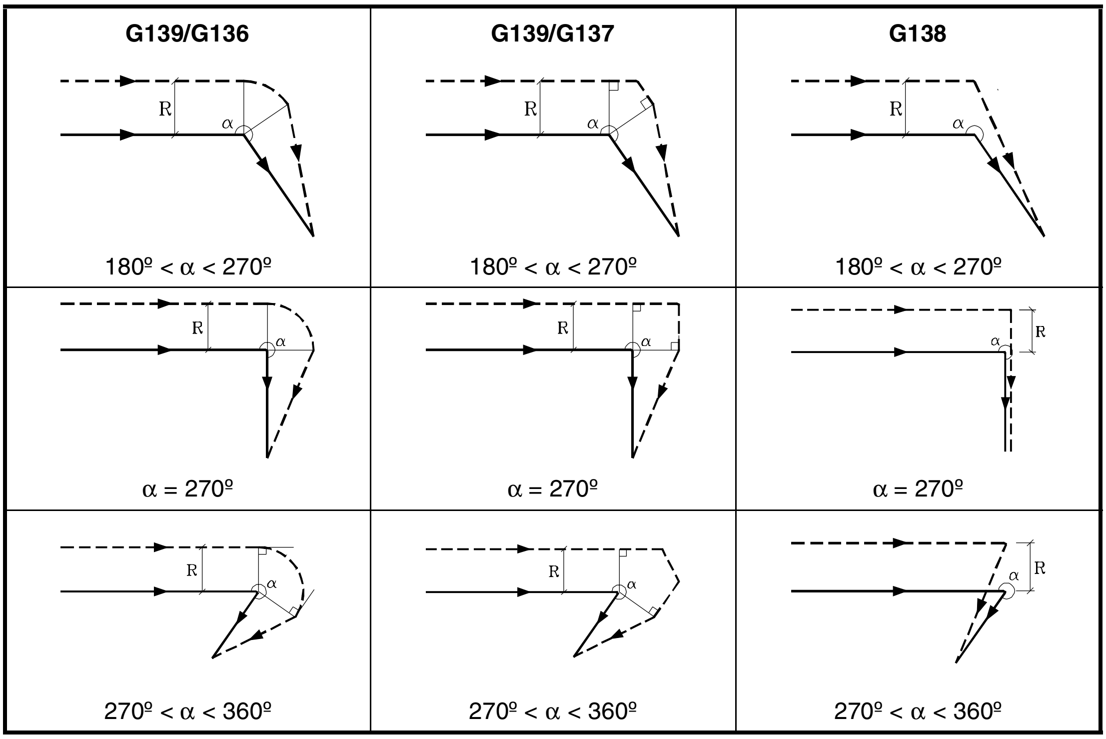


STRAIGHT-TO-STRAIGHT path

When the angle between the paths is smaller or equal to 180º, the way radius compensation is canceled is independent from the G136/G137 and G138/G139 functions selected.

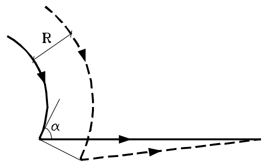
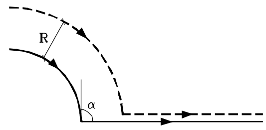
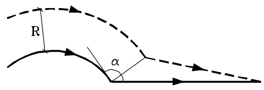
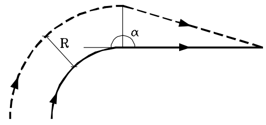


When the angle between paths is greater than 180º, the way radius compensation is canceled depends on the functions selected for type of ending (G138/G139) and type of transition (G136/G137).

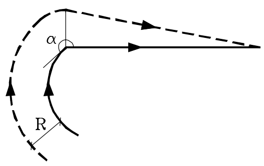
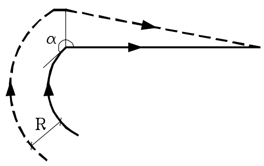
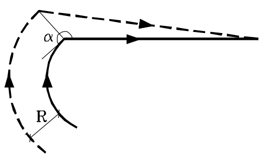
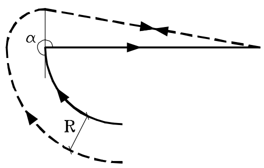
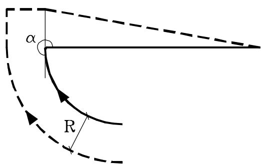
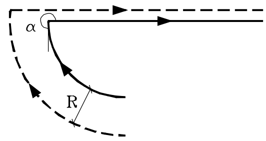
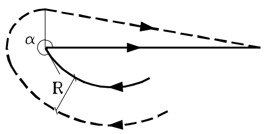
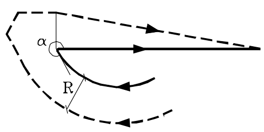
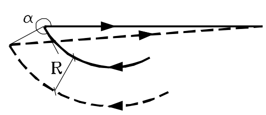


ARC-TO-STRAIGHT path

When the angle between the tangent of the arc and the straight path is smaller or equal to 180º, the way radius compensation is canceled is independent from the G136/G137 and G138/G139 functions selected.

 0º < α < 90º	 α = 90º
 90º < α < 180º	 α = 180º

When the angle between the tangent of the arc and the straight line is greater than 180º, the way radius compensation is canceled depends on the type of ending (G138/G139) and type of transition selected (G136/G137).

G139/G136	G139/G137	G138
 180º < α < 270º	 180º < α < 270º	 180º < α < 270º
 α = 270º	 α = 270º	 α = 270º
 270º < α < 360º	 270º < α < 360º	 270º < α < 360º

9.2 Tool length compensation

Tool length compensation is applied on to the axis indicated by the instruction "#TOOL AX", or when missing, to the longitudinal axis designated by selecting the plane.

If G17, tool lengh compensation is aplied to Z axis.

If G18, tool lengh compensation is aplied to Y axis.

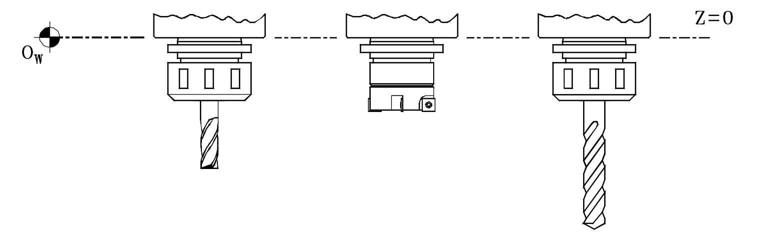
If G19, tool lengh compensation is aplied to X axis.

Whenever any of functions G17, G18 or G19 is executed, the CNC assumes the axis perpendicular to the selected plane as the new longitudinal axis. If, then, "#TOOL AX" is executed, the new selected longitudinal axis replaces the previous one.

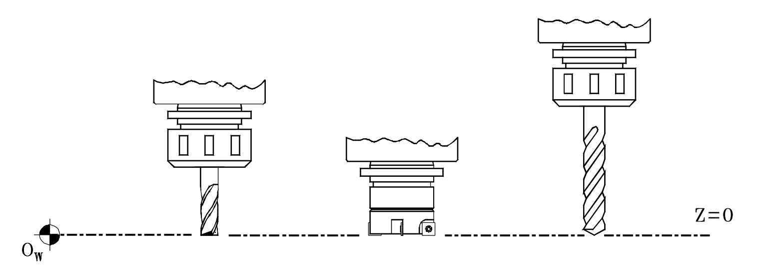
Programming

Tool length compensation is activated when selecting a tool offset.

- To activate this compensation, program "D<n>", where <n> is the tool offset number that contains the tool dimensions that will be used as compensation values.
- To cancel this compensation, program "D0".



Positioning of several tools at position "0", with tool length compensation off.



Positioning of several tools at position "0", with tool length compensation on.

Once one of these codes has been executed, tool length compensation will be activated or cancel during the next movement of the longitudinal axis.

CANNED CYCLES

10

10.1 General concepts

Some canned cycles are edited in ISO code (described in this chapter) and others are generated in conversational mode (described in chapter *"12 Cycle editor"*)

The canned cycles edited in ISO code are defined using a "G" function and its relevant parameters.

G81	Drilling canned cycle.
G82	Drilling canned cycle with variable peck (drilling step).
G83	Deep hole drilling canned cycle with constant peck (drilling step).
G84	Tapping canned cycle.
G85	Reaming canned cycle.
G86	Boring canned cycle.
G87	Rectangular pocket canned cycle.
G88	Circular pocket canned cycle.

Other functions related to canned cycles:

G80	Canned cycle cancellation.
G98	The tool, after the canned cycle is done, returns to the starting plane.
G99	The tool, after the canned cycle is done, returns to the reference plane.

Machining cycles may be executed in any plane.

10.1.1 Canned cycle definition

The canned cycle may be defined anywhere in the program, in the main program as well as in a subroutine.

It is defined using the relevant "G" function and its associated parameters.

Executing a canned cycle does not change the history of the previous "G" functions and maintains the spindle turning direction. If it is stopped, it starts clockwise (M03).

10.1.2 Influence zone of a canned cycle

The canned cycle is modal. Once defined, by program or MDI, it stays active until its cancellation is programmed or until one the conditions that cancels it occurs.

If inside the influence zone of the canned cycle, while active, a motion block is executed, it makes the programmed move and then executes the machining operation corresponding to the canned cycle.

Example:

T1 D1 M6	
G0 G90 Z25 S1000 M3 M8 M41 F200	
G5 X15 Y15	(Movement to X15 Y15)
G99 G81 Z2 I-20	(Defines and executes the drilling canned cycle)
X85	(Movement and new drilling in X85 Y15)
Y85	(Movement and new drilling in X85 Y85)
G80	(Canned cycle cancellation)
X15	(Movement to X15 Y85) There is no drilling)
M30	

10.1.3 Canned cycle cancellation

A cycle is cancelled when:

- Executing function G80.
- Defining a new canned cycle.
- Selecting another longitudinal axis, with G20 or with #TOOL AX
- Homing.
- Selecting a new work plane.
- After executing M02, M30 or after an Emergency or Reset.

10.1.4 Work planes

There are two coordinates in the machining cycles along the longitudinal axis that are described next because they are important:

- Starting plane (Zi).

  Tool coordinate (position) when defining the cycle.

- Reference plane (Z).

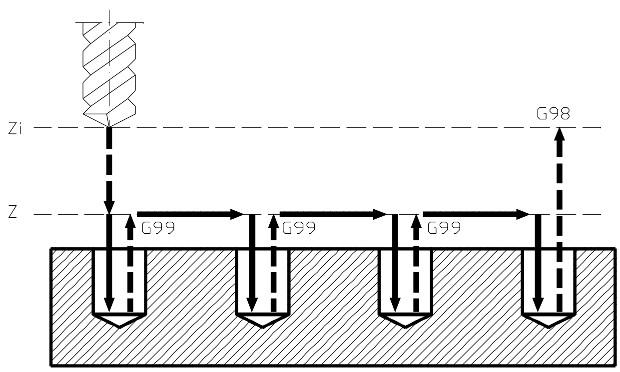
  Coordinate near the part, it is programmed when defining the cycle.

Functions G98 and G99 indicate where the tool returns after machining.

G98	Return (withdraw) to the starting plane (Zi).
G99	Return (withdraw) to the reference plane (Z).

Both functions are modal and G98 is assumed by default.

Example:



G99 G1 X0 Y0	(Movement)
G81 Z I K	(Defines and executes the drilling canned cycle)
X1 Y1	(Move and drill)
X2 Y2	(Move and drill)
G98 X3 Y3	(Move and drill)
G80	(Canned cycle cancellation)

10.1.5 Programming order

Preparatory (G), technological (F, S) and auxiliary (M, H) functions must be defined before defining the canned cycle.

Functions G98, G99 and the positioning move to the machining point must also be programmed before.

Example:

```
T1 D1 M6
    Selects tool 1 and offset 1.
G0 G90 X0 Y0 Z25
    It moves the tool, in rapid, to X0 Y0 Z25.
N10 G99 G1 X60 I30 F1000 S2000 M4
    Moves in G1 to the machining point X60 Y0.
    The starting plane will be Z25.
    The machining will have a withdrawal to the reference plane (G99).
N11 G81 Z2 I-20
    Drill in X60 Y0.
    Withdraw to Z2, reference plane (G98 active).
    Maintains the conditions prior to the cycle (G1 F1000 S2000 M4).
G98 G2 X160 I50
    Circular interpolation (G2) to X160 Y0 Z25.
    Drilled at that point.
    Withdraw to the starting plane (Z25).
M30
```

Blocks N10 (movement) and N11 (canned cycle definition) can also be defined as a single block being the canned cycle definition at the end.

```
N10 G99 G1 X60 I30 F1000 S2000 M4
N11 G81 Z2 I-20

N10 G99 G1 X60 I30 F1000 S2000 M4 G81 Z2 I-20
```

When defining a new canned cycle inside the influence zone of another active cycle, use the following methods:

N10	G81 Z2 I-20	N10	G81 Z2 I-20
	X160 I50 F3000		X160 I50 F3000
N20	G80		
N30	G1 X200 Y200	N30	G1 X200 Y200 G83 Z2 I-2 J5
N31	G83 Z2 I-2 J5		
	X220		X220
	M30		M30

In the example on the left, block N20 must be programmed to cancel the active canned cycle. Otherwise, block N30 will execute the active cycle defined in N10.

In the example on the right, there is no need to program block N20. The active canned cycle defined in N10 is canceled when defining a new one in N30. When executing block N30, it first moves the axes to X200 Y200 and then it executes the canned cycle G83.

10.1.6 Programming in other planes

The following examples show how to drill in X and Y in both directions.

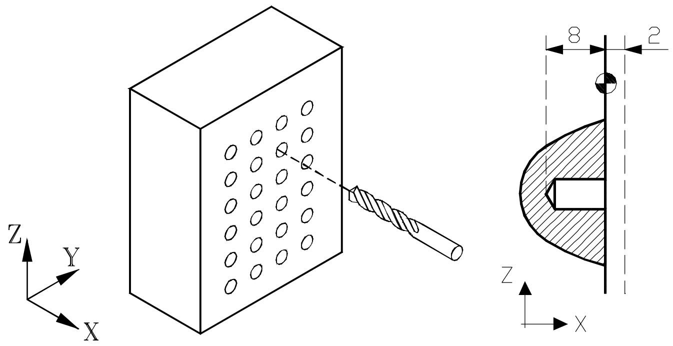
Function G81 defines the drilling canned cycle. It is defined with parameters:

X/Y/Z	Reference coordinate along the longitudinal axis.
I	Drilling depth.
K	Dwell at the bottom.

In the following examples, the part surface has a 0 coordinate, the holes are 8 mm deep and the reference coordinate is 2 mm above the surface.

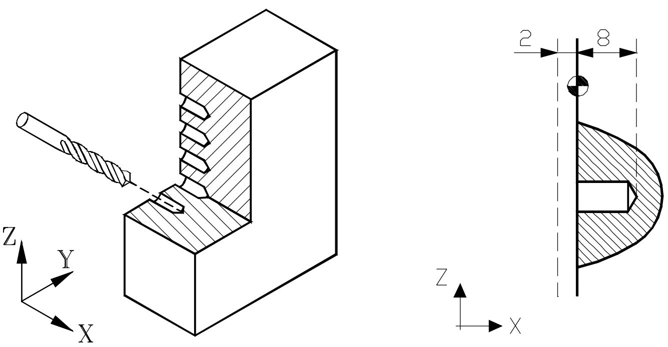
For each type of machine and machining operation the tool's longitudinal axis must be selected using the #TOO AX instruction so the CNC knows the machining direction.

Example 1:



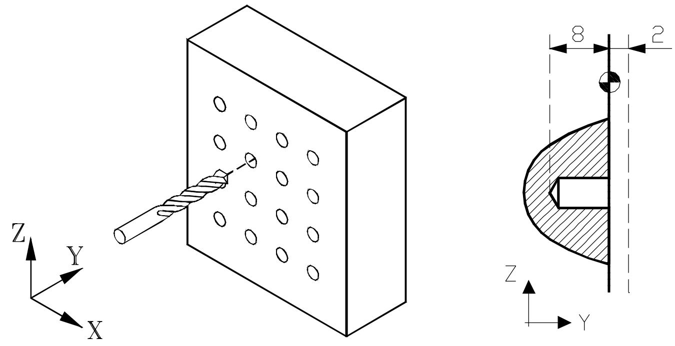

```
G19
#TOOL AX [X+]
G1 X25 F1000 S1000 M3
G81 X2 I-8 K1
```

Example 2:



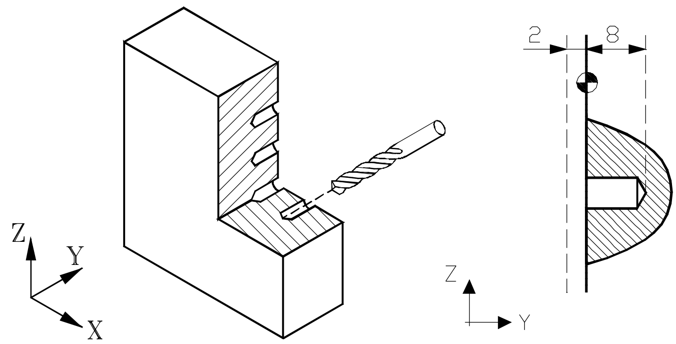

```
G19
#TOOL AX [X-]
G1 X-25 F1000 S1000 M3
G81 X-2 I8 K1
```

Example 3:




```
G18
#TOOL AX [Y-]
G1 Y25 F1000 S1000 M3
G81 Y2 I-8 K1
```

Example 4:




```
G18
#TOOL AX [Y+]
G1 Y-25 F1000 S1000 M3
G81 Y-2 I8 K1
```

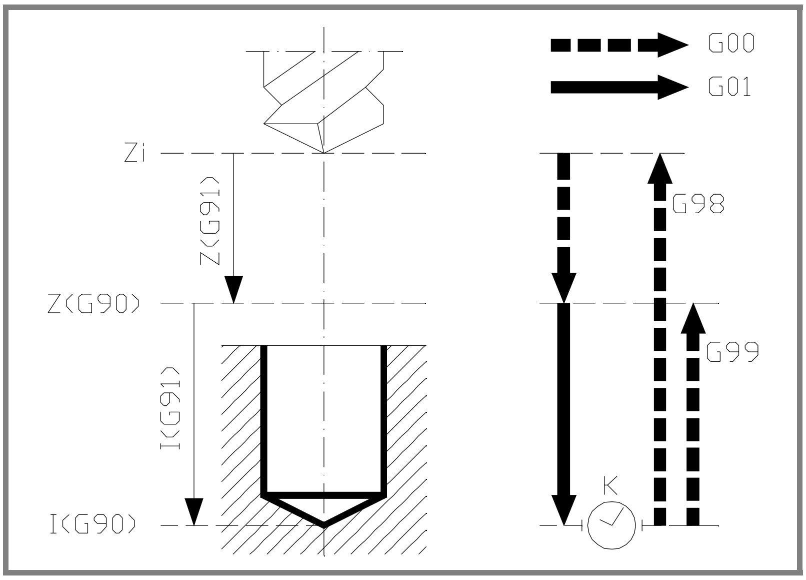
If working in the U V plane and the tool is located on the longitudinal axis X2, it is programmed as follows:

```
#SET AX [U,V,X2]
#TOOL AX [X2+]
G1 X2=25 F1000 S1000
G81 X2=2 I-8 K1
```

10.2 G81. Drilling canned cycle

Programming format in Cartesian coordinates:

G81 Z I K



Parameter definition:

Z Reference plane.

In G90, coordinate referred to part zero.

In G91, coordinate referred to starting plane (Zi).

If not programmed, it assumes as reference plane the current position of the tool (Z=Zi).

I Drilling depth.

In G90, coordinate referred to part zero.

In G91, coordinate referred to reference plane (Z).

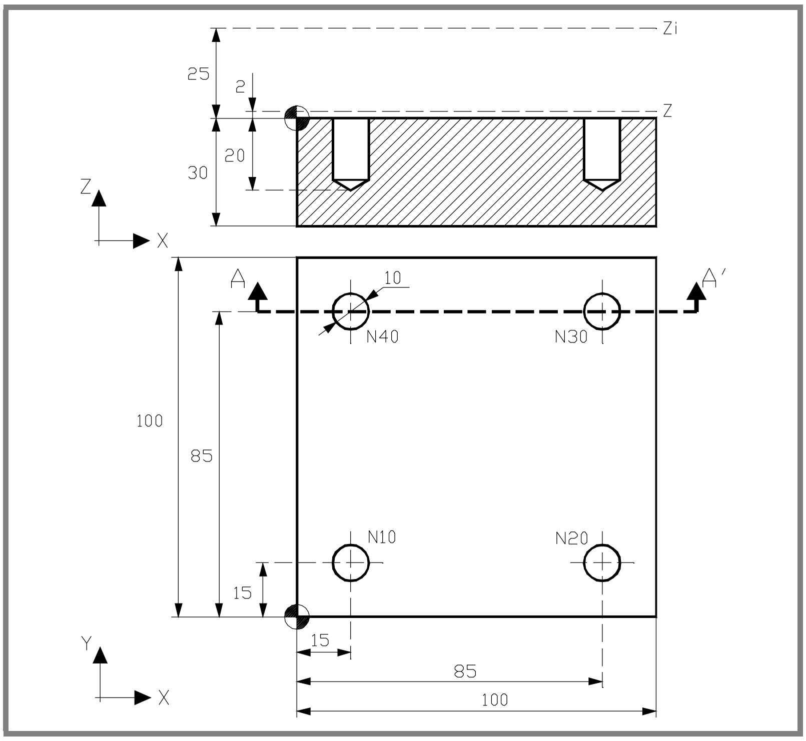
K Delay, in seconds, between the drilling and the withdrawal movement.

If not programmed, it assumes K0.

Basic operation:

1. If the spindle was previously running, it maintains the turning direction. If it is stopped, it starts clockwise (M03).
2. Rapid movement (G0) of the longitudinal axis from the starting plane (Zi) to the reference plane (Z).
3. Drill the hole. Movement of the longitudinal axis at work feedrate, to the bottom of the hole programmed in "I".
4. Dwell, in seconds, if it has been programmed.
5. Rapid withdrawal (G0) to the starting plane (Zi) if function G98 is active or to the reference plane (Z) if function G99 is active.

10.2.1 Programming example



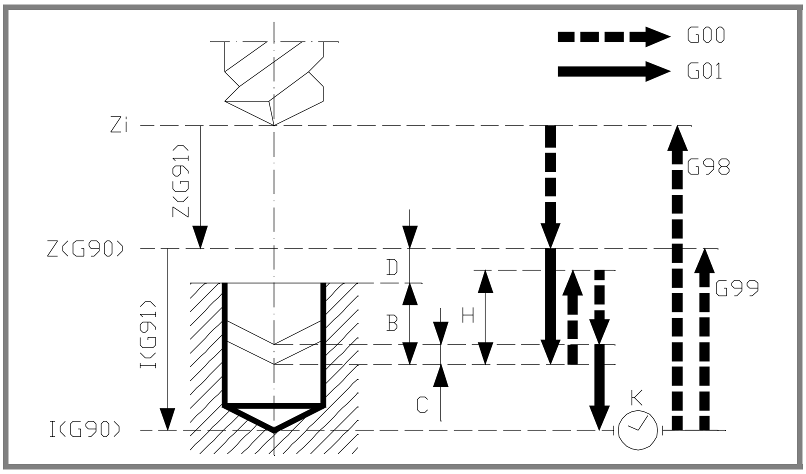
Absolute programming:	
	T1 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 X15 Y15 G81 Z2 I-20
N20	X85
N30	Y85
N40	G98 X15
	M30

Incremental programming:	
	T1 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 G91 X15 Y15 G81 Z-23 I-22
N20	X70
N30	Y70
N40	G98 X-70
	M30

10.3 G82. Drilling canned cycle with variable peck

Programming format in Cartesian coordinates:

G82 Z I D B H C J K R L



Parameter definition:

Z Reference plane.

In G90, coordinate referred to part zero.

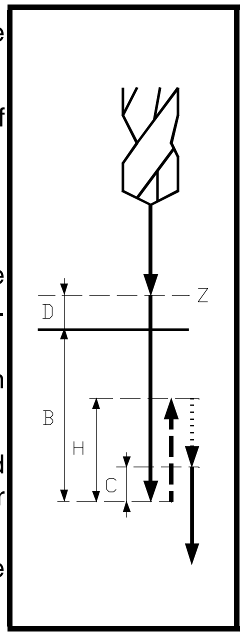
In G91, coordinate referred to starting plane (Zi).

If not programmed, it assumes as reference plane the current position of the tool (Z=Zi).

I Drilling depth.

In G90, coordinate referred to part zero.

In G91, coordinate referred to reference plane (Z).



D Distance between the reference plane and the part surface.

If not programmed, it assumes a value of 0.

B Drilling peck (step).

All the pecks have this value, except the last one that is adjusted to the total depth.

H Distance or coordinate it returns to, in rapid (G0), after each drilling step.

"J" other than 0 means the distance and "J=0" indicates the relief coordinate or absolute coordinate it withdraws to.

If not programmed, it returns to the reference plane.

C Approach coordinate.

It defines the rapid approach (G0) distance of the longitudinal axis from the previous drilling peck to carry out a new drilling peck.

If not programmed, it assumes 1 mm.

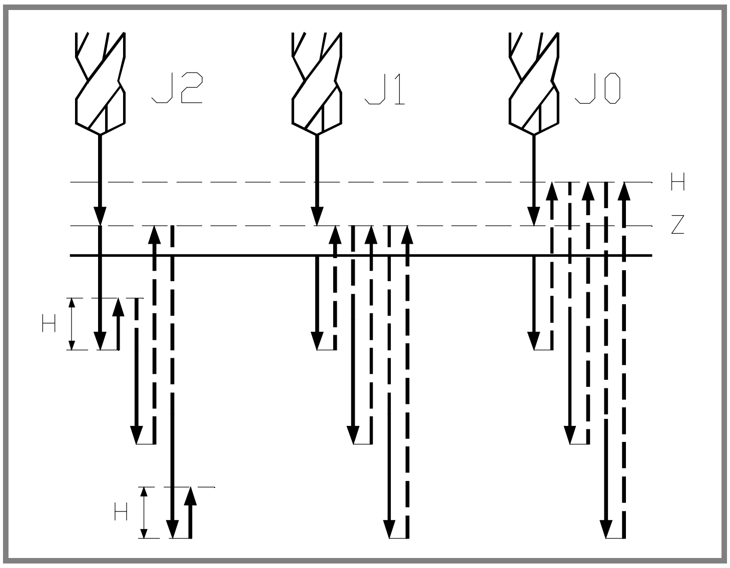
It issues an error message if "C=0" is programmed.

J It defines after how many drilling pecks the tool returns in rapid (G0) to the reference plane (Z).

With "J" greater than 1, after each peck, the tool returns the distance indicated by "H" and every "J" pecks to the reference plane (Z).

With "J=1", it returns to the reference plane (Z) after each peck.

If "J" is not programmed or "J=0" is programmed, it returns to the relief coordinate indicated by "H" after each peck.



K Dwell, in seconds, at the bottom of the hole.

If not defined, it assumes a value of 0.

R Factor that increases or reduces the drilling peck (step) "B".

The first peck will be "B", the second "RB", the third "R(RB)" and so on.

If it is not programmed or "R=0" is programmed, it assumes "R=1". With "R=1", all the drilling pecks will have the value of "B".

L Minimum value for the drilling peck. It is used with "R" values other than 1. If not programmed or programmed with a 0 value, it assumes the value of 1 mm.

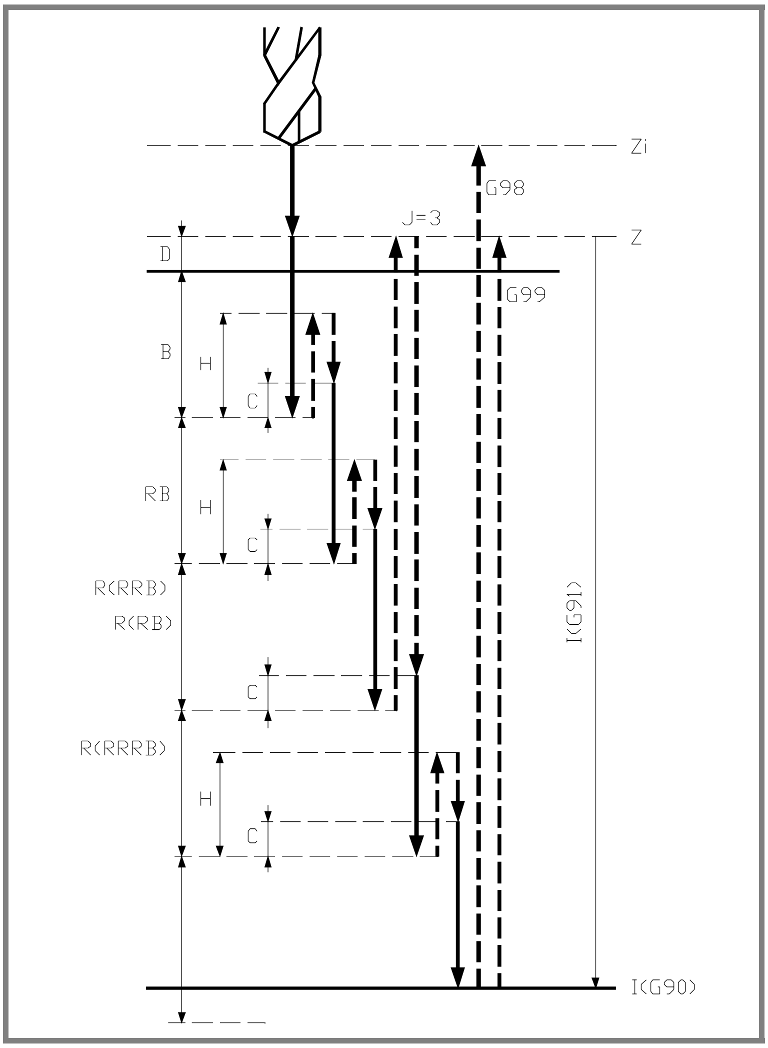
Basic operation:

1. If the spindle was previously running, it maintains the turning direction. If it is stopped, it starts clockwise (M03).
2. Rapid movement (G0) of the longitudinal axis from the starting plane (Zi) to the reference plane (Z).
3. First drilling penetration, at work feedrate. The distance indicated by "B", from the part surface.
4. Drilling loop until reaching the total drilling depth programmed in "I".
   - Rapid withdrawal (G0).

     With "J=1", it returns to the reference plane (Z) after each peck.

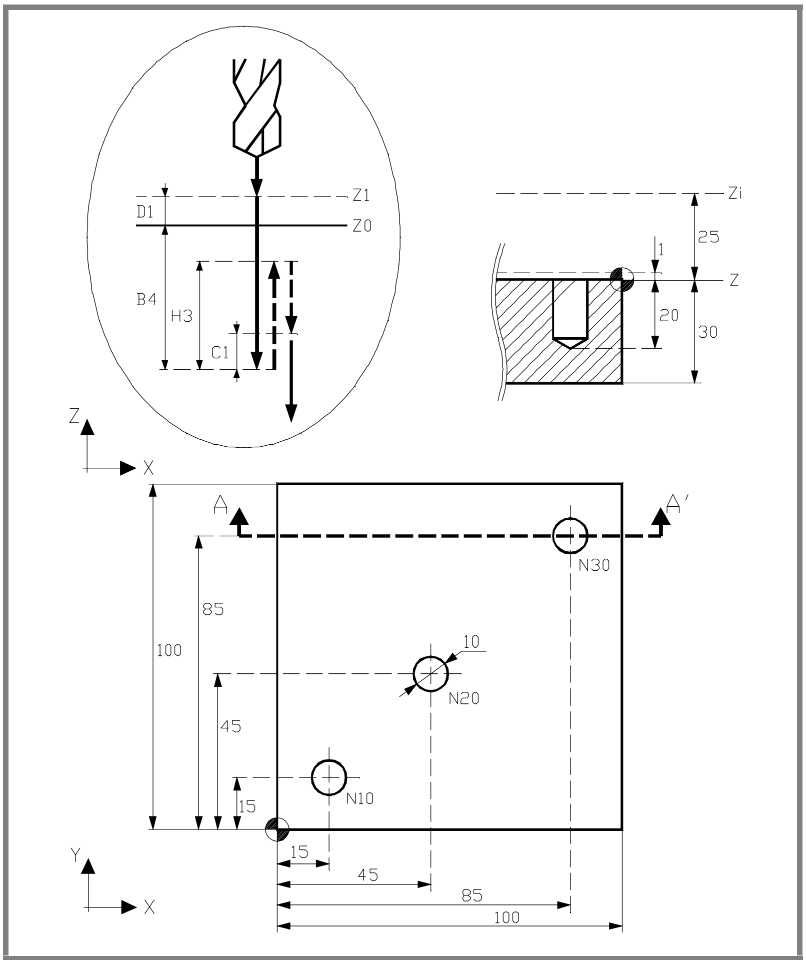
     If "J" is not programmed or "J=0" is programmed, it returns to the relief coordinate indicated by "H" after each peck.

     With "J" greater than 1, after each peck, the tool returns the distance indicated by "H" and every "J" pecks to the reference plane (Z).
   - Rapid approach (G0) to a distance "C" or up to 1 mm from the previous drilling step (peck).
   - New drilling peck, at work feedrate. The distance indicated by "B" and "R".



5. Dwell at the bottom of the hole. The time indicated by "K" in seconds.
6. Rapid withdrawal (G0) to the starting plane (Zi) if function G98 is active or to the reference plane (Z) if function G99 is active.

10.3.1 Programming example



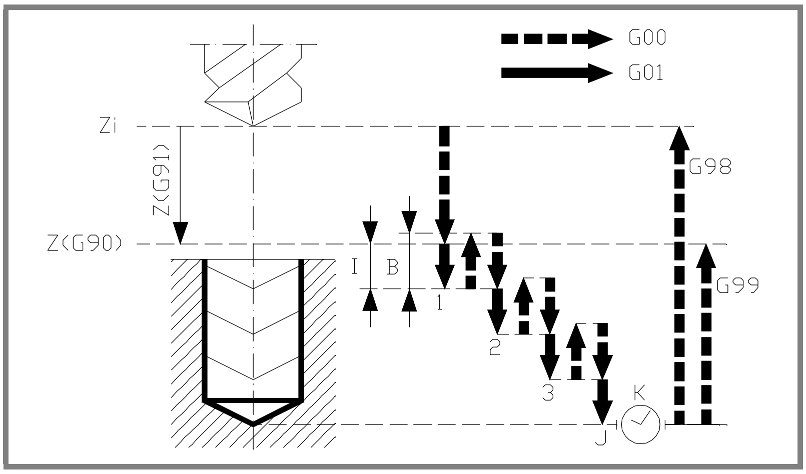
Absolute programming:	
	T2 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 X15 Y15
	G82 Z1 I-20 D1 B4 H3 C1 J3 K1 R0.8 L3
N20	X45 Y45
N30	G98 X85 Y85
	M30

Incremental programming:	
	T2 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 G91 X15 Y15
	G82 Z-24 I-21 D1 B4 H3 C1 J3 K1 R0.8 L3
N20	X30 Y30
N30	G98 X40 Y40
	M30

10.4 G83. Deep-hole drilling canned cycle with constant peck

Programming format in Cartesian coordinates:

G83 Z I J B K



Parameter definition:

Z Reference plane.

In G90, coordinate referred to part zero.

In G91, coordinate referred to starting plane (Zi).

If not programmed, it assumes as reference plane the current position of the tool (Z=Zi).

I Drilling peck (step).

The sign indicates the machining direction. Positive towards plus coordinate and negative towards minus. In the figure "I-".

J Number of pecks required by the drilling operation.

B Rapid withdraw (G0) distance after each drilling step.

If not programmed, it returns to the reference plane.

K Dwell, in seconds, at the bottom of the hole.

If not defined, it assumes a value of 0.

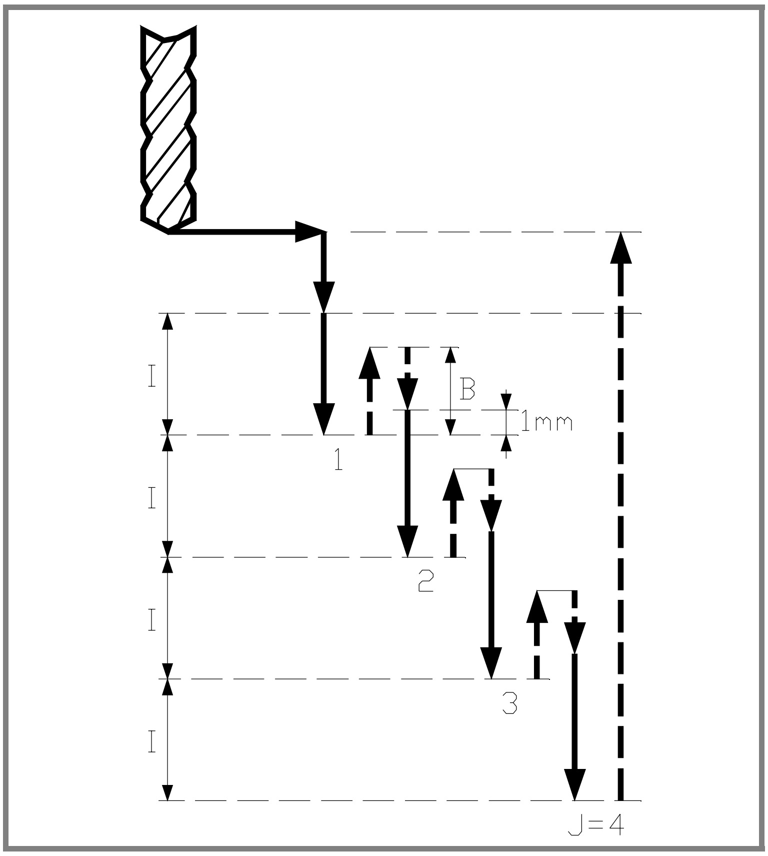
Basic operation:

1. If the spindle was previously running, it maintains the turning direction. If it is stopped, it starts clockwise (M03).

2. Rapid movement (G0) of the longitudinal axis from the starting plane (Zi) to the reference plane (Z).

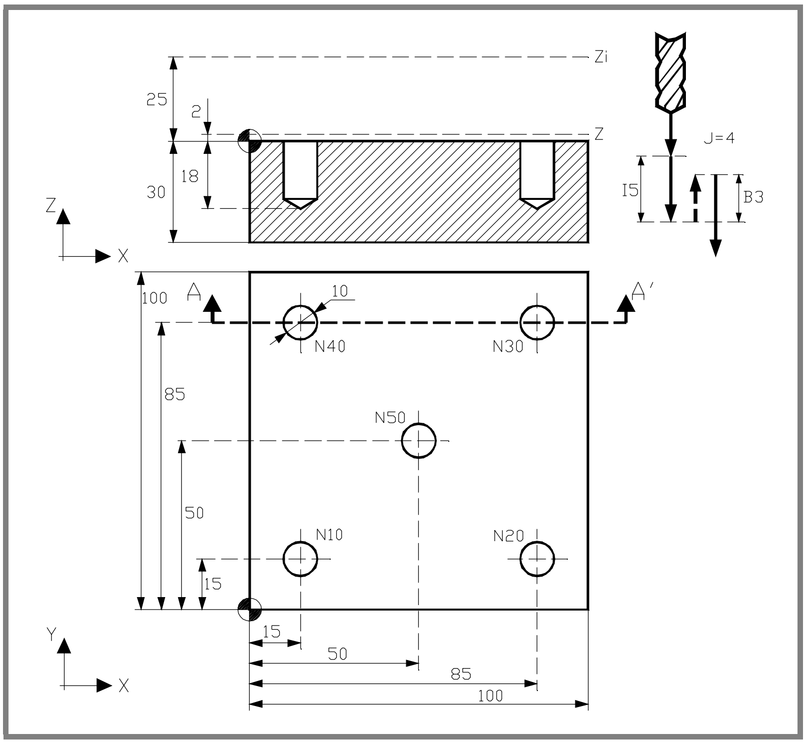
3. Drilling loop. The following steps are repeated "J" times.

- Drilling peck, at work feedrate. The distance indicated by "I".
- Rapid withdrawal (G0). The "B" distance or to the reference plane.
- Rapid approach (G0) up to 1 mm from the previous drilling step (peck).



4. Dwell at the bottom of the hole. The time indicated by "K" in seconds.
5. Rapid withdrawal (G0) to the starting plane (Zi) if function G98 is active or to the reference plane (Z) if function G99 is active.

10.4.1 Programming example



Absolute programming:	
	T3 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 X15 Y15
	G83 Z2 I-5 J4 B3 K1
N20	X85
N30	Y85
N40	X15
N50	G98 X50 Y50
	M30

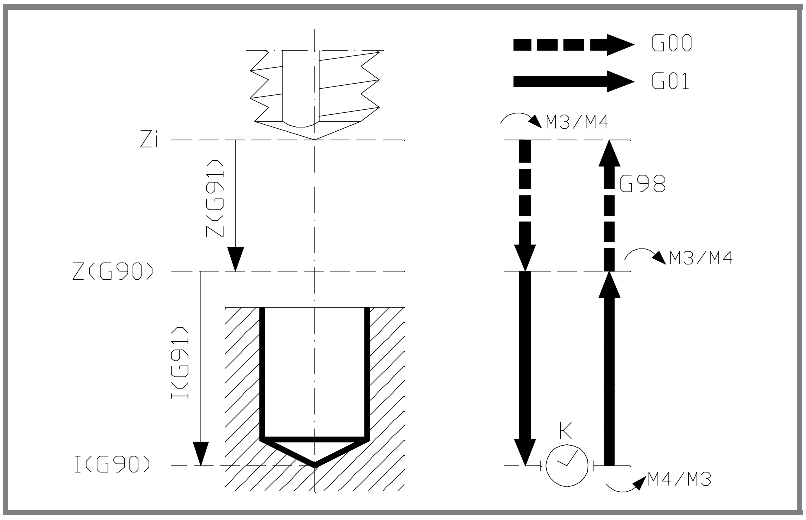
Incremental programming:	
	T3 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 G91 X15 Y15
	G83 Z-23 I-5 J4 B3 K1
N20	X70
N30	Y70
N40	X -70
N50	G98 X35 Y-35
	M30

10.5 G84. Tapping canned cycle

Both tapping with a clutch and rigid tapping are possible. For rigid tapping, the spindle must have a motor-drive system and a spindle encoder.

Programming format in Cartesian coordinates:

G84 Z I K R



Parameter definition:

Z Reference plane.

In G90, coordinate referred to part zero.

In G91, coordinate referred to starting plane (Zi).

If not programmed, it assumes as reference plane the current position of the tool (Z=Zi).

I Tap depth.

In G90, coordinate referred to part zero.

In G91, coordinate referred to reference plane (Z).

K Delay, in seconds, between the tapping and the withdrawal movement.

If not programmed, it assumes K0.

R Type of tapping.

R0: normal tapping.

R1: rigid tapping.

Basic operation:

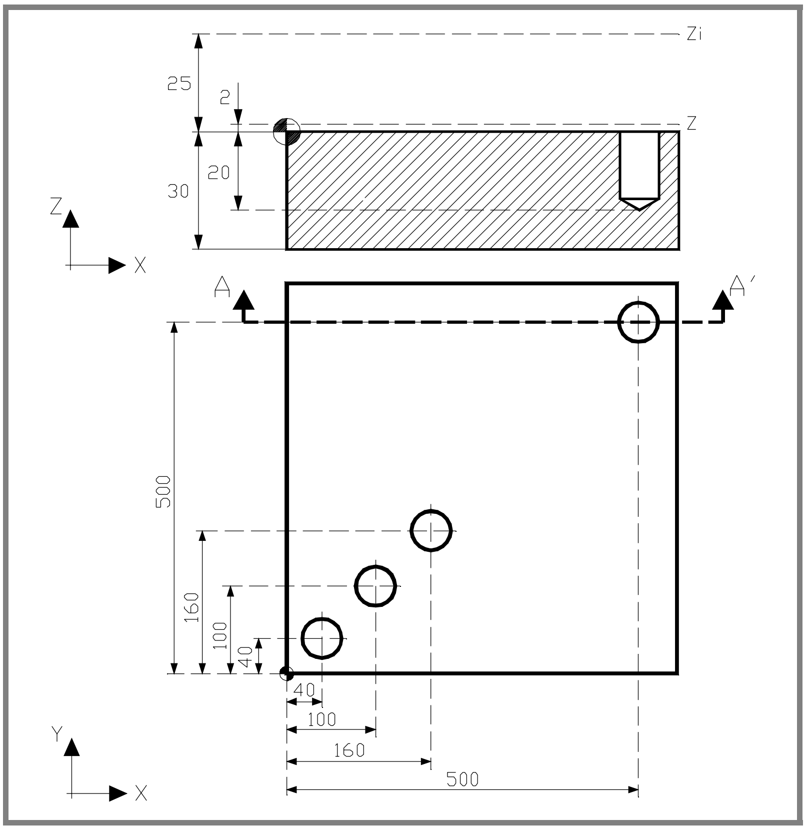
1. If the spindle was previously running, it maintains the turning direction. If it is stopped, it starts clockwise (M03).
2. Rapid movement (G0) of the longitudinal axis from the starting plane (Zi) to the reference plane (Z).
3. Tapping. It is executed at 100% of the feedrate "F" and spindle speed "S" programmed. Tapping cannot be interrupted.
4. If "K" other than 0, spindle stop (M05) and dwell.
5. Reverse the spindle turning direction.

   Withdrawal, exit the tap, to the reference plane. At 100% of the feedrate "F" and spindle speed "S" programmed. The thread exit cannot be interrupted.
6. Depending on the type of tap programmed.

R=0	Reverse the spindle turning direction restoring the initial turning direction.
R=1	Spindle orientation (M19).

7. If function G98 is active, rapid withdraw to the starting plane (Zi).

10.5.1 Programming example



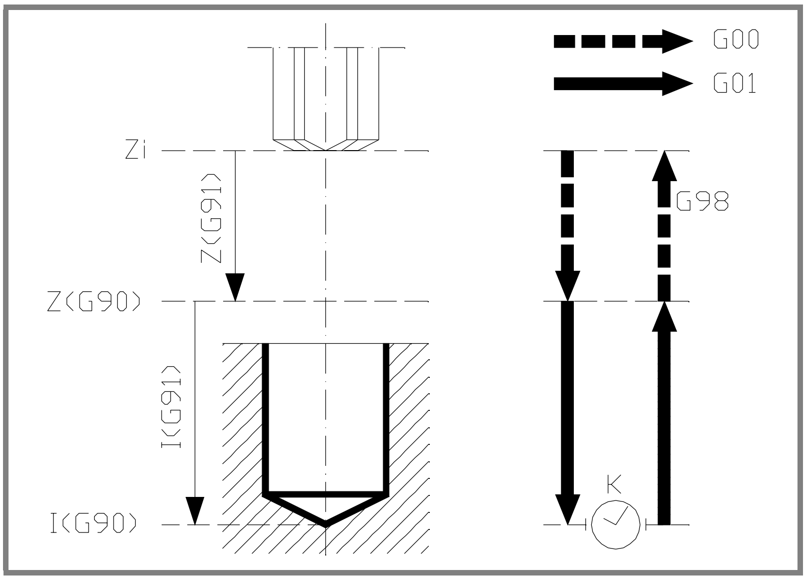

```
Absolute programming:
        T4 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10     G99 X40 Y40 G84 Z2 I-20 K1 R0
N20     X100 Y100
N30     X160 Y160
N40     G98 X500 Y500
        M30
```

```
Incremental programming:
        T4 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10     G99 G91 X40 Y40 G84 Z-23 I-22 K1 R0
        $FOR P0=1,2,1
        X60 Y60
        $ENDFOR
        G98 X340 Y340
        M30
```

10.6 G85. Reaming canned cycle

Programming format in Cartesian coordinates:

G85 Z I K



Parameter definition:

Z Reference plane.

In G90, coordinate referred to part zero.

In G91, coordinate referred to starting plane (Zi).

If not programmed, it assumes as reference plane the current position of the tool (Z=Zi).

I Reaming depth.

In G90, coordinate referred to part zero.

In G91, coordinate referred to reference plane (Z).

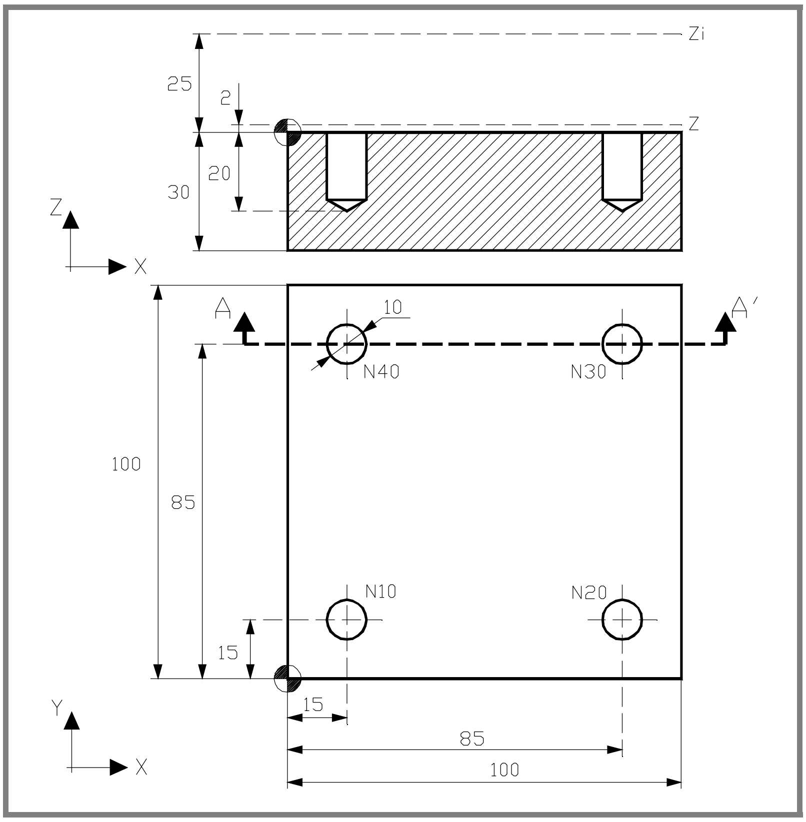
K Delay, in seconds, between the reaming and the withdrawal movement.

If not programmed, it assumes K0.

Basic operation:

1. If the spindle was previously running, it maintains the turning direction. If it is stopped, it starts clockwise (M03).
2. Rapid movement (G0) of the longitudinal axis from the starting plane (Zi) to the reference plane (Z).
3. Reaming the hole. Movement of the longitudinal axis at work feedrate, to the bottom of the hole programmed in "I".
4. Dwell, in seconds, if it has been programmed.
5. Withdrawal, at work feedrate (G01) up to the reference plane (Z).
6. If function G98 is active, rapid withdraw to the starting plane (Zi).

10.6.1 Programming example



Absolute programming:

```
      T5 D1 M6
      G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10   G99 X15 Y15 G85 Z2 I-20
N20   X85
N30   Y85
N40   G98 X15
      M30
```

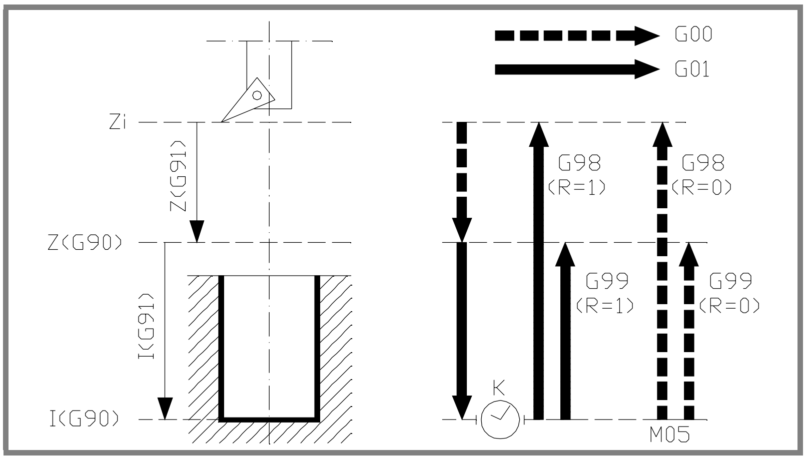
Incremental programming:

```
      T5 D1 M6
      G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10   G99 G91 X15 Y15 G85 Z-23 I-22
N20   X70
N30   Y70
N40   G98 X-70
      M30
```

10.7 G86. Boring canned cycle

Programming format in Cartesian coordinates:

G86 Z I K R



Parameter definition:

Z Reference plane.

In G90, coordinate referred to part zero.

In G91, coordinate referred to starting plane (Zi).

If not programmed, it assumes as reference plane the current position of the tool (Z=Zi).

I Boring depth.

In G90, coordinate referred to part zero.

In G91, coordinate referred to reference plane (Z).

K Delay, in seconds, between the boring and the withdrawal movement.

If not programmed, it assumes K0.

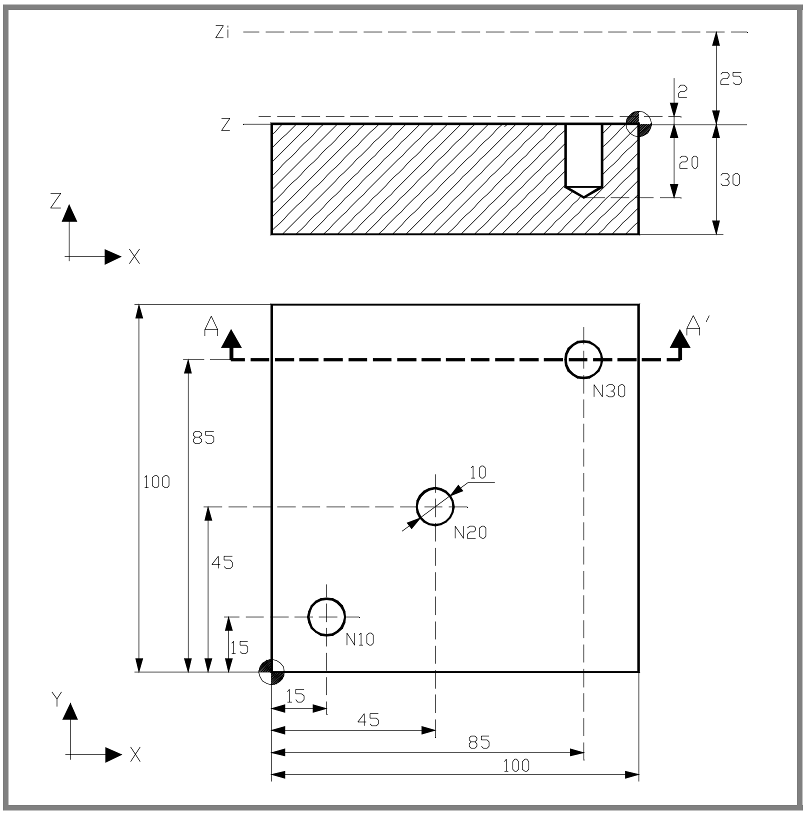
R Type of withdrawal: R0 rapid (G0), R1 at work feedrate (G01). By default, R0.

Basic operation:

1. If the spindle was previously running, it maintains the turning direction. If it is stopped, it starts clockwise (M03).
2. Rapid movement (G0) of the longitudinal axis from the starting plane (Zi) to the reference plane (Z).
3. Boring the hole. Movement of the longitudinal axis at work feedrate, to the bottom of the hole programmed in "I".
4. Dwell, in seconds, if it has been programmed.
5. If "R=0" has been programmed, the spindle stops (M05).
6. Withdrawal to the starting plane (Zi) if function G98 is active or to the reference plane (Z) if function G99 is active.

   In rapid (G0) if "R=0" and at work feedrate (G01) if "R=1".

10.7.1 Programming example



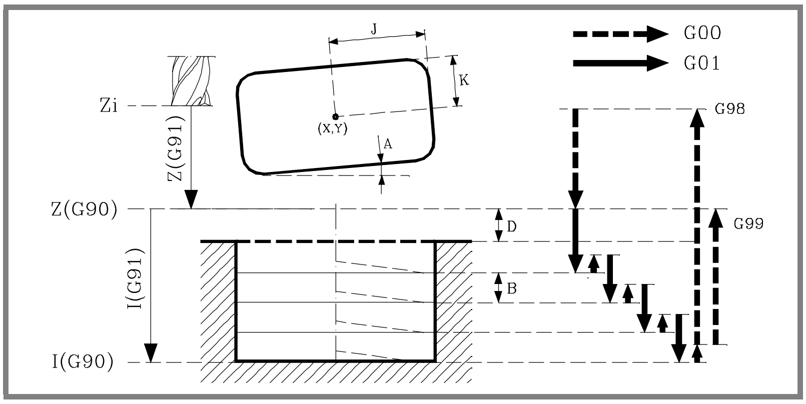
Absolute programming with R=0:	
	T6 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 X15 Y15 G86 Z2 I-20 K3 R0
N20	X45 Y45
N30	G98 X85 Y85
	M30

Incremental programming with R=1:	
	T6 D1 M6
	G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F200
N10	G99 G91 X15 Y15 G86 Z-23 I-22 K3 R1
N20	X30 Y30
N30	G98 X40 Y40
	M30

10.8 G87. Rectangular pocket canned cycle.

Programming format in Cartesian coordinates:

G87 Z I D A J K M Q B C L H V



Parameter definition:

Z Reference plane.

In G90, coordinate referred to part zero.

In G91, coordinate referred to starting plane (Zi).

If not programmed, it assumes as reference plane the current position of the tool (Z=Zi).

I Pocket depth.

In G90, coordinate referred to part zero.

In G91, coordinate referred to reference plane (Z).

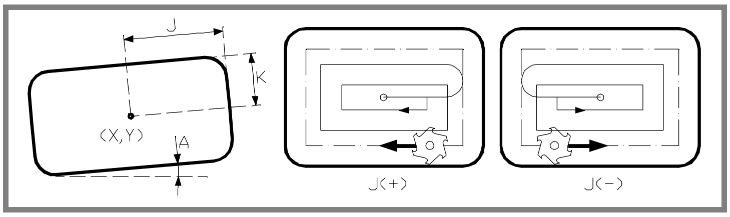
D Distance between the reference plane and the part surface. If not programmed, it assumes a value of 0.

A Angle, in degrees, between the pocket and the abscissa axis. If not programmed, it assumes a value of 0.

J Half length of the pocket.

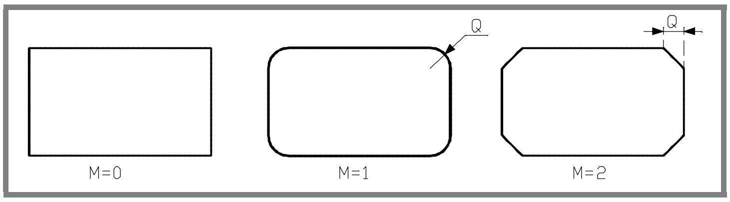
The sign indicates the pocket machining direction:

(J+) clockwise, (J-) counterclockwise.



K Half width of the pocket.

M Type of corner. (0) square, (1) rounded, (2) chamfered. If not programmed, it assumes a value of 0.

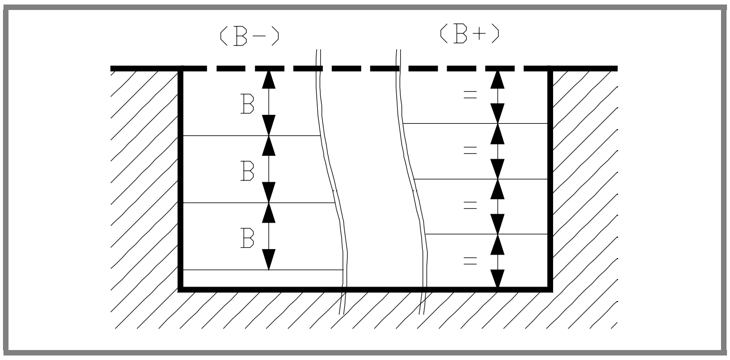


Q Rounding radius or chamfer size.

B Depth of pass.

If programmed with a positive sign (B+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.

If programmed with a negative sign (B-), the pocket is machined with the given pass (step) except the last pass that machines the rest.

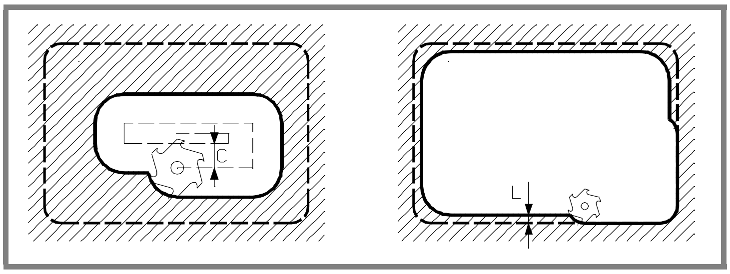


C Milling pass or width.

If not programmed or programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

If it is the same as parameter "J" or "K" (half length/half width of the pocket) it only runs the finishing pass.

If programmed with a value greater than the tool diameter, the CNC issues the relevant error message.



L Finishing pass.

If not programmed or programmed with a 0 value, it does not run the finishing pass.

H Feedrate for the finishing pass. If not programmed or programmed with a 0 value, it is carried out at the roughing feedrate.

V Tool penetrating feedrate. If not programmed or programmed with a 0 value, it is carried out at 50% of the feedrate in the plane.

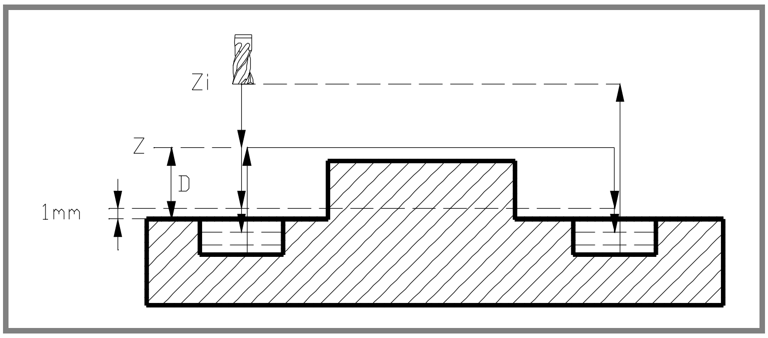
Basic operation:

1. If the spindle was previously running, it maintains the turning direction. If it is stopped, it starts clockwise (M03).

2. Rapid movement (G0) of the longitudinal axis from the starting plane (Zi) to the reference plane (Z).

3. Rapid movement (G0) of the longitudinal axis up to 1 mm off the part surface.

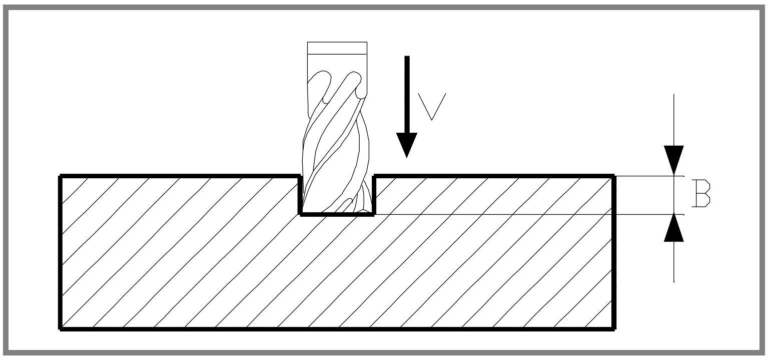
The movement permits, as in the case of the figure, a fast approach to the machining surface when the safety coordinate (Z) is far away from the surface.



4. Penetration. The longitudinal axis penetrates into the part the distance indicated by "B" and at the feedrate indicated by "V".

5. Milling of the pocket surface at work feedrate in the passes defined by "C" up to a distance "L" (finishing pass) from the pocket wall.

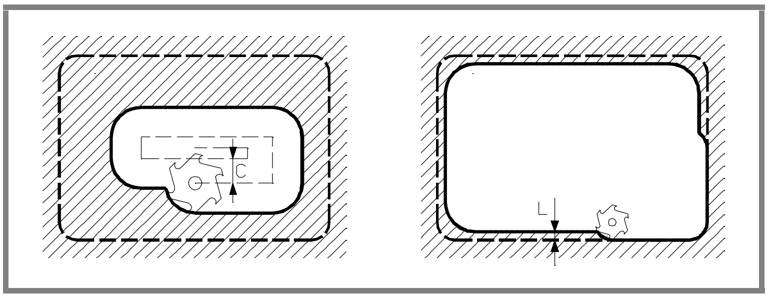
It is carried out in the direction indicated by parameter "J".



6. Finishing milling, "L" amount, at the work feedrate defined by "H".

In order to obtain good part finish when machining the pocket walls, the finishing passes are carried out with tangential entry and exit.

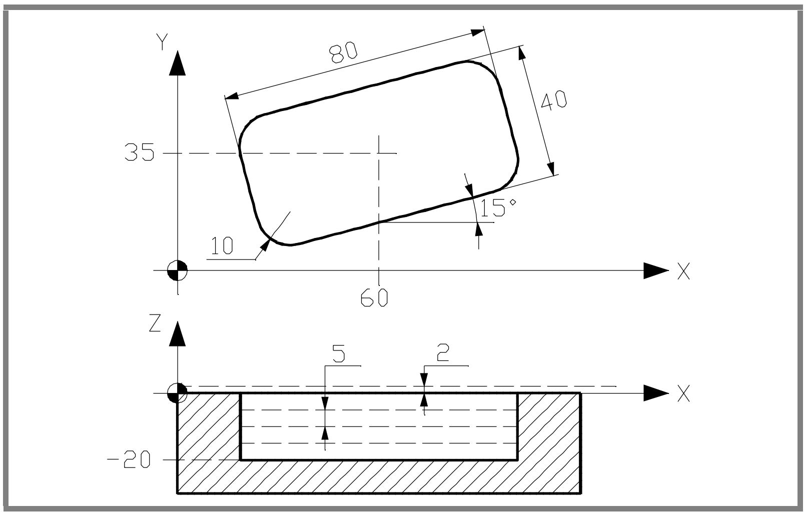
7. Rapid withdrawal (G0) to the center of the pocket, 1 mm off the machined surface.



8. New milling surfaces until reaching the total depth of the pocket.
   - Penetration, at the feedrate indicated in "F" up to a distance "B" from the previous surface.
   - Milling of the new surface following the steps indicated in points 5, 6 and 7.
9. Withdrawal to the starting plane (Zi) if function G98 is active or to the reference plane (Z) if function G99 is active.

10.8.1 Programming example

To machine a 80x40 pocket centered at (X60, Y35) and rotated 15º. The pocket surface is at Z0 and is to be emptied down to Z-20. The reference plane is located at Z2.



G90 G0 X60 Y35
G87 Z2 I-20 D2 A15 J40 K20 ·····

The pocket corners are to be rounded with a 10 mm radius.

G87 Z2 I-20 D2 A15 J40 K20 **M1 Q10** ·····

The penetrating pass is 5 mm and it is carried out at a feedrate of 50 mm/min.

G87 Z2 I-20 D2 A15 J40 K20 M1 Q10 **B5** ····· **V50**

The milling is carried out with a 5 mm wide roughing pass and at a feedrate of 800 mm/min. Since the milling feedrate must be selected before the execution of the cycle, it is defined in the previous block.

G90 G0 X60 Y35 **F800**
G87 Z2 I-20 D2 A15 J40 K20 M1 Q10 B5 **C5** ····· V50

It will leave a finishing stock of 1 mm that will be machined at a feedrate of 300 mm/min.

G87 Z2 I-20 D2 A15 J40 K20 M1 Q10 B5 C5 **L1 H300** V50

We now show how to execute a pocket and repeated in several positions (X200 Y135) and (X350 Y235).

Absolute programming:

```
        T7 D1 M6
        G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F800
N10     G99 X60 Y35
        G87 Z2 I-20 D2 A15 J40 K20 M1 Q10 B5 C5 L1 H300 V50
N20     X200 Y135
N30     G98 X350 Y235
        M30
```

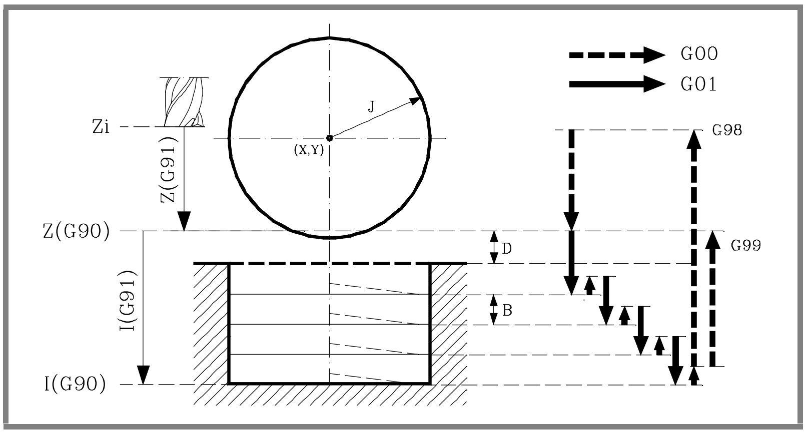
Incremental programming:

```
      T7 D1 M6
      G0 G90 X0 Y0 Z25 S1000 M3 M8 M41 F800
N10   G99 G91 X60 Y35
      G87 Z-23 I-45 D2 A15 J40 K20 M1 Q10 B5 C5 L1 H300 V50
N20   X140 Y100
N30   G98 X150 Y100
      M30
```

10.9 G88. Circular pocket canned cycle

Programming format in Cartesian coordinates:

G88 Z I D J B C L H V



Parameter definition:

Z Reference plane.

In G90, coordinate referred to part zero.
In G91, coordinate referred to starting plane (Zi).
If not programmed, it assumes as reference plane the current position of the tool (Z=Zi).

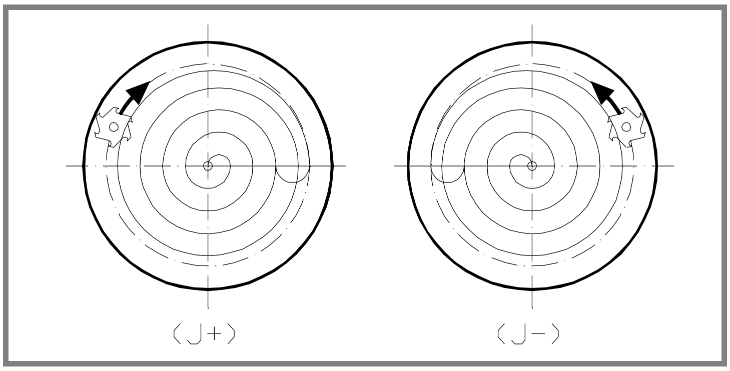
I Pocket depth.

In G90, coordinate referred to part zero.
In G91, coordinate referred to reference plane (Z).

D Distance between the reference plane and the part surface. If not programmed, it assumes a value of 0.

J Pocket radius.

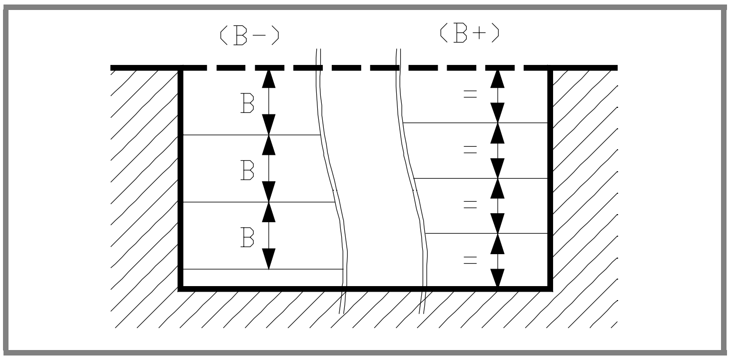
The sign indicates the pocket machining direction:
(J+) clockwise, (J-) counterclockwise.



B Depth of pass.

If programmed with a positive sign (B+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.

If programmed with a negative sign (B-), the pocket is machined with the given pass (step) except the last pass that machines the rest.

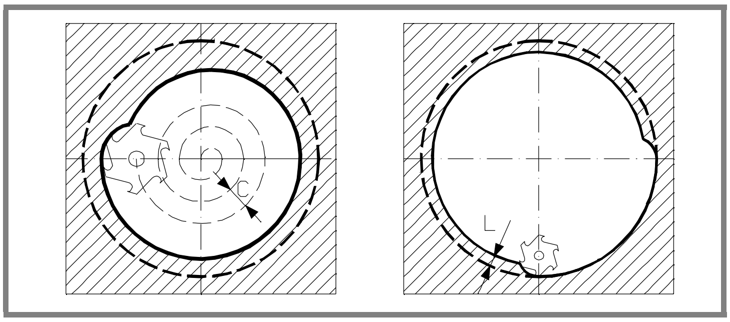


C Milling pass or width.

If not programmed or programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

If it is the same as parameter "J" (pocket radius), it only runs the finishing pass.

If programmed with a value greater than the tool diameter, the CNC issues the relevant error message.



L Finishing pass.

If not programmed or programmed with a 0 value, it does not run the finishing pass.

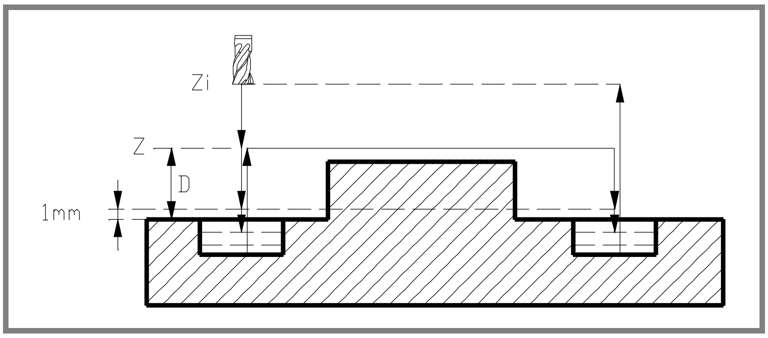
H Feedrate for the finishing pass. If not programmed or programmed with a 0 value, it is carried out at the roughing feedrate.

V Tool penetrating feedrate. If not programmed or programmed with a 0 value, it is carried out at 50% of the feedrate in the plane.

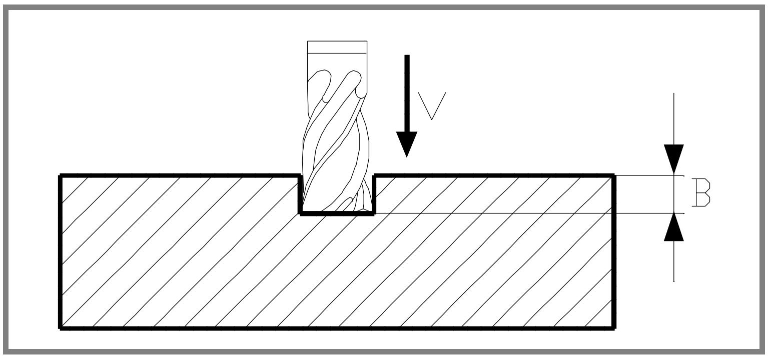
Basic operation:

1. If the spindle was previously running, it maintains the turning direction. If it is stopped, it starts clockwise (M03).
2. Rapid movement (G0) of the longitudinal axis from the starting plane (Zi) to the reference plane (Z).
3. Rapid movement (G0) of the longitudinal axis up to 1 mm off the part surface.

   The movement permits, as in the case of the figure, a fast approach to the machining surface when the safety coordinate (Z) is far away from the surface.

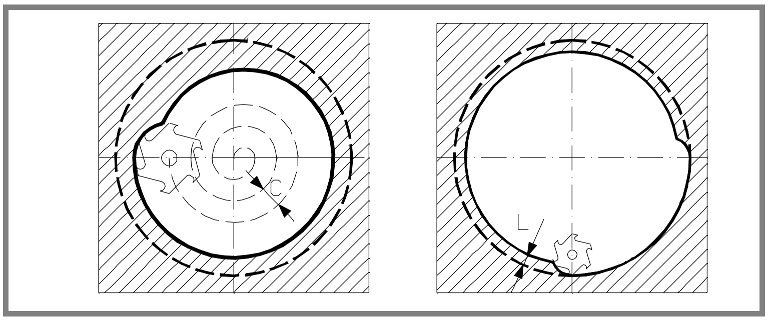


4. Penetration. The longitudinal axis penetrates into the part the distance indicated by "B" and at the feedrate indicated by "V".



5. Milling of the pocket surface at work feedrate in the passes defined by "C" up to a distance "L" (finishing pass) from the pocket wall.

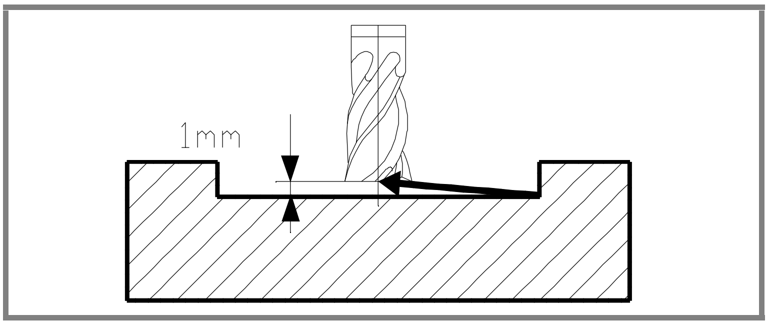
   It is carried out in the direction indicated by parameter "J".



6. Finishing milling, "L" amount, at the work feedrate defined by "H".

   In order to obtain good part finish when machining the pocket walls, the finishing passes are carried out with tangential entry and exit.

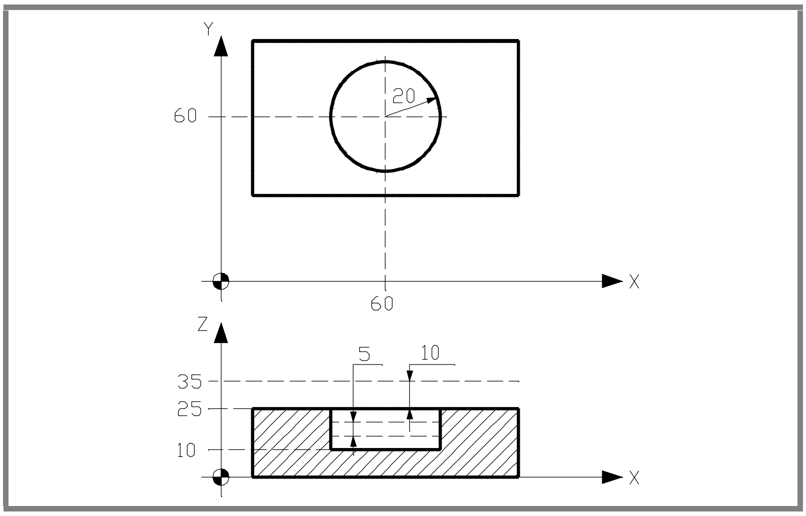
7. Rapid withdrawal (G0) to the center of the pocket, 1 mm off the machined surface.



8. New milling surfaces until reaching the total depth of the pocket.
   - Penetration, at the feedrate indicated in "F" up to a distance "B" from the previous surface.
   - Milling of the new surface following the steps indicated in points 5, 6 and 7.
9. Withdrawal to the starting plane (Zi) if function G98 is active or to the reference plane (Z) if function G99 is active.

10.9.1 Programming example

To machine a 20 mm radius pocket centered in (X60, Y60). The pocket surface is at Z25 and it is to be emptied down to Z10. The reference plane is at Z35.



G90 G0 X60 Y60

G88 Z35 I10 D10 J20 ·····

The penetrating pass is 5 mm and it is carried out at a feedrate of 50 mm/min.

G88 Z35 I10 D10 J20 **B5** ····· **V50**

The milling is carried out with a 5 mm wide roughing pass and at a feedrate of 800 mm/min. Since the milling feedrate must be selected before the execution of the cycle, it is defined in the previous block.

G90 G0 X60 Y60 **F800**

G88 Z35 I10 D10 J20 B5 **C5** ····· V50

It will leave a finishing stock of 1 mm that will be machined at a feedrate of 300 mm/min.

G88 Z35 I10 D10 J20 B5 C5 **L1** **H300** V50

We now show how to execute a pocket and repeated in several positions (X200 Y135) and (X350 Y235).

Absolute programming:

```
      T8 D1 M6
      G0 G90 X0 Y0 Z45 S1000 M3 M8 M41 F800
N10   G99 X60 Y60
      G88 Z35 I10 D10 J20 B5 C5 L1 H300 V50
N20   X200 Y135
N30   G98 X350 Y235
      M30
```

Incremental programming:	
	T8 D1 M6
	G0 G90 X0 Y0 Z45 S1000 M3 M8 M41 F800
N10	G99 G91 X60 Y60
	G87 Z-10 I-35 D10 J20 B5 C5 L1 H300 V50
N20	X140 Y75
N30	G98 X150 Y100
	M30

MULTIPLE MACHINING

11

The programmer selects the type of machining that could be any canned cycle.

Programming

The machining paths are defined by the following functions:

G160	Multiple machining in a straight line.
G161	Multiple machining in a parallelogram pattern.
G162	Multiple machining in a grid pattern.
G163	Multiple machining in a full circle.
G164	Multiple machining in an arc.
G165	Machining programmed with an arc-chord.

These functions may be executed in any work plane and must be defined every time they are used because they not modal.

The machining operation to be repeated MUST BE active. In other words, these functions will only make sense if they are under the influence (affected by) a canned cycle.

Follow these steps to carry out a multiple machining operation:

1. Move the tool to the first point where the multiple machining will take place.
2. Define the canned cycle to be repeated at all the points.
3. Define the multiple machining operation to be carried out.

Considerations

All the machining operations programmed with these functions are carried out under the same working conditions (T, D, F, S) that were selected when the canned cycle was defined.

Once the programmed multiple machining has been executed, the program will restore the history that it had before starting the machining operation, the canned cycle will even remain active. F now being the feedrate for the feedrate programmed for the canned cycle.

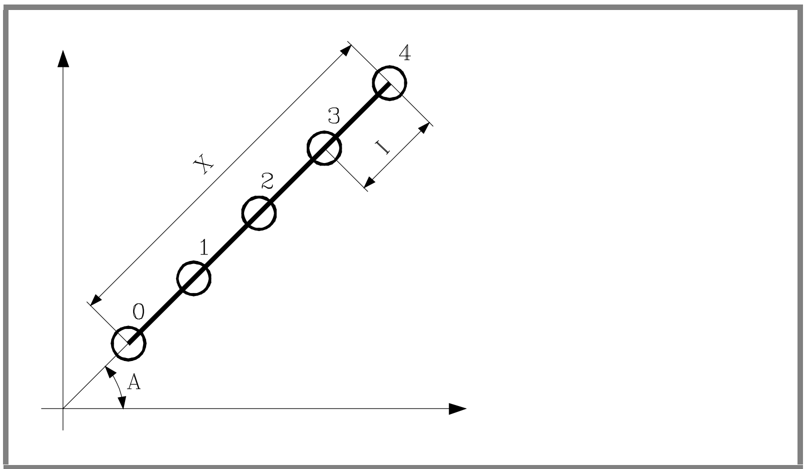
Likewise, the tool will be positioned at the last point where the programmed machining operation was carried out.

A detailed description is given of the multiple machining operations assuming in all of them that the work plane is formed by the X and Y axis.

11.1 G160. Multiple machining in straight line

The programming format for this cycle is:

```
G160  A | X I | P Q R S T U V
        | X K |
        | I K |
```



A Angle, in degrees of the machining path with respect to the abscissa axis.

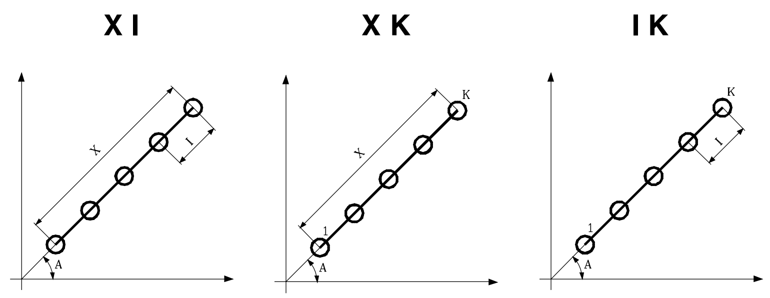
If not programmed, a value of A = 0 is assumed.

When defining the machining operation, only two of parameters "X", "I" and "K" are required.

X Length of the machining path.

I Step between machining operations.

K Total number of machining operations in the section, including that of the machining definition point.



When selecting the "X-I" format, bear in mind that the resulting number of machining operations must be an integer, otherwise, the CNC will issue the relevant error message.

P,Q,R,S,T,U,V These parameters are optional and are used to indicate at which points or between which points of the ones programmed the machining operation will NOT be carried out.

Thus, programming "P7" means that no machining operation takes place at point 7. Programming "Q10.013" means that no machining takes place at points 10, 11, 12 and 13.

When defining a set of points (Q10.013), bear in mind that the last point must be defined with three digits because, for example, "Q10.13" is the same as programming "Q10.130".

The programming order for these parameters is "P" "Q" "R" "S" "T" "U" "V" and the numbering sequence for the points assigned to them must also be respected; In other words, the numbering sequence of the points assigned to "Q" must be greater than the one of those assigned to "P" and smaller than the one for those assigned to "R".

Example:

Correct programming	P5.006 Q12.015 R20.022
Wrong programming	P5.006 Q20.022 R12.015

If these parameters are not programmed, the CNC executes the machining operation at all the points of the programmed path.

Basic operation

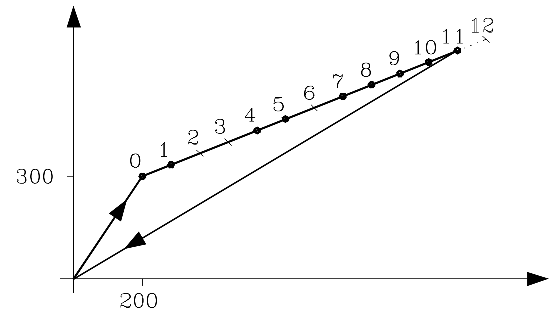
Multiple machining is executed as follows:

1. The multiple machining calculates the next programmed point to machine.
2. Rapid movement (G00) to that point.
3. The multiple machining will execute the selected canned cycle after the movement.
4. The CNC will repeat steps 1-2-3 until completing the programmed multiple machining operation.

   After completing the multiple machining, the tool will remain positioned at the last point of the programmed path where the machining operation took place.

11.1.1 Programming example

Programming example assuming that the work plane is formed by the X and Y axes, that the Z axis is the longitudinal axis and that the starting point is X0 Y0 Z0:




```
G00 G91 X200 Y300 F100 S500
G98 G81 Z-8 I-22
G160 A30 X1200 I100 P2.003 Q6 R12
G80
G90 X0 Y0
M30
```

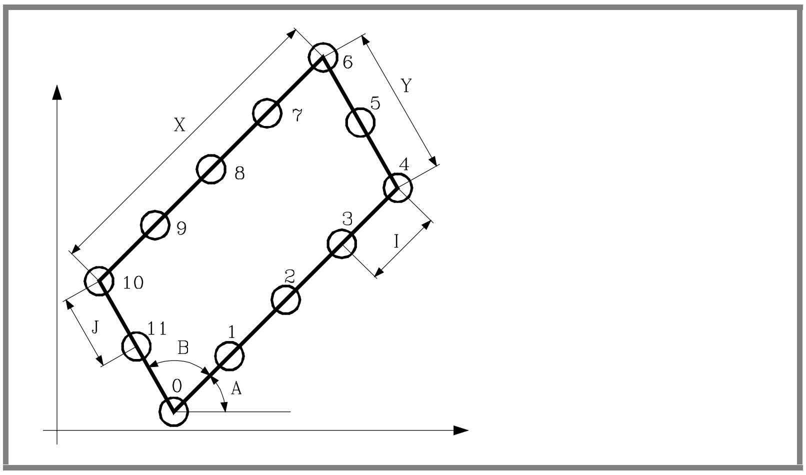
The multiple machining definition block may also be defined as follows:

```
G160 A30 X1200 K13 P2.003 Q6 R12
G160 A30 I100 K13 P2.003 Q6 R12
```

11.2 G161. Multiple machining in rectangular pattern

The programming format for this cycle is:

G161 A B	X I	Y J	P Q R S T U V
	X K	Y D	
	I K	J D	



A Angle, in degrees of the machining path with respect to the abscissa axis.

If not programmed, a value of A = 0 is assumed.

B Angle between both machining paths.

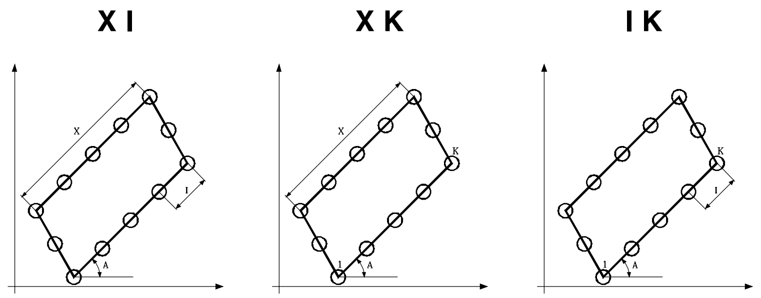
If not programmed, a value of B = 90 is assumed.

When defining the length of the parallelogram, only two of parameters "X", "I" and "K" are required.

X Length of the parallelogram.

I Step between machining operations along the path.

K Total number of machining operations along the path, including that of the machining definition point.



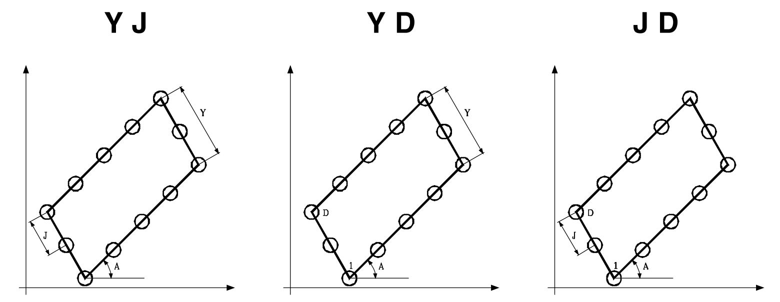
When selecting the "X-I" format, bear in mind that the resulting number of machining operations must be an integer, otherwise, the CNC will issue the relevant error message.

When defining the width of the parallelogram, only two of parameters "Y", "J" and "D" are required.

Y Width of the parallelogram.

J Step between machining operations along the path.

D Total number of machining operations along the path, including that of the machining definition point.



When selecting the "Y-J" format, bear in mind that the resulting number of machining operations must be an integer, otherwise, the CNC will issue the relevant error message.

P,Q,R,S,T,U,V These parameters are optional and are used to indicate at which points or between which points of the ones programmed the machining operation will NOT be carried out.

Thus, programming "P7" means that no machining operation takes place at point 7. Programming "Q10.013" means that no machining takes place at points 10, 11, 12 and 13.

When defining a set of points (Q10.013), bear in mind that the last point must be defined with three digits because, for example, "Q10.13" is the same as programming "Q10.130".

The programming order for these parameters is "P" "Q" "R" "S" "T" "U" "V" and the numbering sequence for the points assigned to them must also be respected; In other words, the numbering sequence of the points assigned to "Q" must be greater than the one of those assigned to "P" and smaller than the one for those assigned to "R".

Example:

Correct programming	P5.006 Q12.015 R20.022
Wrong programming	P5.006 Q20.022 R12.015

If these parameters are not programmed, the CNC executes the machining operation at all the points of the programmed path.

Basic operation

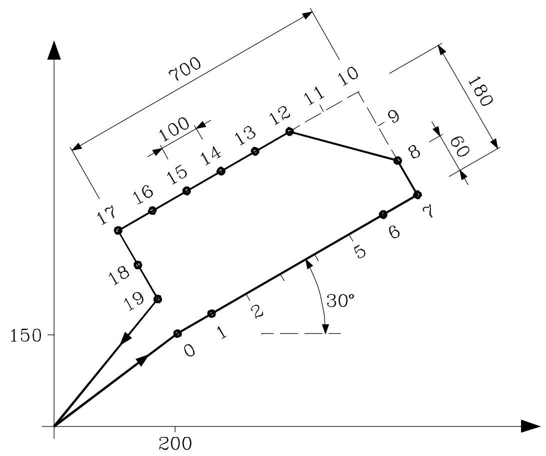
Multiple machining is executed as follows:

1. The multiple machining calculates the next programmed point to machine.
2. Rapid movement (G00) to that point.
3. The multiple machining will execute the selected canned cycle after the movement.
4. The CNC will repeat steps 1-2-3 until completing the programmed multiple machining operation.

   After completing the multiple machining, the tool will remain positioned at the last point of the programmed path where the machining operation took place.

11.2.1 Programming example

Programming example assuming that the work plane is formed by the X and Y axes, that the Z axis is the longitudinal axis and that the starting point is X0 Y0 Z0:



G00 G91 X100 Y150 F100 S500

G98 G81 Z-8 I-22

G161 A30 X700 I100 Y180 J60 P2.005 Q9.011

G80

G90 X0 Y0

M30

The multiple machining definition block may also be defined as follows:

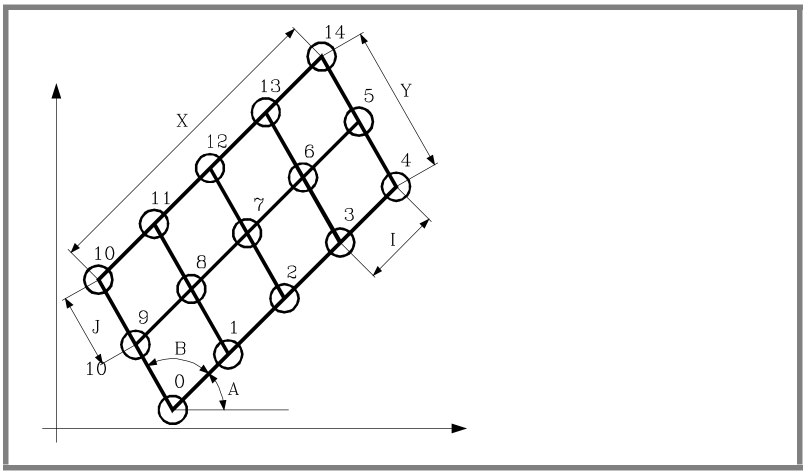
G161 A30 X700 K8 J60 D4 P2.005 Q9.011

G161 A30 I100 K8 Y180 D4 P2.005 Q9.011

11.3 G162. Multiple machining in grid pattern

The programming format for this cycle is:

G162 A B	X I	Y J	P Q R S T U V
	X K	Y D	
	I K	J D	



A Angle, in degrees of the machining path with respect to the abscissa axis.

If not programmed, a value of A = 0 is assumed.

B Angle between both machining paths.

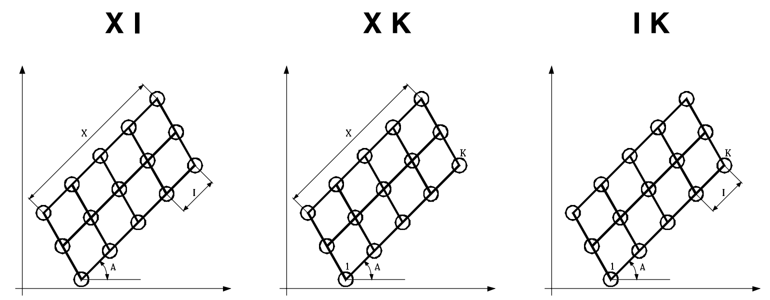
If not programmed, a value of B = 90 is assumed.

When defining the length of the grid only two of parameters "X", "I" and "K" are required.

X Length of the drid.

I Step between machining operations along the path.

K Total number of machining operations along the path, including that of the machining definition point.



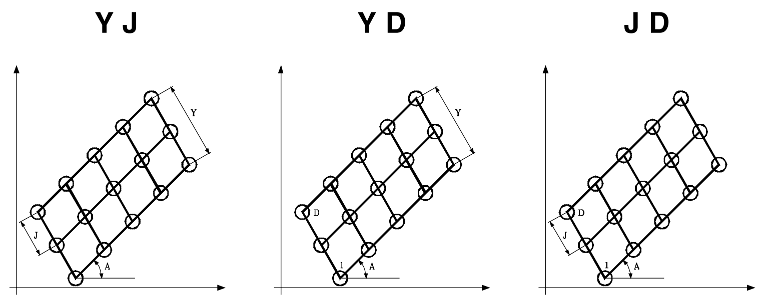
When selecting the "X-I" format, bear in mind that the resulting number of machining operations must be an integer, otherwise, the CNC will issue the relevant error message.

When defining the width of the grid, only two of parameters "Y", "J" and "D" are required.

Y Width of the drid.

J Step between machining operations along the path.

D Total number of machining operations along the path, including that of the machining definition point.



When selecting the "Y-J" format, bear in mind that the resulting number of machining operations must be an integer, otherwise, the CNC will issue the relevant error message.

P,Q,R,S,T,U,V These parameters are optional and are used to indicate at which points or between which points of the ones programmed the machining operation will NOT be carried out.

Thus, programming "P7" means that no machining operation takes place at point 7. Programming "Q10.013" means that no machining takes place at points 10, 11, 12 and 13.

When defining a set of points (Q10.013), bear in mind that the last point must be defined with three digits because, for example, "Q10.13" is the same as programming "Q10.130".

The programming order for these parameters is "P" "Q" "R" "S" "T" "U" "V" and the numbering sequence for the points assigned to them must also be respected; In other words, the numbering sequence of the points assigned to "Q" must be greater than the one of those assigned to "P" and smaller than the one for those assigned to "R".

Example:

Correct programming	P5.006 Q12.015 R20.022
Wrong programming	P5.006 Q20.022 R12.015

If these parameters are not programmed, the CNC executes the machining operation at all the points of the programmed path.

Basic operation

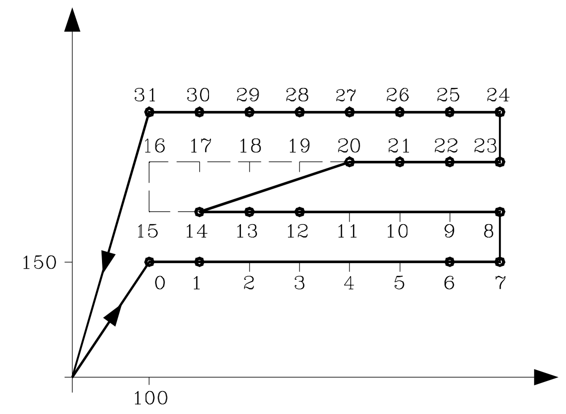
Multiple machining is executed as follows:

1. The multiple machining calculates the next programmed point to machine.
2. Rapid movement (G00) to that point.
3. The multiple machining will execute the selected canned cycle after the movement.
4. The CNC will repeat steps 1-2-3 until completing the programmed multiple machining operation.

   After completing the multiple machining, the tool will remain positioned at the last point of the programmed path where the machining operation took place.

11.3.1 Programming example

Programming example assuming that the work plane is formed by the X and Y axes, that the Z axis is the longitudinal axis and that the starting point is X0 Y0 Z0:




```
G00 G91 X100 Y150 F100 S500
G98 G81 Z-8 I-22
G162 X700 I100 Y180 J60 P2.005 Q9.011 R15.019
G80
G90 X0 Y0
M30
```

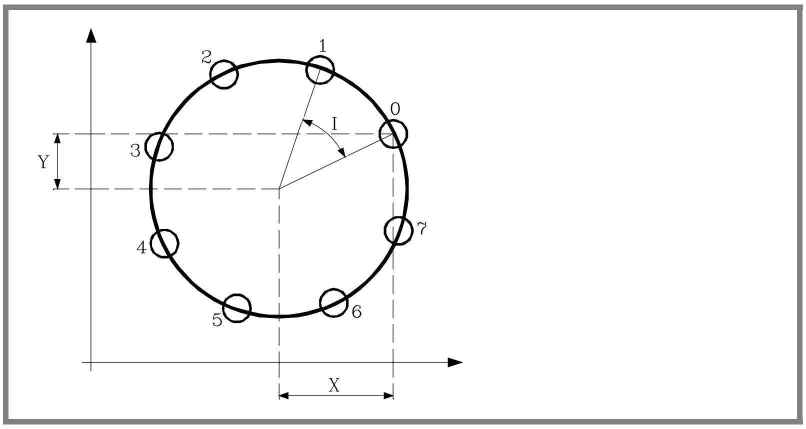
The multiple machining definition block may also be defined as follows:

```
G162 X700 K8 J60 D4 P2.005 Q9.011 R15.019
G162 I100 K8 Y180 D4 P2.005 Q9.011 R15.019
```

11.4 G163. Multiple machining in a full circle

The programming format for this cycle is:

G163 X Y | I | C F P Q R S T U V
         | K |



Parameters "X" and "Y" define the center of the circle, same as "I" and "J" in circular interpolations (G02, G03).

X Distance from the starting point to the center along the abscissa axis.

Y Distance from the starting point to the center along the ordinate axis.

When defining the machining operation, only one of parameters "I" and "K" is required. If the angular step is programmed, bear in mind that the total angular movement must be 360º, otherwise, the CNC will issue the relevant error message.

I Angular step between machining operations.

When the movement between points is done in G00 or G01, the sign indicates the direction: "I+" counterclockwise and "I-" clockwise.

K Total number of machining operations including that of the machining definition point.

When the movement between points is done in G00 or G01, the machining operation is carried out counterclockwise.

C It indicates how it will move between the machining points. If not programmed, a value of C = 0 is assumed.

C=0 In rapid (G00).

C=1 Linear interpolation (G01).

C=2 In clockwise circular interpolation (G02).

C=3 In counterclockwise circular interpolation (G03).

F Feedrate for the movement between points. It will only be valid for "C" values other than zero.

P,Q,R,S,T,U,V These parameters are optional and are used to indicate at which points or between which points of the ones programmed the machining operation will NOT be carried out.

Thus, programming "P7" means that no machining operation takes place at point 7. Programming "Q10.013" means that no machining takes place at points 10, 11, 12 and 13.

When defining a set of points (Q10.013), bear in mind that the last point must be defined with three digits because, for example, "Q10.13" is the same as programming "Q10.130".

The programming order for these parameters is "P" "Q" "R" "S" "T" "U" "V" and the numbering sequence for the points assigned to them must also be respected; In other words, the numbering sequence of the points assigned to "Q" must be greater than the one of those assigned to "P" and smaller than the one for those assigned to "R".

Example:

Correct programming	P5.006 Q12.015 R20.022
Wrong programming	P5.006 Q20.022 R12.015

If these parameters are not programmed, the CNC executes the machining operation at all the points of the programmed path.

Basic operation

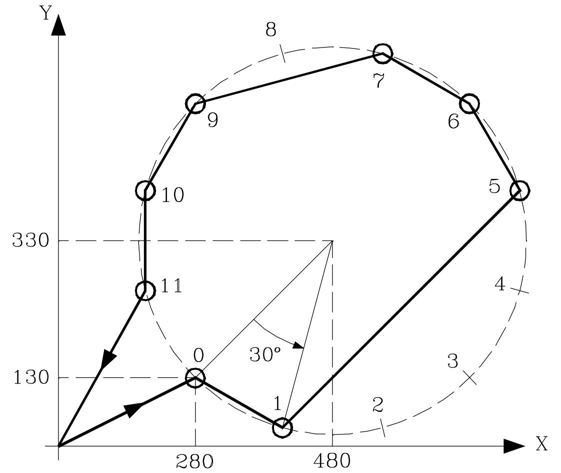
Multiple machining is executed as follows:

1. The multiple machining calculates the next programmed point to machine.
2. Movement to that point at the feedrate programmed with "C" (G00, G01, G02 or G03).
3. The multiple machining will execute the selected canned cycle after the movement.
4. The CNC will repeat steps 1-2-3 until completing the programmed multiple machining operation.

   After completing the multiple machining, the tool will remain positioned at the last point of the programmed path where the machining operation took place.

11.4.1 Programming example

Programming example assuming that the work plane is formed by the X and Y axes, that the Z axis is the longitudinal axis and that the starting point is X0 Y0 Z0:




```
G00 G91 X280 Y130 F100 S500
G98 G81 Z-8 I-22
G163 X200 Y200 I30 C1 F200 P2.004 Q8
G80
G90 X0 Y0
M30
```

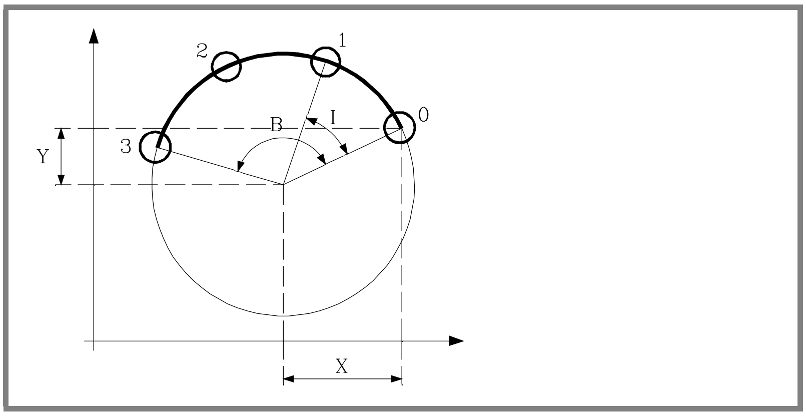
The multiple machining definition block may also be defined as follows:

```
G163 X200 Y200 K12 C1 F200 P2.004 Q8
```

11.5 G164. Multiple machining in arc pattern

The programming format for this cycle is:

G164 X Y B | I | C F P Q R S T U V
K



Parameters "X" and "Y" define the center of the circle, same as "I" and "J" in circular interpolations (G02, G03).

X Distance from the starting point to the center along the abscissa axis.

Y Distance from the starting point to the center along the ordinate axis.

B Angular distance in degrees of the machining path.

When defining the machining operation, only one of parameters "I" and "K" is required. If the angular step is programmed, bear in mind that the total angular movement must be the programmed angular distance "B", otherwise, the CNC will issue the relevant error message.

I Angular step between machining operations.

When the movement between points is done in G00 or G01, the sign indicates the direction: "I+" counterclockwise and "I-" clockwise.

K Total number of machining operations including that of the machining definition point.

When the movement between points is done in G00 or G01, the machining operation is carried out counterclockwise.

C It indicates how it will move between the machining points. If not programmed, a value of C = 0 is assumed.

C=0 In rapid (G00).

C=1 Linear interpolation (G01).

C=2 In clockwise circular interpolation (G02).

C=3 In counterclockwise circular interpolation (G03).

F Feedrate for the movement between points. It will only be valid for "C" values other than zero.

P,Q,R,S,T,U,V These parameters are optional and are used to indicate at which points or between which points of the ones programmed the machining operation will NOT be carried out.

Thus, programming "P7" means that no machining operation takes place at point 7. Programming "Q10.013" means that no machining takes place at points 10, 11, 12 and 13.

When defining a set of points (Q10.013), bear in mind that the last point must be defined with three digits because, for example, "Q10.13" is the same as programming "Q10.130".

The programming order for these parameters is "P" "Q" "R" "S" "T" "U" "V" and the numbering sequence for the points assigned to them must also be respected; In other words, the numbering sequence of the points assigned to "Q" must be greater than the one of those assigned to "P" and smaller than the one for those assigned to "R".

Example:

Correct programming	P5.006 Q12.015 R20.022
Wrong programming	P5.006 Q20.022 R12.015

If these parameters are not programmed, the CNC executes the machining operation at all the points of the programmed path.

Basic operation

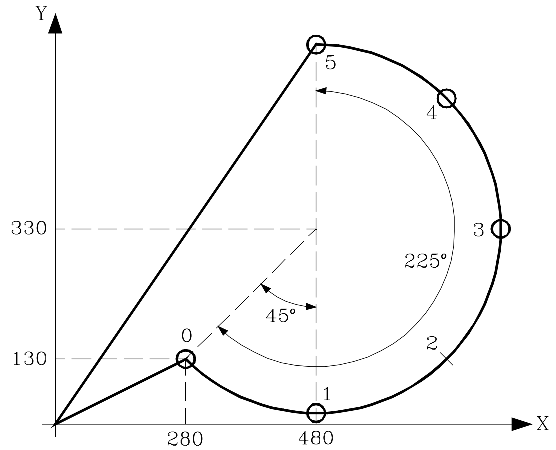
Multiple machining is executed as follows:

1. The multiple machining calculates the next programmed point to machine.
2. Movement to that point at the feedrate programmed with "C" (G00, G01, G02 or G03).
3. The multiple machining will execute the selected canned cycle after the movement.
4. The CNC will repeat steps 1-2-3 until completing the programmed multiple machining operation.

   After completing the multiple machining, the tool will remain positioned at the last point of the programmed path where the machining operation took place.

11.5.1 Programming example

Programming example assuming that the work plane is formed by the X and Y axes, that the Z axis is the longitudinal axis and that the starting point is X0 Y0 Z0:



G00 G91 X280 Y130 F100 S500

G98 G81 Z-8 I-22

G164 X200 Y200 B225 I45 C3 F200 P2

G80

G90 X0 Y0

M30

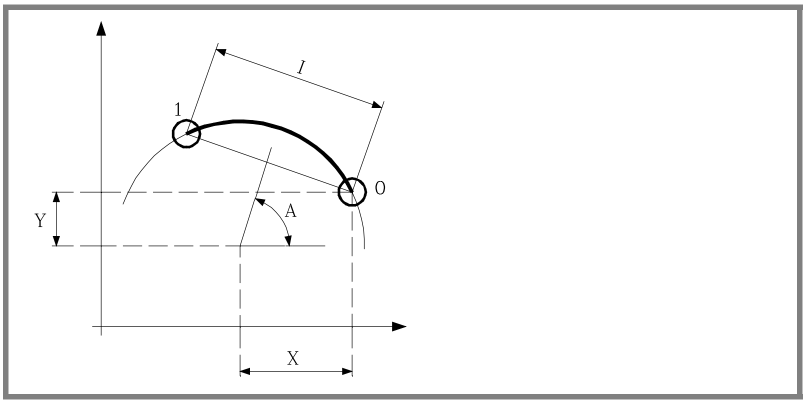
The multiple machining definition block may also be defined as follows:

G164 X200 Y200 B225 K6 C3 F200 P2

11.6 G165. Multiple machining in a chord pattern

With this function, it is possible to execute the active machining operation at the point programmed with an arch chord. Only one machining operation will be executed and its programming format is:

G165 X Y | A | C F
| I |



Parameters "X" and "Y" define the center of the circle, same as "I" and "J" in circular interpolations (G02, G03).

X Distance from the starting point to the center along the abscissa axis.

Y Distance from the starting point to the center along the ordinate axis.

When defining the machining operation, only one of parameters "A" and "I" is required.

A Angle, in degrees of the perpendicular bisector of the chord with respect to the abscissa axis.

I Length of the chord.

When the movement between points is done in G00 or G01, the sign indicates the direction: "I+" counterclockwise and "I-" clockwise.

C It indicates how it will move between the machining points. If not programmed, a value of C = 0 is assumed.

C=0 In rapid (G00).

C=1 Linear interpolation (G01).

C=2 In clockwise circular interpolation (G02).

C=3 In counterclockwise circular interpolation (G03).

F Feedrate for the movement between points. It will only be valid for "C" values other than zero.

Basic operation

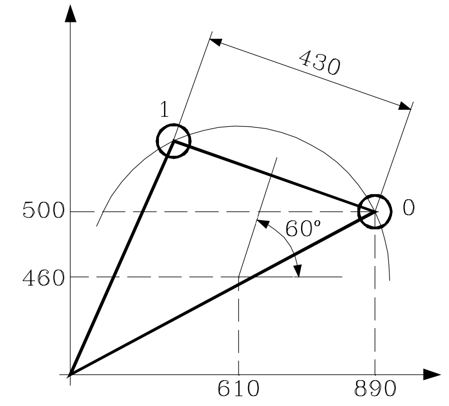
Multiple machining is executed as follows:

1. The multiple machining calculates the programmed point to machine.
2. Movement to that point at the feedrate programmed with "C" (G00, G01, G02 or G03).
3. The multiple machining will execute the selected canned cycle after the movement.

   After the multiple machining, the tool will remain positioned at the programmed point.

11.6.1 Programming example

Programming example assuming that the work plane is formed by the X and Y axes, that the Z axis is the longitudinal axis and that the starting point is X0 Y0 Z0:



G00 G91 X890 Y500 F100 S500

G98 G81 Z-8 I-22

G165 X-280 Y-40 A60 C1 F200

G80

G90 X0 Y0

M30

The multiple machining definition block may also be defined as follows:

G165 X-280 Y-40 I430 C1 F200

CYCLE EDITOR

12

12.1 General concepts

The cycles integrated into the cycle editor are grouped as follows:

Machining canned cycles.

- Drilling:
  Center punching, Drilling 1, Drilling 2
- Tapping.
- Reaming
- Boring
  Boring 1, Boring 2
- Pockets
  Pocket: Simple, Rectangular, Circular, Pre-emptied, 2D, 3D
- Bosses
  Boss: Rectangular, Circular
- Surface milling
- Profile milling
  Point-to-point profile, Profile
- Slot milling

Multiple machining.

- Linear.
- Arc.
- Rectangle.
- Grid.
- Random (several points defined by the user).

Multiple machining may be associated with canned cycles so it can be repeated in several points.

Execution

While executing these canned cycles, the CNC shows the following "G" functions in the window for active functions.

G281	Center punching.
G282	Drilling 1.
G283	Drilling 2.
G284	Tapping.
G285	Reaming.
G286	Boring 1.
G297	Boring 2.
G287	Rectangular pocket.
G288	Circular pocket.
G289	Simple pocket.
G296	Pre-emptied pocket.
G291	Rectangular boss.
G292	Circular boss.
G290	Surface milling.
G293	Point-to-point profile.
G294	Profile.
G295	Slot milling.

12.1.1 Associate a multiple machining operation with a canned cycle

A multiple machining operation may be associated with the following cycles:

- Center punching, Drilling 1, Drilling 2, Tapping, Reaming, Boring 1, Boring 2.
- simple, rectangular, circular and pre-emptied pocket.
- rectangular and circular boss.

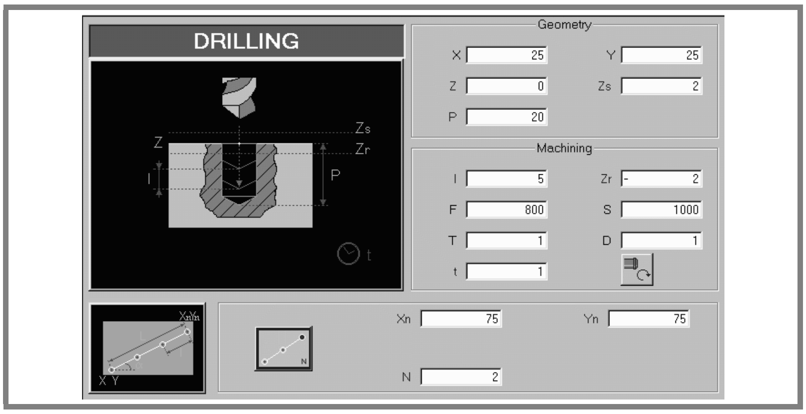
However, no multiple machining operation may be associated with the following cycles.

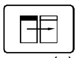
- 2D and 3D pockets, Surface milling, Profile, Point-to-point profile and Slot milling.

To associate a multiple machining operation with a cycle:

1. Select and define the canned cycle.
2. Press the "Multiple" softkey.
3. Select the desired multiple machining operation.

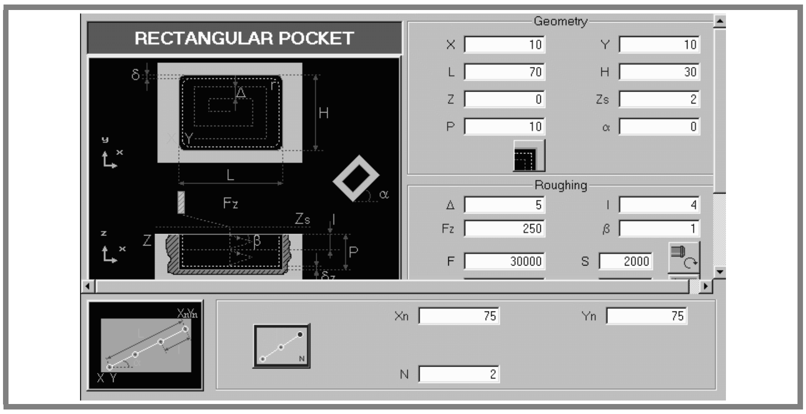
The next figure shows the Drilling 1 cycle (top) with a multiple linear machining operation associated to it (bottom).



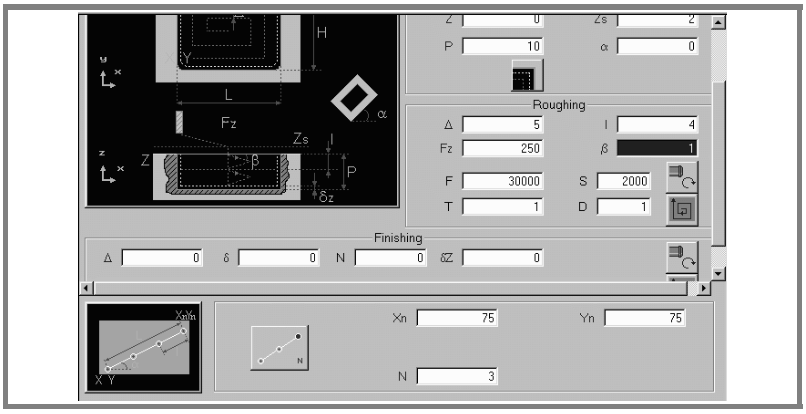

(a)

To edit the data of the canned cycle or of the multiple machining operation, select the relevant window using the (a) key.

When the canned cycle takes up the whole screen, the multiple machining operation is super-imposed on it as shown in the figure.



In these cases, while editing the cycle data, the top window is shifted automatically to show the data.



The canned cycle editing windows are generic. They do not depend on the active work plane.

The canned cycles have no work plane associated to them, they are executed in the current active work plane.

The same nomenclature as for the G17 work plane has been used.

X abscissa axis.

Y ordinate axis.

Z longitudinal axis.

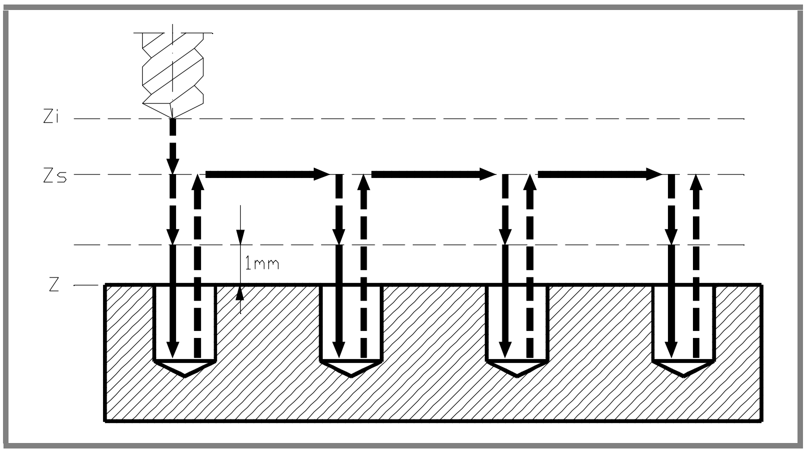
When working in another plane, one must:

- *Select the proper work plane.*
  G17, G18, G19 or instruction #SET AX.
- *Select longitudinal axis and machining direction.*
  Instruction #TOOL AX.
- *Program the cycles considering the previous nomenclature.*

12.1.2 Machining movements

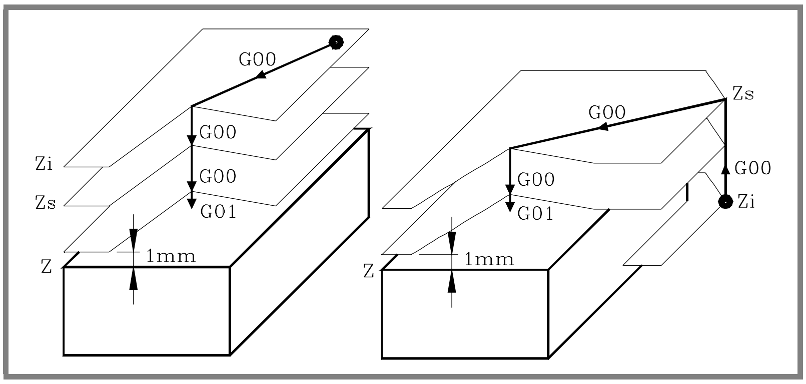
There are four work planes in all the operations:

- Starting plane or tool position when calling the cycle Zi). No need to define it.
- Safety plane. It is used for the first approach and to move the tool between machining operations. It is defined by cycle parameter Zs.
- Part approaching plane. No need to define it. It is calculated by the CNC, 1 mm off the part surface.
- Part surface. It is defined using the Z parameter.



When executing the cycle, the tool moves in rapid (G0) to the safety plane (Zs):

- If the starting plane is above the safety plane (left figure), it first moves on X, Y and then on Z.
- If the starting plane is below the safety plane (right figure), it first moves on Z up to the safety plane and then on X, Y.

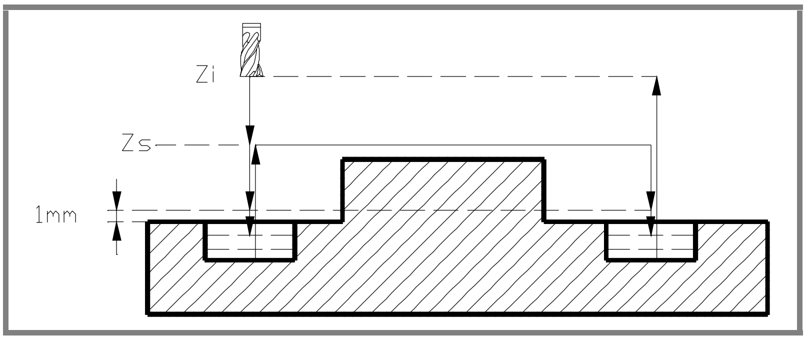


Then, it moves in rapid (G0) to the approach plane and finally at working feedrate to carry out the machining operation.

Once the machining operation has concluded, the tool returns to the safety plane (Zs).

If it has a multiple machining associated to it, the tool moves in XY, along the safety plane (Zs), up to the next point to be machined.

The approach plane permits, as in the case of the figure, a fast approach to the machining surface when the safety plane (Zs) is far away from the part surface.



12.1.3 Selecting data, profiles and icons

Data selection.

To enter or modify a data, it must be selected; i.e. it must have the editing focus on it.

The parameters of the cycles may be selected with the [🡄] [🡆] [🡅] [🡇] keys or with the direct access keys. The first data of each group may also be selected by pressing the page-up and page-down keys.

The direct access keys correspond to the name of the parameters; [F] for feedrates, [T] for tools, etc. Every time the same key is pressed, it selects the next data of the same type.

Data entry.

Place the cursor in the relevant window, key in the desired value and press [ENTER]. If [ENTER] is not pressed, the new value will not be assumed.

If the Teach-in mode is selected, the current position of the machine may be associated with a coordinate. Place the cursor in the relevant window and press the [RECALL] key.

For the X axis parameters, it will take the coordinate of the first axis of the channel where the edit-simulation mode is active. For the Y axis parameters, the coordinate of the second axis and for the Z axis parameters, the coordinate of the third one.

Changing the state of an icon.

Place the cursor on the desired icon and press the space bar.

Select - Define a profile.

Place the cursor in the relevant window.

To select one, press the [🡇] key to expand the list of defined profiles and select one or key in its name.

To define a new one, key in the desired name or press the [RECALL] key. It accesses the profile editor.

To modify an existing one, key its name or press the [RECALL] key. It accesses the profile editor.

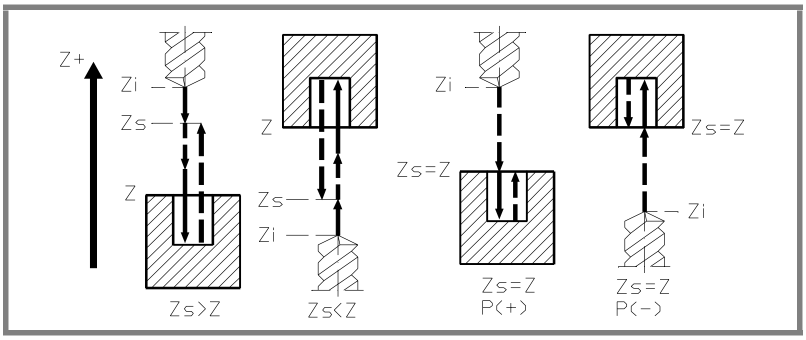
12.1.4 Value applied when the value of a parameter is 0

Machining direction:

Z and Zs set the machining direction.

If Z=Zs, the direction is set by the sign of P (total depth).

If P(+) direction towards Z(-) and if P(-) towards Z(+).



Penetration step I=0:

When programming I=0, it assumes as step the cutting length assigned to the tool in the tool table.

An error will be issued if the table value is also 0.

Penetration feedrate Fz=0:

When programming Fz=0, the roughing and finishing penetration takes place at half the milling feedrate "F" selected for each operation.

Penetration angles β=0 and θ=0:

In both cases, when programming 0, it takes the value assigned to the table in the tool table.

If the table value is also 0, it penetrates vertically, without inclination, 90° angle.

Finishing passes or number of penetrations N=0:

When programming N=0, it carries out the least amount of passes possible, considering the cutting length assigned to the tool in the tool table.

In pockets and bosses (except in 2D and 3D pockets), if the table value is also 0, it checks the roughing and finishing tools. If it is the same, the wall finishing is carried with tangential entry and exit at each penetration after the roughing operation.

An error will be issued if they are different.

12.1.5 Simulate a canned cycle

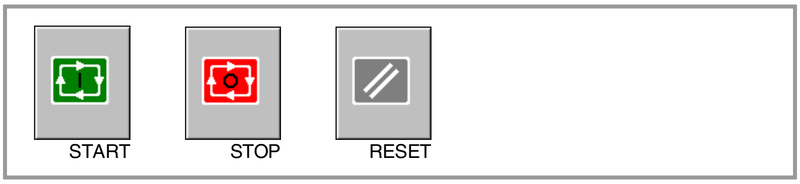
At the canned cycle editor, it is possible to simulate the cycle being edited without having to simulate the whole part-program. During simulation, another canned cycle may be viewed and edited and it is also possible to return to the program editor.

If the cycle editor is included in the automatic operating mode, it will not be possible to simulate a cycle.

Simulating a cycle

Pressing the [START] icon begins the simulation of the cycle that is being edited. The simulation may be interrupted with the [STOP] icon or canceled with the [RESET] icon.

The simulation graphics is always superimposed on the help graphics of the main cycle. If the cycle has a positioning associated with it, the graphics is superimposed on the main cycle; in the case of a 2D pocket with drilling, on the pocket.



Once the simulation has started, it is maintained until the cycle is over or the [RESET] icon is pressed. Even when changing cycles or returning to the program editor during simulation, the previous cycle is still in effect during the simulation.

Cycle simulation window

The graphics window (in simulation) is activated by pressing the [START] icon and is canceled by pressing the [RESET] icon. This window is placed over the cycle help graphics; it may be expanded to full screen (or shrink it again) using the key combination [CTRL]+[G].

The lower left corner of the window indicates the name of the cycle and the simulation channel, which will be the channel of the program editor from which the cycle editor has been called.

Configuring the graphic environment

When activating or selecting the graphics window, the horizontal softkey menu shows the available graphic options. For further information on the graphic options, see the chapter on the edit-simulation mode of the operation manual.

Some graphic options can also be edited manually. The editing area is only shown when the window is expanded ([CTRL]+[G]).

The simulated graphics are maintained until erased; i.e. starting to simulate a new cycle does not erase the previous graphics.

Best area for displaying the graphics

The display are may be established from the softkey menu associated with the simulation graphics window or may be left up to the CNC to periodically calculate the best area.

While the graphics window is visible, the key combination [CTRL]+[D] activates the calculation of the best area. From that moment on and until quitting the cycle editor, the CNC periodically calculates the best display are for the graphics.

When quitting the graphics, it will assume as the new display area the one calculated last.

Window for simulation and data editing

While the graphics window is selected, it may be switched to the cycle parameter area using the direct access keys. If the parameter belongs to a positioning cycle, firs press [CTRL]+[F2] (window change)

If the cycle simulated at full screen, the cycle editor may also be accessed by pressing the [ESC] key. To select the graphics window again, use the key combination [CTRL]+[G] or [SHIFT]+[G] or [G].

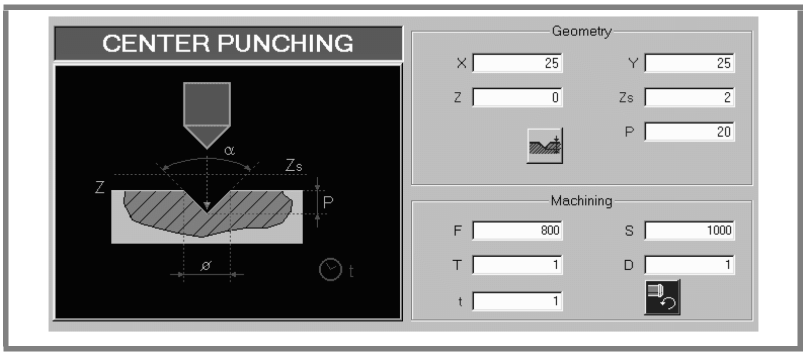
The horizontal softkey menu will show the graphic options when the graphics window has the focus and those of the cycle editor if otherwise.

The simulation in progress is not interrupted while editing data. If the cycle data is changed during simulation, they will be assumed for the next simulation of the cycle; i.e. after RESETting the simulation in progress once it has finished or after a STOP and RESET to abort it.

Summary of the hotkeys while simulating a cycle.

[CTRL]+[F2]	In the parameter window, it toggles between the cycle parameters and the positioning parameters.
[CTRL]+[G]	It selects the graphics window. It shrinks or expands the graphics window. It shows the dialog area for the graphics data.
[CTRL]+[D]	It activates the periodic calculation of the best display area.
[SHIFT]+[G] [G]	It shows the graphics window when a simulation is running and the parameter editing window is active.
[ESC]	If the graphics are shown at full screen, it shows the cycle editor screen.

12.2 Center punching

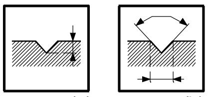


Geometric parameters:

X, Y Machining point.

Z Part surface coordinate.

Zs Safety plane coordinate.


(a) (b)

Depth programming type (icon).

P Total depth. With icon[(a)].

τo Center-punching angle. With icon[(b)].

ϕ Center-punching diameter. With icon[(b)].

With Z=Zs and icon[(b)] the machining direction is always towards Z(-)

Machining parameters:

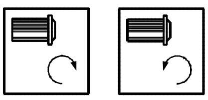
F Feedrate.

S Spindle speed.

T Tool.

D Tool offset.

t Dwell at the bottom, in seconds.


(c) (d)

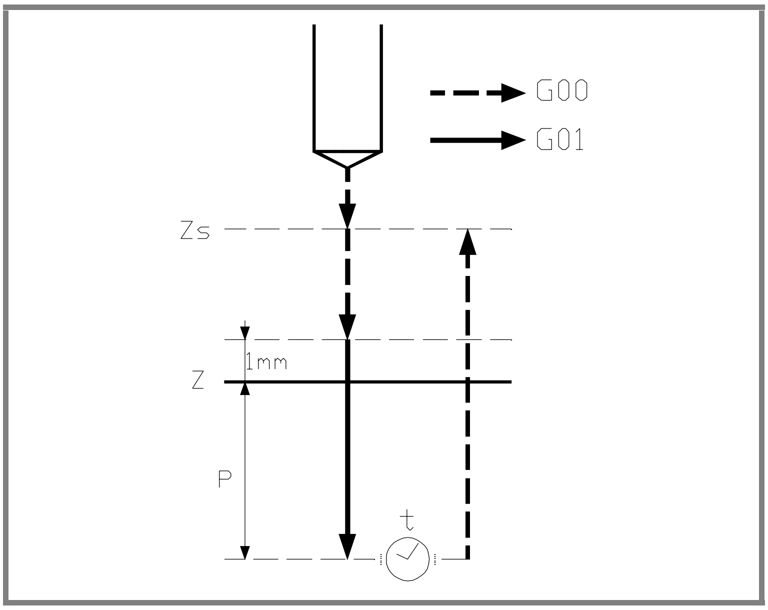
Spindle turning direction (icon).

Clockwise with icon[(c)] and counterclockwise with icon[(d)]

Basic operation:

1. It starts the spindle in the requested direction.
2. Rapid movement (G0), up to the XY point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.

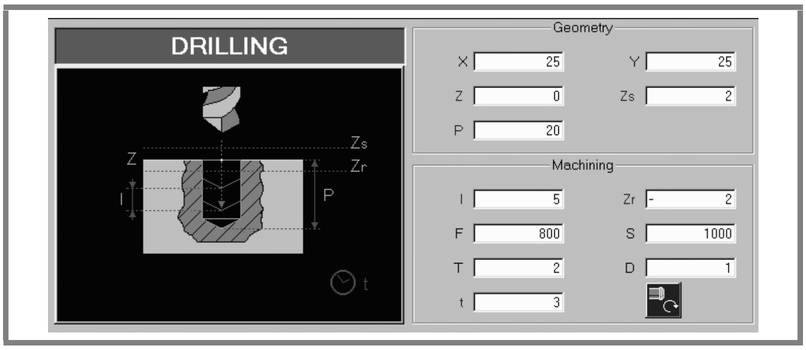


3. Rapid movement (G0) up to the approach plane.
4. Penetration at feedrate "F".
5. Dwell "t".
6. Rapid withdrawal (G0) up to the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

7. Rapid movement (G0) to the next point.
8. Repeats steps 3, 4, 5, 6.

12.3 Drilling 1.



Geometric parameters:

X, Y Machining point.

Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

Machining parameters:

I Penetration step. The drilling takes place with the given step, except the last step that machines the rest.

Zr Relief coordinate it returns to, in rapid (G0), after each drilling step.

If it has not reached the "Zr" coordinate, it returns to the approach plane.

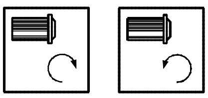
F Feedrate.

S Spindle speed.

T Tool.

D Tool offset.

t Dwell at the bottom, in seconds.



(a) (b)

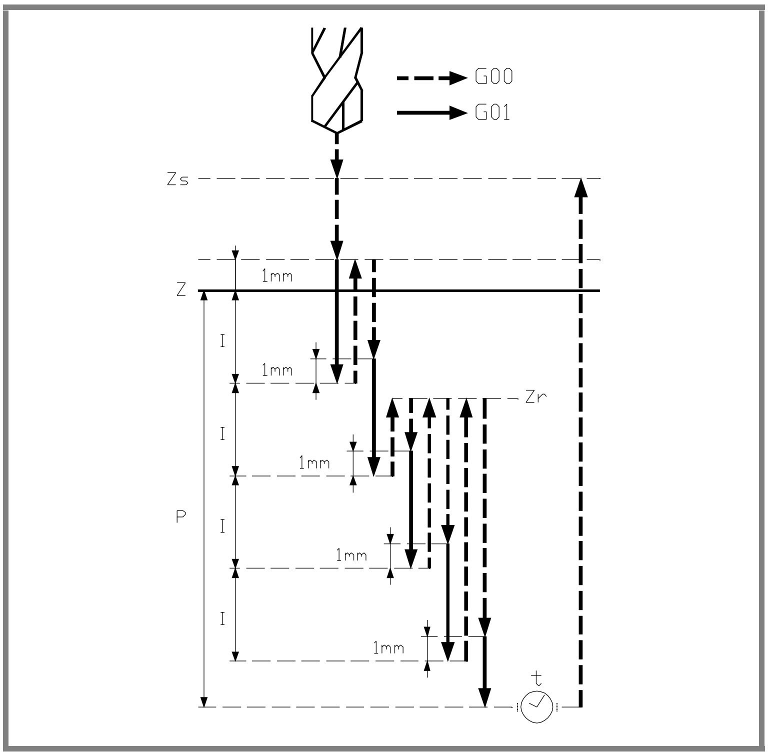
Spindle turning direction (icon).

Clockwise with icon(a) and counterclockwise with icon(b)

Basic operation:

1. It starts the spindle in the requested direction.
2. Rapid movement (G0), up to the XY point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.



3. Rapid movement (G0) up to the approach plane.
4. It penetrates the distance "I" at the feedrate "F".
5. Drilling loop until reaching the total depth "P".

   5.1. Rapid withdrawal (G0) up to the relief coordinate Zr.

   If it has not reached the "Zr" coordinate yet, it returns to the approach plane.

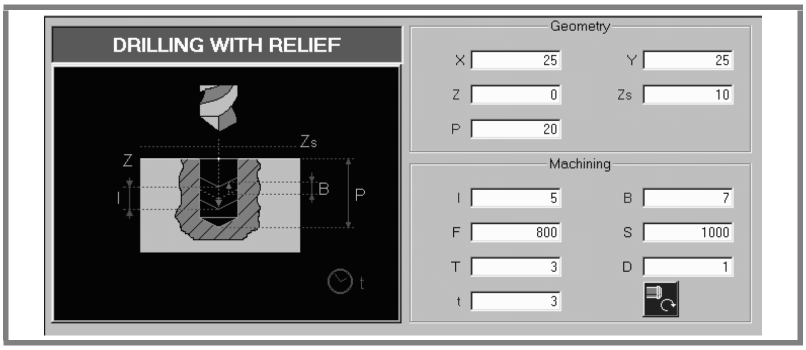
   5.2. Rapid approach (G0) up to 1 mm from the previous drilling step (peck).

   5.3. It penetrates the distance "I" at the feedrate "F".
6. Dwell "t".
7. Rapid withdrawal (G0) up to the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

8. Rapid movement (G0) to the next point.
9. Drills a new hole, steps 3, 4, 5, 6, 7.

12.4 Drilling 2.



Geometric parameters:

X, Y Machining point.

Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

Machining parameters:

I Penetration step. The drilling takes place with the given step, except the last step that machines the rest.

B Relief distance (it withdraws), in rapid (G0), after each drilling step.

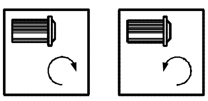
F Feedrate.

S Spindle speed.

T Tool.

D Tool offset.

t Dwell at the bottom, in seconds.



(a) (b)

Spindle turning direction (icon).

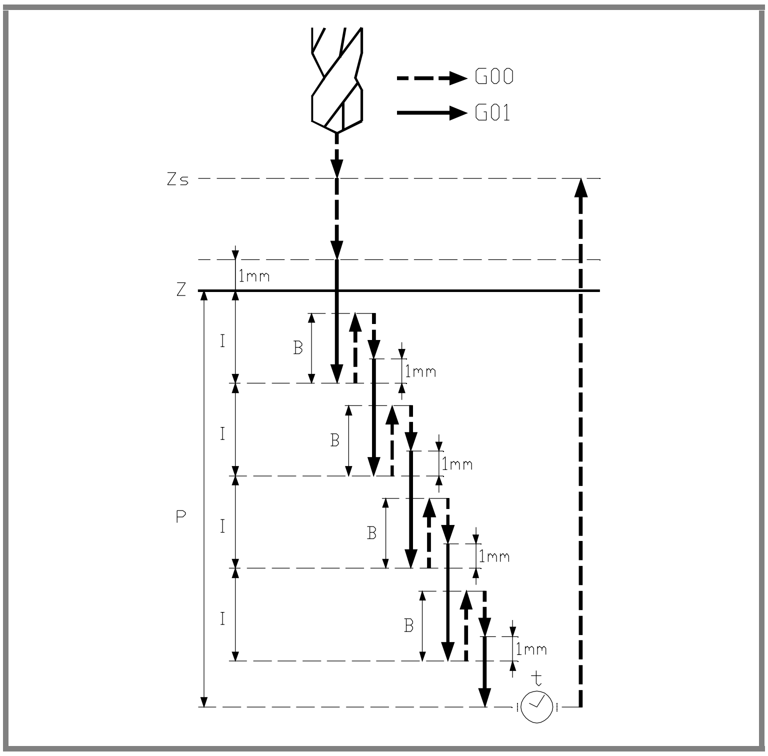
Clockwise with icon(a) and counterclockwise with icon(b)

Basic operation:

1. It starts the spindle in the requested direction.

2. Rapid movement (G0), up to the XY point and the safety plane (Zs).

Depending on the starting plane, it first moves in XY and then in Z or vice versa.



3. Rapid movement (G0) up to the approach plane.

4. It penetrates the distance "I" at the feedrate "F".

5. Drilling loop until reaching the total depth "P".

- **5.1.** It withdraws in rapid (G0) the relief distance "B".
- **5.2.** Rapid approach (G0) up to 1 mm from the previous drilling step (peck).
- **5.3.** It penetrates the distance "I" at the feedrate "F".

6. Dwell "t".

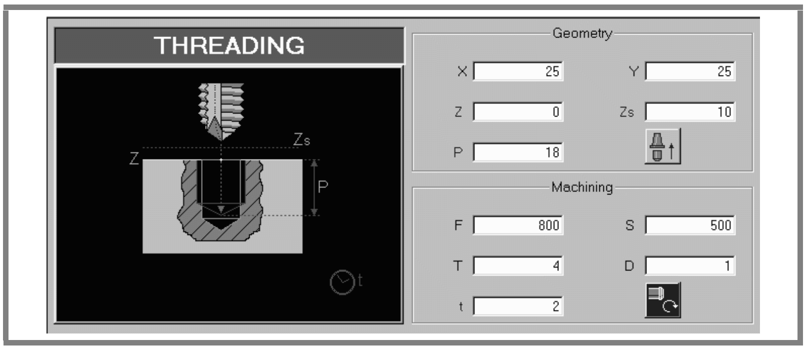
7. Rapid withdrawal (G0) up to the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

8. Rapid movement (G0) to the next point.

9. Repeats steps 3, 4, 5, 6, 7.

12.5 Tapping.



Geometric parameters:

X, Y Machining point.

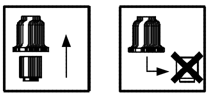
Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

Kf Feedrate factor for the exit.

Rigid tapping allows a rapid exit from the tap maintaining always the synchronism between the feedrate and the speed. The withdrawal feedrate is multiplied by this factor (Kf) and the speed adapts to the new feedrate.


(a) (b)

Type of tapping (icon).

Tapping with a clutch[(a)].
Rigid tapping[(b)].

Machining parameters:

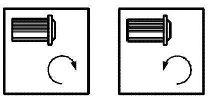
F Feedrate.

S Spindle speed.

T Tool.

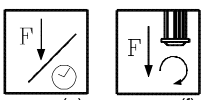
D Tool offset.

t Dwell at the bottom, in seconds.


(c) (d)

Spindle turning direction (icon).

Clockwise with icon[(c)] and counterclockwise with icon[(d)]



(e) (f)

Type of feedrate (icon).

In mm/min or (inch/min)[(e)].
In mm/vuelta[(f)].

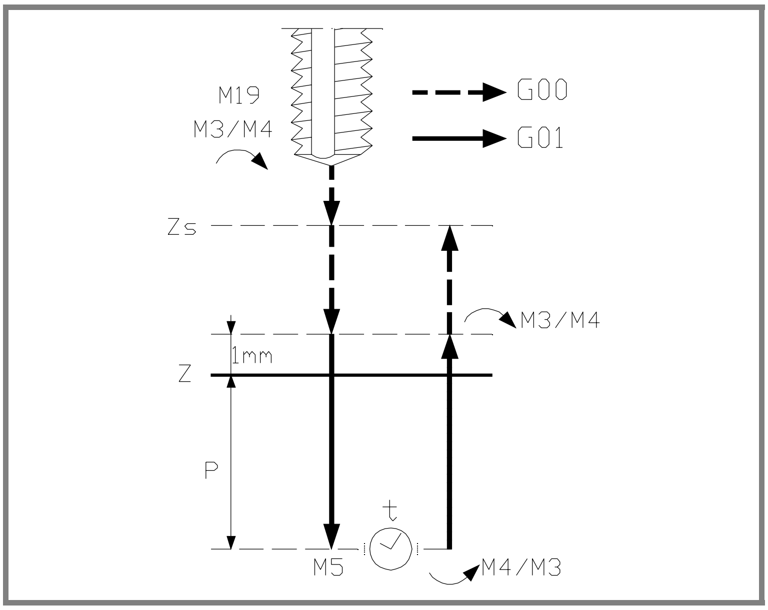
Basic operation:

1. If rigid tapping, it orients the spindle (M19).

If tapping with clutch, it starts the spindle in the requested direction.

2. Rapid movement (G0), up to the XY point and the safety plane (Zs).

Depending on the starting plane, it first moves in XY and then in Z or vice versa.



3. Rapid movement (G0) up to the approach plane.

4. Tapping. It is executed at 100% of the feedrate "F" and spindle speed "S" programmed. Tapping with a clutch cannot be interrupted. In rigid tapping, the feedrate override percentage may be changed and even stopped (0% override).

5. If "t" other than 0, spindle stop (M05) and dwell.

6. If tapping with a clutch, it reverses the spindle turning direction.

7. Withdrawal, exit the tap, to the approach plane.

At 100% of the feedrate "F" and spindle speed "S" programmed. The thread exit cannot be interrupted when tapping with a clutch. In rigid tapping, the feedrate override percentage may be changed and even stopped (0% override).

8. If tapping with a clutch, it reverses the spindle turning direction (restores the initial one).

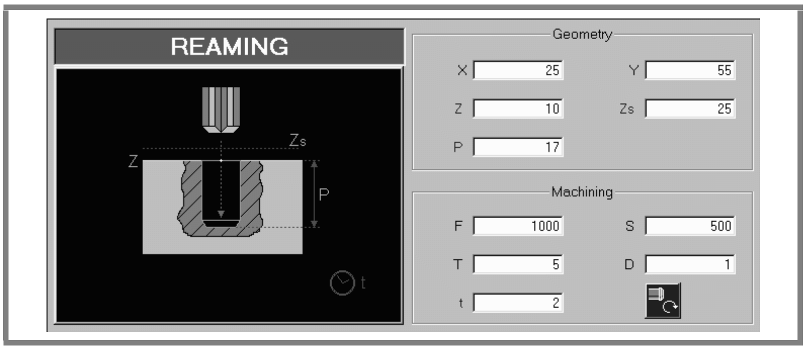
9. Rapid withdrawal (G0) up to the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

10. Rapid movement (G0) to the next point.

11. Repeats steps 3, 4, 5, 6, 7, 8, 9.

12.6 Reaming

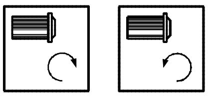


Geometric parameters:

X, Y	Machining point.
Z	Part surface coordinate.
Zs	Safety plane coordinate.
P	Total depth.

Machining parameters:

F	Feedrate.
S	Spindle speed.
T	Tool.
D	Tool offset.
t	Dwell at the bottom, in seconds.


(a) (b)

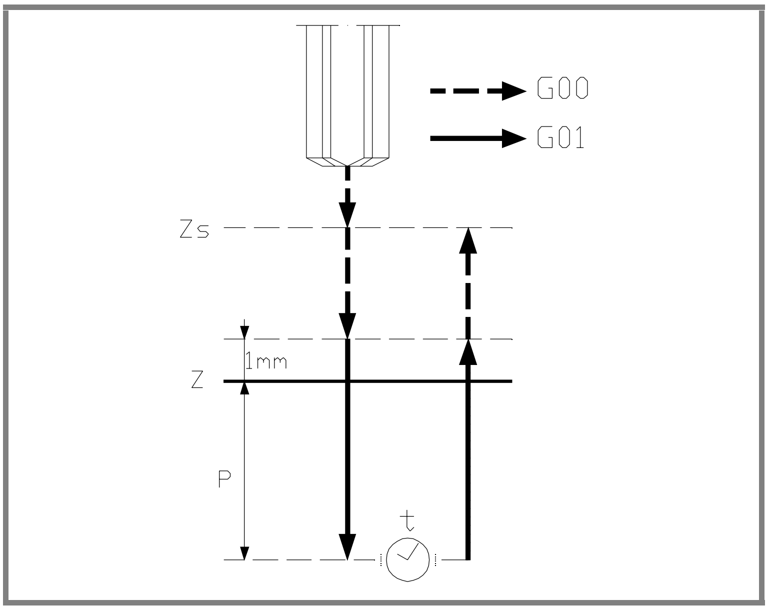
Spindle turning direction (icon).

Clockwise with icon[(a)] and counterclockwise with icon[(b)]

Basic operation:

1. It starts the spindle in the requested direction.
2. Rapid movement (G0), up to the XY point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.

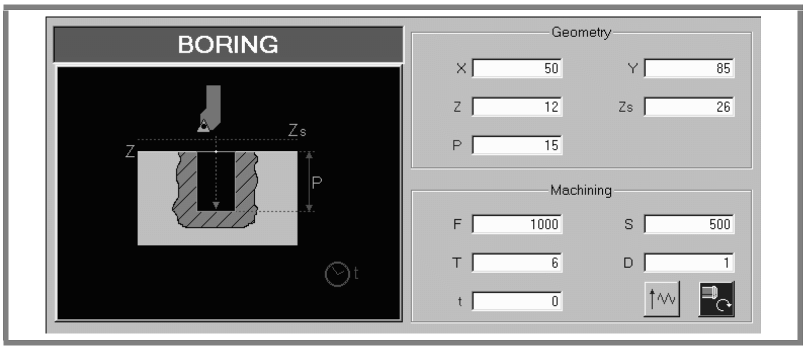


3. Rapid movement (G0) up to the approach plane.
4. Penetration at feedrate "F".
5. Dwell "t".
6. Withdrawal, at feedrate "F", to the approach plane.
7. Rapid movement (G0) up to the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

8. Rapid movement (G0) to the next point.
9. Repeats steps 3, 4, 5, 6, 7.

12.7 Boring 1.

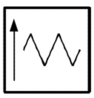


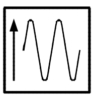
Geometric parameters:

X, Y	Machining point.
Z	Part surface coordinate.
Zs	Safety plane coordinate.
P	Total depth.

Machining parameters:

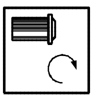
F	Feedrate.
S	Spindle speed.
T	Tool.
D	Tool offset.
t	Dwell at the bottom, in seconds.

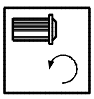

(a)


(b)

Type of withdrawal (icon).

At feedrate "F" and the spindle turning. Icon(a).
In rapid (G0) with the spindle stopped. Icon(b).


(c)


(d)

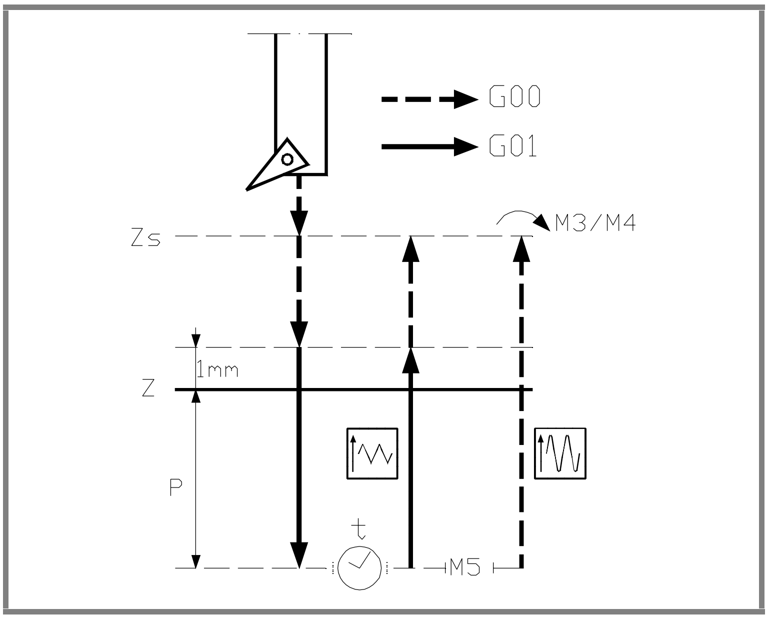
Spindle turning direction (icon).

Clockwise with icon(c) and counterclockwise with icon(d)

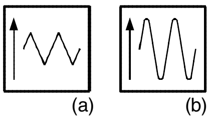
Basic operation:

1. It starts the spindle in the requested direction.
2. Rapid movement (G0), up to the XY point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.



3. Rapid movement (G0) up to the approach plane.
4. Penetration at feedrate "F".
5. Dwell "t".
6. If the icon(b) was defined, it stops the spindle (M05).
7. Withdrawal.

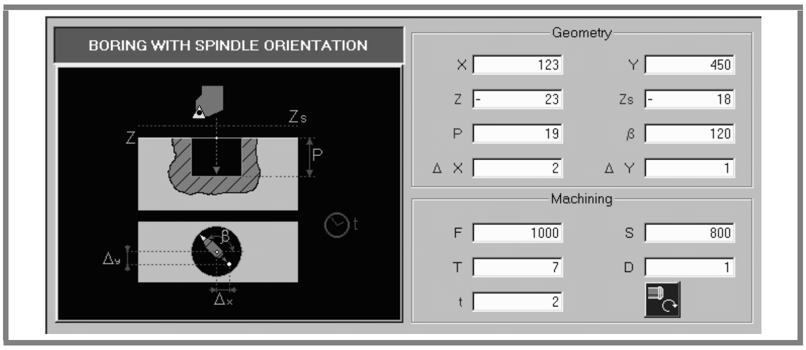


- If the icon(a) was defined, it first withdraws at feedrate "F" to the approach plane (at 1 mm above the surface Z) and then in rapid (G0) to the safety plane Zs.
- If the icon(b) was defined, it withdraws in rapid (G0) to the safety plane Zs and then starts the spindle in the direction it was turning.

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

8. Rapid movement (G0) to the next point.
9. Repeats steps 3, 4, 5, 6, 7.

12.8 Boring 2.



Geometric parameters:

X, Y Machining point.

Z Part surface coordinate.

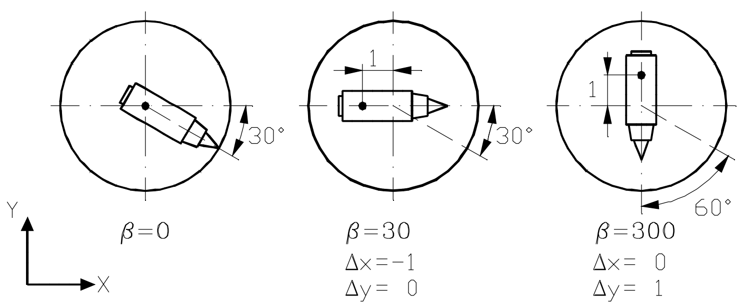
Zs Safety plane coordinate.

P Total depth.

β Spindle position, in degrees, for the withdrawal.

Δx, Δy Distance the tool must move to get the cutter off the wall before withdrawing.

The following example shows how to use parameters β, Δx and Δy. The spindle rest position (I0 position) is at -30º with respect to the X axis.



Machining parameters:

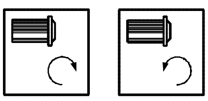
F Feedrate.

S Spindle speed.

T Tool.

D Tool offset.

t Dwell at the bottom, in seconds.


(a) (b)

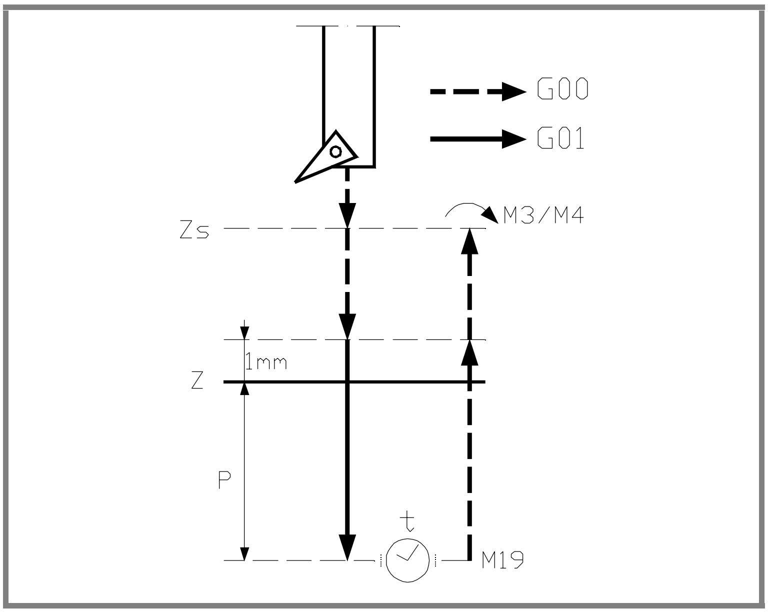
Spindle turning direction (icon).

Clockwise with icon(a) and counterclockwise with icon(b)

Basic operation:

1. It starts the spindle in the requested direction.
2. Rapid movement (G0), up to the XY point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.



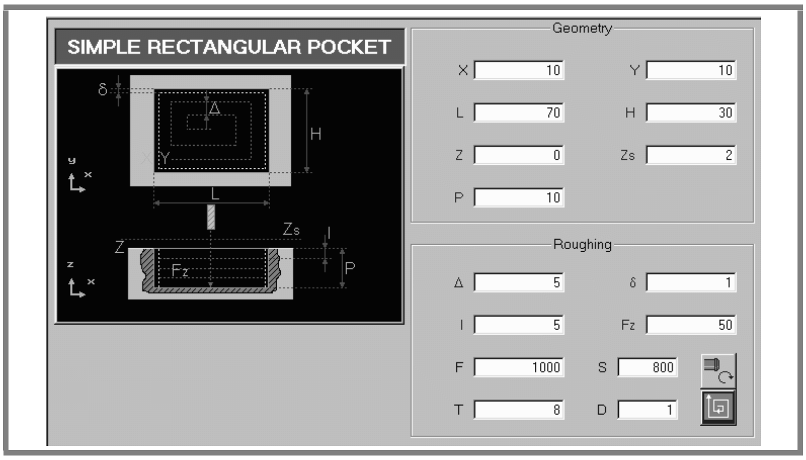
3. Rapid movement (G0) up to the approach plane.
4. Penetration at feedrate "F".
5. Dwell "t".
6. The spindle stops and the tool is oriented in the "β" position (M19).
7. It gets the cutter off the wall. It moves the distance indicated by "Δx, Δy".
8. Rapid withdrawal (G0) up to the approach plane.
9. The tool returns to its position (XY) and starts the spindle in the direction it was turning.
10. Rapid movement (G0) up to the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

11. Rapid movement (G0) to the next point.
12. Repeats steps 3, 4, 5, 6, 7, 8, 9, 10.

12.9 Simple pocket.

The way the roughing and finishing blocks of this cycle are joined will be the one previously set by the user with the instructions #HSC, G5, G50 or G7. We recommend to use #HSC or G5 controlling the shape of the corner with the instruction #ROUNDPAR.

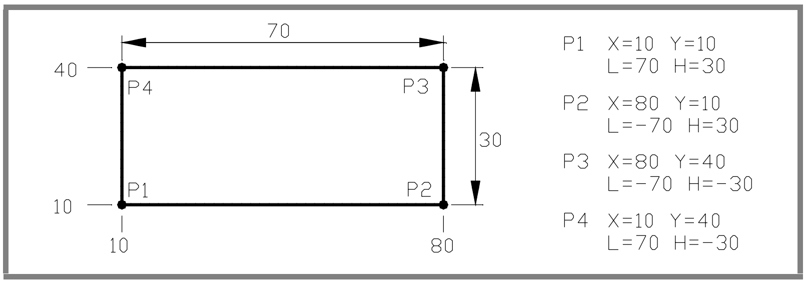


Geometric parameters:

X, Y Pocket corner.

L, H Pocket dimensions.

The sign indicates the orientation referred to the XY point.



Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

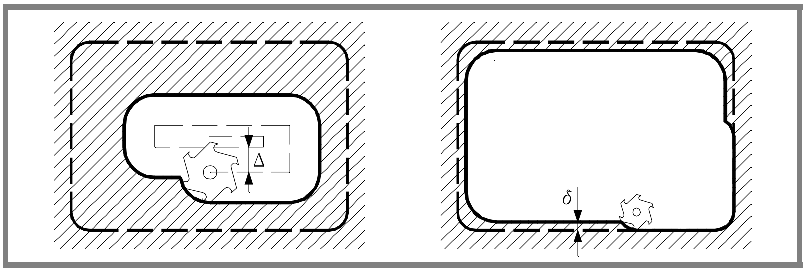
Machining parameters:

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

δ Finishing stock on the side walls.

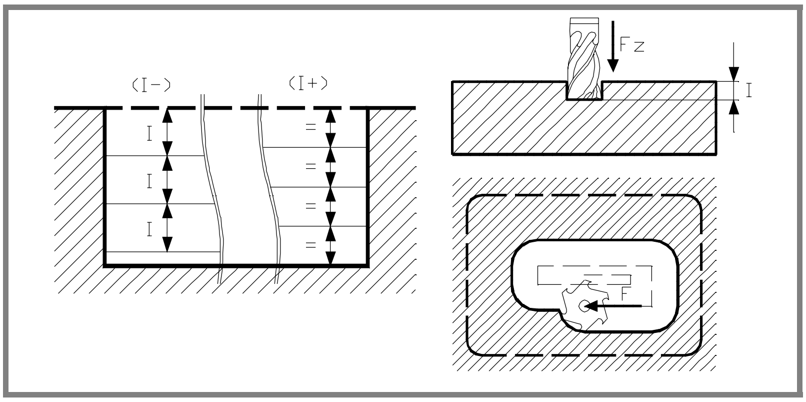


I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the pocket is machined with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

Fz Penetration feedrate.

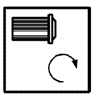


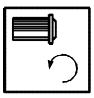
F Surface milling feedrate.

S Spindle speed.

T Tool.

D Tool offset.

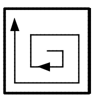

(a)

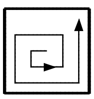

(b)

Spindle turning direction (icon).

Clockwise with icon(a).

Counterclockwise with icon(b).


(c)


(d)

Machining direction (icon).

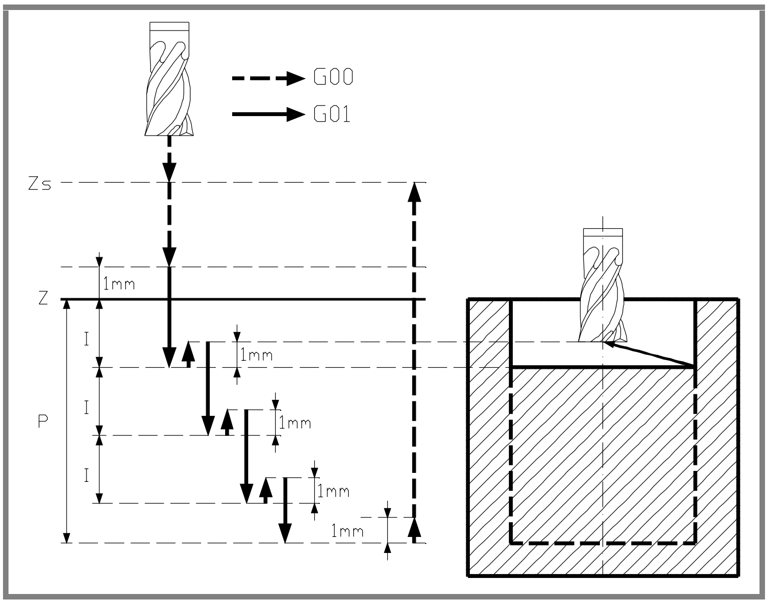
Clockwise with icon(c).

Counterclockwise with icon(d).

Basic operation:

1. It starts the spindle in the requested direction.
2. Rapid movement (G0) to the center of the pocket and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.



3. Rapid movement (G0) up to the approach plane.
4. First penetration, the "Fz" feedrate, the amount "I".
5. Milling of the pocket surface.

   Roughing is carried out at feedrate "F" with the passes defined by "Δ" and up to a distance "δ" from the pocket wall.

   The finishing pass "δ" is carried out with tangential entry and exit and at feedrate "F".
6. Rapid withdrawal (G0) to the center of the pocket in the approach plane.
7. New milling surfaces until reaching the total depth of the pocket.

   7.1. Penetration, at the feedrate indicated in "Fz" up to a distance "I" from the previous surface.

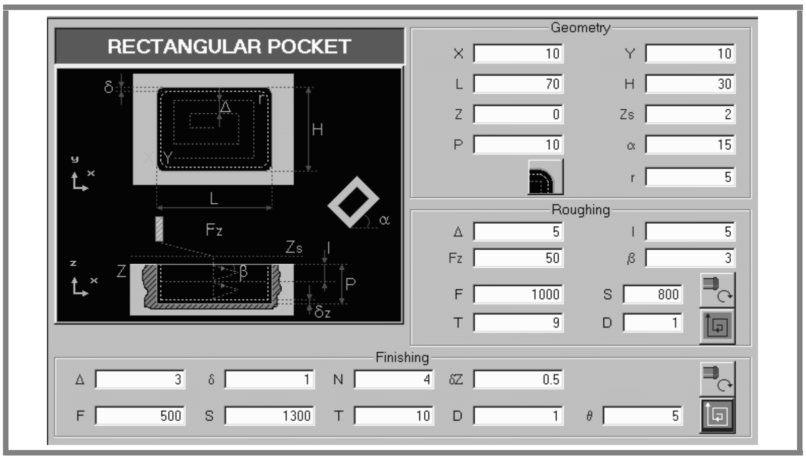
   7.2. Milling of the new surface following the steps indicated in points 5 and 6.
8. Rapid withdrawal (G0) up to the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

9. Rapid movement (G0) to the next point.
10. Repeats steps 3, 4, 5, 6, 7, 8.

12.10 Rectangular pocket

The way the roughing and finishing blocks of this cycle are joined will be the one previously set by the user with the instructions #HSC, G5, G50 or G7. We recommend to use #HSC or G5 controlling the shape of the corner with the instruction #ROUNDPAR.

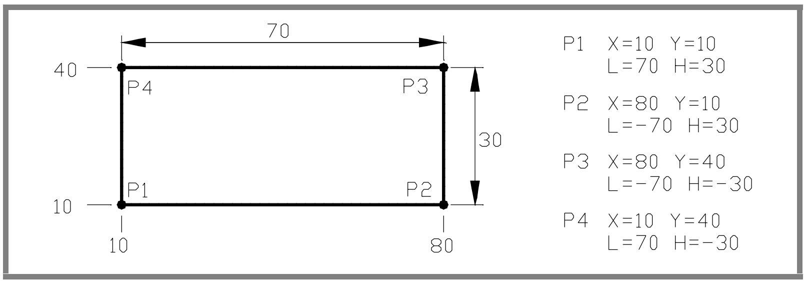


Geometric parameters:

X, Y Pocket corner.

L, H Pocket dimensions.

The sign indicates the orientation referred to the XY point.

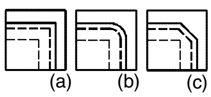


Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

το Angle, in degrees, between the pocket and the abscissa axis. The turn is carried out on the defined corner, X,Y point.



Type of corner (icon).

Square corner with icon[(a)].
Rounded corner with icon[(b)].
Chamfered corner with icon[(c)].

r Rounding radius or chamfer size.

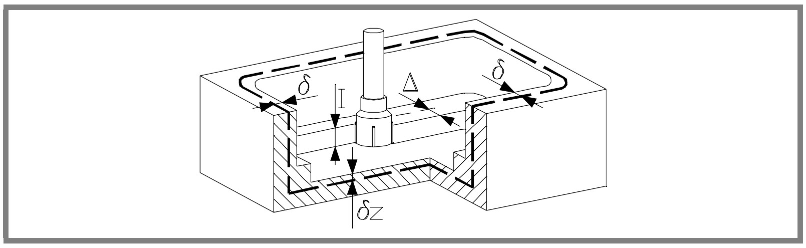
Roughing parameters:

The roughing operation empties the pocket leaving the following finishing stocks:

δ Finishing stock on the side walls.

δz Finishing stock at the bottom of the pocket.

Both stocks are defined as finishing parameters.



The roughing operation defining parameters are:

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

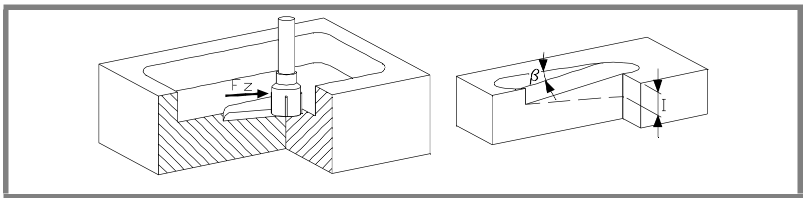
If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the pocket is machined with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

Fz Penetration feedrate.



β Penetrating angle.

The penetration is carried out in zigzag, starting and ending at the center of the pocket.

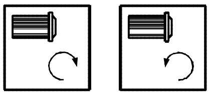
If defined with a value greater than the one assigned to the tool in the tool table, it assumes the table value.

F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

If programmed T=0, there is no roughing.

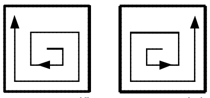


(d) (e)

Spindle turning direction (icon).

Clockwise with icon(d).

Counterclockwise with icon(e).



(f) (g)

Machining direction (icon).

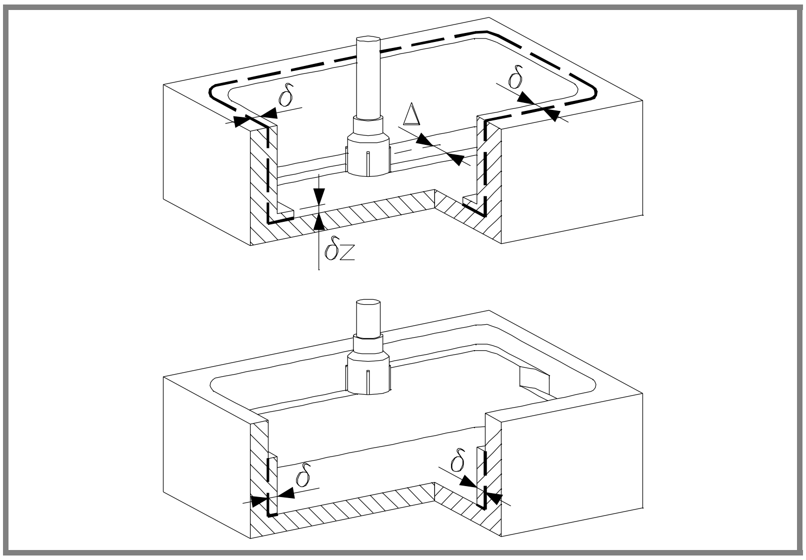
Clockwise with icon(f).

Counterclockwise with icon(g).

Finishing parameters:

The finishing operation is carried out in two stages.

First, it machines the bottom of the pocket and then the side walls, with tangential entry and exit.



The finishing operation defining parameters are:

δ Finishing stock on the side walls.

δz Finishing stock at the bottom of the pocket.

Δ Milling pass or width at the bottom of the pocket.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

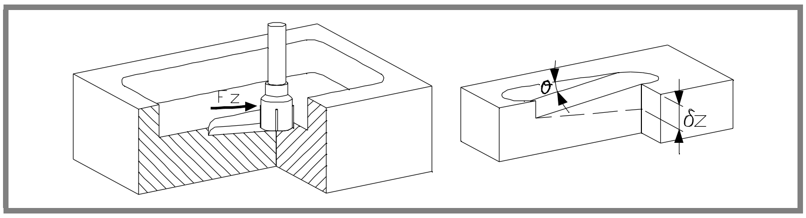
If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

N Number of penetration passes (steps) for the side finishing. If the resulting step is greater than the cutting length assigned to the table in the tool table, the step will be limited to that value.

θ Penetrating angle.

The penetration is carried out at the feedrate set by roughing parameter "Fz" starting and ending at the center of the pocket.

If defined with a value greater than the one assigned to the tool in the tool table, it assumes the table value.



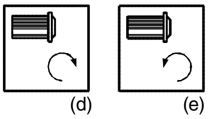
F Surface and side milling feedrate.

S Spindle speed.

T Finishing tool.

If programmed T=0, there is no finishing.

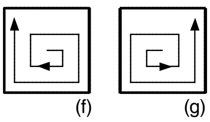
D Tool offset.



Spindle turning direction (icon).

Clockwise with icon(d).

Counterclockwise with icon(e).



Machining direction (icon).

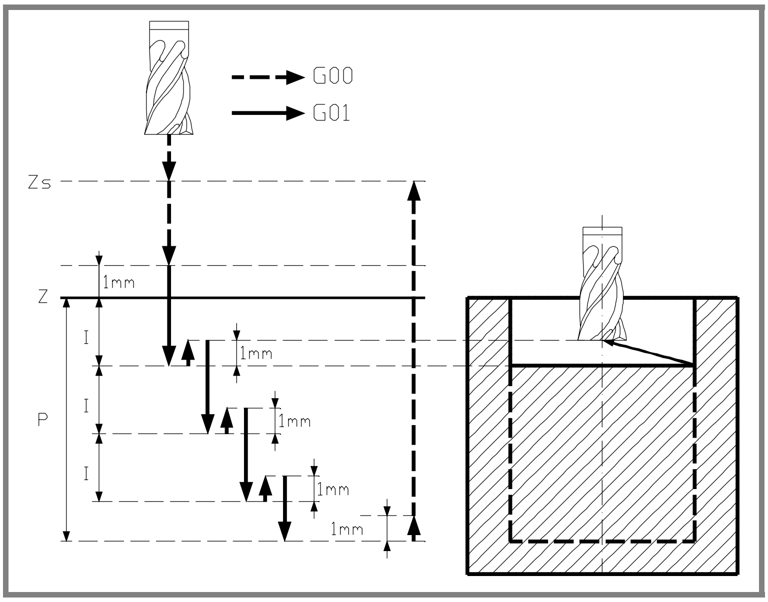
Clockwise with icon(f).

Counterclockwise with icon(g).

Basic operation:

1. It selects the roughing tool and starts the spindle in the requested direction.
2. Rapid movement (G0) up to the safety plane (Zs) positioning at the center of the pocket.

   Depending on the tool position, it first moves in XY and then in Z or vice versa.



3. Rapid movement (G0) up to the approach plane.

4. Roughing operation.

It is carried out in layers, until reaching the total depth minus the finishing stock at the bottom "δz".

4.1. Penetration "I" at feedrate "Fz" at an angle "β".

4.2. Milling of the pocket surface up to a distance "δ" from the pocket wall. It is carried out at feedrate "F" and, if necessary, it recalculates the pass (Δ) so all the passes are identical.

4.3. Rapid withdrawal (G0) to the center of the pocket, 1 mm off the machined surface.

5. Rapid withdrawal (G0) up to the safety plane (Zs).

6. It selects the finishing tool and it approaches in rapid (G0) down to 1 mm from the roughed out bottom.

7. Finishing of the bottom of the pocket.

7.1. Penetration at feedrate "Fz" at an angle "θ".

7.2. Milling of the bottom of the pocket up to a distance "δ" from the pocket wall. It is carried out at finishing feedrate "F" and, if necessary, it recalculates the finishing pass (Δ) so all the passes are identical.

8. Withdrawal, in rapid (G0), to the center of the pocket in the approach plane (1 mm off the "Z" surface).

9. Finishing of the side walls.

It is carried out in "N" passes at the finishing feedrate "F" and with tangential entry and exit.

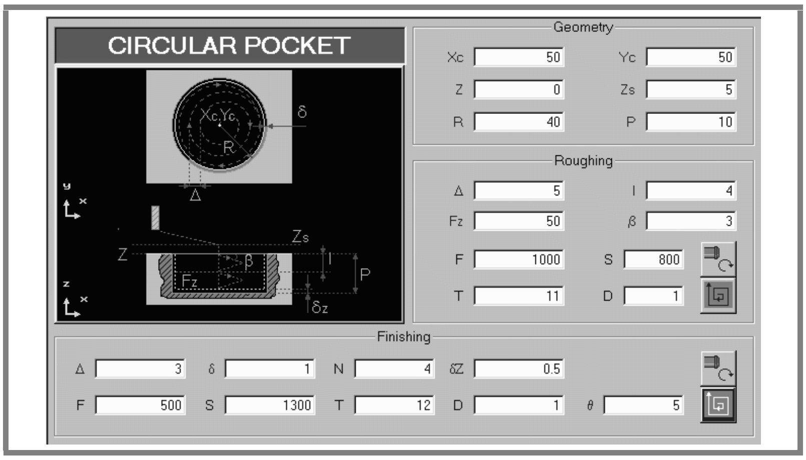
10. Rapid withdrawal (G0) to the center of the pocket in the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

11. Rapid movement (G0) to the next point.

12. Repeats steps 3, 4, 5, 6, 7, 8, 9, 10.

12.11 Circular pocket



Geometric parameters:

Xc, Yc Center of the pocket.

R Pocket radius.

Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

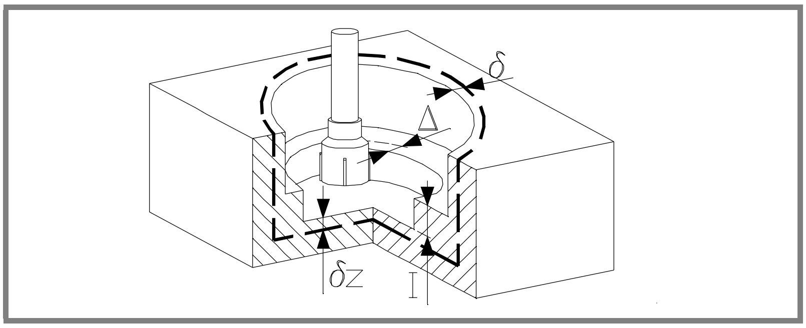
Roughing parameters:

The roughing operation empties the pocket leaving the following finishing stocks:

δ Finishing stock on the side walls.

δz Finishing stock at the bottom of the pocket.

Both stocks are defined as finishing parameters.



The roughing operation defining parameters are:

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

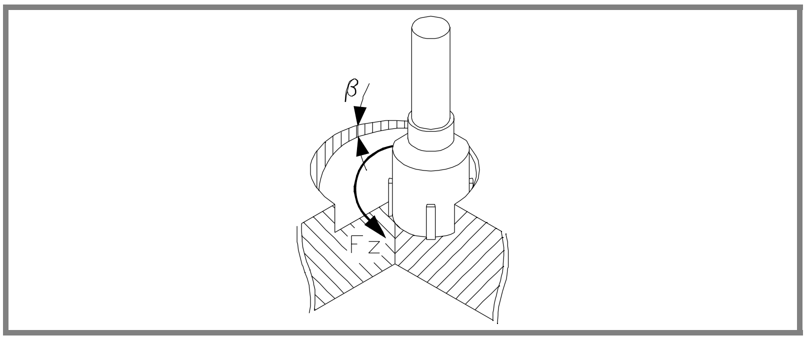
If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the pocket is machined with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

Fz Penetration feedrate.



β Penetrating angle.

The penetration is carried out along a helical path, starting and ending at the center of the pocket.

If defined with a value greater than the one assigned to the tool in the tool table, it assumes the table value.

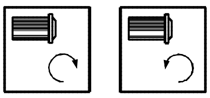
F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

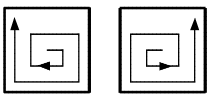
If programmed T=0, there is no roughing.

D Tool offset.


(a) (b)

Spindle turning direction (icon).

Clockwise with icon(a).
Counterclockwise with icon(b).


(c) (d)

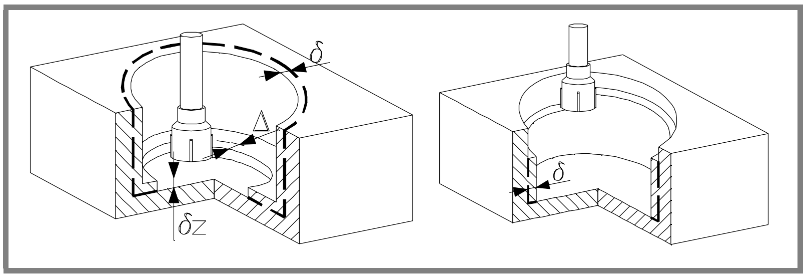
Machining direction (icon).

Clockwise with icon(c).
Counterclockwise with icon(d).

Finishing parameters:

The finishing operation is carried out in two stages.

First, it machines the bottom of the pocket and then the side walls, with tangential entry and exit.



The finishing operation defining parameters are:

δ Finishing stock on the side walls.

δz Finishing stock at the bottom of the pocket.

Δ Milling pass or width at the bottom of the pocket.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

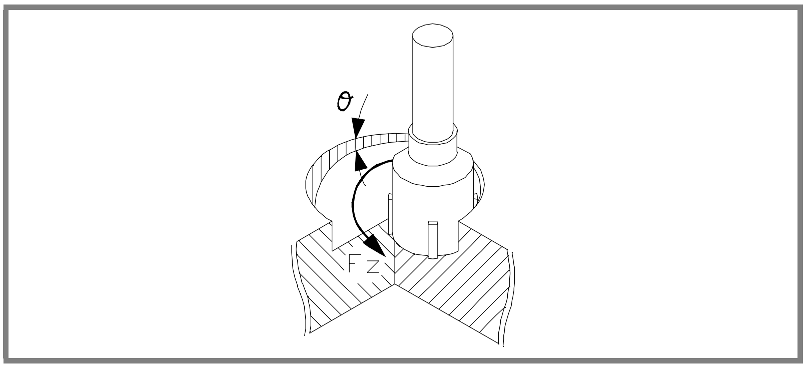
If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

N Number of penetration passes (steps) for the side finishing. If the resulting step is greater than the cutting length assigned to the table in the tool table, the step will be limited to that value.

θ Penetrating angle.

The penetration is carried out along a helical path at the feedrate set by roughing parameter "Fz" starting and ending at the center of the pocket.

If defined with a value greater than the one assigned to the tool in the tool table, it assumes the table value.



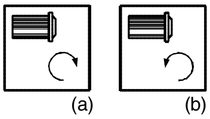
F Surface and side milling feedrate.

S Spindle speed.

T Finishing tool.

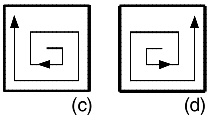
If programmed T=0, there is no finishing.

D Tool offset.



Spindle turning direction (icon).

Clockwise with icon(a).
Counterclockwise with icon(b).



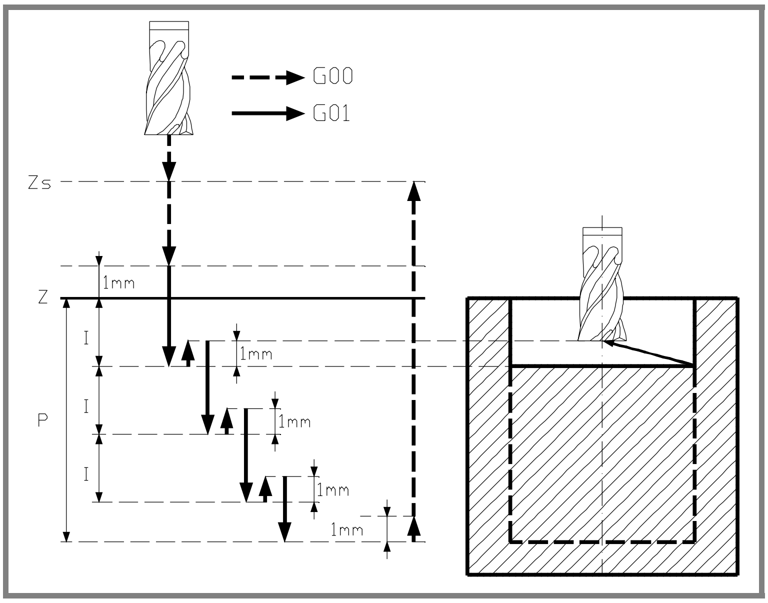
Machining direction (icon).

Clockwise with icon(c).
Counterclockwise with icon(d).

Basic operation:

1. It selects the roughing tool and starts the spindle in the requested direction.
2. Rapid movement (G0) to the center of the pocket and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.
3. Rapid movement (G0) up to the approach plane.



4. Roughing operation.

   It is carried out in layers, until reaching the total depth minus the finishing stock at the bottom "δz".

   4.1. Penetration "I" at feedrate "Fz" at an angle "β".

   4.2. Milling of the pocket surface up to a distance "δ" from the pocket wall. It is carried out at feedrate "F" and, if necessary, it recalculates the pass (Δ) so all the passes are identical.

   4.3. Rapid withdrawal (G0) to the center of the pocket, 1 mm off the machined surface.

5. Rapid withdrawal (G0) up to the safety plane (Zs).

6. It selects the finishing tool and it approaches in rapid (G0) down to 1 mm from the roughed out bottom.

7. Finishing of the bottom of the pocket.

   7.1. Penetration at feedrate "Fz" at an angle "θ".

   7.2. Milling of the bottom of the pocket up to a distance "δ" from the pocket wall. It is carried out at finishing feedrate "F" and, if necessary, it recalculates the finishing pass (Δ) so all the passes are identical.

8. Rapid withdrawal (G0) to the center of the pocket in the approach plane.

9. Finishing of the side walls.

   It is carried out in "N" passes at the finishing feedrate "F" and with tangential entry and exit.

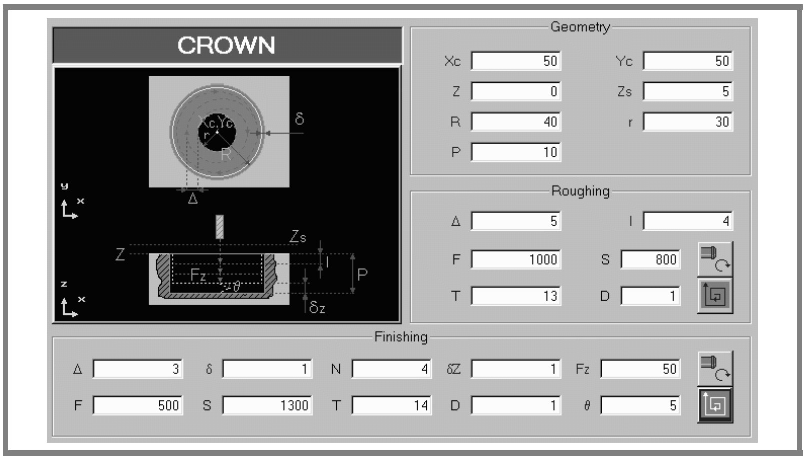
10. Rapid withdrawal (G0) to the center of the pocket in the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

11. Rapid movement (G0) to the next point.

12. Repeats steps 3, 4, 5, 6, 7, 8, 9, 10.

12.12 Pre-emptied pocket



Geometric parameters:

Xc, Yc Center of the pocket.

R Pocket radius.

r Pre-emptying radius.

Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

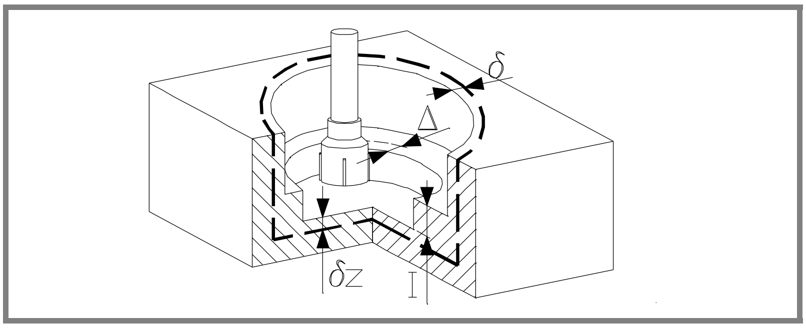
Roughing parameters:

The roughing operation empties the pocket leaving the following finishing stocks:

δ Finishing stock on the side walls.

δz Finishing stock at the bottom of the pocket.

Both stocks are defined as finishing parameters.



The roughing operation defining parameters are:

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the pocket is machined with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

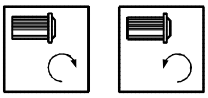
F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

If programmed T=0, there is no roughing.

D Tool offset.

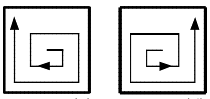


(a) (b)

Spindle turning direction (icon).

Clockwise with icon[(a)].

Counterclockwise with icon[(b)].



(c) (d)

Machining direction (icon).

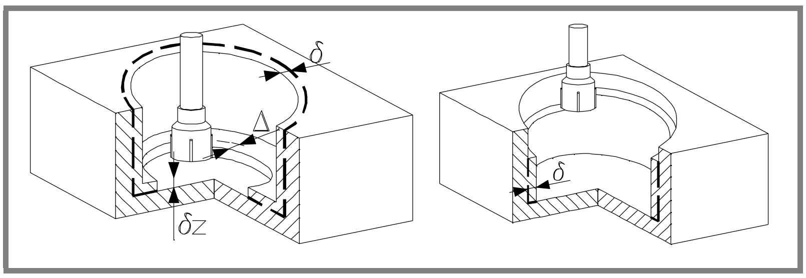
Clockwise with icon[(c)].

Counterclockwise with icon[(d)].

Finishing parameters:

The finishing operation is carried out in two stages.

First, it machines the bottom of the pocket and then the side walls, with tangential entry and exit.



The finishing operation defining parameters are:

δ Finishing stock on the side walls.

δz Finishing stock at the bottom of the pocket.

Δ Milling pass or width at the bottom of the pocket.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

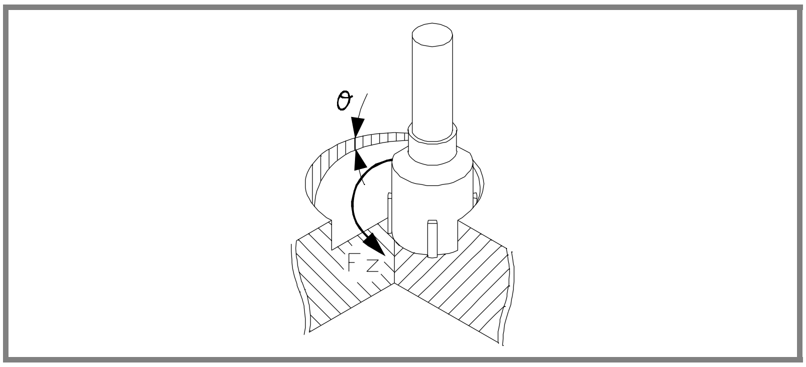
N Number of penetration passes (steps) for the side finishing. If the resulting step is greater than the cutting length assigned to the table in the tool table, the step will be limited to that value.

Fz Penetration feedrate.

θ Penetrating angle.

The penetration is carried out along a helical path at the feedrate set by finishing parameter "Fz" starting and ending at the center of the pocket.

If defined with a value greater than the one assigned to the tool in the tool table, it assumes the table value.



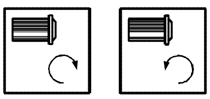
F Surface and side milling feedrate.

S Spindle speed.

T Finishing tool.

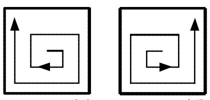
If programmed T=0, there is no finishing.

D Tool offset.


(a) (b)

Spindle turning direction (icon).

Clockwise with icon(a).
Counterclockwise with icon(b).


(c) (d)

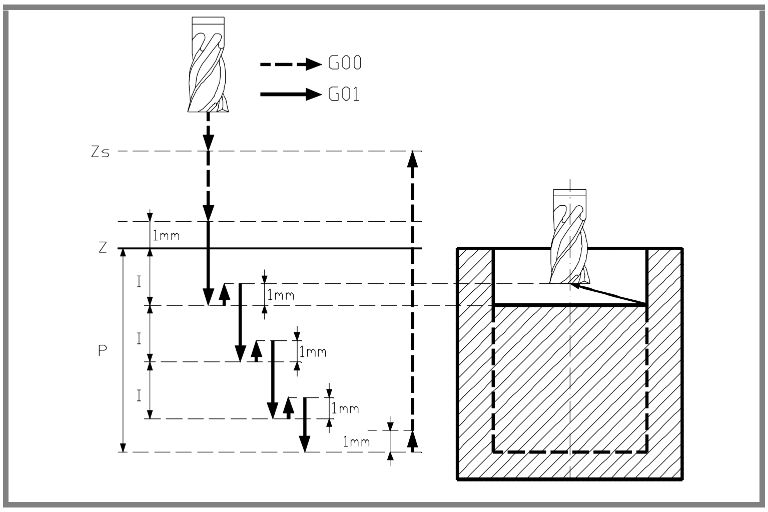
Machining direction (icon).

Clockwise with icon(c).
Counterclockwise with icon(d).

Basic operation:

1. It selects the roughing tool and starts the spindle in the requested direction.
2. Rapid movement (G0) to the center of the pocket and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.

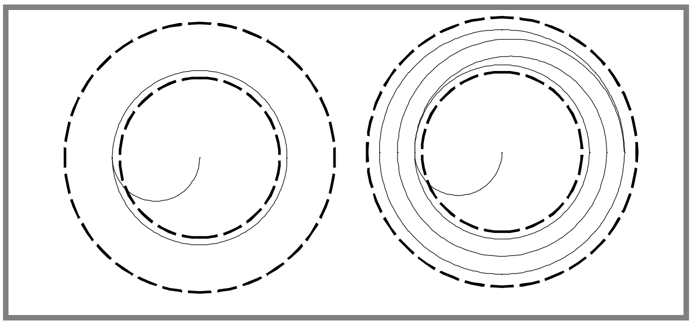


3. Rapid movement (G0) up to the approach plane.
4. Roughing operation.

   It is carried out in layers, until reaching the total depth minus the finishing stock at the bottom "δz".

   4.1. Penetration "I".

   4.2. Approach to the pre-emptied side with tangential entry.



   4.3. Milling of the pocket surface up to a distance "δ" from the pocket wall. It is carried out at feedrate "F" and, if necessary, it recalculates the pass (Δ) so all the passes are identical.

   4.4. Rapid withdrawal (G0) to the center of the pocket, 1 mm off the machined surface.

5. Rapid withdrawal (G0) up to the safety plane (Zs).
6. It selects the finishing tool and it approaches in rapid (G0) down to 1 mm from the roughed out bottom.

7. Finishing of the bottom of the pocket.

7.1. Penetration at feedrate "Fz" at an angle "θ".

7.2. Milling of the bottom of the pocket up to a distance "δ" from the pocket wall. It is carried out at finishing feedrate "F" and, if necessary, it recalculates the finishing pass (Δ) so all the passes are identical.

8. Withdrawal, in rapid (G0), to the center of the pocket in the approach plane (1 mm off the "Z" surface).

9. Finishing of the side walls.

It is carried out in "N" passes at the finishing feedrate "F" and with tangential entry and exit.

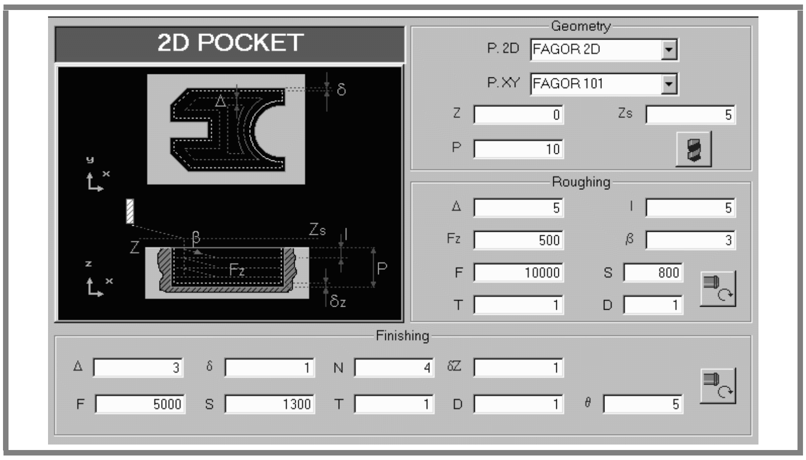
10. Rapid withdrawal (G0) to the center of the pocket in the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

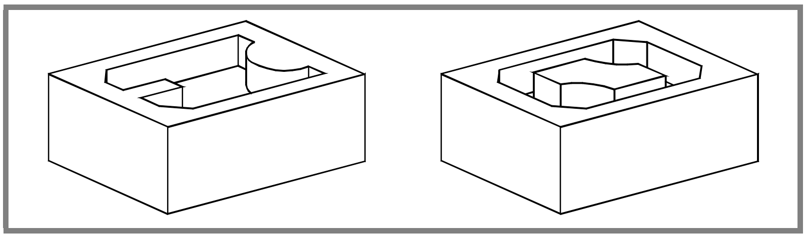
11. Rapid movement (G0) to the next point.

12. Repeats steps 3, 4, 5, 6, 7, 8, 9, 10.

12.13 2D pocket



A pocket consists of an outside contour and a number of inside contours called islands. All the walls of 2D pockets are vertical.



It is recommended to previously define the #ROUNDPAR instruction in order to obtain a good finish because the finishing passes are carried out in G05.

Geometric parameters:

The composition of the pocket and the profile in the plane is stored in \ Cnc8070\ Users\ Profile.

pocket.P2D	Pocket composition.
profile.PXY	Plane profile.

P.2D Name of the 2D pocket.

Once the pocket configuration has been validated, the CNC associates the geometry of the pocket to its name.

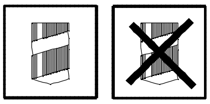
P.XY Name of the plane profile.

The profile must indicate the pocket's outside contour and those of the islands.

Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.



(a) (b)

Drilling (icon).

It indicates whether drilling[(a)] takes place before machining the pocket or not[(b)]. It should be used when the roughing tool cannot machine downwards.

Press the "Drilling" softkey to access the drilling cycle and after defining it, press the "End" softkey to return to the 2D pocket cycle.

The diameter of the drilling tool must not exceed the radius of the roughing tool: or that of the roughing at the bottom if there is no roughing operation.

The cycle calculates the drilling point depending on the programmed profile and the roughing tool.

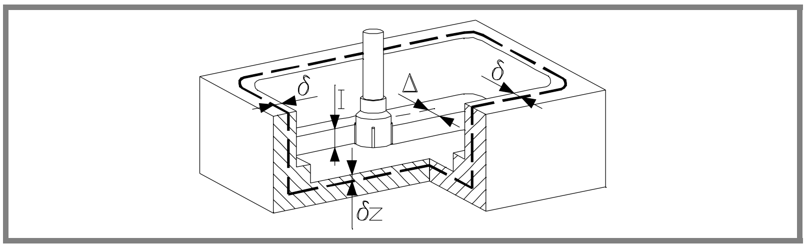
Roughing parameters:

The roughing operation empties the pocket leaving the following finishing stocks:

δ Finishing stock on the side walls.

δz Finishing stock at the bottom of the pocket.

Both stocks are defined as finishing parameters.



The roughing operation defining parameters are:

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the pocket is machined with the given pass (step) except the last pass that machines the rest.

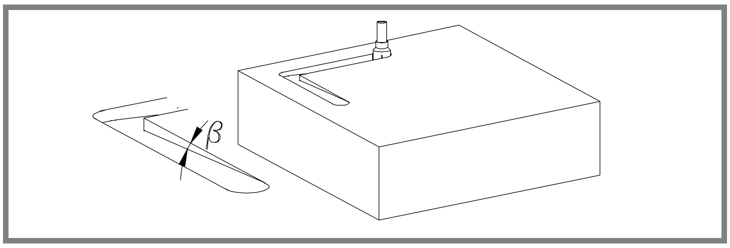
In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

Fz Penetration feedrate.

β Penetrating angle.

The penetration is carried out maintaining this angle until the corresponding depth is reached.

If defined with a value greater than the one assigned to the tool in the tool table, it assumes the table value.

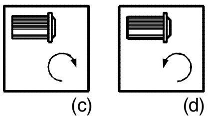


F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

If programmed T=0, there is no roughing.



Spindle turning direction (icon).

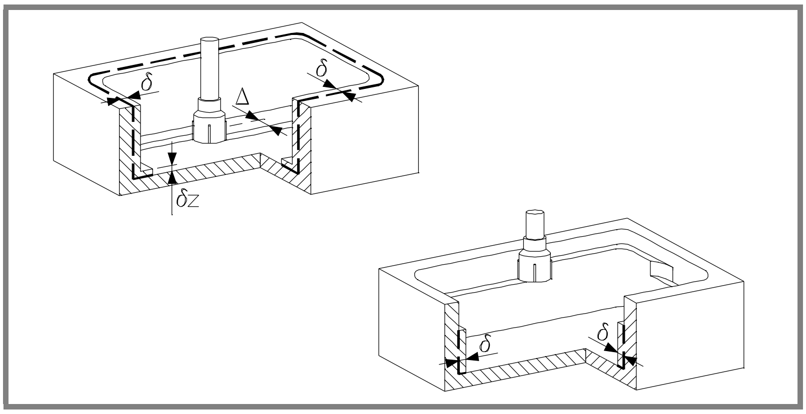
Clockwise with icon (c).

Counterclockwise with icon (d).

Finishing parameters:

The finishing operation is carried out in two stages.

First, it machines the bottom of the pocket and then the side walls, with tangential entry and exit.



The finishing operation defining parameters are:

δ Finishing stock on the side walls.

δz Finishing stock at the bottom of the pocket.

Δ Milling pass or width at the bottom of the pocket.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

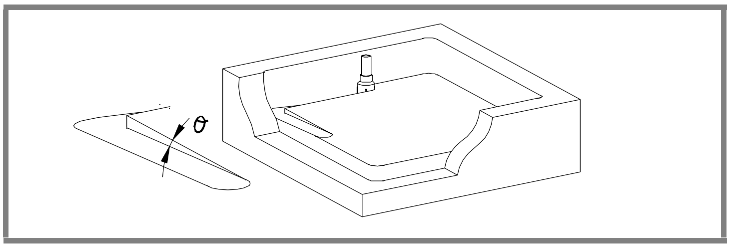
N Number of penetration passes (steps) for the side finishing.

When programming a 0 value, it carries out the least amount of passes possible, considering the cutting length assigned to the tool in the tool table.

θ Penetrating angle.

The penetration is carried out at the feedrate set by roughing parameter "Fz" maintaining this angle until reaching the corresponding depth.

If defined with a value greater than the one assigned to the tool in the tool table, it assumes the table value.



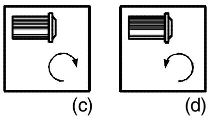
F Surface and side milling feedrate.

S Spindle speed.

T Finishing tool.

If programmed T=0, there is no finishing.

D Tool offset.



Spindle turning direction (icon).

Clockwise with icon(c).

Counterclockwise with icon(d).

Executable pocket file.

To simulate or execute this type of pockets, the CNC uses an executable file with geometry information. This file is generated the first time the pocket is simulated or executed. If from the editor, any data of the pocket geometry or the used tool, is modified, the CNC will generate this file again.

In versions prior to V2.00, the user generated the executable file from the editor before inserting the cycle. From version V2.00 on, it is no longer necessary, the CNC is in charge of generating the executable file when necessary.

The executable files are stored in the directory CNC8070 \Users \Pocket with the name of the pocket (parameter P.2D) and the extension *C2D*. These files must not be deleted, moved to another location or tampered with in any way. If when executing or simulating the pocket, the CNC cannot find these files, it will generate them.

Overall, a 2D pocket consists of the following files.

pocket.P2D	Pocket composition.
profile.PXY	Plane profile.
pocket.C2D	Executable file.

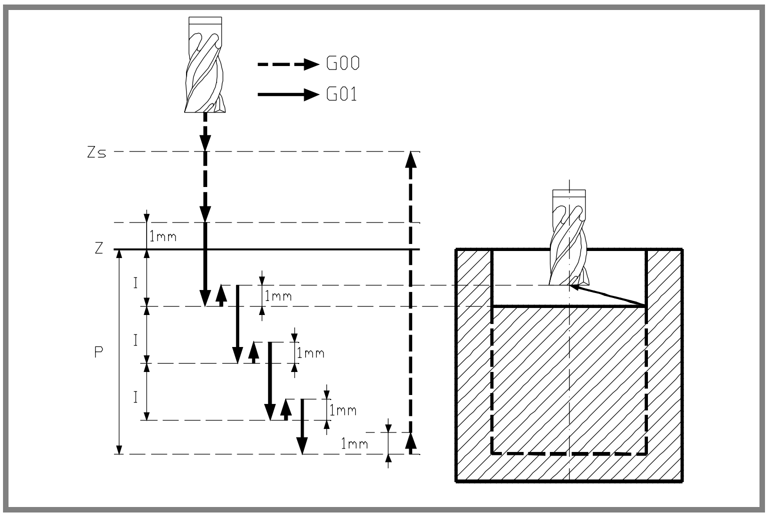
The executable file is also updated after a software update and when executing or simulating a pocket,

Basic operation:

The CNC calculates the initial coordinate depending on the geometry of the pocket and the tool radius.

1. Drilling operation. Only if it has beenprogrammed.
2. It selects the roughing tool and starts the spindle in the requested direction.
3. Rapid movement (G0) to the roughing starting point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.



4. Rapid movement (G0) up to the approach plane.
5. Roughing operation.

   It is carried out in layers, until reaching the total depth minus the finishing stock at the bottom "δz".

   5.1. Penetration "I" at feedrate "Fz" at an angle "β".
   5.2. Milling of the pocket surface up to a distance "δ" from the pocket wall. It is carried out at feedrate "F" and, if necessary, it recalculates the pass (Δ) so all the passes are identical.

        It is carried out following paths concentric to the profile, in the same direction as the outside profile was defined. The islands are machined in the opposite direction.
   5.3. Rapid withdrawal (G0) up to 1 mm off the machined surface.
6. Rapid withdrawal (G0) up to the safety plane (Zs).
7. It selects the finishing tool and it approaches in rapid (G0) down to 1 mm from the roughed out bottom.
8. Finishing of the bottom of the pocket.
   8.1. Penetration at feedrate "Fz" at an angle "θ".
   8.2. Milling of the bottom of the pocket up to a distance "δ" from the pocket wall.

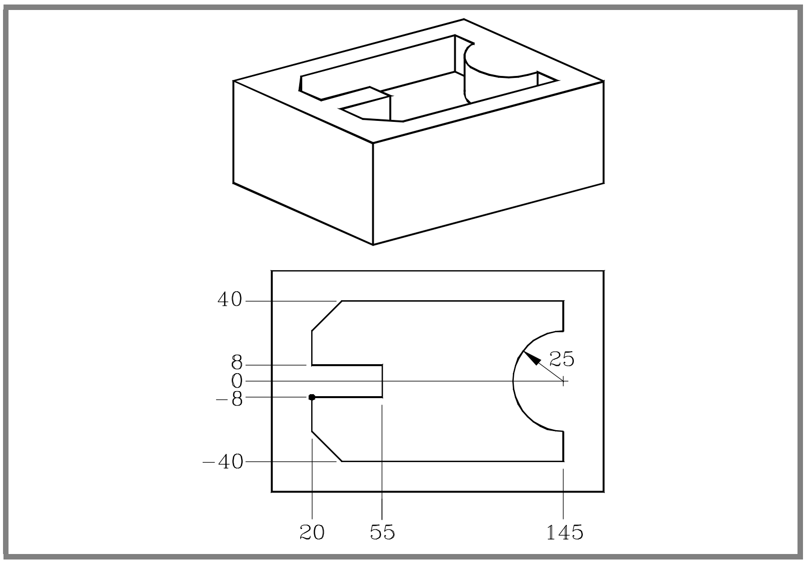
        It is carried out at finishing feedrate "F" and, if necessary, it recalculates the finishing pass (Δ) so all the passes are identical.

        It is carried out following paths concentric to the profile, in the same direction as the outside profile was defined. The islands are machined in the opposite direction.
9. Rapid withdrawal (G0) up to the approach plane.
10. Finishing of the side walls.

    It is carried out in "N" passes at the finishing feedrate "F" and with tangential entry and exit.

    The outside profile in the same direction that was defined and the islands in the opposite direction.
11. Rapid withdrawal (G0) up to the safety plane (Zs).

12.13.1 Examples of how to define 2D profiles



Profile P.XY	FAGOR 101	Recall

Configuration:

Abscissa axis: X	Ordinate axis: Y
Autozoom: Yes	Validate

Profile:

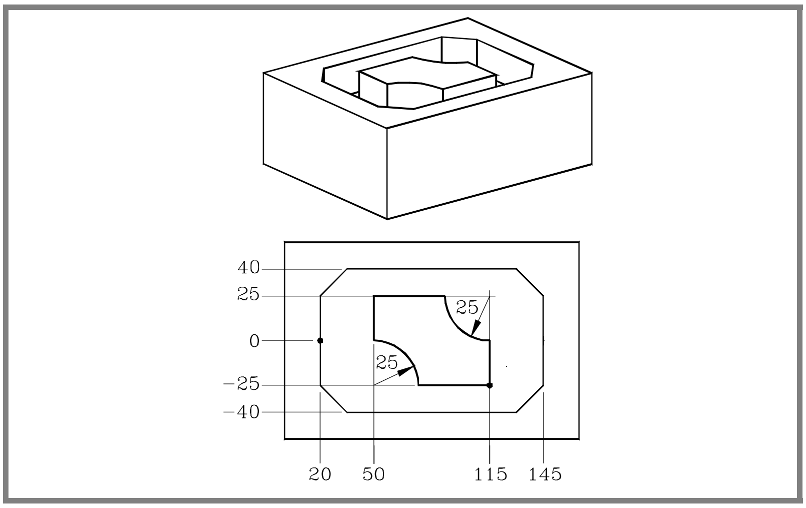
Starting point	X 20	Y -8		Validate
Straight	X 20	Y -40		Validate
Straight	X 145	Y -40		Validate
Straight	X 145	Y -25		Validate
Clockwise arc	Xf 145	Yf 25	R 25	Validate
Straight	X 145	Y 40		Validate
Straight	X 20	Y 40		Validate
Straight	X 20	Y 8		Validate
Straight	X 55	Y 8		Validate
Straight	X 55	Y -8		Validate
Straight	X 20	Y -8		Validate

Corners

Chamfer	
Select the lower left corner	Enter
Chamfer 15	Enter
Select the upper left corner	Enter
Chamfer 15	Enter
	Escape

End:

Save profile



Profile P.XY	FAGOR 102	Recall

Configuration:

Abscissa axis: X	Ordinate axis: Y
Autozoom: Yes	Validate

Profile (outside profile):

Starting point	X 20	Y 0	Validate
Straight	X 20	Y -40	Validate
Straight	X 145	Y -40	Validate
Straight	X 145	Y 40	Validate
Straight	X 20	Y 40	Validate
Straight	X 20	Y 0	Validate

Corners

Chamfer	
Select the lower left corner	Enter
Chamfer 15	Enter
Select the lower right corner	Enter
Chamfer 15	Enter
Select the upper right corner	Enter
Chamfer 15	Enter
Select the upper left corner	Enter
Chamfer 15	Enter
	Escape

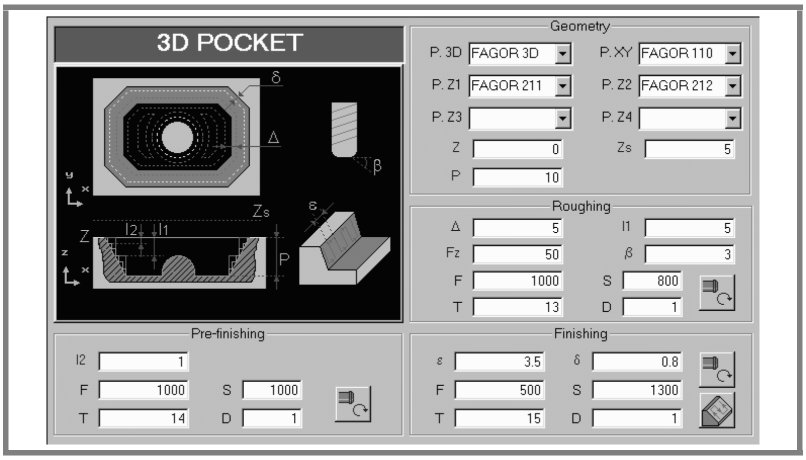
New profile (island):

Starting point	X 115	Y -25		Validate
Straight	X 115	Y 0		Validate
Clockwise arc	Xf 90	Yf 25		
	Xc 115	Yc 25	R 25	Validate
Straight	X 50	Y 25		Validate
Straight	X 50	Y 0		Validate
Clockwise arc	Xf 75	Yf -25		
	Xc 50	Yc -25	R 25	Validate
Straight	X 115	Y -25		Validate

End:

Save profile

12.14 3D pocket

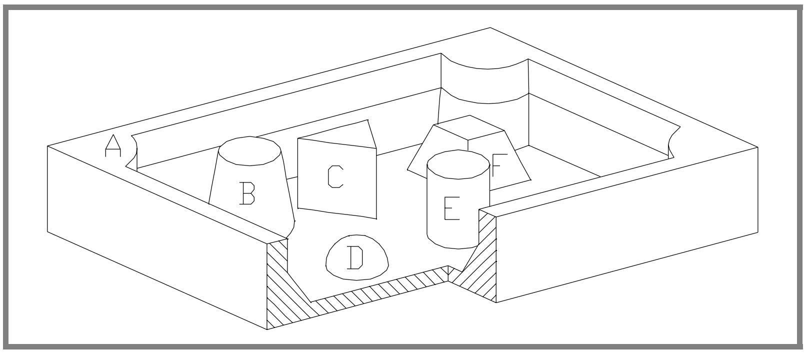


A pocket consists of an outside contour and a number of inside contours called islands.

As opposed to 2D pockets, whose walls are vertical, 3D pockets may be defined with a depth profile different for each contour (up to a maximum of 4 different ones).

The surface profile defines all the contours, the outside one and the inside ones (islands).

The first 4 contours defined in the surface profile may be assigned their own depth profiles. The rest of the profiles will be vertical.



The 3D pocket of the figure has 2 contours with "vertical profile" (C and E) and 4 contours with "non-vertical profile" (A, B, D and F).

Since only 4 contours may be defined with "non-vertical profile", contours A, B, D, F must be defined first and contours C, E at the end.

It is recommended to previously define the #ROUNDPAR instruction in order to obtain a good finish because the finishing passes are carried out in G05.

Geometric parameters:

The composition of the pocket and the plane and depth profiles are stored in \ Cnc8070\ Users\ Profile.

pocket.P3D	Pocket composition.
profile.PXY	Plane profile.
profile.PXZ	Depth profile.

P.3D Name of the 3D pocket.

Once the pocket configuration has been validated, the CNC associates the geometry of the pocket to its name (surface profile and depth profiles).

P.XY Name of the surface profile or plane profile.

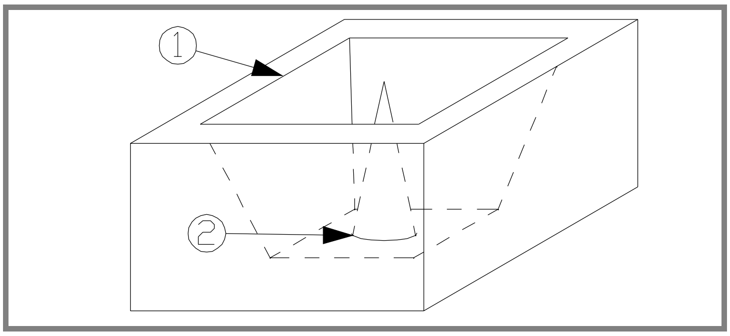
It must indicate all the contours.

For the outside contour, the one for the surface (1).

For the islands, the one for the base (2).

All the contours must be closed and must not intersect themselves.

Remember that the order in which the contours are defined is very important.



P.Z1 P.Z2 P.Z3 P.Z4

Name of the depth profiles.

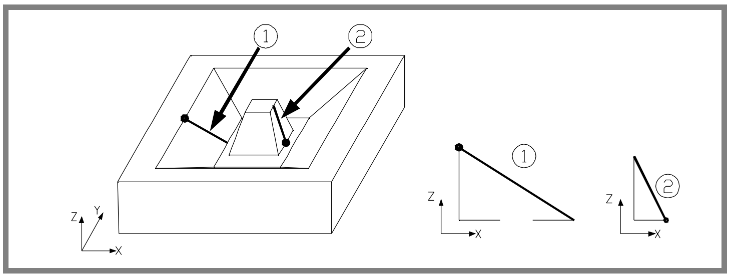
They corresponds to the first 4 contours defined in the surface profile, the number indicates the order.

To define the depth profile, use one of the axis of the plane and the perpendicular axis.

Use the same point to define the beginning of the contour and the beginning of the depth contour.

For the outside contour, one for the surface (1).

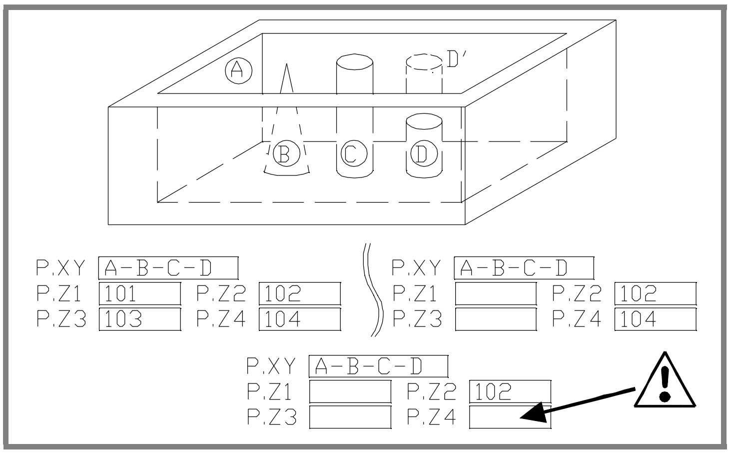
For the islands, one for the base (2).



All the profiles must be open and without direction changes along their travel (not zigzagging).

Vertical depth profiles for the outside contour and for the islands that reach the surface plane need not be programmed.

The figure shows three programming examples.



When defining the contours in the surface profile, all these cases follow the sequence A-B-C-D.

The top left-hand example defines all the depth profiles: Z1(A), Z2(B), Z3(C), Z4(D).

The top right-hand example has left out all the vertical depth profiles: Z1(A), Z3(C).

The lower example is programmed wrong because none of the vertical profiles have been defined.

If the profile of the island (D) is not defined, the cycle interprets that the island reaches the surface plane and will machine the island (D').

Z Part surface coordinate.

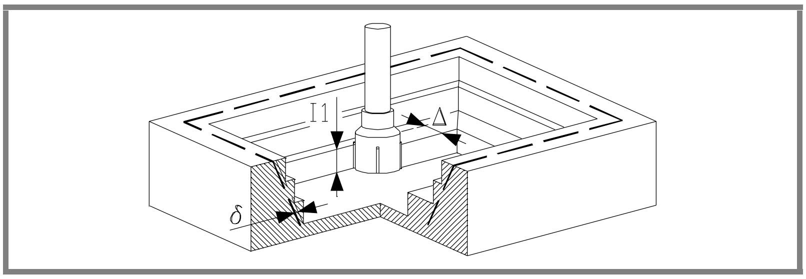
Zs Safety plane coordinate.

P Total depth.

Roughing parameters:

The roughing operation empties the pocket leaving the finishing stock δ on the side walls:

This stock is defined as finishing parameter.



The roughing operation defining parameters are:

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

I1 Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the pocket is machined with the given pass (step) except the last pass that machines the rest.

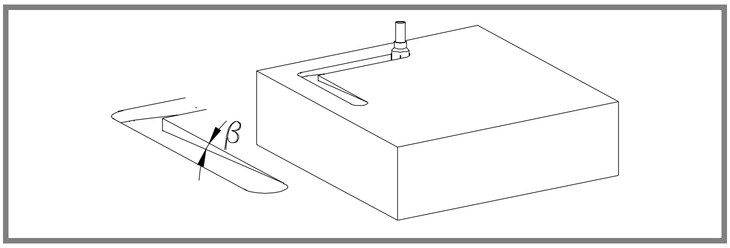
In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

Fz Penetration feedrate.

β Penetrating angle.

The penetration is carried out maintaining this angle until the corresponding depth is reached.

If defined with a value greater than the one assigned to the tool in the tool table, it assumes the table value.



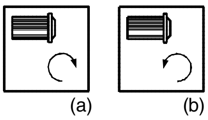
F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

If programmed T=0, there is no roughing.

D Tool offset.

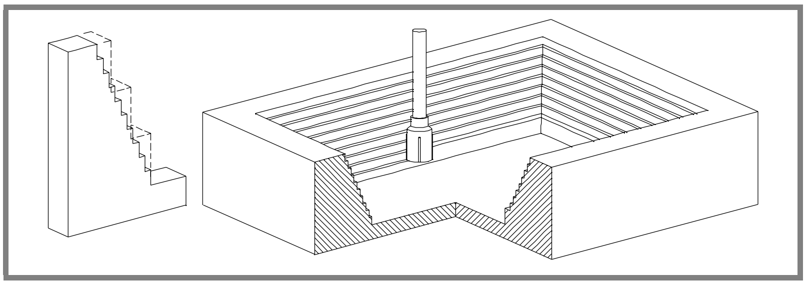


Spindle turning direction (icon).

Clockwise with icon(a).
Counterclockwise with icon(b).

Pre-finishing parameters:

This operation minimizes the ridges remaining on the side walls after the roughing operation while maintaining the finishing stock δ.



The pre-finishing operation defining parameters are:

I2 Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the pocket is machined with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

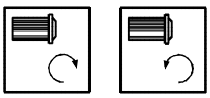
F Milling feedrate.

S Spindle speed.

T Pre-finishing tool.

If programmed T=0, there is no pre-finishing.

D Tool offset.


(a) (b)

Spindle turning direction (icon).

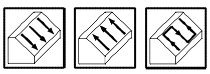
Clockwise with icon(a).
Counterclockwise with icon(b).

Finishing parameters:

The finishing operation takes into account the geometry of the tool tip. It compensates the tool tip radius defined in the table.

δ Finishing stock on the side walls.

ε Milling pass or width for the side walls.


(c) (d) (e)

Machining direction for the side walls (icon).

Always down(c), always up(d), in zig-zag(e).

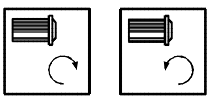
F Milling feedrate.

S Spindle speed.

T Finishing tool.

If programmed T=0, there is no finishing.

D Tool offset.


(a) (b)

Spindle turning direction (icon).

Clockwise with icon(a).
Counterclockwise with icon(b).

Executable pocket file.

To simulate or execute this type of pockets, the CNC uses an executable file with geometry information. This file is generated the first time the pocket is simulated or executed. If from the editor, any data of the pocket geometry or the used tool, is modified, the CNC will generate this file again.

In versions prior to V2.00, the user generated the executable file from the editor before inserting the cycle. From version V2.00 on, it is no longer necessary, the CNC is in charge of generating the executable file when necessary.

The executable files are stored in the directory CNC8070 \Users \Pocket with the name of the pocket (parameter P.3D) and the extension *C3D*. These files must not be deleted, moved to another location or tampered with in any way. If when executing or simulating the pocket, the CNC cannot find these files, it will generate them.

Overall, a 2D pocket consists of the following files.

pocket.P3D	Pocket composition.
profile.PXY	Plane profile.
profile.PXZ	Depth profile.
pocket.C3D	Executable file.

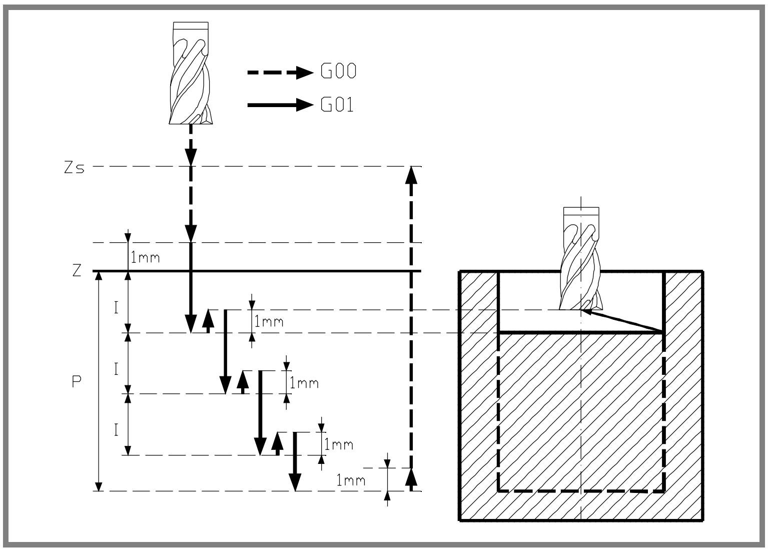
The executable file is also updated after a software update and when executing or simulating a pocket,

Basic operation:

The CNC calculates the initial coordinate depending on the geometry of the pocket and the tool radius.

1. It selects the roughing tool and starts the spindle in the requested direction.
2. Rapid movement (G0) to the roughing starting point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.



3. Rapid movement (G0) up to the approach plane.

4. Roughing operation.

It is carried out in layers until the total depth is reached.

4.1. Penetration "I1" at feedrate "Fz" at an angle "β".

4.2. Milling of the pocket surface up to a distance "δ" from the pocket wall. It is carried out at feedrate "F" and, if necessary, it recalculates the pass (Δ) so all the passes are identical.

It is carried out following paths concentric to the profile, in the same direction as the outside profile was defined. The islands are machined in the opposite direction.

4.3. Rapid withdrawal (G0) up to 1 mm off the machined surface.

5. Rapid withdrawal (G0) up to the approach plane.

6. It selects the pre-finishing tool and starts the spindle in the requested direction.

7. Pre-finishing operation for the side walls.

It is carried out with the pass indicated by "I2" and at the pre-finishing feedrate "F".

The outside profile in the same direction that was defined and the islands in the opposite direction.

8. Rapid withdrawal (G0) up to the approach plane.

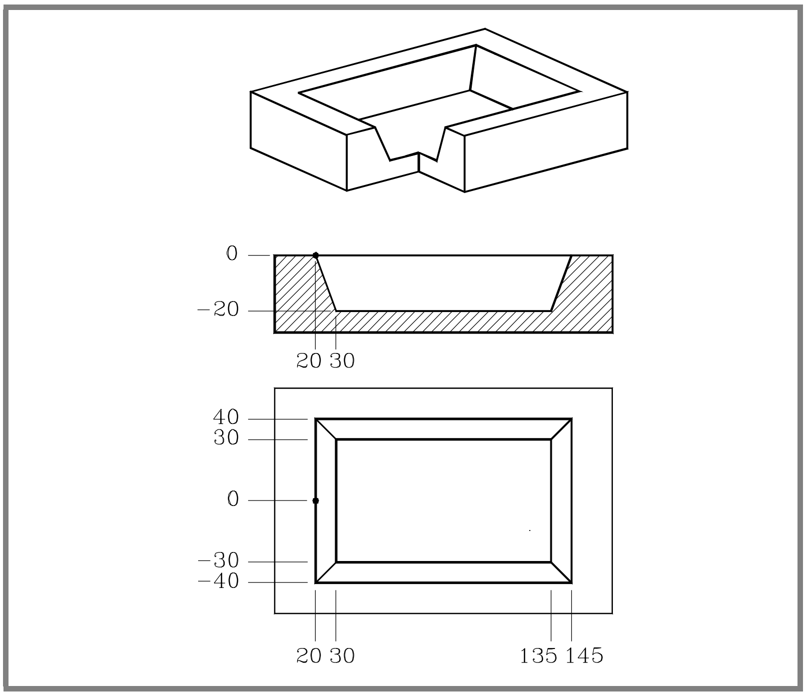
9. It selects the finishing tool and starts the spindle in the requested direction.

10. Finishing of the side walls.

It is carried out with the pass "ε" and direction indicated by the icon.

Rapid withdrawal (G0) up to the safety plane (Zs).

12.14.1 Examples of how to define 3D profiles



Pocket P.3D	FAGOR-A	

Profile P.XY	FAGOR 110	Recall

Configuration:

Abscissa axis: X	Ordinate axis: Y
Autozoom: Yes	Validate

Profile (outside profile):

Starting point	X 20	Y 0	Validate
Straight	X 20	Y -40	Validate
Straight	X 145	Y -40	Validate
Straight	X 145	Y 40	Validate
Straight	X 20	Y 40	Validate
Straight	X 20	Y 0	Validate

End:

Save profile

Profile P.Z1	FAGOR 211	Recall

Configuration:

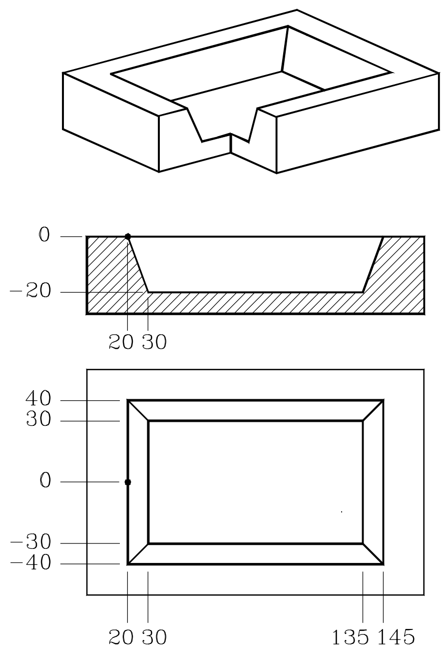
Abscissa axis: X	Ordinate axis: Z
Autozoom: Yes	Validate

Profile (depth profile):

Starting point	X 20	Z0	Validate
Straight	X 30	Z -20	Validate

End:

Save profile



Pocket P.3D	FAGOR-B	

Profile P.XY	FAGOR 120	Recall

Configuration:

Abscissa axis: X	Ordinate axis: Y
Autozoom: Yes	Validate

Profile (outside profile):

Starting point	X 20	Y 0	Validate
Straight	X 20	Y -40	Validate
Straight	X 145	Y -40	Validate
Straight	X 145	Y 40	Validate
Straight	X 20	Y 40	Validate
Straight	X 20	Y 0	Validate

New profile (island):

Circle	X 62.5	Y0	Xc 82.5	Yc 0	Validate

End:

Save profile

Profile P.Z1	FAGOR 221	Recall

Configuration:

Abscissa axis: X	Ordinate axis: Z
Autozoom: Yes	Validate

Profile (outside depth profile):

Starting point	X 20	Z 0	Validate
Straight	X 30	Z -20	Validate

End:

Save profile

Profile P.Z2	FAGOR 222	Recall

Configuration:

Abscissa axis: X	Ordinate axis: Z
Autozoom: Yes	Validate

Profile (island depth profile):

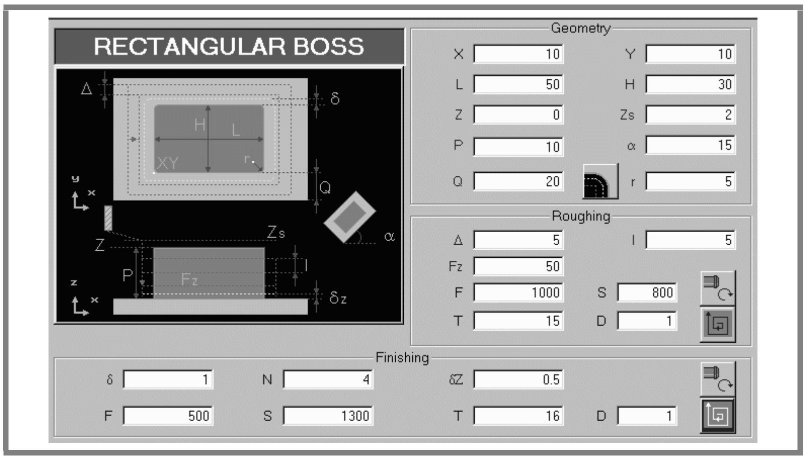
Starting point	X 62.5	Z -20	Validate
Straight	X 77.5	Z 0	Validate

End:

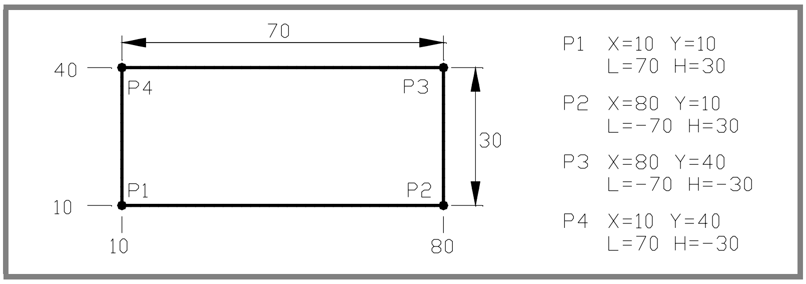
Save profile

12.15 Rectangular Boss

The way the roughing and finishing blocks of this cycle are joined will be the one previously set by the user with the instructions #HSC, G5, G50 or G7. We recommend to use #HSC or G5 controlling the shape of the corner with the instruction #ROUNDPAR.



Geometric parameters:



X, Y Corner of the boss.

L, H Boss dimensions.

The sign indicates the orientation referred to the XY point.

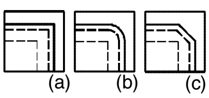
Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

τo Angle, in degrees, between the boss and the abscissa axis. The turn is carried out on the defined corner, X,Y point.

Q Amount of stock to be removed.



Type of corner (icon).

Square corner with icon(a).
Rounded corner with icon(b).
Chamfered corner with icon(c).

r Rounding radius or chamfer size.

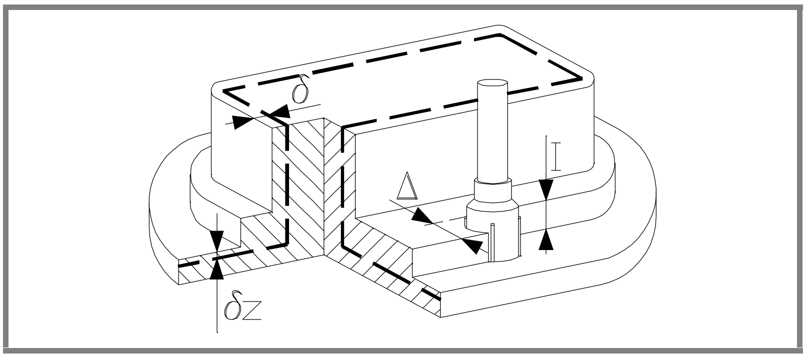
Roughing parameters:

The roughing operation machines the boss leaving the following finishing stocks:

δ Finishing stock on the side walls.

δz Finishing stock at the base of the boss.

Both stocks are defined as finishing parameters.



The roughing operation defining parameters are:

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the boss is machined with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

Fz Penetration feedrate.

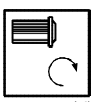
F Surface milling feedrate.

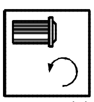
S Spindle speed.

T Roughing tool.

If programmed T=0, there is no roughing.

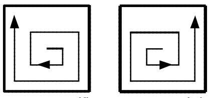
D Tool offset.


(d)


(e)

Spindle turning direction (icon).

Clockwise with icon(d).
Counterclockwise with icon(e).


(f) (g)

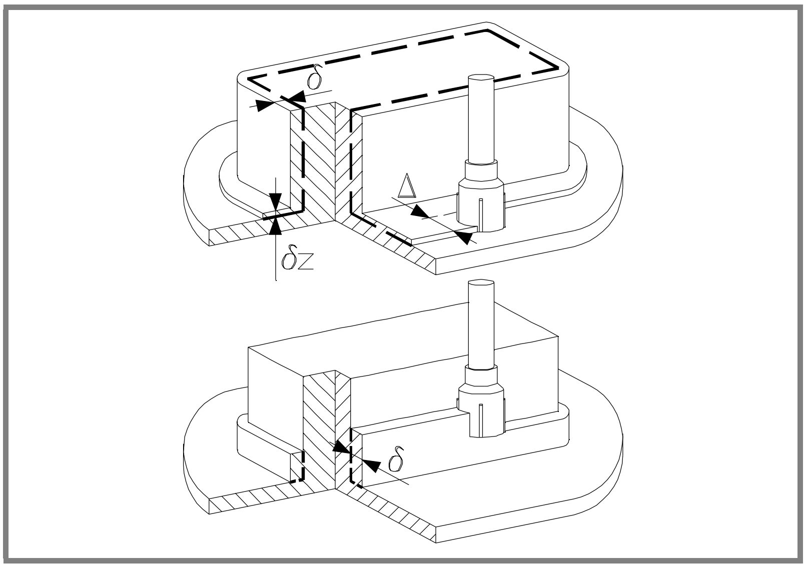
Machining direction (icon).

Clockwise with icon(f).
Counterclockwise with icon(g).

Finishing parameters:

The finishing operation is carried out in two stages.

First, it machines the base of the boss and then the side walls, with tangential entry and exit.



The finishing operation defining parameters are:

δ Finishing stock on the side walls.

δz Finishing stock at the base of the boss.

N Number of penetration passes (steps) for the side finishing. If the resulting step is greater than the cutting length assigned to the table in the tool table, the step will be limited to that value.

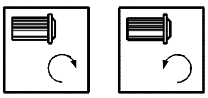
F Surface and side milling feedrate.

S Spindle speed.

T Finishing tool.

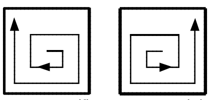
If programmed T=0, there is no finishing.

D Tool offset.


(d) (e)

Spindle turning direction (icon).

Clockwise with icon(d).
Counterclockwise with icon(e).


(f) (g)

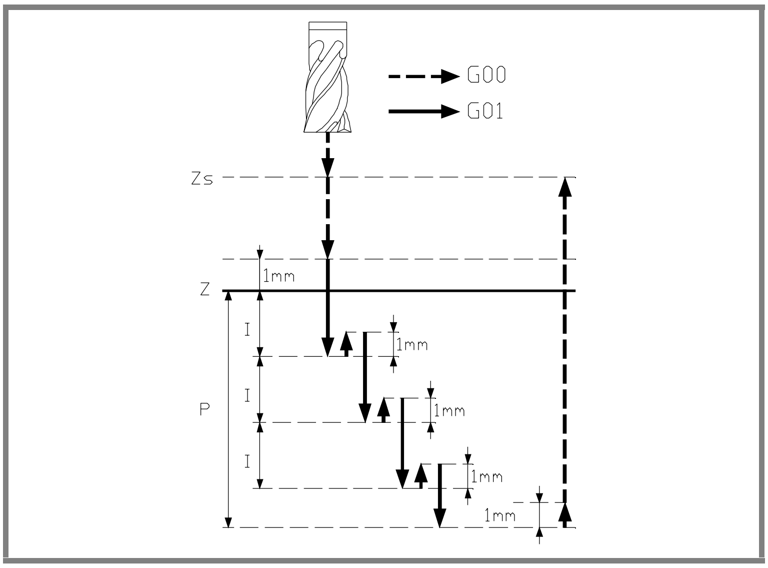
Machining direction (icon).

Clockwise with icon(f).
Counterclockwise with icon(g).

Basic operation:

1. It selects the roughing tool and starts the spindle in the requested direction.
2. Rapid movement (G0) to the roughing starting point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.
3. Rapid movement (G0) up to the approach plane.



4. Roughing operation.

   It is carried out in layers, until reaching the total depth minus the finishing stock at the base "δz".

   4.1. Penetration "I" at feedrate "Fz".

   4.2. Milling of the boss surface up to a distance "δ" from the side wall. It is carried out at feedrate "F" and, if necessary, it recalculates the pass (Δ) so all the passes are identical.

   4.3. Rapid withdrawal (G0) to the starting point.
5. Rapid withdrawal (G0) up to the safety plane (Zs).
6. It selects the finishing tool and it approaches in rapid (G0) down to 1 mm from the last roughing operation.
7. Finishing of the base of the boss.

   7.1. Penetration at feedrate "Fz".

   7.2. Milling of the base of the boss up to a distance "δ" from the side wall. It is carried out at the finishing feedrate "F" and with the roughing pass.
8. Rapid withdrawal (G0) to the starting point in the approach plane.
9. Finishing of the side walls.

   It is carried out in "N" passes at the finishing feedrate "F" and with tangential entry and exit.

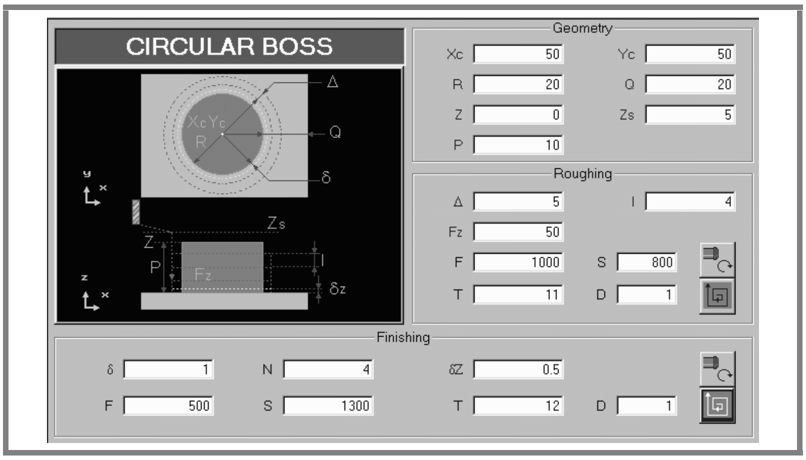
10. Rapid withdrawal (G0) up to the safety plane (Zs).

If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

11. Rapid movement (G0) to the next point.

12. Repeats steps 3, 4, 5, 6, 7, 8, 9, 10.

12.16 Circular boss



Geometric parameters:

Xc, Yc Center of the boss.

R Boss radius.

Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

Q Amount of stock to be removed.

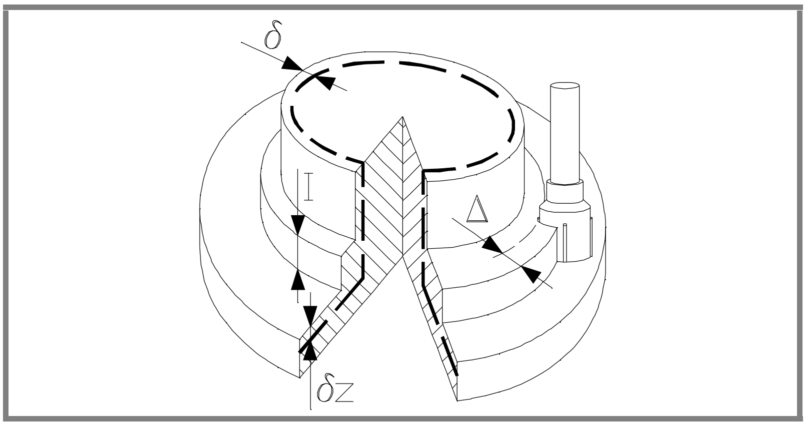
Roughing parameters:

The roughing operation machines the boss leaving the following finishing stocks:

δ Finishing stock on the side walls.

δz Finishing stock at the base of the boss.

Both stocks are defined as finishing parameters.



The roughing operation defining parameters are:

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the boss is machined with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

Fz Penetration feedrate.

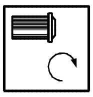
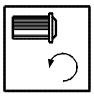
F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

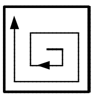
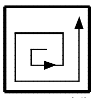
If programmed T=0, there is no roughing.

D Tool offset.


(a)

(b)

Spindle turning direction (icon).

Clockwise with icon(a).
Counterclockwise with icon(b).


(c)

(d)

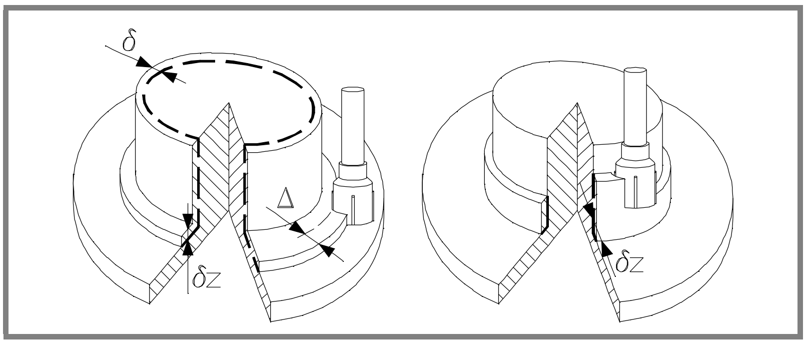
Machining direction (icon).

Clockwise with icon(c).
Counterclockwise with icon(d).

Finishing parameters:

The finishing operation is carried out in two stages.

First, it machines the base of the boss and then the side walls, with tangential entry and exit.



The finishing operation defining parameters are:

δ Finishing stock on the side walls.

δz Finishing stock at the base of the boss.

N Number of penetration passes (steps) for the side finishing. If the resulting step is greater than the cutting length assigned to the table in the tool table, the step will be limited to that value.

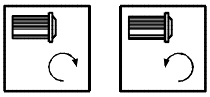
F Surface and side milling feedrate.

S Spindle speed.

T Finishing tool.

If programmed T=0, there is no finishing.

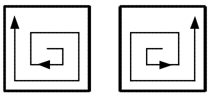
D Tool offset.



(a) (b)

Spindle turning direction (icon).

Clockwise with icon(a).
Counterclockwise with icon(b).



(c) (d)

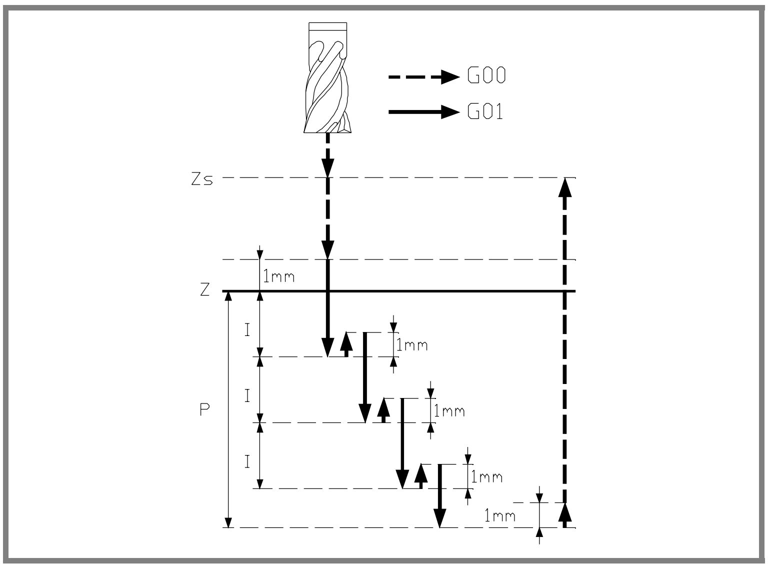
Machining direction (icon).

Clockwise with icon(c).
Counterclockwise with icon(d).

Basic operation:

1. It selects the roughing tool and starts the spindle in the requested direction.
2. Rapid movement (G0) to the roughing starting point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.



3. Rapid approach (G0) up to 1 mm off the surface "Z".

4. Roughing operation.

It is carried out in layers, until reaching the total depth minus the finishing stock at the base "δz".

4.1. Penetration "I" at feedrate "Fz".

4.2. Milling of the boss surface up to a distance "δ" from the side wall. It is carried out at feedrate "F" and, if necessary, it recalculates the pass (Δ) so all the passes are identical.

4.3. Rapid withdrawal (G0) to the starting point.

5. Rapid withdrawal (G0) up to the safety plane (Zs).

6. It selects the finishing tool and it approaches in rapid (G0) down to 1 mm from the last roughing operation.

7. Finishing of the base of the boss.

7.1. Penetration at feedrate "Fz".

7.2. Milling of the base of the boss up to a distance "δ" from the side wall. It is carried out at the finishing feedrate "F" and with the roughing pass.

8. Rapid withdrawal (G0) to the starting point in the approach plane.

9. Finishing of the side walls.

It is carried out in "N" passes at the finishing feedrate "F" and with tangential entry and exit.

10. Rapid withdrawal (G0) up to the safety plane (Zs).

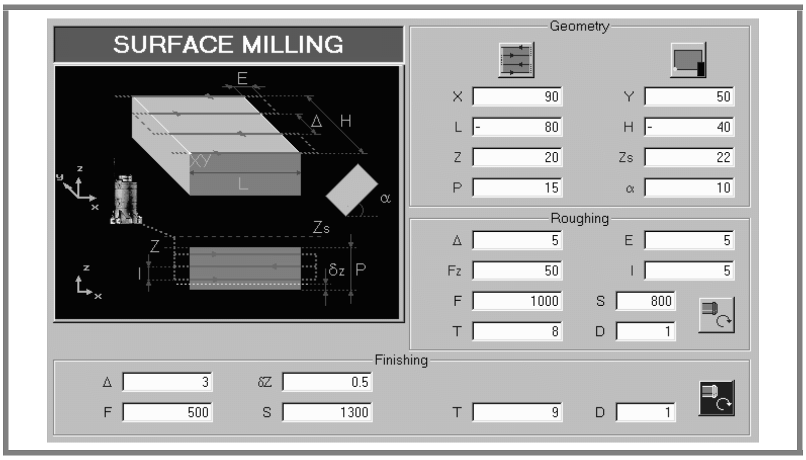
If it has a multiple machining operation associated with it, it executes the following steps as often as necessary:

11. Rapid movement (G0) to the next point.

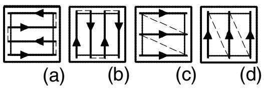
12. Repeats steps 3, 4, 5, 6, 7, 8, 9, 10.

12.17 Surface milling

The way the roughing and finishing blocks of this cycle are joined will be the one previously set by the user with the instructions #HSC, G5, G50 or G7. We recommend to use #HSC or G5 controlling the shape of the corner with the instruction #ROUNDPAR.

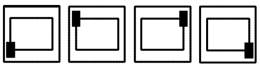


Geometric parameters:



Machining direction (icon).

Bidirectional in X(a), Bidirectional in Y(b).
Unidirectional in X(c), Unidirectional in Y(d).



Corner where the surface milling begins (icon).

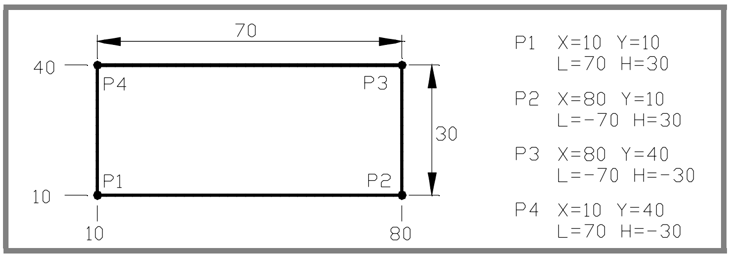
Any of the 4 corners may be selected.

X, Y, L, H

Surface to be milled.

Define one of the corners (X, Y), the length (L) and the width (H) of the surfaced to be milled.

The (X, Y) point needs not coincide with the corner selected to begin machining. The sign of L and H indicates the orientation with respect to the XY point.



Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

α Angle, in degrees, between the surface and the abscissa axis. The turn is carried out on the defined corner, X,Y point.

Roughing parameters:

The roughing operation leaves a finishing stock δz defined as finishing parameter.

The roughing operation defining parameters are:

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

E Overshooting distance of the tool off the surface being milled.

Fz Penetration feedrate.

I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the milling is carried out with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

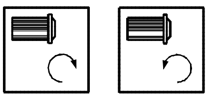
F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

If programmed T=0, there is no roughing.

D Tool offset.



Spindle turning direction (icon).

Clockwise with icon(e).

Counterclockwise with icon(f).

Finishing parameters:

δz Finishing stock.

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

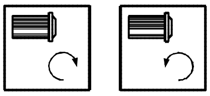
F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

If programmed T=0, there is no roughing.

D Tool offset.


(e) (f)

Spindle turning direction (icon).

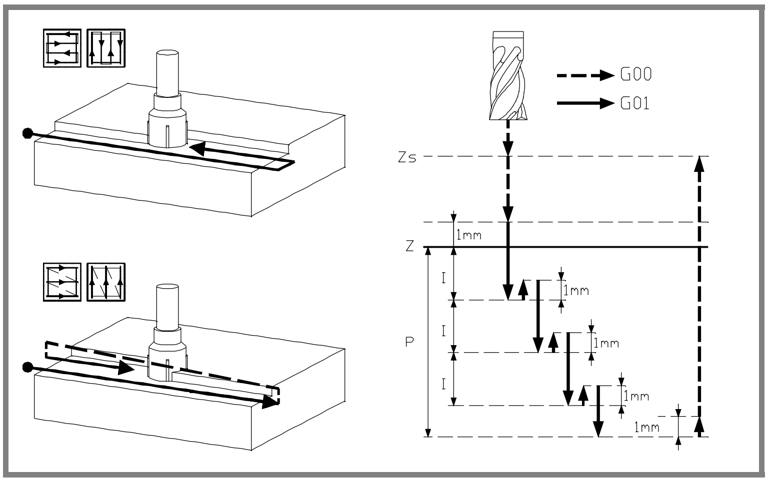
Clockwise with icon(e).

Counterclockwise with icon(f).

Basic operation:

1. It selects the roughing tool and starts the spindle in the requested direction.
2. Rapid movement (G0) to the roughing starting point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.

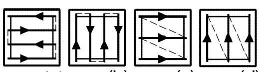


3. Rapid movement (G0) up to the approach plane.
4. Roughing operation.

   It is carried out in layers, until reaching the total depth minus the finishing distance "δz".

   4.1. Penetration "I" at feedrate "Fz".

   4.2. Milling at feedrate "F" and, if necessary, it recalculates the pass (Δ) so all the passes are identical.

   
   (a) (b) (c) (d)

   In bidirectional milling(a)(b), all the movements are at feedrate "F".

   In unidirectional milling(c)(d), the movements between two consecutive milling passes are carried out in rapid and at 1 mm above the part.

   4.3. Rapid withdrawal (G0) up to 1 mm above the part.

   4.4. Rapid movement (G0) to the starting point.

5. Rapid withdrawal (G0) up to the safety plane (Zs).

6. Finishing.

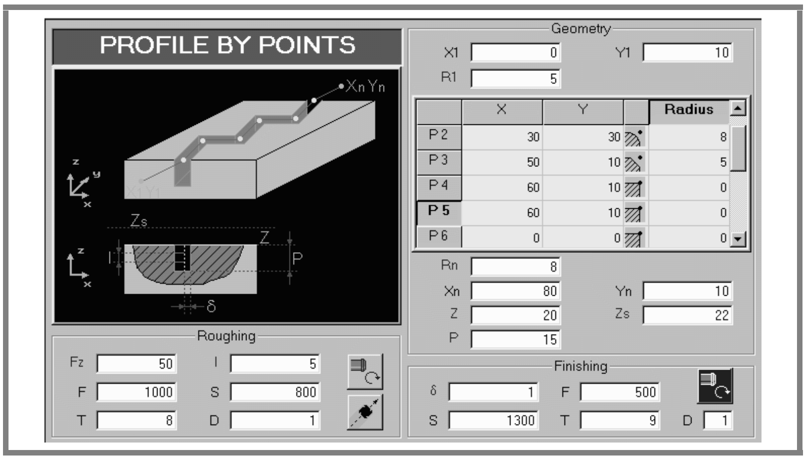
6.1. Penetration at feedrate "Fz".

6.2. Milling at finishing feedrate "F" and, if necessary, it recalculates the finishing pass (Δ) so all the passes are identical.

7. Rapid withdrawal (G0) up to the safety plane (Zs).

12.18 Point-to-point profile

The way the roughing and finishing blocks of this cycle are joined will be the one previously set by the user with the instructions #HSC, G5, G50 or G7. We recommend to use #HSC or G5 controlling the shape of the corner with the instruction #ROUNDPAR.

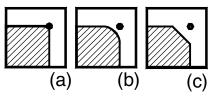


Geometric parameters:

X1, Y1 Profile entry point

R1 Radius of the tangential entry to the profile

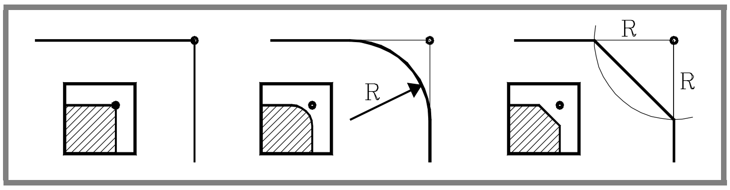
P1..P12 Points of the profile.



All intermediate points P2 to P11 have an icon to indicate the type of corner: square[(a)], rounded[(b)] or chamfered[(c)].

For rounded or chamfered corners, indicate the rounding radius or chamfer size.

When not using all 12 points, define the first unused point with the same coordinates as those of the last point of the profile.



Rn Radius of the tangential exit from the profile

Xn, Yn Profile exit point

Z Part surface coordinate.

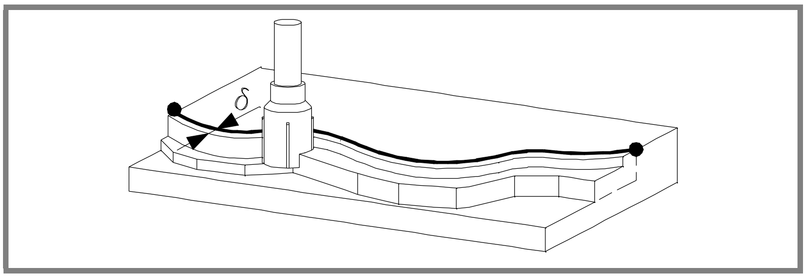
Zs Safety plane coordinate.

P Total depth.

Roughing parameters:

The roughing operation mills the profile leaving the finishing stock δ.

This stock is defined as finishing parameter.



The roughing operation defining parameters are:

Fz Penetration feedrate.

I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the milling is carried out with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

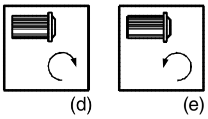
F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

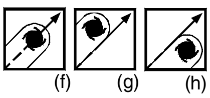
If programmed T=0, there is no roughing.

D Tool offset.



Spindle turning direction (icon).

Clockwise with icon(d).
Counterclockwise with icon(e).



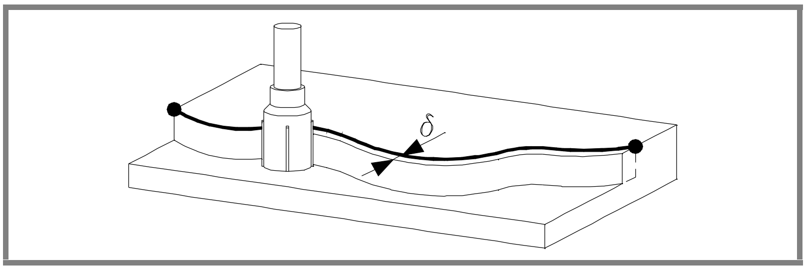
Tool radius compensation (icon).

Without compensation(f).
Left-hand compensation(g).
Right-hand compensation(h).

Finishing parameters:

In order to carry out the finishing operation, the roughing must be defined with tool radius compensation.

The operation removes the finishing stock (δ).



The roughing operation defining parameters are:

δ Finishing stock on the side walls.

When working without tool radius compensation, there is no finishing operation, the finishing stock (δ) is ignored.

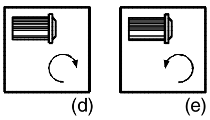
F Milling feedrate.

S Spindle speed.

T Finishing tool.

If programmed T=0, there is no finishing.

D Tool offset.



Spindle turning direction (icon).

Clockwise with icon(d).

Counterclockwise with icon(e).

Basic operation:

1. It selects the roughing tool and starts the spindle in the requested direction.
2. Rapid movement (G0), up to the XY point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.
3. Rapid movement (G0) up to the approach plane.
4. Roughing operation.

   It is carried out in layers until the total depth is reached.

   4.1. Penetration "I" at feedrate "Fz".

   4.2. Profile milling at feedrate "F" and tangential entry if it has been programmed.

   If it was defined with tool radius compensation, the milling is carried out at a "δ" distance from the wall.

   4.3. Exit to point XnYn with tangential exit if it has been programmed.

   4.4. Rapid withdrawal (G0) up to the safety plane (Zs).

4.5. Rapid movement to the starting point X1Y1.

5. It selects the finishing tool and starts the spindle in the requested direction.

6. Finishing operation.

7. Penetration to the bottom at feedrate "Fz".

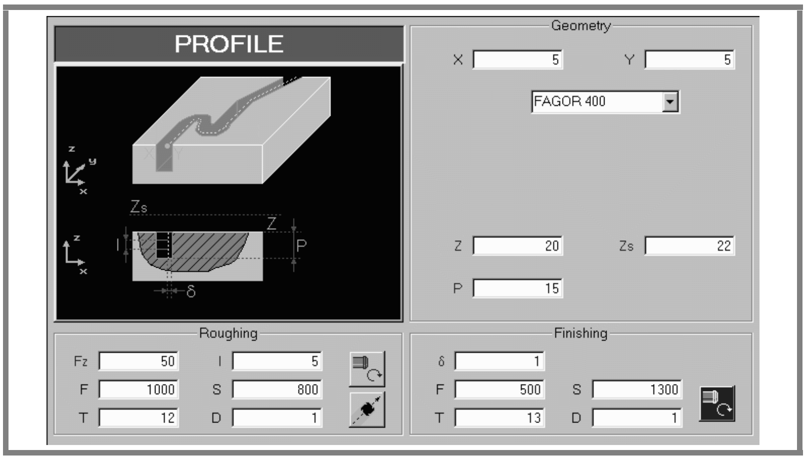
   7.1. Profile milling at feedrate "F" and tangential entry if it has been programmed.

   7.2. Exit to point XnYn with tangential exit if it has been programmed.

8. Rapid withdrawal (G0) up to the safety plane (Zs).

12.19 Profile

The way the roughing and finishing blocks of this cycle are joined will be the one previously set by the user with the instructions #HSC, G5, G50 or G7. We recommend to use #HSC or G5 controlling the shape of the corner with the instruction #ROUNDPAR.



Geometric parameters:

X, Y Profile entry point

Name of the profile.

To machine with tangential entry and exit, define these values inside the profile.

Z Part surface coordinate.

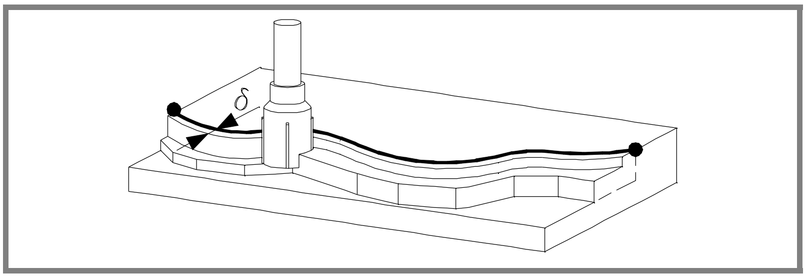
Zs Safety plane coordinate.

P Total depth.

Roughing parameters:

The roughing operation mills the profile leaving the finishing stock δ.

This stock is defined as finishing parameter.



The roughing operation defining parameters are:

Fz Penetration feedrate.

I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the milling is carried out with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

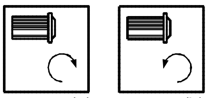
F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

If programmed T=0, there is no roughing.

D Tool offset.



(a) (b)

Spindle turning direction (icon).

Clockwise with icon[(a)].
Counterclockwise with icon[(b)].

(c) (d) (e)

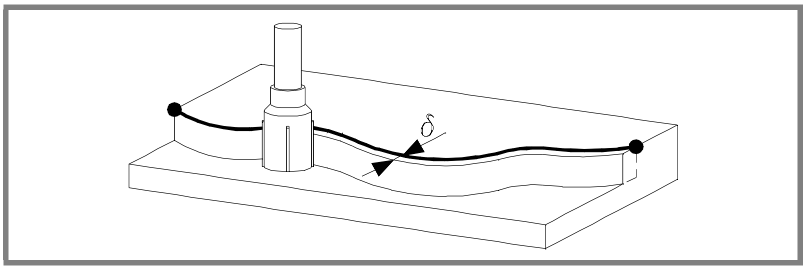
Tool radius compensation (icon).

Without compensation[(c)].
Left-hand compensation[(d)].
Right-hand compensation[(e)].

Finishing parameters:

In order to carry out the finishing operation, the roughing must be defined with tool radius compensation.

This operation removes the finishing stock (δ).



The roughing operation defining parameters are:

δ Finishing stock on the side walls.

When working without tool radius compensation, the stock (δ) is ignored . In this case, the tool center travel is the same when roughing as when finishing.

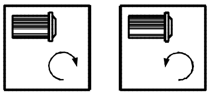
F Milling feedrate.

S Spindle speed.

T Finishing tool.

If programmed T=0, there is no finishing.

D Tool offset.


(a) (b)

Spindle turning direction (icon).

Clockwise with icon[a].
Counterclockwise with icon[b].

Basic operation:

1. It selects the roughing tool and starts the spindle in the requested direction.
2. Rapid movement (G0), up to the XY point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.
3. Rapid movement (G0) up to the approach plane.
4. Roughing operation.

   It is carried out in layers until the total depth is reached.

   4.1. Penetration "I" at feedrate "Fz".

   4.2. Profile milling at feedrate "F".

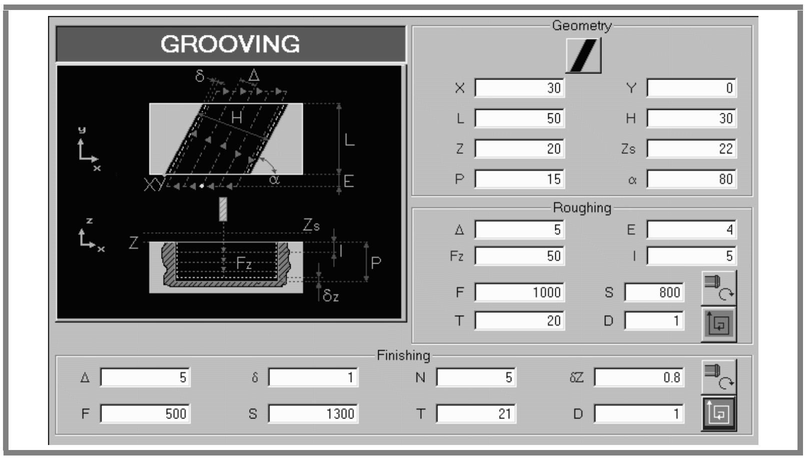
   If it was defined with tool radius compensation, the milling is carried out at a "δ" distance from the wall.

   4.3. Rapid withdrawal (G0) up to the safety plane (Zs).

   4.4. Rapid movement to the starting point X1Y1.
5. It selects the finishing tool and starts the spindle in the requested direction.
6. Finishing operation.
7. Penetration to the bottom at feedrate "Fz".
   - Profile milling at feedrate "F".
8. Rapid withdrawal (G0) up to the safety plane (Zs).

12.20 Slot milling

The way the roughing and finishing blocks of this cycle are joined will be the one previously set by the user with the instructions #HSC, G5, G50 or G7. We recommend to use #HSC or G5 controlling the shape of the corner with the instruction #ROUNDPAR.



Geometric parameters:

Type of slot milling (icon).

There are 6 possible types.
4 for slot mill each corner of the part.
2 for milling a slot across the part.

X, Y Corner where the slot is to be milled.

L, H Slot dimensions.

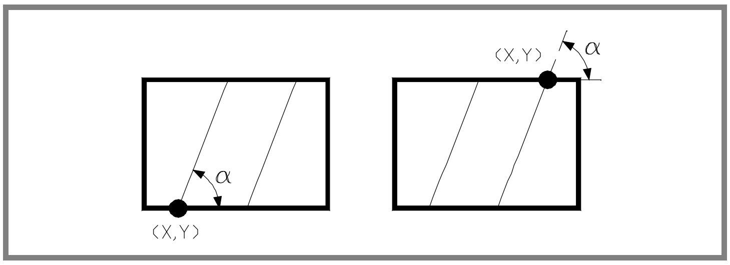
The sign indicates the orientation referred to the XY point.

Z Part surface coordinate.

Zs Safety plane coordinate.

P Total depth.

τo Angle, in degrees, between the slot and the abscissa axis. The turn is carried out on the defined corner, X,Y point.



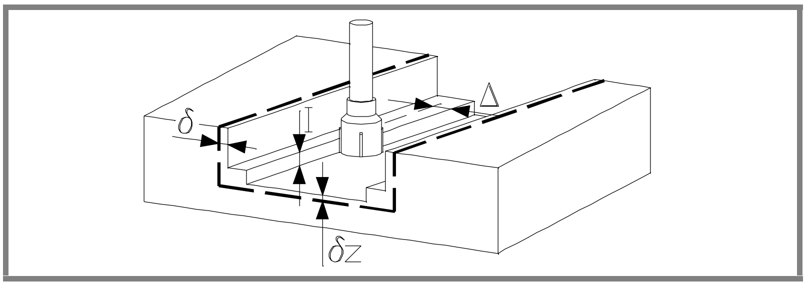
Roughing parameters:

The roughing operation leaves the following finishing stocks:

δ Finishing stock on the side walls.

δz	Finishing stock at the bottom of the pocket.

Both stocks are defined as finishing parameters.



The roughing operation defining parameters are:

Δ Maximum milling pass or width.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

E Overshooting distance of the tool off the surface being milled.

Fz Penetration feedrate.

I Penetration step.

- If programmed with a positive sign (I+), the cycle recalculates the step so all the penetrations are identical with the same value as or smaller than the one programmed.
- If programmed with a negative sign (I-), the slot milling is carried out with the given pass (step) except the last pass that machines the rest.

In either case, the cycle limits the step to the cutting length assigned to the tool in the tool table.

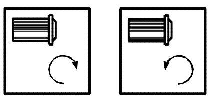
F Surface milling feedrate.

S Spindle speed.

T Roughing tool.

If programmed T=0, there is no roughing.

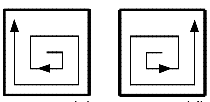
D Tool offset.


(a) (b)

Spindle turning direction (icon).

Clockwise with icon(a).

Counterclockwise with icon(b).


(c) (d)

Machining direction (icon).

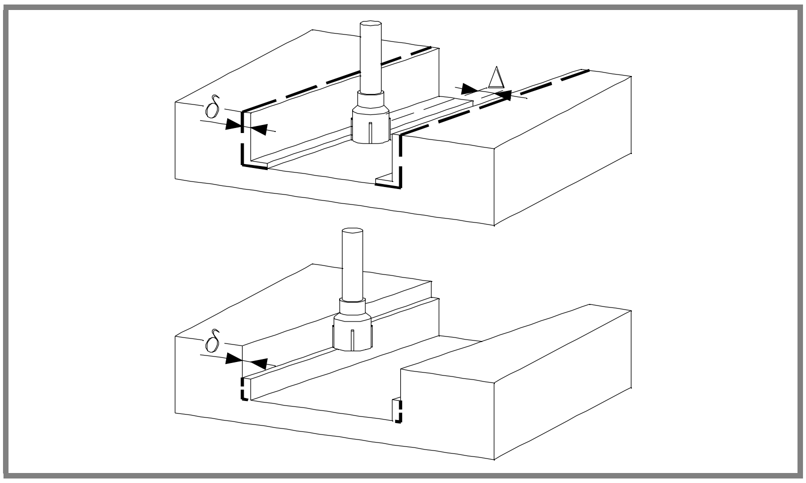
Clockwise with icon(c).

Counterclockwise with icon(d).

Finishing parameters:

The finishing operation is carried out in two stages.

First, it machines the bottom of the slot and then the side walls, with tangential entry and exit.



The finishing operation defining parameters are:

δ Finishing pass on the side walls.

δz Finishing pass at the bottom.

Δ Milling pass or width at the bottom of the slot.

The cycle recalculates the pass so that all the passes are identical, with the same value as or smaller than the one programmed.

If programmed with a 0 value, it assumes a value of 3/4 of the diameter of the selected tool.

N Number of penetration passes (steps) for the side finishing. If the resulting step is greater than the cutting length assigned to the table in the tool table, the step will be limited to that value.

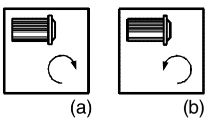
F Surface and side milling feedrate.

S Spindle speed.

T Finishing tool.

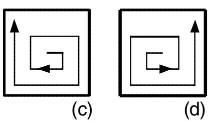
If programmed T=0, there is no finishing.

D Tool offset.



Spindle turning direction (icon).

Clockwise with icon[(a)].
Counterclockwise with icon[(b)].



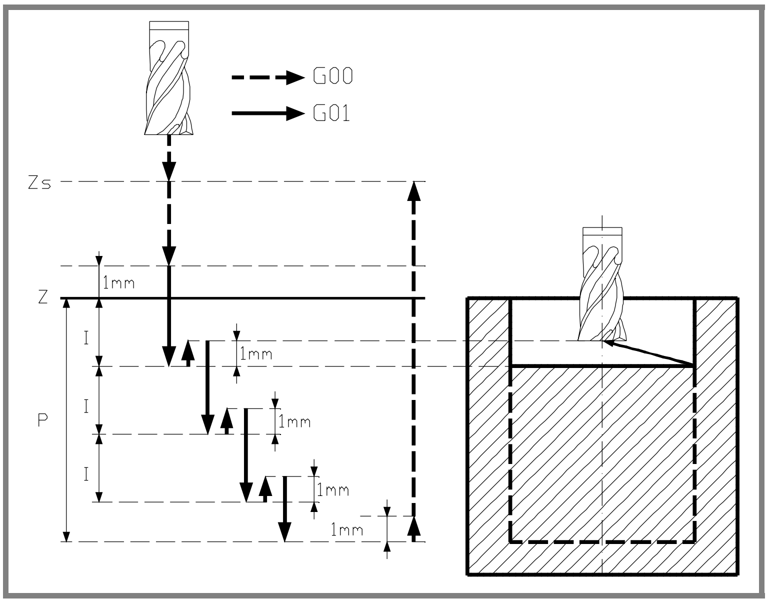
Machining direction (icon).

Clockwise with icon[(c)].
Counterclockwise with icon[(d)].

Basic operation:

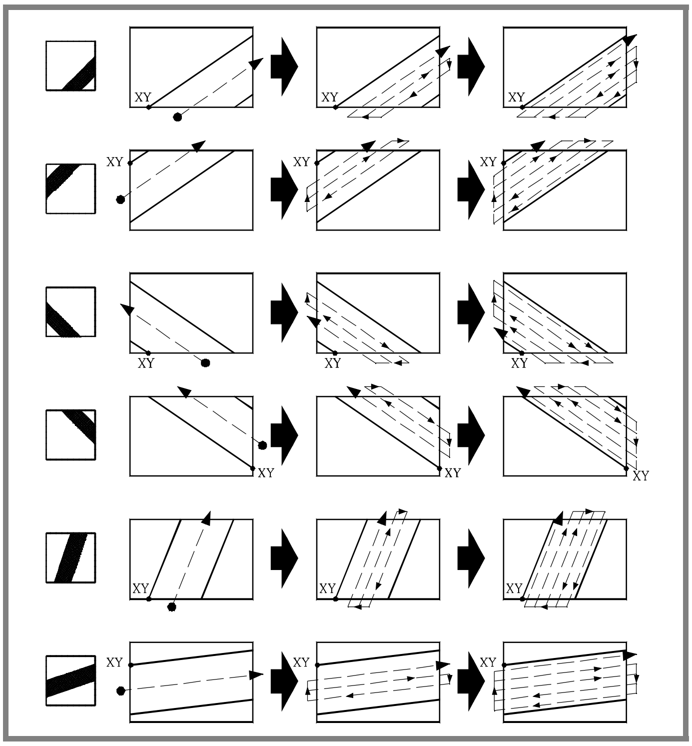
1. It selects the roughing tool and starts the spindle in the requested direction.
2. Rapid movement (G0) to the roughing starting point and the safety plane (Zs).

   Depending on the starting plane, it first moves in XY and then in Z or vice versa.
3. Rapid movement (G0) up to the approach plane.



4. Roughing operation.

   It is carried out in layers, until reaching the total depth minus the finishing distance "δz".



4.1. Penetration "I" at feedrate "Fz".

4.2. Slot milling of the boss surface up to a distance "δ" from the side wall. It is carried out at feedrate "F" and, if necessary, it recalculates the pass (Δ) so all the passes are identical.

4.3. Rapid withdrawal (G0) up to the safety plane (Zs).

4.4. Rapid movement (G0) to the starting point.

4.5. Rapid approach (G0) up to 1 mm off the machined surface.

5. Rapid withdrawal (G0) up to the safety plane (Zs).

6. It selects the finishing tool and it approaches in rapid (G0) down to 1 mm from the roughed out bottom.

7. Finishing of the bottom of the slot.

7.1. Penetration at feedrate "Fz".

7.2. Milling of the bottom of the slot up to a distance "δ" from the pocket wall. It is carried out at finishing feedrate "F" and, if necessary, it recalculates the finishing pass (Δ) so all the passes are identical.

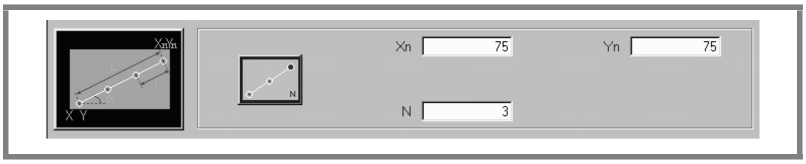
8. Rapid withdrawal (G0) up to the safety plane (Zs).

9. Finishing of the side walls.

It is carried out in "N" passes at the finishing feedrate "F".

10. Rapid withdrawal (G0) up to the safety plane (Zs).

12.21 Multiple machining in a straight line



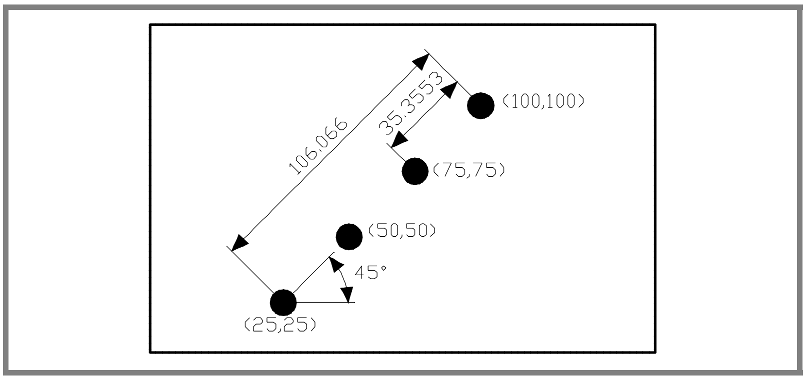
Definition format (icon).

There are 5 different ways to define the machining operation. To select the desired one, place the cursor on the icon and press the space bar.

The number of machining operations "N" must also include the one for the cycle defining point.

Programming example:

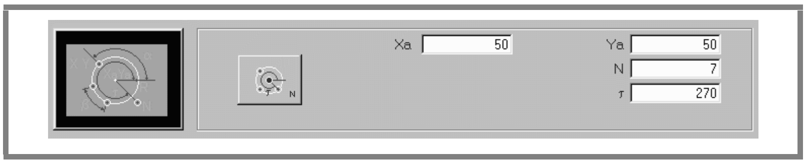
The canned cycle defined at point X25, Y25 is to be repeated at the rest of the points.



We now show the 5 possible ways to define it.

1)	Coordinates of the end point	Xn 100, Yn 100
	Total number of machining operations	N 4
2)	Angle of the path	α 45
	Distance to travel	L 106.066
	Total number of machining operations	N 4
3)	Angle of the path	α 45
	Total number of machining operations	N 4
	Distance between machining operations	I 35.3553
4)	Coordinates of the end point	Xn 100, Yn 100
	Distance between machining operations	I 35.3553
5)	Angle of the path	α 45
	Distance to travel	L 106.066
	Distance between machining operations	I 35.3553

12.22 Multiple machining in an arc



Definition format (icon).

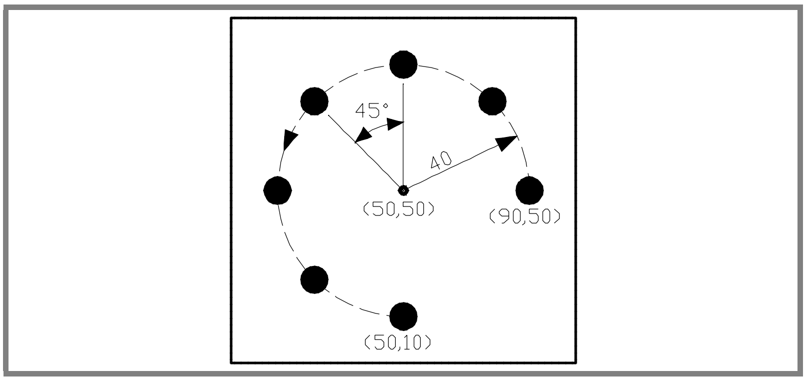
> There are 9 different ways to define the machining operation. To select the desired one, place the cursor on the icon and press the space bar.

The movement in arc is made counterclockwise. To do it clockwise, define the angular distance between machining operations β with a negative sign.

The number of machining operations "N" must also include the one for the cycle defining point.

Programming example:

The canned cycle defined at point X90, Y50 is to be repeated at the rest of the points.

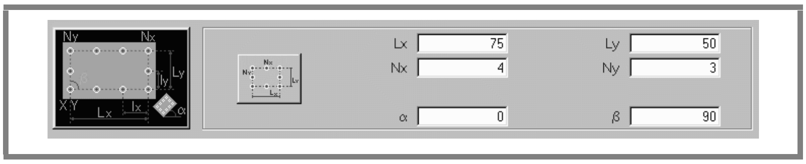


We now show the 9 possible ways to define it.

1)	Center coordinates	Xa 50, Ya 50
	Total number of machining operations	N 7
	Angle of the end point	τ 270
2)	Center coordinates	Xa 50, Ya 50
	Total number of machining operations	N 7
	Angular distance between machining operations	β 45
3)	Radius	R 40
	Total number of machining operations	N 7
	Angle of the starting point	α 0
	Angle of the end point	τ 270

4)	Radius	R 40
	Total number of machining operations	N 7
	Angle of the starting point	α 0
	Angular distance between machining operations	β 45
5)	Center coordinates	Xa 50, Ya 50
	Angle of the end point	τ 270
	Angular distance between machining operations	β 45
6)	Radius	R 40
	Angle of the starting point	α 0
	Angle of the end point	τ 270
	Angular distance between machining operations	β 45
7)	Center coordinates	Xa 50, Ya 50
	Radius	R 40
	Total number of machining operations	N 7
	Angle of the starting point	α 0
	Angular distance between machining operations	β 45
8)	Center coordinates	Xa 50, Ya 50
	Radius	R 40
	Total number of machining operations	N 7
	Angle of the starting point	α 0
	Angle of the end point	τ 270
9)	Center coordinates	Xa 50, Ya 50
	Radius	R 40
	Angle of the starting point	α 0
	Angle of the end point	τ 270
	Angular distance between machining operations	β 45

12.23 Multiple machining in a parallelogram pattern



Definition format (icon).

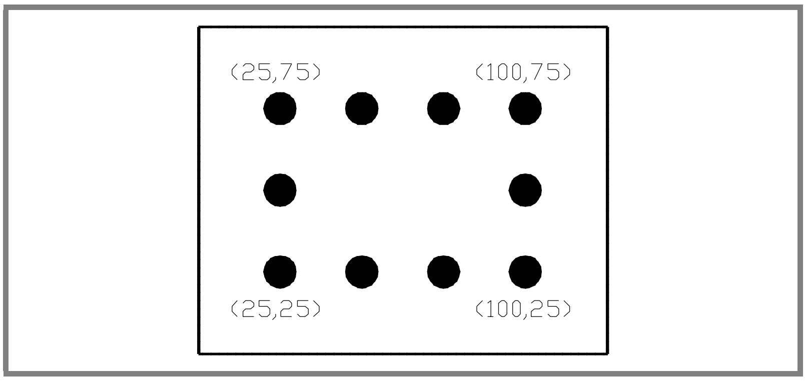
There are 3 different ways to define the machining operation. To select the desired one, place the cursor on the icon and press the space bar.

The cycle assumes the lower left point as the starting point. If it is not, define with the proper sign the distances between holes Ix and Iy.

The number of machining operations "N" must also include the one for the cycle defining point.

Programming example:

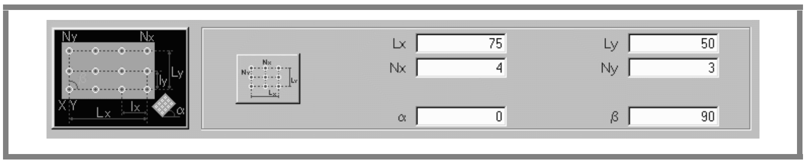
The canned cycle defined at point X25, Y25 is to be repeated at the rest of the points.



We now show the 3 possible ways to define it.

1)	Lengths in X, Y	Lx 75, Ly 50
	Number of machining operations in X and Y	Nx 4, Ny 3
	Rotation angle	α 0
	Angle between paths	β 90
2)	Number of machining operations in X and Y	Nx 4, Ny 3
	Distance between machining operations in X and Y	Ix 25, Iy 25
	Rotation angle	α 0
	Angle between paths	β 90
3)	Lengths in X, Y	Lx 75, Ly 50
	Distance between machining operations in X and Y	Ix 25, Iy 25
	Rotation angle	α 0
	Angle between paths	β 90

12.24 Multiple machining in a grid pattern



Definition format (icon).

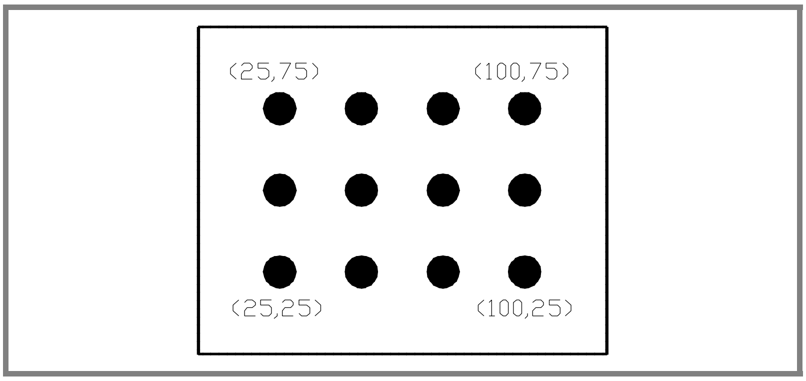
There are 3 different ways to define the machining operation. To select the desired one, place the cursor on the icon and press the space bar.

The cycle assumes the lower left point as the starting point. If it is not, define with the proper sign the distances between holes Ix and Iy.

The number of machining operations "N" must also include the one for the cycle defining point.

Programming example:

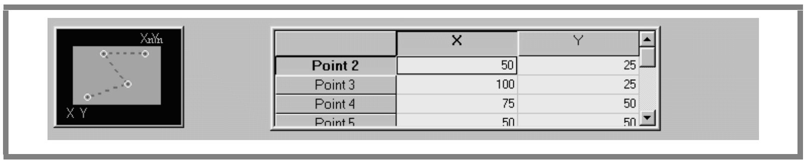
The canned cycle defined at point X25, Y25 is to be repeated at the rest of the points.



We now show the 3 possible ways to define it.

1)	Lengths in X, Y	Lx 75, Ly 50
	Number of machining operations in X and Y	Nx 4, Ny 3
	Rotation angle	α 0
	Angle between paths	β 90
2)	Number of machining operations in X and Y	Nx 4, Ny 3
	Distance between machining operations in X and Y	Ix 25, Iy 25
	Rotation angle	α 0
	Angle between paths	β 90
3)	Lengths in X, Y	Lx 75, Ly 50
	Distance between machining operations in X and Y	Ix 25, Iy 25
	Rotation angle	α 0
	Angle between paths	β 90

12.25 Random multiple machining



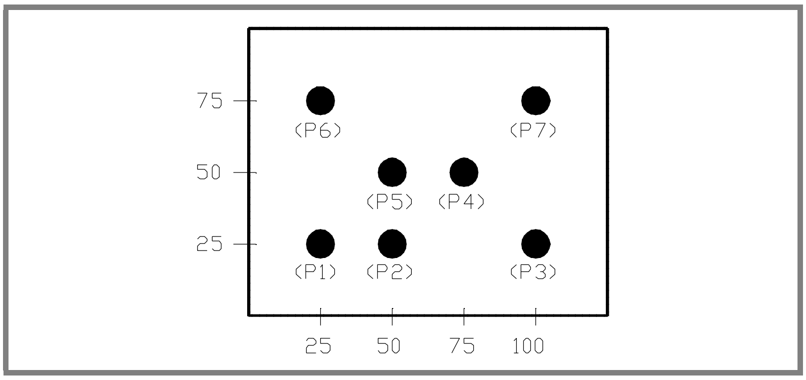
The starting point is the cycle defining point.

The rest of the points (P2) to (P12) must be defined in the area for multiple machining.

When not using all the points, define the first unused point with the same coordinates as those of the last point of the profile.

Programming example:

The canned cycle defined at point X25, Y25 is to be repeated at the rest of the points.



The canned cycle is defined at point (P1) X25, Y25

The rest of the points (P2) to (P7) must be defined in the area for multiple machining.

Since there are only 7 points, you must define (P8) = (P7).

(P2)	X 50	Y 25
(P3)	X 100	Y 25
(P4)	X 75	Y 50
(P5)	X 50	Y 50
(P6)	X 25	Y 75
(P7)	X 100	Y 75
(P8)	X 100	Y 75

COORDINATE TRANSFORMATION

13

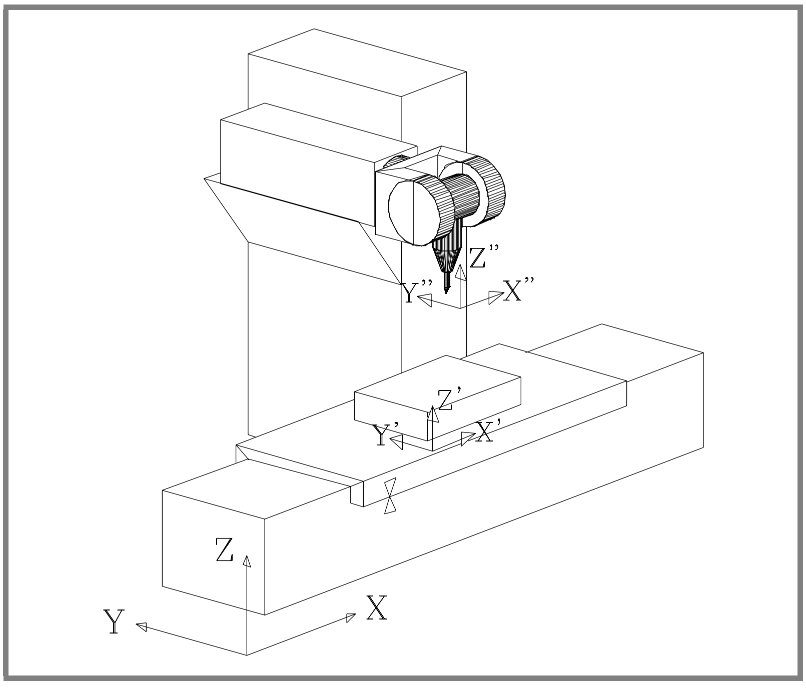
The description of the general coordinate transformation is divided into these basic functionalities:

- Selection of the kinematics. #KIN ID instruction.
- Definition and selection of the machining coordinate system (incline plane). #CS instruction.
- Definition and selection of the fixture coordinate system. #ACS instruction.
- RTCP (Rotating Tool Center Point) transformation. #RTCP instruction.
- Orient the tool perpendicular to the work plane (parallel to the third axis). #TOOL ORI instruction.
- Tool length compensation adaptation implicit in the program. #TLC instruction.

For clarity's sake, the following examples show three coordinate systems:

XYZ	Machine coordinate system.
X' Y' Z'	Part coordinate system.
X" Y" Z"	Tool coordinate system.

When no transformation has been made and the spindle is in the starting position, the three coordinate systems coincide.

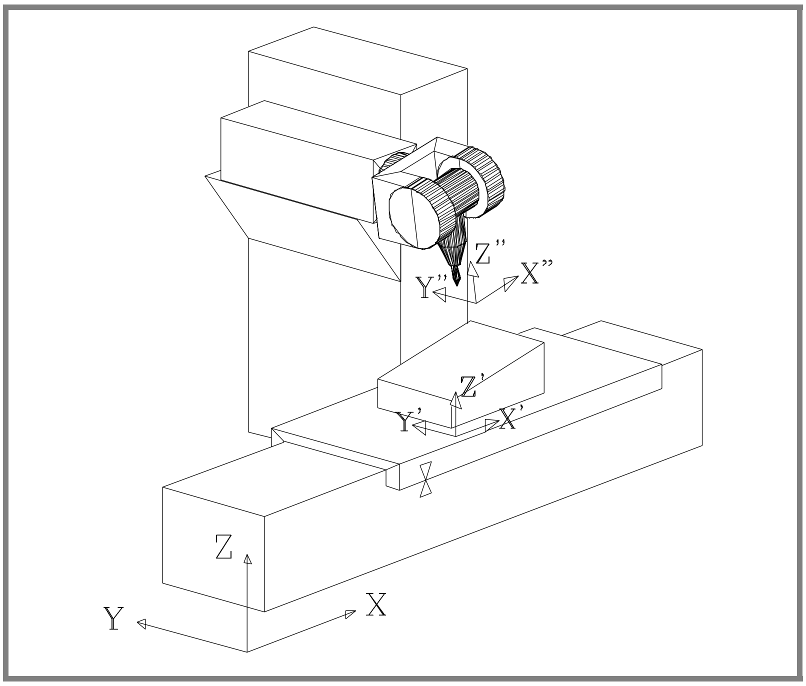


FAGOR

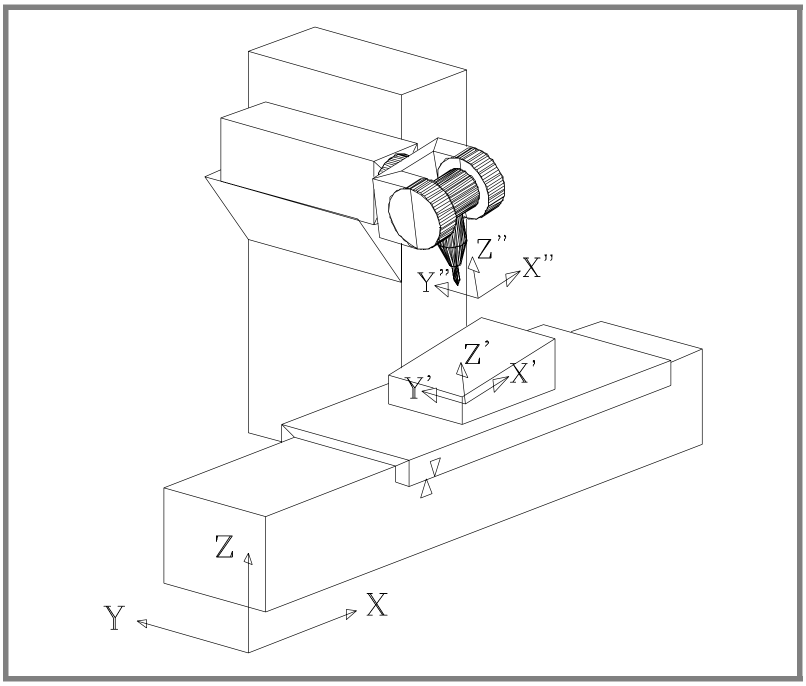
CNC 8070

(SOFT V02.0x)

When turning the spindle, the tool coordinate system (X" Y" Z") changes.



If besides this, a new machining coordinate system is selected (#CS instruction) or fixture coordinate system (#ACS instruction) the part coordinate system will also change (X' Y' Z').

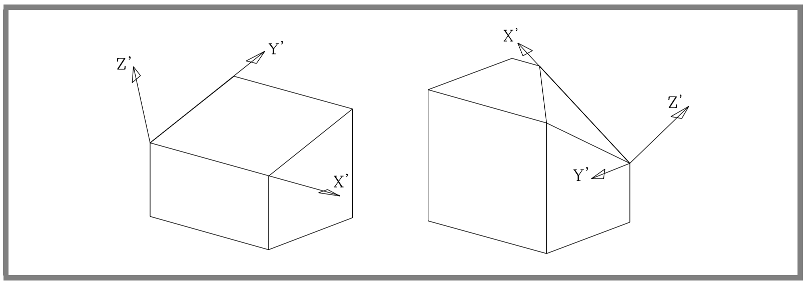


13.1 Movement in an incline plane

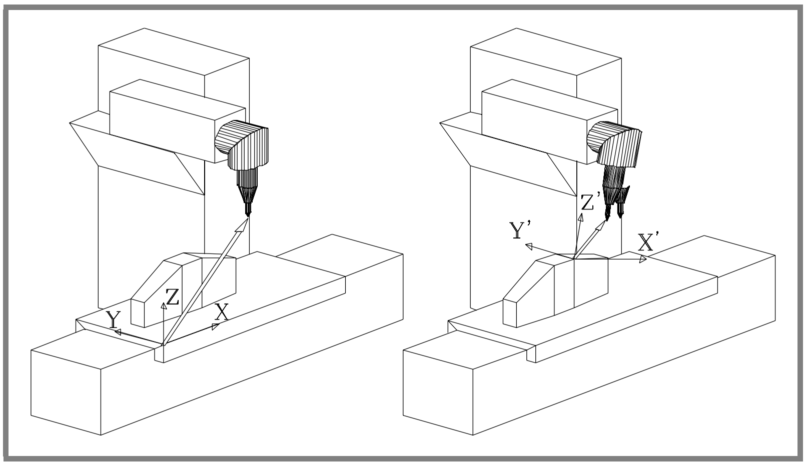
An incline plane is any plane in space resulting from the coordinate transformation of the XYZ axes.

Any plane in space may be selected to carry out machining operations in it.

To define the incline plane corresponding to the machining operations, use the #CS and #ACS instructions that are described later on in this chapter.

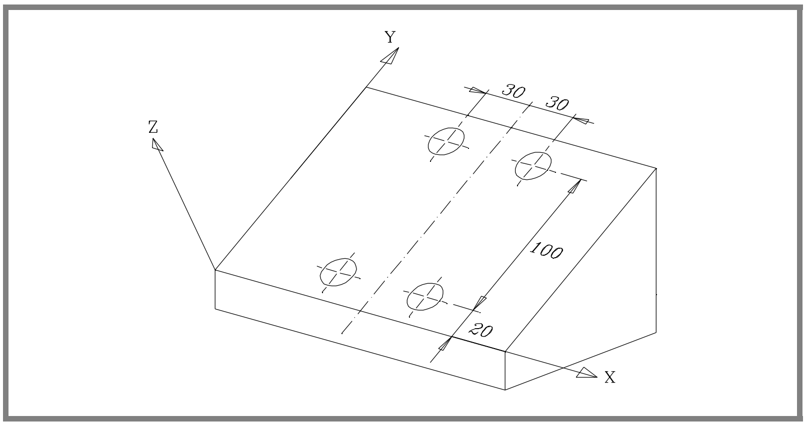


The new coordinates (right figure) are referred to the new part zero assuming that the tool is positioned perpendicular to the new plane.

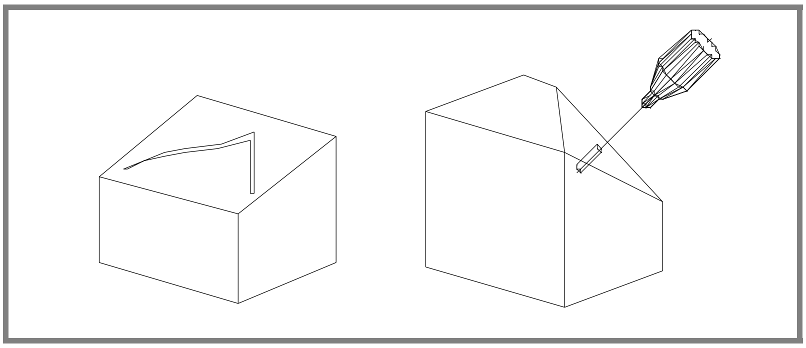


To place the tool at that position, use the #TOOL ORI instruction (***apartado 13.8***) or the kynetics related variables (***apartado 13.8***) that indicate the position that each rotary axis of the spindle head must occupy.

From this moment on, the programming and the X, Y movements are carried out along the selected plane and those of the Z axis will be perpendicular to it.



To orient the tool and work with it perpendicular to the plane, use the instruction #TOOL ORI that is described later on in this chapter.



13.2 Kinematics selection (#KIN ID)

The OEM may set up to 6 different kinematics for the machine. Each one of them indicates the type of spindle being used, its characteristics and dimensions.

To work with coordinate transformation, the kinematics being used must be indicated.

Usually, the OEM defines the kinematics number being used by default by means of general machine parameter KINID.

If there is only one and it has been set as the default kinematics, the (#KIN ID) instruction does not have to be programmed.

Format to activate a particular kinematics:

#KIN ID [n]   n: Kinematics number

Format to activate the kinematics that the OEM has defined as the default kinematics:

#KIN ID

Functions #RTCP, #TLC and #TOOL ORI must always be activated after selecting a kinematics.

The kinematics cannot be changed while function #RTCP or #TLC is active.

Example:

N50 #KIN ID[2]	(Activating kinematics Nr 2)
N60 #RTCP ON	(Activating RTCP with kinematics 2)
...	
N70 #RTCP OFF	(Turn RTCP off)
N80 M30	

13.3 Coordinate systems (#CS) (#ACS)

With the #CS instruction, up to 5 coordinate systems may be defined, stored, activated an deactivated.

With the #ACS instruction, up to 5 fixture coordinate systems may be defined, stored, activated and deactivated. It is used to compensate for workpiece inclination due to the fixtures used to secure them.

Both instructions use the same programming format and may be used independently or combined as indicated in the following section.

Format to define and store:

#CS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

#ACS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

Format to define, store and activate:

#CS ON [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

#ACS ON [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

Format to define and activate (without storing):

It may be used, until canceled, as any other coordinate system stored in memory.

Only one of them may be defined; to define another one, the previous one must be canceled.

#CS ON [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

#ACS ON [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

Format to deactivate and delete all the current #CS or #ACS and define and activate a new one:

#CS NEW [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

#ACS NEW [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

Format to deactivate and delete all the current #CS or #ACS and define and activate a new one (without storing):

#CS NEW [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

#ACS NEW [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]

Format to assume and store the current coordinates as a #CS or a #ACS:

#CS DEF ACT [n]

#ACS DEF ACT [n]

Format to activate one that has been stored:

#CS ON [n]

#ACS ON [n]

Format to activate the one stored last:

#CS ON

#ACS ON

Format to deactivate the one activated last:

#CS OFF

#ACS OFF

Format to deactivate all the activated #CS or #ACS:

#CS OFF ALL

#ACS OFF ALL

Meaning of the parameters that use both instructions:

[n]	Coordinate system number (1..5). Up to 5 different ones may be defined and stored to be activated at any time.
MODE m	Definition mode used (1..6). They are described next.
V1...V3	Components of the translation vector.
φ1...φ3	Rotation angles.
0/1	0/1 value, only in the 3, 4, 5 modes.

The #CS and #ACS are kept active after a Reset or an M30. When turning the CNC off, they are deactivated and all the information stored is deleted.

Since the coordinate origin is referred to the current part zero, it could happen that when activating a #CS or #ACS previously stored, the coordinate origin of the plane is not the desired one.

This happens if the part zero is modified between the definition and the application of the #CS or #ACS.

While being a #CS or #ACS activated, new part zeros may be preset in the plane. These values are valid only until the #CS or #ACS is deactivated.

Several #ACS and #CS coordinate systems may be combined. When activating a new one, it is added to the current coordinate system (see ***apartado 13.4****).*

It is recommended to start the program with #CS NEW or #ACS NEW to avoid undesired planes. This happens, for example, after interrupting the program and resuming execution.

Programming example:

```
#CS NEW [3] [MODE 1,2,15,5,2,3,4.5]
    (It deletes the current CS)
    (It defines it and stores it as CS3)
#CS DEF [2] [MODE 1,P1,15,5,2,3,4.5]
    (It defines it and stores it as CS2)
#CS DEF [5] [MODE 2,0,1,2,0,30,30]
    (It defines it and stores it as CS5)
#CS ON
    (It activates the CS programmed last, the CS5)
#CS OFF
    (It cancels the CS5)
#CS ON [3]
    (It activates the CS3)
#CS DEF [2] [MODE 1,1,1.2,1.3,0,0,33]
    (It redefines the stored CS2, the CS3 stays active)
M30
```

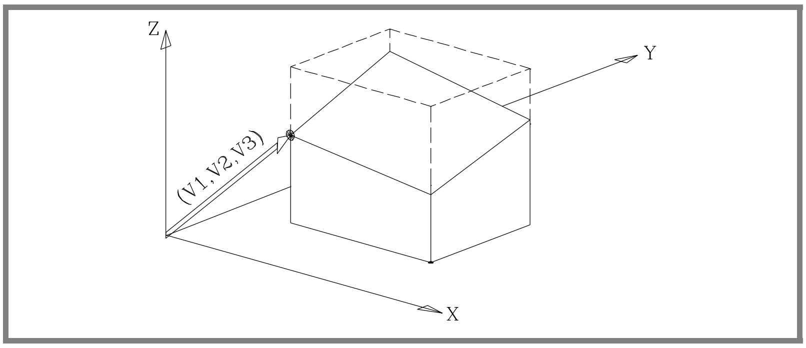
13.3.1 Coordinate system definition MODE 1

#CS DEF [n] [MODE 1, V1, V2, V3, φ1, φ2, φ3]

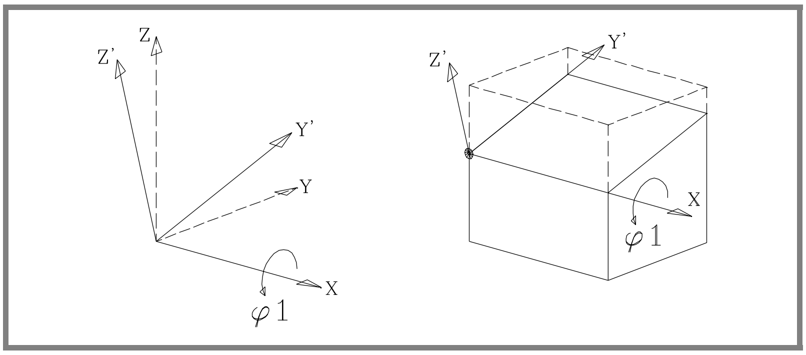
#ACS DEF [n] [MODE 1, V1, V2, V3, φ1, φ2, φ3]

It defines the incline plane resulting from rotating the amounts indicated in φ1, φ2, φ3 respectively first around the 1st axis, then around the 2nd axis and finally around the 3rd axis.

V1, V2, V3 — Define the coordinate origin of the incline plane with respect to the current part zero.

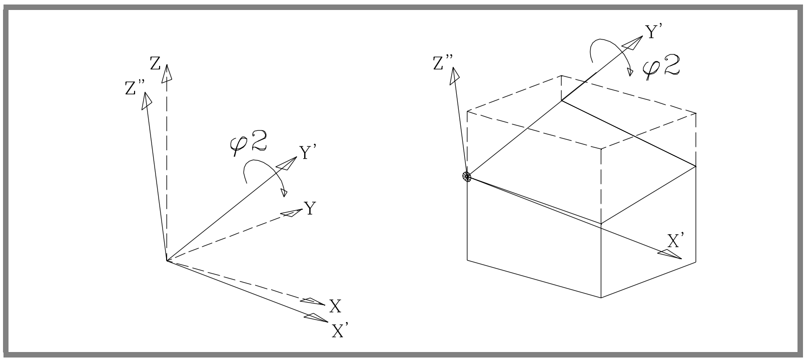


φ1, φ2, φ3 — Define the incline plane resulting from having rotated first around the 1st axis (X), the amount indicated by φ1.



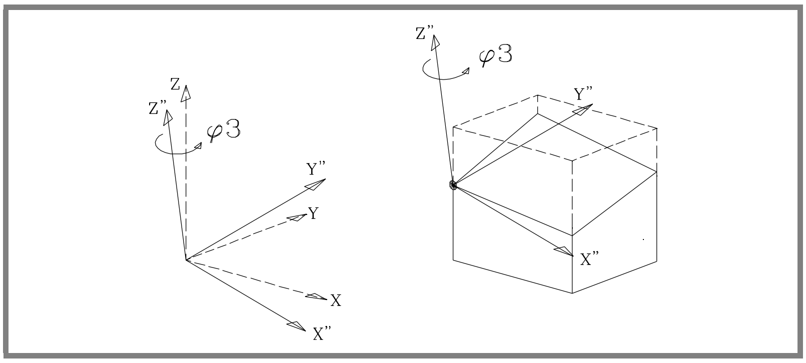
In the figure, the new coordinate system resulting from this transformation is called X Y' Z' because the Y, Z axes have been rotated.

Then, rotate around the 2nd axis (Y'), the φ2 amount.



In the figure, the new coordinate system resulting from this transformation is called X' Y' Z'' because the X, Z axes have been rotated.

And last, rotate around the Z'' axis the amount indicated by φ3.



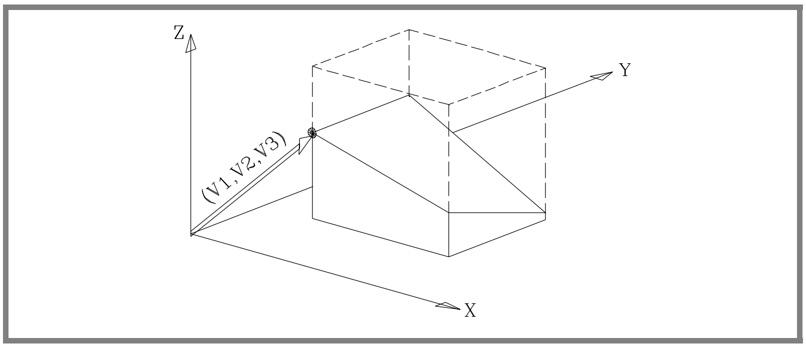
13.3.2 Coordinate system definition MODE 2

#CS DEF [n] [MODE 2, V1, V2, V3, φ1, φ2, φ3]

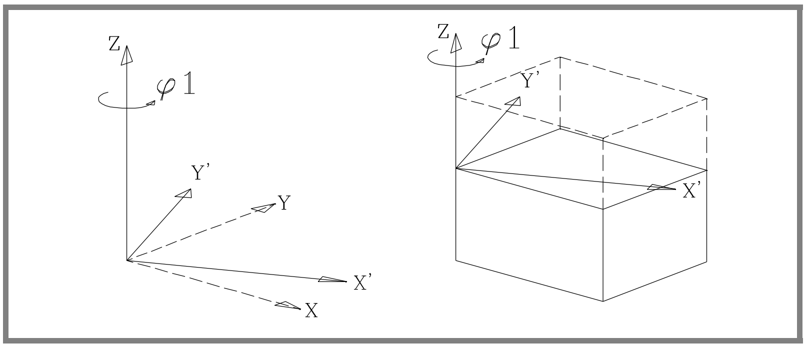
#ACS DEF [n] [MODE 2, V1, V2, V3, φ1, φ2, φ3]

They define, in spherical coordinates, the incline plane resulting from having rotated around the 3rd axis, then around the 2nd one and then again around the 3rd axis the amounts indicated by φ1, φ2, φ3 respectively.

V1, V2, V3 — Define the coordinate origin of the incline plane with respect to the current part zero.

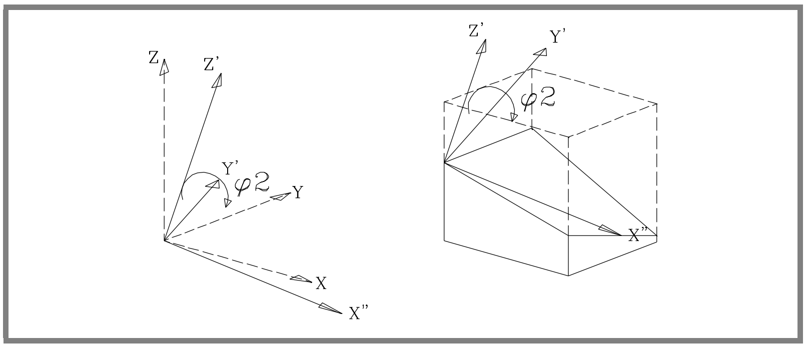


φ1, φ2, φ3 — Define the incline plane resulting from having rotated first around the 3rd axis (Z), the amount indicated by φ1.



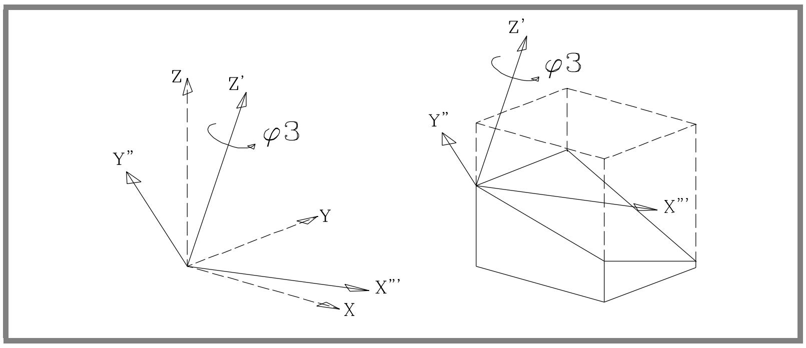
In the figure, the new coordinate system resulting from this transformation is called X' Y' Z because the X, Y axes have been rotated.

Then, it must be rotated around the Y' axis the φ2 amount.



In the figure, the new coordinate system resulting from this transformation is called X'' Y' Z' because the X, Z axes have been rotated.

And last, rotate around the Z' axis the amount indicated by φ3.



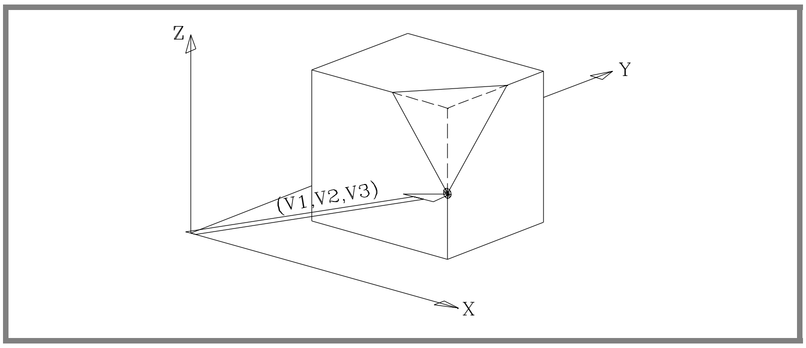
13.3.3 Coordinate system definition MODE 3

#CS DEF [n] [MODE 3, V1, V2, V3, φ1, φ2, φ3, 0/1]

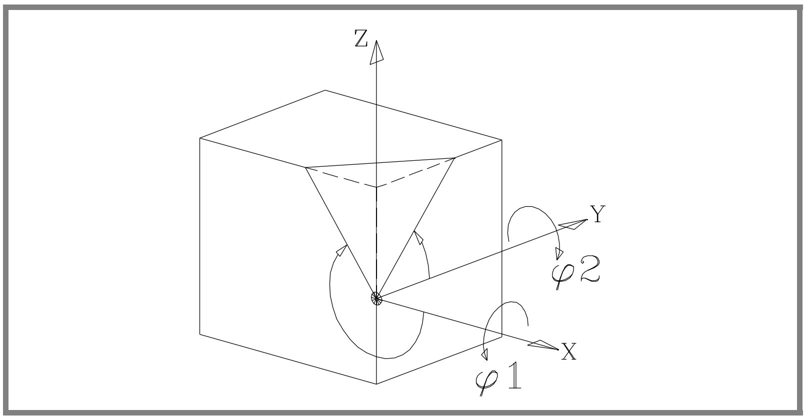
#ACS DEF [n] [MODE 3, V1, V2, V3, φ1, φ2, φ3, 0/1]

The incline plane is defined with the angles it forms with respect to the 1st and 2nd axes (X Y) of the machine's coordinate system.

V1, V2, V3 — Define the coordinate origin of the incline plane with respect to the current part zero.



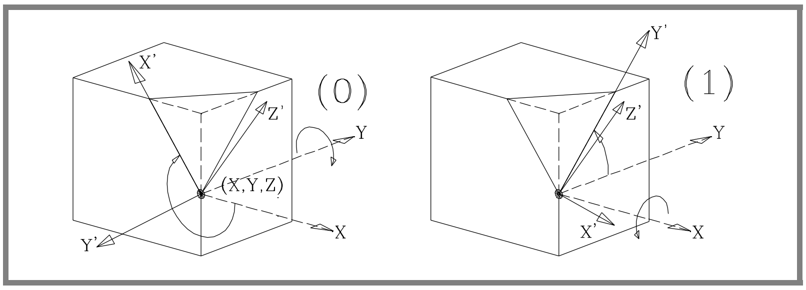
φ1, φ2 — Define the angles that the incline plane forms with the 1st and 2nd axes (X Y) of the machine's coordinate system.



0/1 — Defines which of the axes of the new plane (X' Y') is aligned with the edge.

If <0> the X' axis and if <1> the Y' axis.

If not programmed, it assumes <0>.



φ3 — Permits defining and applying a coordinate rotation in the new cartesian plane X' Y'.

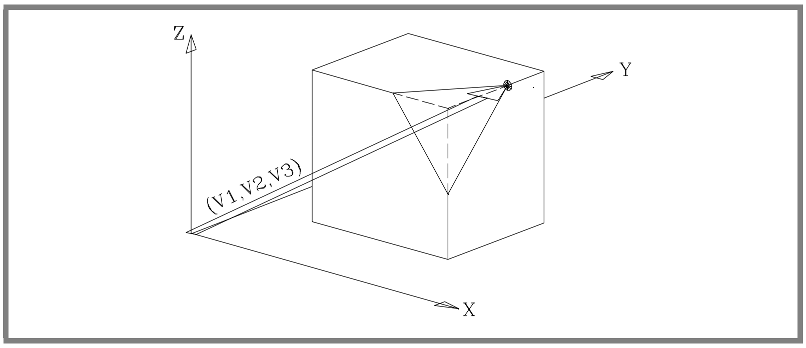
13.3.4 Coordinate system definition MODE 4

#CS DEF [n] [MODE 4, V1, V2, V3, φ1, φ2, φ3, 0/1]

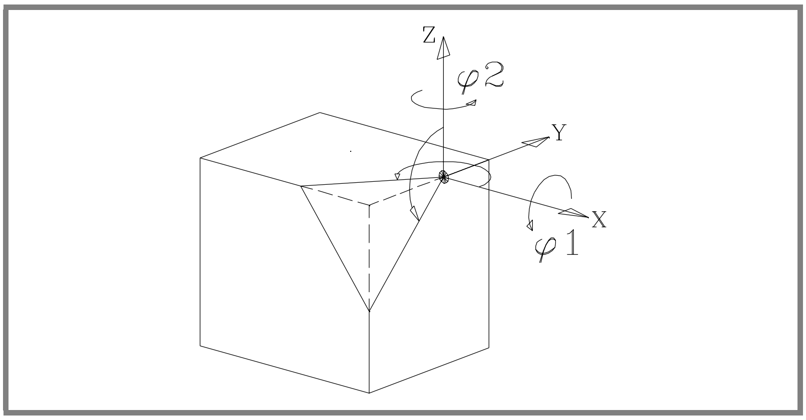
#ACS DEF [n] [MODE 4, V1, V2, V3, φ1, φ2, φ3, 0/1]

The incline plane is defined with the angles it forms with respect to the 1st and 3rd axes (X Z) of the machine's coordinate system.

V1, V2, V3 — Define the coordinate origin of the incline plane with respect to the current part zero.



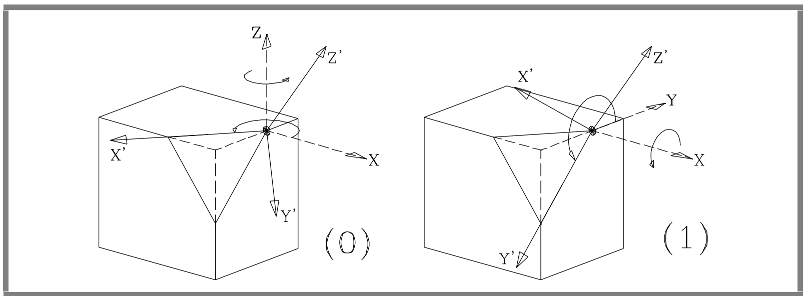
φ1, φ2 — Define the angles that the incline plane forms with the 1st and 3rd axes (X Z) of the machine's coordinate system.



0/1 — Defines which of the axes of the new plane (X' Y') is aligned with the edge.

If <0> the X' axis and if <1> the Y' axis.

If not programmed, it assumes <0>.



φ3 — Permits defining and applying a coordinate rotation in the new cartesian plane X' Y'.

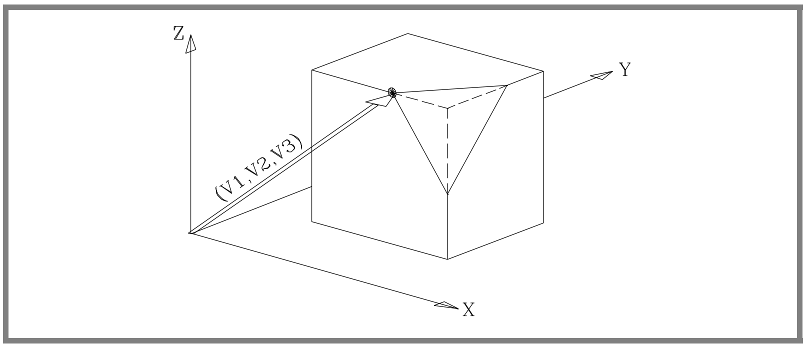
13.3.5 Coordinate system definition MODE5

#CS DEF [n] [MODE 5, V1, V2, V3, φ1, φ2, φ3, 0/1]

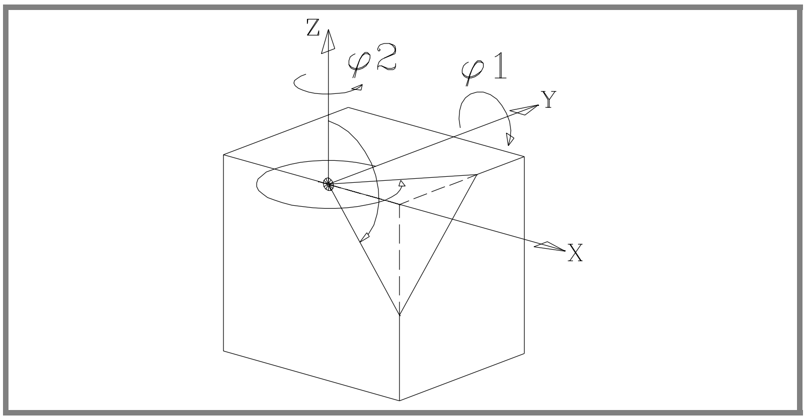
#ACS DEF [n] [MODE 5, V1, V2, V3, φ1, φ2, φ3, 0/1]

The incline plane is defined with the angles it forms with respect to the 2nd and 3rd axes (Y Z) of the machine's coordinate system.

V1, V2, V3 — Define the coordinate origin of the incline plane with respect to the current part zero.



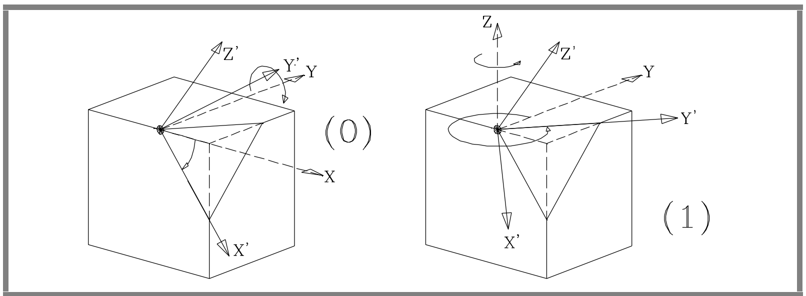
φ1, φ2 — Define the angles that the incline plane forms with the 2nd and 3rd axes (Y Z) of the machine's coordinate system.



0/1 — Defines which of the axes of the new plane (X' Y') is aligned with the edge.

If <0> the X' axis and if <1> the Y' axis.

If not programmed, it assumes <0>.



φ3 — Permits defining and applying a coordinate rotation in the new cartesian plane X' Y'.

13.3.6 Coordinate system definition MODE6

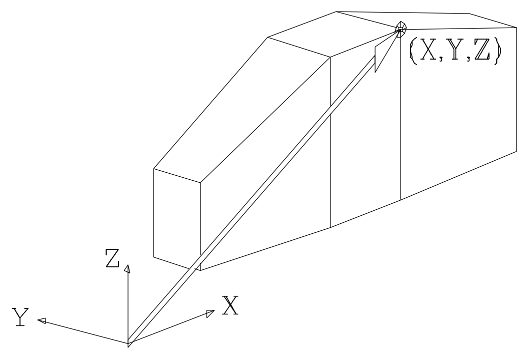
In order to use this definition, while setting up the machine, the tool position when it is parallel to the Z axis of the machine must be set as the spindle's rest position.

#CS DEF [n] [MODE 6, V1, V2, V3, φ1]

#ACS DEF [n] [MODE 6, V1, V2, V3, φ1]

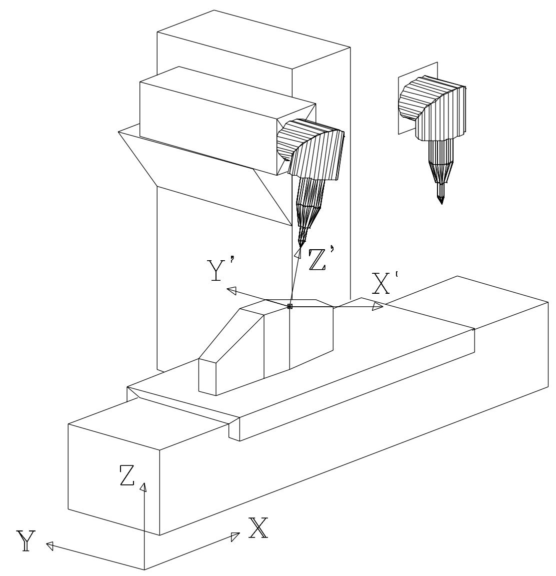
It defines a new work plane (incline plane) perpendicular to the direction of the tool.

V1, V2, V3	Define the coordinate origin of the incline plane with respect to the current part zero.



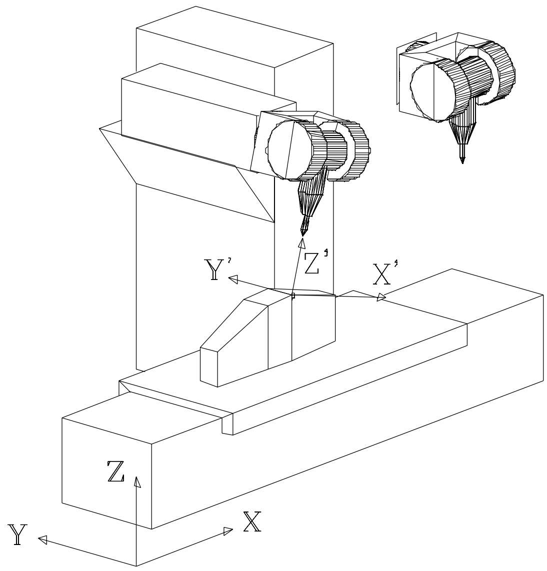
The new work plane assumes the orientation of the tool's coordinate system.

Example:



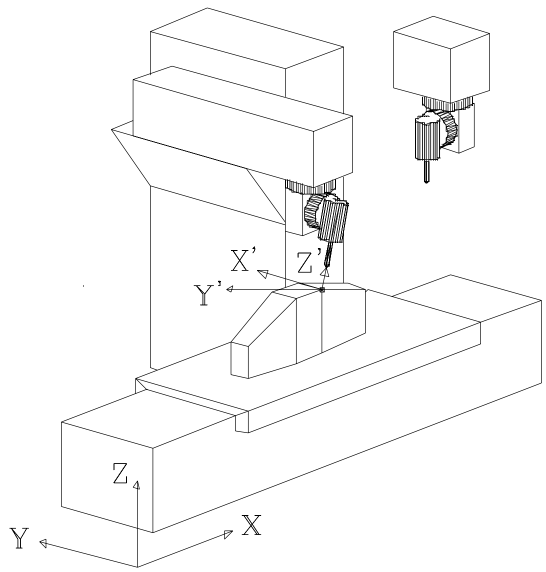
On this machine, only the main rotary axis has rotated. See the rest position of the spindle at the top right side.

Example:



On this machine, only the main rotary axis has rotated. See the rest position of the spindle at the top right side.

Example:



On the contrary, on this machine, to achieve the same tool orientation, both the main and secondary rotary axes have rotated. See the rest position of the spindle at the top right side.

The main axis has rotated 90º and, therefore, the X' Y' axes of the plane are rotated 90º.

φ1	Permits defining and applying a coordinate rotation in the new cartesian plane X' Y'.

If on the last machine, we wanted to orient the X', Y' axes like in the other two cases, we would have to program the following:

```
#CS DEF [n] [MODE 6, V1, V2, V3, -90]
```

13.4 How to combine several coordinate systems

Several #ACS and #CS coordinate systems may be combined to construct new coordinate systems.

For example, the #ACS inclination generated by a fixture on the part may be combined with the #CS coordinate system that defines the incline plane of the part to be machined.

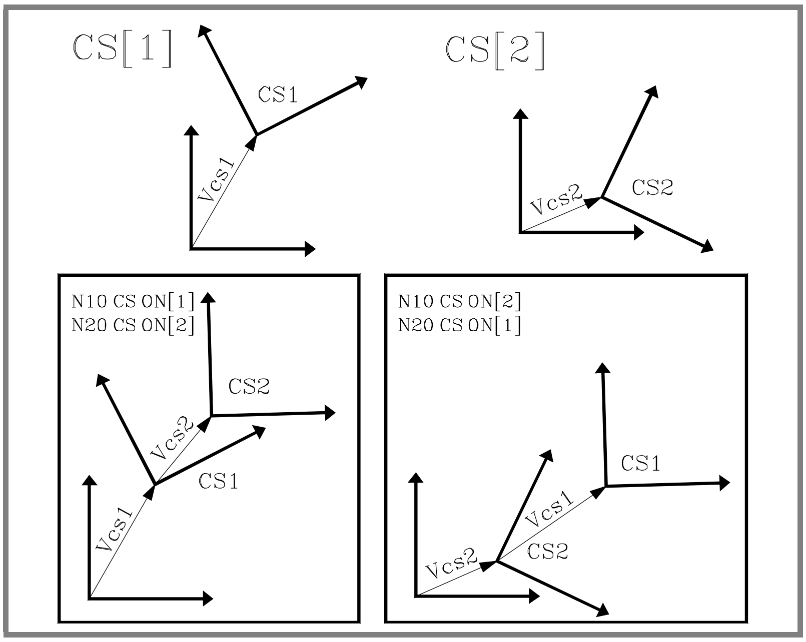
Up to ten #ACS or #CS coordinate systems may be combined. The CNC acts as follows:

First, it checks the #ACS and applies them sequentially in the programmed order, resulting in an #ACS transformation.

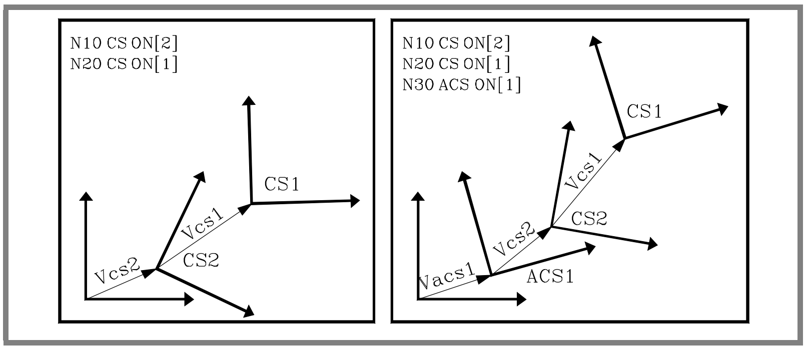
Then, it checks the #CS and applies them sequentially in the programmed order, resulting in a #CS transformation.

And last, it applies the resulting #CS over the resulting #ACS to obtain the new coordinate system.

The result of the combination depends on the order they are activated as may be observed in the figure below.



Every time a #ACS or #CS is activated, the resulting coordinate system is recalculated as can be observed in the figure below.



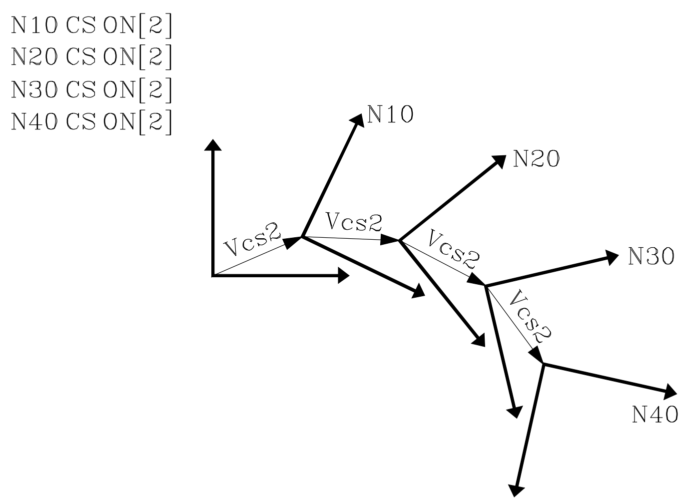
The #ACS OFF and #CS OFF instructions deactivate the last #ACS or #CS activated, respectively.

Example:

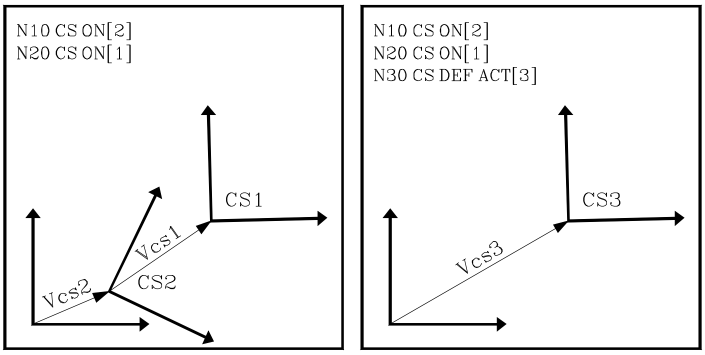
```
N100 #CS ON [1]          (CS[1])
N110 #ACS ON [2]         (ACS[2] + CS[1])
N120 #ACS ON [1]         (ACS[2] + ACS[1] + CS[1])
N130 #CS ON [2]          (ACS[2] + ACS[1] + CS[1] + CS[2])
N140 #ACS OFF            (ACS[2] + CS[1] + CS[2])
N140 #CS OFF             (ACS[2] + CS[1])
N150 #CS ON [3]          (ACS[2] + CS[1] + CS[3])
N160 #ACS OFF ALL        (CS[1] + CS[3])
N170 #CS OFF ALL
M30
```

A #ACS or #CS coordinate system may be activated several time.

Example:



The figure below shows an example of the instruction #CS DEF ACT [n] to assume and store the current coordinate system as a #CS.

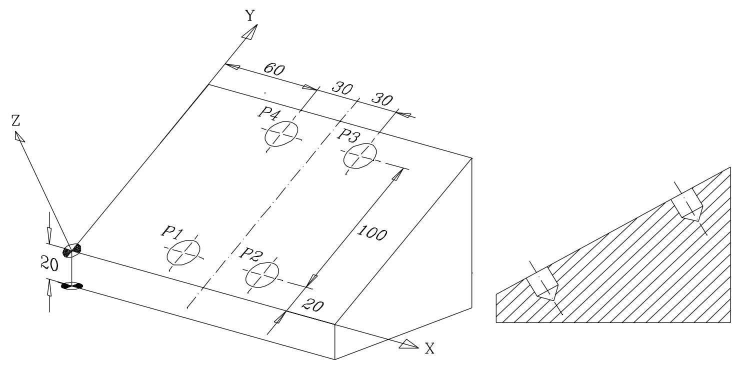


13.5 Tool perpendicular to the plane (#TOOL ORI)

The #TOOL ORI instruction is used to position the tool perpendicular to the work plane.

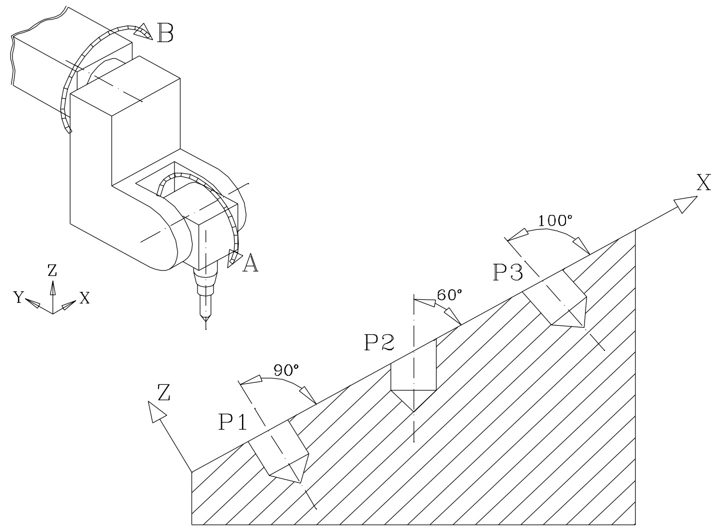
After executing the #TOOL ORI instruction, the tool is positioned perpendicular to the plane, parallel to the 3rd axis of the active coordinate system at the first motion programmed next.

Example:



#CS ON [1] [MODE 1, 0, 0, 20, 30, 0, 0, 0]	
	(Defines the incline plane)
#TOOL ORI	(Perpendicular tool, request)
G90 G0 X60 Y20 Z3	(Position at point P1)
	(The spindle orients perpendicular to the plane during this positioning move)
G1 G91 Z-13 F1000	(Drilling)
G0 Z13	(Withdrawal)
G0 G90 X120 Y20	(Position at point P2)
G1 G91 Z-13 F1000	(Drilling)
G0 Z13	(Withdrawal)
G0 G90 X120 Y120	(Position at point P3)
G1 G91 Z-13 F1000	(Drilling)
G0 Z13	(Withdrawal)
G0 G90 X60 Y120	(Position at point P4)
G1 G91 Z-13 F1000	(Drilling)
G0 Z13	(Withdrawal)
M30	

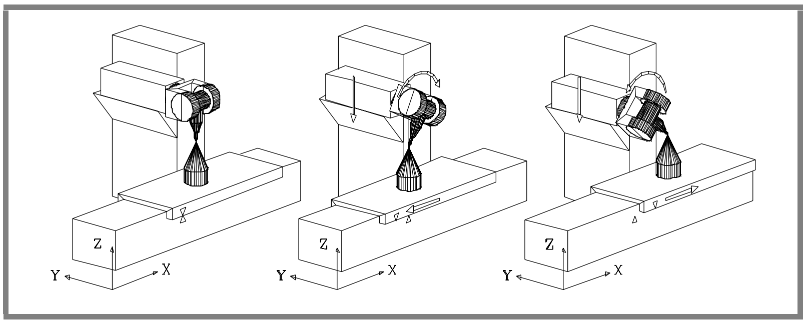
The following example shows how to drill three holes with different inclination in the same plane:



#CS ON [1] [MODE .....]	(Defines the incline plane)
#TOOL ORI	(Perpendicular tool, request)
G0 <P1>	(Movement to point P1)
(The spindle orients perpendicular to the plane during this positioning move)	
G1 G91 Z-10 F1000	(Drilling)
G0 Z10	(Withdrawal)
G0 <P2>	(Movement to point P2)
G90 B0	(Orients the tool with machine coordinates)
#MCS ON	(Programming in machine coordinates)
G1 G91 Z-10 F1000	(Drilling)
G0 Z10	(Withdrawal)
#MCS OFF	(End of programming in machine coordinates. It recovers plane coordinates)
G0 <P3>	(Movement to point P3)
G90 B-100	(Positions the tool at 100º)
#CS OFF	
#CS ON [2] [MODE6 .....]	(Defines the incline plane perpendicular to the tool)
G1 G91 Z-10 F1000	(Drilling)
G0 Z30	(Withdrawal)
#CS OFF	
M30	

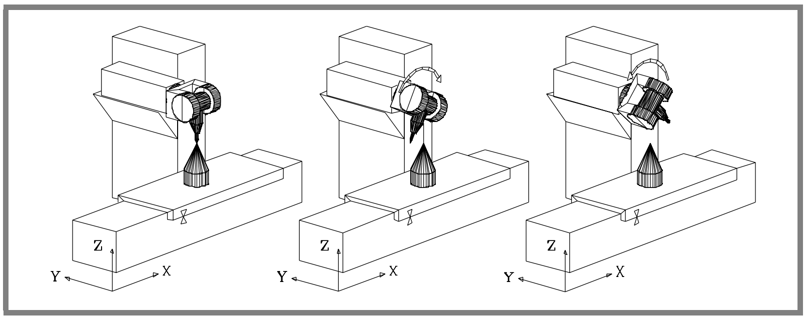
13.6 Using RTCP (Rotating Tool Center Point)

The orientation of the tool may be changed without modifying the position occupied by its tip on the part. The RTCP represents a length compensation in space.



Obviously, the CNC must move several axes in order to maintain the tool tip position at all times.

The figure below shows what happens when turning the spindle when NOT working with RTCP.



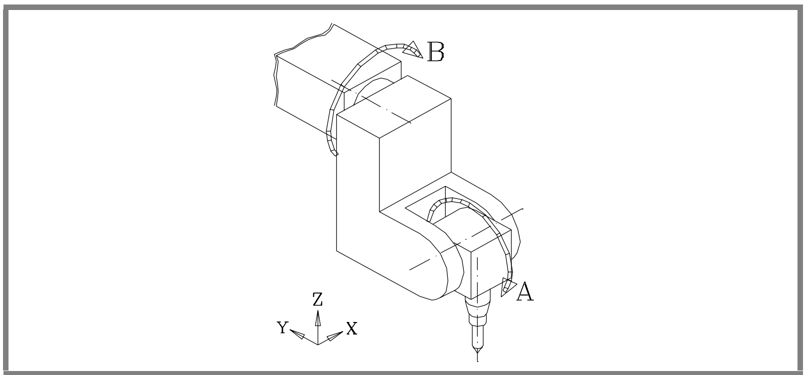
Use the following instructions for working with RTCP transformation:

#RTCP ON	Activate RTCP transformation
#RTCP OFF	Cancel RTCP transformation

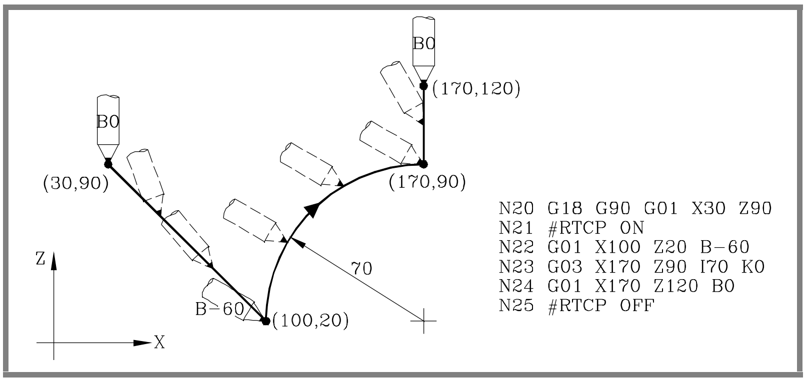
Once RTCP transformation is active, spindle positioning may be combined with linear and circular interpolations.

The RTCP function cannot be selected while the TLC function is active.

The following examples use a double swivel rectangular spindle head:



Example a) Circular interpolation maintaining tool orientation



Block N20 selects the ZX plane (G18) and positions the tool at the starting point (30,90).

Block N21 turns RTCP on.

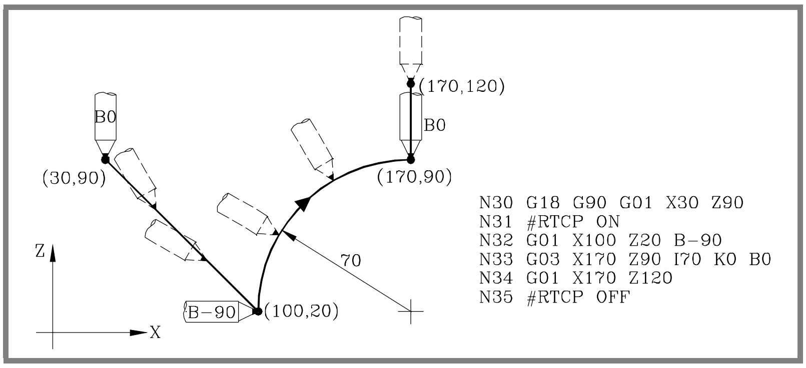
Block N22 contains a movement to point (100,20) and a tool orientation from 0º to -60º. The CNC interpolates the X, Z and B axes in such a way that the tool is being oriented along the movement.

Block N23 makes a circular interpolation to point (170,90) maintaining the same tool orientation along the whole path.

Block N24 contains a movement to point (170,120) and a tool orientation from -60º to 0º. The CNC interpolates the X, Z and B axes in such a way that the tool is being oriented along the movement.

Block N25 turns RTCP off.

Example b) Circular interpolation with tool perpendicular to its path



Block N30 selects the ZX plane (G18) and positions the tool at the starting point (30,90).

Block N31 turns RTCP on.

Block N32 contains a movement to point (100,20) and a tool orientation from 0º to -90º. The CNC interpolates the X, Z and B axes in such a way that the tool is being oriented along the movement.

Block N33 contains a circular interpolation to point (170,90) maintaining the tool perpendicular to the path at all times.

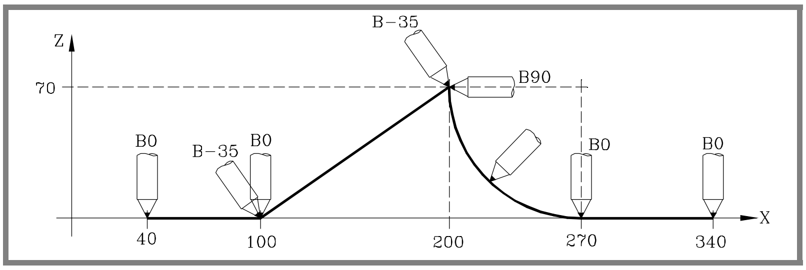
At the starting point, it is oriented to -90º and at the endpoint, it must end be 0º.

The CNC interpolates the X, Z and B axes maintaining the tool perpendicular to its path at all times.

Block N34 moves the tool to point (170,120) maintaining the orientation of 0º.

Block N35 cancels RTCP.

Example c) Machining a profile



G18 G90	Selects the ZX plane (G18)
#RTCP ON	It activates RTCP transformation
G01 X40 Z0 B0 F1000	Positions the tool at (40,0) oriented to (0º)
X100	Movement to (100,0) with tool oriented to (0º)
B-35	Orients the tool to (-35º)
X200 Z70	Movement to (200,70) with tool oriented to (-35º)
B90	Orients the tool to (90º)
G02 X270 Z0 R70 B0	Circular interpolation to (270,0) maintaining the tool perpendicular to the path.
G01 X340	Movement to (340.0) with tool oriented to (0º)
#RTCP OFF	It cancels RTCP transformation

13.6.1 Considerations about the RTCP function

In order to work with RTCP transformation, the X, Y, Z axes must be defined, they must form a trihedron and be linear. X, Y and Z may be GANTRY axes.

The RTCP transformation is kept active even after executing M02 or M30, after an Emergency or a Reset and after turning the CNC off.

While RTCP is on, the following operations are possible:

- Zero offsets G54-G59, G159.
- Presetting (G92).
- Movements in continuous / incremental jog and handwheel.

Home search (G74) is not allowed if the RTCP transformation is active.

When working with incline planes and RTCP transformation, it is recommended to follow this programming order (sequence):

#RTCP ON	(Turn RTCP on)
#CS ON	(Define the incline plane)
#TOOL ORI	(Tool perpendicular to the plane)
G	(Start machining)
	(End machining)
#CS OFF	(Cancel the incline plane)
#RTCP OFF	(Turn RTCP off)
M30	(End of part-program)

RTCP should be turned on first because it allows orienting the tool without modifying the tool tip position.

13.7 Tool length compensation (#TLC)

It must be used with CAD-CAM generated programs and the tool to be used does not have the same dimensions.

The #TLC compensates for the length difference; but it does not compensate for the radius difference.

CAD-CAM programs take the tool length into consideration and generate the coordinates for the tool base.

When using the #TLC function (Tool Length Compensation) the CNC compensates the length difference between both tools, the actual (real) one and the theoretical (calculated) one.

To work with tool length compensation (#TLC), use the following instructions:

#TLC ON [n]	Turn TLC on. n: Tool length difference (real - theoretical).
#TLC OFF	Turn TLC off.

The TLC function cannot be selected while the RTCP function is active.

Examples:

N10 #TLC ON [1.5]	(Turn TLC on with a tool that is 1.5mm. longer)
N100 #TLC OFF	(Turn TLC off)
N200 #TLC ON [-2]	(Turn TLC on with a tool that is 2mm. shorter)
N300 #TLC OFF	(Turn TLC off)
N200 M30	

13.8 Kinematics related variables

These variables indicate the position occupied by the rotary axes of the spindle head and the one (target) they must occupy in order to position the tool perpendicular to the defined plane.

They are very useful when the spindle is not fully motorized (mono-rotary or manual spindles).

Variables that indicate the position of the rotary axes. They can be read and written (R/W) and are given in degrees.

(V.)G.POSROTF	Main rotary axis position.
(V.)G.POSROTS	Secondary rotary axis position.

Variables that indicate the position the rotary axes must occupy in order for the tool to be perpendicular to the define work plane. They are read-only (R) and are given in degrees.

Here are the two possible solutions for swivel spindles:

The one involving the shortest movement of the main rotary axis with respect to the zero position.

(V.)G.TOOLORIF1	Position of the main rotary axis in order to position perpendicular to the incline plane.
(V.)G.TOOLORIS1	Position of the secondary rotary axis in order to position perpendicular to the incline plane.

The one involving the longest movement of the main rotary axis with respect to the zero position.

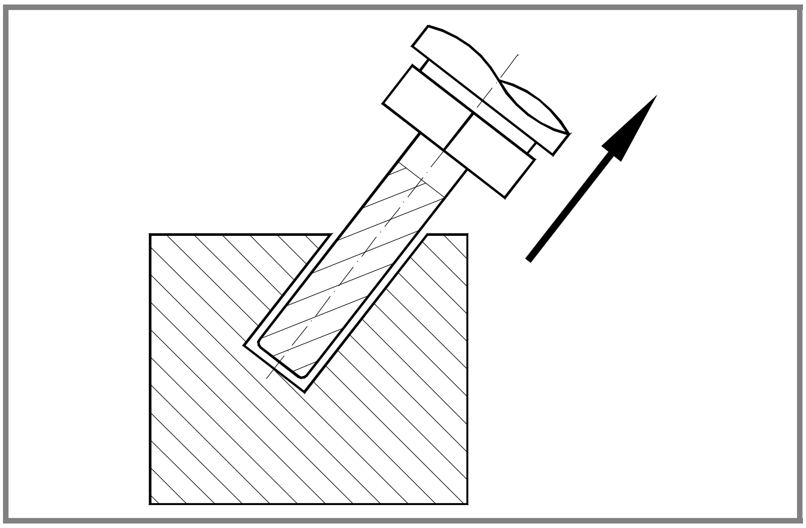
(V.)G.TOOLORIF2	Position of the main rotary axis in order to position perpendicular to the incline plane.
(V.)G.TOOLORIS2	Position of the secondary rotary axis in order to position perpendicular to the incline plane.

The CNC updates the (V.)G.TOOLORI* variables every time a new plane is selected using the instructions #CS or #ACS.

13.9 How to withdraw the tool when losing the plane

If the CNC is turned off and back on while working with kinematics, the work plane that was selected gets lost.

If the tool is inside the part, proceed as follows to withdraw it:



Use the #KIN ID [n] instruction to select the kinematics that was being used.

Use the coordinate system definition MODE6 so the CNC selects a plane perpendicular to the direction of the tool as the work plane.

#CS ON [n] [MODE 6, 0, 0, 0, 0]

Move the tool along the longitudinal axis until it is away from the part.

This movement may be made in jog mode or by program, for example, G0 G91 Z20.

CNC VARIABLES

14

14.1 Understanding the description of the variables

PRG / PLC / INT – Access to variables

The internal CNC variables may be accessed from the part program, MDI, PLC and from any application (for example FGUIM). This chapter uses the following abbreviations to indicate where these variables may be accessed from:

PRG	From the part-program or MDI.
PLC	from the PLC.
INT	From any application (interface). For example FGUIM.

Each variable must indicate whether it can only be read (R) or read and written (R/W).

LIN / ROT / CAB / ANA / SER – Variables related to the axes and drives

For variables associated with the axes, they indicate the type of axis and the drive associated with the variable.

Lin	Linear axis
Rot	Rotary axis
Spd	Spindle
Ana	Analog drive
Ser	Sercos drive.

When using Sercos drives, it will indicate whether the variable is valid or not when the drive works in position mode (P) or velocity mode (S) or in both (P/S).

EXEC – Access to the variable during preparation or execution

The CNC reads several blocks ahead (preparation) of the one being executed in order to calculate in advance the path to follow. This prior reading is known as "block preparation".

Certain variables are accessed during block preparation whereas others are evaluated when they are executed. The latter interrupt block preparation.

(V.)G.PRGF	Feedrate by program in G94. Evaluated during preparation (before executed).
(V.)G.FREAL	Actual (real) CNC feedrate. Evaluated when being executed.

For variables accessed from PRG, the "Exec" column indicates whether the variable is read or written during block preparation or when being executed.

Yes	When being executed. It interrupts block preparation.
No	During preparation.

Accessing the variables from PLC or INT always interrupts block preparation.

Interrupting block preparation may result in compensated paths different from the ones programmed, undesired joints when working with small sections, interruptions when working with look-ahead, jerky axis movement, etc.

Use the `#FLUSH` instruction to force the evaluation of a variable when it is being executed.

Sync / Asyn – Synchronous or asynchronous access from the PLC.

PLC access to the variable, both for reading and writing, may be either synchronous or asynchronous. A synchronous access is resolved immediately whereas an asynchronous access takes several PLC cycles to resolve.

The asynchronous variables are:

- The tool variables will be read asynchronously when the tool is neither the active one nor in the magazine.
- The tool variables will be written asynchronously whether the tool is the active one or not.

Example of how to access asynchronous variables

Reading of the radius value of offset ·1· of tool ·9· when it is not in the tool magazine.

```
<condition> AND NOT M11 = CNCRD (TM.TORT.[9][1], R11, M11)
```

The `M11` mark is set to "1" at the beginning of the operation and it keeps its value until the end of the operation.

```
DFD M11 AND CPS R11 EQ 3 = ···
```

It waits for the consultation to end before evaluating the data.

Examples of how to access synchronous variables:

```
<condition> = CNCRD (G.FREAL, R12, M12)
CPS R12 GT 2000 = ···
```

There is no need to wait for consulting the data because the synchronous variables are resolved immediately.

```
<condition> = CNCWR (R13, PLC.TIMER, M13)
```

It resets the clock enabled by the PLC with the value contained in register R13.

14.1.1 Access to numeric values from the PLC

When accessing from the PLC numeric values that may have decimals, it must be borne in mind that the values are given as follows.

Coordinates

They will be given in ten-thousandths if they are in mm or hundred-thousandths if they are inches.

For 1 mm.	the reading is 10000.
For 1 inch	the reading is 100000.
For 1 degree	the reading is 10000.

Feedrate of the axes

They will be given in ten-thousandths if they are in mm or hundred-thousandths if they are inches.

For 1 mm/min.	the reading is 10000.
For 1 inch/min.	the reading is 100000.

Spindle speed

They will be given in ten-thousandths.

With G97 for 1 rpm.	the reading is 10000.
With G96, for 1 m/min.	the reading is 10000.
With G96, for 1 foot/min.	the reading is 10000.
With G196 for 1 rpm.	the reading is 10000.
With M19, for 1 º/min.	the reading is 10000.

Percentages

The real value will be given in tenths or in hundredths depending on the variable. If not indicated otherwise, it will read the actual value. If not so, it will indicate if the variable will be read in tenths (x10) or in hundredths (x100).

For 1%	the reading is 1.
For 1%	(x10) the reading is 10.
For 1%	(x100) the reading is 100.

Time

They will be given in thousandths.

For 1 second	the reading is 1000.

Voltage

The variables associated with the machine parameter table return the actual value (in millivolts). For the rest of the variables (in volts), the reading will appear in ten-thousandths.

For 1 Volt	the reading is 10000.

14.1.2 Accessing the variables in a single-channel system

Name of the variables

The generic mnemonic associated with the variables is written as follows.

```
(V.){prefix}.{variable}
```

The mnemonic associated with each variable starts with a (V.). Use these characters (except the parenthesis) when accessing from PRG; but do not use them when accessing from INT and PLC.

Mnemonic	PRG	PLC / INT
(V.)MPG.NAXIS	V.MPG.NAXIS	MPG.NAXIS

Axis and spindle parameters

Axis and spindle variables are identified with the prefix –A.–. When these variables refer to a spindle, they may also be accessed with the prefix –SP.–.

```
(V.)A.{variable}.{axis/spindle}
(V.)SP.{variable}.{spindle}
```

The variables of the machine parameters with –MPA.– prefix can also be accessed using the –SP– prefix when referring to a spindle.

```
(V.)MPA.{variable}.{axis/spindle}
(V.)SP.{variable}.{spindle}
```

In these variables one must indicate which axis or spindle they refer to. The axis may be referred to by its name or logic number; the spindle may be referred to by its name, logic number or index in the spindle system.

Identifying the axes and the spindles.

In variables with the prefix –A.– and –MPA.–, the axes and the spindles are identified with their logic number.

- For the axes, the logic number sets the order `AXISNAME`.
- For spindles, the logic number is given by the sum of `NAXIS` + orden `SPDLNAME`.

In variables with the prefix –SP.–, the spindles are identified with their index in the system, according to the order `SPDLNAME`.

Variables of the master spindle

They are special variables that may be used to access the data of the master spindle without knowing its name or number. They are meant for displaying data and programming cycles.

The variables are identified with the prefix –SP.– but without indicating the spindle.

`(V.)SP.{var}`	Variable of the master spindle.

Mnemonic	Axis	Spindle	Master spindle
(V.)A.POS.Xn	V.A.POS.X V.A.POS.1	V.A.POS.S V.SP.POS.S V.A.POS.6 V.SP.POS.2	V.SP.POS
(V.)MPA.AXISTYPE.Xn	V.MPA.AXISTYPE.X V.MPA.AXISTYPE.1	V.MPA.AXISTYPE.S V.SP.AXISTYPE.S V.MPA.AXISTYPE.6 V.SP.AXISTYPE.2	V.SP.AXISTYPE

14.1.3 Accessing the variables of a single-channel system

Name of the variables

The generic mnemonic associated with the variables is written as follows.

```
(V.)[channel].{prefix}.{variable}
```

The mnemonic associated with each variable starts with a (V.). Use these characters (except the parenthesis) when accessing from PRG; but do not use them when accessing from INT and PLC.

For these variables, you must indicate the channel they belong to (the first channel is number 1 and "0" is not a valid number). The brackets must be programmed.

Mnemonic	PRG	PLC / INT
(V.)[n].G.FREAL	V.[1].G.FREAL	[1].G.FREAL

Programming the channel is optional. If no channel is indicated, it will assume the following:

PRG	Channel where it is being executed.
PLC	First channel or main channel.
INT	Active channel.

Axis and spindle parameters

Axis and spindle variables are identified with the prefix –A.–. When these variables refer to a spindle, they may also be accessed with the prefix –SP.–.

```
(V.)[n].A.{variable}.{axis/spindle}
(V.)[n].SP.{variable}.{spindle}
```

The variables of the machine parameters with –MPA.– prefix can also be accessed using the –SP– prefix when referring to a spindle.

```
(V.)MPA.{variable}.{axis/spindle}
(V.)SP.{variable}.{spindle}
```

In these variables one must indicate which axis or spindle they refer to. The axis may be referred to by its name or logic number; the spindle may be referred to by its name, logic number or the spindle system index or channel index.

Identifying the axes and the spindles.

In variables with the prefix –A.– and –MPA.–, the axes and the spindles are identified with their logic number.

- For the axes, the logic number sets the order `AXISNAME`.
- For spindles, the logic number is given by the sum of `NAXIS` + orden `SPDLNAME`.

In variables with the prefix –SP.–, the spindles are identified with their channel index or with their system index.

- When reading from the program interface or PLC, the spindle is identified with its system index according to the order `SPDLNAME`.
- When reading from the program interface (INT), the spindle is identified with its channel index according to the order `CHSPDLNAME`.

Access to common variables for axis and spindle

Accessing variables by their name

When referring to the axis or spindle by its name, programming the channel they are in is not a determining factor; thus, programming them in this case is irrelevant. When programming the channel, if the axis or spindle is not in it, its programming is ignored.

`(V.)A.{var}.X`	Axis variable with that name.
`(V.)A.{var}.S`	Spindle variable with that name.
`(V.)SP.{var}.S2`	Spindle variable with that name.

Accessing variables by their logic number

Depending on whether the channel number is programmed or not, the mnemonic has a different meaning depending on whether it is access from PRG, PLC or INT.

Accessing from PRG or PLC when not indicating the channel number.

`V.A.{var}.m`	Axis or spindle variable with logic number *m*.
`V.SP.{var}.m`	Spindle variable with *m* index in the system.

Accessing from INT when not indicating the channel number.

`A.{var}.m`	Axis variable with *m* index in the active channel.
`SP.{var}.m`	Spindle variable with *m* index in the active channel.

Accessing from PRG, PLC or INT when indicating the channel number.

`(V.)[1].A.{var}.m`	Axis variable with *m* index in the channel. (n=1 corresponds to the first axis of the channel)
`(V.)[2].SP.{var}.m`	Spindle variable with *m* index in the channel. (n=1 corresponds to the first spindle of the channel)

When indicating the channel number, the spindle variables cannot be accessed using the –A.– prefix.

Accessing the exclusive spindle variables

Accessing variables by their name

The access and behavior are the same as if it were an axis and spindle variable.

Accessing variables by their logic number

Depending on whether the channel number is programmed or not, the mnemonic has a different meaning depending on whether it is access from PRG, PLC or INT.

The access from PRG or PLC when not indicating the channel number is the same as if it were an axis and spindle variable.

`(V.)A.{var}.m`	Spindle variable with logic number *m*.
`(V.)SP.{var}.m`	Spindle variable with *m* index in the system.

Accessing from INT when not indicating the channel number. The spindle variables cannot be accessed from the interface using the –A.– prefix.

`V.SP.{var}.m`	Spindle variable with *m* index in the active channel.

Accessing from PRG, PLC or INT when indicating the channel number. The spindle variables cannot be accessed using the –A.– prefix.

`(V.)[n].SP.{var}.m`	Spindle variable with *m* index in the *n* channel .

Variables of the master spindle

They are special variables that may be used to access the data of the master spindle of each channel without knowing its name, logic number or index. They are meant for displaying data and programming cycles.

The variables are identified with the prefix; but without indicating the number nor the name of the spindle.

`(V.)[n].SP.{var}`	Variable of the channel master spindle *n*.

If the channel is not programmed, it assumes the default channel, which in each is:

PRG	Channel where it is being executed.
PLC	First channel or main channel.
INT	Active channel.

14.2 Related to general machine parameters

These variables are read-only (R) synchronous and are evaluated execution time.

They have generic names.

- Replace the "x" letter with the axis number.
- Replace the letters "i" and "m" with numbers keeping the brackets.

(V.)MPG.AXISNAMEx	V.MPG.AXISNAME2	V.MPG.AXISNAME3
(V.)MPG.MASTERAXIS[i]	V.MPG.MASTERAXIS[1]	V.MPG.MASTERAXIS[2]

CHANNEL CONFIGURATION		PRG	PLC	INT
(V.)MPG.NCHANNEL	Number of CNC channels.	R	R	R

AXIS CONFIGURATION		PRG	PLC	INT
(V.)MPG.NAXIS	Number of axes governed by the CNC	R	R	R
(V.)MPG.AXISNAMEx	Name of the "n" logic axis	—	—	R
(V.)MPG.TMASTERAXIS[i]	Tandem [i]. Logic number of the master axis	R	R	R
(V.)MPG.TSLAVEAXIS[i]	Tandem [i]. Logic number of the slave axis	R	R	R
(V.)MPG.TORQDIST[i]	Tandem [i]. Torque distribution	R	R	R
(V.)MPG.PRELOAD[i]	Tandem [i]. Preload	R	R	R
(V.)MPG.PRELFITI[i]	Tandem [i]. Time to apply the preload	R	R	R
(V.)MPG.TPROGAIN[i]	Tandem [i]. Proportional gain	R	R	R
(V.)MPG.TINTIME[i]	Tandem [i]. Integral gain	R	R	R
(V.)MPG.TCOMPLIM[i]	Tandem [i]. Compensation Limit	R	R	R
(V.)MPG.MASTERAXIS[i]	Gantry [i]. Logic number of the master axis	R	R	R
(V.)MPG.SLAVEAXIS[i]	Gantry [i]. Logic number of the slave axis	R	R	R
(V.)MPG.WARNCOUPE[i]	Gantry [i]. Maximum difference to issue a warning	R	R	R
(V.)MPG.MAXCOUPE[i]	Gantry [i]. Maximum difference allowed	R	R	R
(V.)MPG.DIFFCOMP[i]	Gantry [i]. Error difference compensation. "0" = No "1"= Yes	R	R	R

The PLC reading of *TORQDIST*, *PRELOAD*, *TPROGAIN* and *TCOMPLIM* comes in hundredths (x100). Ver ***"Access to numeric values from the PLC"*** en la página 358.

SPINDLE CONFIGURATION		PRG	PLC	INT
(V.)MPG.NSPDL	Number of spindles governed by the CNC	R	R	R
(V.)MPG.SPDLNAMEx	Name of the "x" spindle	—	—	R

TIME SETTING		PRG	PLC	INT
(V.)MPG.LOOPTIME	Loop time	R	R	R
(V.)MPG.PRGFREQ	Frequency of the PRG module (in cycles)	R	R	R

CAN AND SERCOS BUS CONFIGURATION		PRG	PLC	INT
(V.)MPG.SERBRATE	Sercos transmission speed "0" = 4Mbps "1" = 2Mbps	R	R	R
(V.)MPG.SERPOWSE	Sercos optical power	R	R	R
(V.)MPG.CANLENGTH	Can bus cable length (in meters) "0" = Up to 20 "1" = Up to 30 "2" = Up to 40 "3" = Up to 50 "4" = Up to 60 "5" = Up to 70 "6" = Up to 80 "7" = Up to 90 "8" = Up to 100 "9" >100	R	R	R

DEFAULT CONDITIONS		PRG	PLC	INT
(V.)MPG.INCHES	Default work units "0" = mm "1" = inch	R	R	R

RELATED TO ARITHMETIC PARAMETERS		PRG	PLC	INT
(V.)MPG.MAXLOCP	Maximum local arithmetic parameter	R	R	R
(V.)MPG.MINLOCP	Minimum local arithmetic parameter	R	R	R
(V.)MPG.MAXGLBP	Maximum global arithmetic parameter	R	R	R
(V.)MPG.MINGLBP	Minimum global arithmetic parameter	R	R	R
(V.)MPG.ROPARMAX	Maximum global read-only arithmetic parameter	R	R	R
(V.)MPG.ROPARMIN	Minimum global read-only arithmetic parameter	R	R	R
(V.)MPG.MAXCOMP	Maximum common arithmetic parameter	R	R	R
(V.)MPG.MINCOMP	Maximum common arithmetic parameter	R	R	R

CROSS COMPENSATION TABLE		PRG	PLC	INT
(V.)MPG.MOVAXIS[m]	Table [m]. Master axis	R	R	R
(V.)MPG.COMPAXIS[m]	Table [m]. Axis to be compensated	R	R	R
(V.)MPG.NPCROSS[m]	Table [m]. Number of points	R	R	R
(V.)MPG.TYPCROSS[m]	Table [m]. Type of compensation "0" = Real coordinates "1" = Theoretical coordinates	R	R	R
(V.)MPG.BIDIR[m]	Table [m]. Bi-directional compensation "0" = No "1"= Yes	R	R	R
(V.)MPG.REFNEED[m]	Table [m]. Mandatory home search "0" = No "1"= Yes	R	R	R
(V.)MPG.POSITION[m][i]	Table [m]. Master axis position for point [i]	R	R	R
(V.)MPG.POSERROR[m][i]	Table [m]. Error of point [i] in the positive direction	R	R	R
(V.)MPG.NEGERROR[m][i]	Table [m]. Error of point [i] in the negative direction	R	R	R

EXECUTION TIMES		PRG	PLC	INT
(V.)MPG.MINAENDW	Minimum duration of the AUXEND signal	R	R	R
(V.)MPG.REFTIME	Estimated home searching time	R	R	R
(V.)MPG.HTIME	Estimated time for an "H" function	R	R	R
(V.)MPG.DTIME	Estimated time for a "D" function	R	R	R
(V.)MPG.TTIME	Estimated time for a "T" function	R	R	R

NUMBERING OF DIGITAL I/O		PRG	PLC	INT
(V.)MPG.NDIMOD	Total of digital input modules	R	R	R
(V.)MPG.NDOMOD	Total of digital output modules	R	R	R
(V.)MPG.DIMODADDR[n]	Base index of the digital input modules	R	R	R
(V.)MPG.DOMODADDR[n]	Base index of the digital output modules	R	R	R

PROBE		PRG	PLC	INT
(V.)MPG.PROBE	There is a probe for tool calibration "0" = No "1"= Yes	R	R	R
(V.)MPG.PRBDI1	Digital input associated with probe 1	R	R	R
(V.)MPG.PRBDI2	Digital input associated with probe 2	R	R	R
(V.)MPG.PRBPULSE1	Type of pulse of probe 1 "0" = Negative "1" = Positive	R	R	R
(V.)MPG.PRBPULSE2	Type of pulse of probe 2 "0" = Negative "1" = Positive	R	R	R

14.2.1 Channel related

These variables are read-only (R) synchronous and are evaluated execution time.

They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the "x" letter with the axis number.

(V.)[n].MPG.GROUPID	V.[1].MPG.GROUPID	V.[2].MPG.GRUOPID
(V.)[n].MPG.CHAXISNAMEx	V.[2].MPG.CHAXISNAME2	V.[1].MPG.CHAXISNAME3

CHANNEL CONFIGURATION		PRG	PLC	INT
(V.)[n].MPG.GROUPID	Group the channel belongs to	R	R	R
(V.)[n].MPG.CHTYPE	Channel type "0" = CNC "1" = PLC "2" = CNC+PLC	R	R	R
(V.)[n].MPG.HIDDENCH	Hidden channel "0" = No "1"= Yes	R	R	R

CONFIGURING THE AXES OF THE CHANNEL		PRG	PLC	INT
(V.)[n].MPG.CHNAXIS	Number of axes of the channel	R	R	R
(V.)[n].MPG.CHAXISNAMEx	Name of the "n" logic axis	—	—	R

CONFIGURING THE SPINDLES OF THE CHANNEL		PRG	PLC	INT
(V.)[n].MPG.CHNSPDL	Number of spindles of the channel	R	R	R
(V.)[n].MPG.CHSPDLNAMEx	Name of the "x" spindle	—	—	R
(V.)[n].MPG.CAXNAME	Axis working as "C" axis (by default)	—	—	R
(V.)[n].MPG.ALIGNC	"C" axis in diametrical machining "0" = No "1"= Yes	R	R	R

TIME SETTING (CHANNEL)		PRG	PLC	INT
(V.)[n].MPG.PREPFREQ	Number of blocks to prepare per cycle	R	R	R
(V.)[n].MPG.ANTIME	Anticipation time	R	R	R

DEFAULT CONDITIONS		PRG	PLC	INT
(V.)[n].MPG.KINID	Default kinematics number	R	R	R
(V.)[n].MPG.SLOPETYPE	Default acceleration type "1" = Linear "2" = Trapezoidal "3" = Square sine	R	R	R
(V.)[n].MPG.IPLANE	Default work plane "0" = G17 "1" = G18	R	R	R
(V.)[n].MPG.ISYSTEM	Default programming type "0" = G90 "1" = G91	R	R	R
(V.)[n].MPG.IMOVE	Default movement type "0" = G00 "1" = G01	R	R	R
(V.)[n].MPG.IFEED	Default feedrate type "0" = G94 "1" = G95	R	R	R
(V.)[n].MPG.ICORNER	Default corner type "0" = G50 "1" = G05 "2" = G07	R	R	R
(V.)[n].MPG.IRCOMP	Radius compensation mode by default "0" = G136 "1" = G137	R	R	R
(V.)[n].MPG.ROUNDTYPE	Rounding type in G5 (by default) "0" = Chordal error "1" = %feedrate	R	R	R
(V.)[n].MPG.MAXROUND	Maximum rounding error in G5	R	R	R
(V.)[n].MPG.ROUNDFEED	Percentage of feedrate in G5	R	R	R
(V.)[n].MPG.CIRINERR	Absolute radius error	R	R	R
(V.)[n].MPG.CIRINFACT	Percentage of error over the radius	R	R	R
(V.)[n].MPG.MAXOVR	Maximum axis override (%)	R	R	R
(V.)[n].MPG.RAPIDOVR	Override affecting G00 "0" = No "1"= Yes	R	R	R

PLC reading of *CIRINFACT* and *MAXOVR* comes in tenths (a reading of 10 for 1%) Ver ***"Access to numeric values from the PLC"*** en la página 358.

RELATED TO SUBROUTINES		PRG	PLC	INT
(V.)[n].MPG.TOOLSUB	Subroutine associated with "T"	—	—	R
(V.)[n].MPG.REFPSUB	Subroutine associated with G74	—	—	R
(V.)[n].MPG.OEMSUB(1..10)	Subroutines associated with G180 through G189	—	—	R
(V.)[n].MPG.SUBPATH	Program subroutine path	—	—	R

PROBE		PRG	PLC	INT
(V.)[n].MPG.PRB1MIN	Minimum probe coordinate along the abscissa axis	R	R	R
(V.)[n].MPG.PRB1MAX	Maximum probe coordinate along the abscissa axis	R	R	R
(V.)[n].MPG.PRB2MIN	Minimum probe coordinate along the ordinate axis	R	R	R
(V.)[n].MPG.PRB2MAX	Maximum probe coordinate along the ordinate axis	R	R	R
(V.)[n].MPG.PRB3MIN	Minimum probe coordinate along the axis perpendicular to the plane	R	R	R
(V.)[n].MPG.PRB3MAX	Maximum probe coordinate along the axis perpendicular to the plane	R	R	R

14.3 Related to axis machine parameters

When these variables refer to a spindle, they may be accessed using prefix –MPA.– or –SP.– indistinctly.

These variables may be accessed from the program (PRG), PLC and interface (INT), they are read-only (R) synchronous and are evaluated in execution time.

They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the "Xn" character by the name, logic number or index in the channel of the axis or of the spindle.
- Replace the letter "i" with a number keeping the brackets.

(V.)[n].MPA.AXISTYPE.Xn	V.[1].MPA.AXISTYPE.X V.SP.AXISTYPE.S	V.[2].MPA.AXISTYPE.1 V.[3].SP.AXISTYPE.6
(V.)[n].MPA.INCJOGDIST[i].Xn	V.[2].MPA.INCJOGDIST[1].Z	V.[4].MPA.INCJOGDIST[2].3

BELONGING TO THE CHANNEL		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.AXISEXCH	Channel change permission "0" = No "1" = Temporary "2" = Maintained	Yes	Yes	Yes	Yes	P/S

TYPE OF AXIS AND DRIVE		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.DRIVETYPE.Xn	Type of drive "1" = Analog "2" = Sercos "16"=Simulated	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.AXISTYPE.Xn	Type of axis "1" = Linear "2" = Rotary "4" = Spindle	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.DRIVEID.Xn	Sercos drive select (ID)	Yes	Yes	Yes	—	P/S
(V.)[n].MPA.OPMODEP.Xn	Sercos drive operating mode "0" = Position "1" = Velocity	Yes	Yes	Yes	—	P/S
(V.)[n].MPA.FBACKSRC.Xn	Type of axis "0" = Internal "1" = External	Yes	Yes	Yes	—	P/S

HIRTH AXIS		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.HIRTH.Xn	Hirth axis "0" = No "1"= Yes	Yes	Yes	—	Yes	P/S
(V.)[n].MPA.HPITCH.Xn	Hirth axis pitch	Yes	Yes	—	Yes	P/S

AXIS CONFIGURATION FOR LATHE TYPE MACHINES		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.FACEAXIS.Xn	Face axis "0" = No "1"= Yes	Yes	—	—	Yes	P/S
(V.)[n].MPA.LONGAXIS.Xn	Longitudinal axis "0" = No "1"= Yes	Yes	—	—	Yes	P/S

ROTARY AXES		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.AXISMODE.Xn	Work mode "0" = Module "1" = Linear like	—	Yes	—	Yes	P/S
(V.)[n].MPA.UNIDIR.Xn	Unidirectional rotation "0" = No "1"= Yes	—	Yes	—	Yes	P/S
(V.)[n].MPA.SHORTESTWAY.Xn	Via shortest way "0" = No "1"= Yes	—	Yes	—	Yes	P/S

ROTARY AXES AND SPINDLE		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.MODCOMP.Xn	Module compensation "0" = No "1"= Yes	—	Yes	Yes	Yes	S
(V.)[n].MPA.CAXIS.Xn	Works as a "C" axis "0" = No "1"= Yes	—	Yes	Yes	Yes	P/S
(V.)[n].MPA.CAXSET.Xn	Work set for "C" axis	—	Yes	Yes	Yes	P/S

SPINDLE		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.AUTOGEAR.Xn	Automatic gear change "0" = No "1"= Yes	—	—	Yes	Yes	P/S
(V.)[n].MPA.LOSPDLIM.Xn	Lower "rpm OK" percentage	—	—	Yes	Yes	P/S
(V.)[n].MPA.UPSPDLIM.Xn	Upper "rpm OK" percentage	—	—	Yes	Yes	P/S
(V.)[n].MPA.SPDLTIME.Xn	Estimated time for an S function	—	—	Yes	Yes	P/S
(V.)[n].MPA.SPDLSTOP.Xn	M2, M30 and Reset stop the spindle "0" = No "1"= Yes	—	—	Yes	Yes	P/S
(V.)[n].MPA.SREVM05.Xn	G84. Reversal stops the spindle "0" = No "1"= Yes	—	—	Yes	Yes	P/S
(V.)[n].MPA.STEPOVR.Xn	Override step	—	—	Yes	Yes	P/S
(V.)[n].MPA.MINOVR.Xn	Minimum override (%)	—	—	Yes	Yes	P/S
(V.)[n].MPA.MAXOVR.Xn	Maximum override (%)	—	—	Yes	Yes	P/S

PLC reading of *LOSPDLIM*, *UPSPDLIM*, *STEPOVR*, *MINOVR* and *MAXOVR* comes in tenths (a reading of 10 for 1%) Ver ***"Access to numeric values from the PLC"*** en la página 358.

SOFTWARE AXIS LIMITS		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.POSLIMIT.Xn	Positive software limit	Yes	Yes	—	Yes	P/S
(V.)[n].MPA.NEGLIMIT.Xn	Negative software limit	Yes	Yes	—	Yes	P/S
(V.)[n].MPA.SWLIMITTOL.Xn	Software limit tolerance	Yes	Yes	—	Yes	P/S

RUNAWAY PROTECTION		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.TENDENCY.Xn	Activation of tendency test "0" = No "1"= Yes	Yes	Yes	Yes	Yes	P/S

PLC OFFSET		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.PLCOINC.Xn	PLC offset increment per cycle	Yes	Yes	Yes	Yes	P/S

DWELL FOR DEAD AXES		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.DWELL.Xn	Dwell for dead axes	Yes	Yes	Yes	Yes	P/S

RADIUS / DIAMETER		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.DIAMPROG.Xn	Programming in diameters "0" = No "1"= Yes	Yes	—	—	Yes	P/S

HOME SEARCH		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.REFDIREC.Xn	Home search direction "0" = Negative "1" = Positive	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.DECINPUT.Xn	Home switch "0" = No "1"= Yes	Yes	Yes	Yes	Yes	P/S

PROBE		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.PROBEAXIS.Xn	Probing axis	Yes	Yes	—	Yes	P/S
(V.)[n].MPA.PROBERANGE.Xn	Maximum braking distance	Yes	Yes	—	Yes	P/S
(V.)[n].MPA.PROBEFEED.Xn	Probing feedrate	Yes	Yes	—	Yes	P/S
(V.)[n].MPA.PROBEDELAY	Delay for the "probe 1" signal	Yes	Yes	—	Yes	P/S
(V.)[n].MPA.PROBEDELAY	Delay for the "probe 2" signal	Yes	Yes	—	Yes	P/S

TOOL INSPECTION		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.REPOSFEED.Xn	Maximum repositioning feedrate	Yes	Yes	—	Yes	P/S

INDEPENDENT AXIS		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.POSFEED.Xn	Positioning feedrate	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.DSYNCVELW.Xn	Velocity synchronization window	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.DSYNCPOSW.Xn	Position synchronization window	Yes	Yes	Yes	Yes	P/S

JOG MODE		Lin Rot	Spd	Ana	Ser
(V.)[n].MPA.MANPOSSW.Xn	Maximum positive travel with G201	Yes	—	Yes	P/S
(V.)[n].MPA.MANNEGSW.Xn	Maximum negative travel with G201	Yes	—	Yes	P/S
(V.)[n].MPA.JOGFEED.Xn	Continuous JOG mode feedrate	Yes	—	Yes	P/S
(V.)[n].MPA.JOGRAPFEED.Xn	Rapid feed in continuous JOG mode	Yes	—	Yes	P/S
(V.)[n].MPA.MAXMANFEED.Xn	Maximum feed in continuous JOG	Yes	—	Yes	P/S
(V.)[n].MPA.MAXMANACC.Xn	Maximum acceleration in JOG mode	Yes	—	Yes	P/S
(V.)[n].MPA.MANFEEDP.Xn	Maximum % of jog feedrate with G201	Yes	—	Yes	P/S
(V.)[n].MPA.IPOFEEDP.Xn	Maximum % of execution feedrate with G201	Yes	—	Yes	P/S
(V.)[n].MPA.MANACCP.Xn	Maximum % of jog acceleration with G201	Yes	—	Yes	P/S
(V.)[n].MPA.IPOACCP.Xn	Maximum % of execution acceleration with G201	Yes	—	Yes	P/S

JOG MODE. HANDWHEELS		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.MPGRESOL[i].Xn	Dial resolution at the [i] position	Yes	Yes	—	Yes	P/S
(V.)[n].MPA.MPGFILTER.Xn	Filter time for the handwheel	Yes	Yes	—	Yes	P/S

JOG MODE. INCREMENTAL JOG		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.INCJOGDIST[i].Xn	Moving distance at [i] dial position	Yes	Yes	—	Yes	P/S
(V.)[n].MPA.INCJOGFEED[i].Xn	Feedrate at [i] position	Yes	Yes	—	Yes	P/S

LEADSCREW ERROR COMPENSATION		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.LSCRWCOMP.Xn	Leadscrew error compensation "0" = No "1"= Yes	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.NPOINTS.Xn	Number of points in the table	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.TYPLSCRW.Xn	Type of compensation "0" = Real coordinates "1" = Theoretical coordinates	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.BIDIR.Xn	Bi-directional compensation "0" = No "1"= Yes	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.REFNEED.Xn	Mandatory home search "0" = No "1"= Yes	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.POSITION[i].Xn	Master axis position for point [i]	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.POSERROR[i].Xn	Error of point [i] in the positive direction	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.NEGERROR[i].Xn	Error of point [i] in the negative direction	Yes	Yes	Yes	Yes	P/S

FILTERS		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.ORDER[i].Xn	Filter order	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.TYPE[i].Xn	Type of filter "1" = Low passing "2" = Anti-resonance	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.FREQUENCY[i].Xn	Break or center frequency	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.NORBWIDTH[i].Xn	Normal bandwidth	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.SHARE[i].Xn	% of signal going through the filter	Yes	Yes	Yes	Yes	P/S

WORK SETS		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.NPARSETS.Xn	Number of work sets	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.DEFAULTSET.Xn	Default work set (on power-up)	Yes	Yes	Yes	Yes	P/S

14.3.1 Related to gear parameters

These variables may be accessed from the program (PRG), PLC and interface (INT), they are read-only (R) synchronous and are evaluated in execution time.

They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the letter "g" with a gear number keeping the brackets. The first gear is identified with the number 1, "0" is not a valid number.
- Replace the "Xn" character by the name, logic number or index in the channel of the axis or of the spindle.

(V.)[n].MPA.COUNTERID[g].Xn	V.[1].MPA.COUNTERID[1].X	V.[2].MPA.COUNTERID[2].1
(V.)[n].MPA.PITCH[g].Xn	V.[2].MPA.PITCH[1].Z	V.[4].MPA.PITCH[2].3

RESOLUTION		Lin Rot	Spd	Ana	Ser
(V.)[n].MPA.PITCH[g].Xn	Leadscrew pitch	Yes	Yes	Yes	P/S
(V.)[n].MPA.PITCH2[g].Xn	Leadscrew pitch (2nd feedback)	Yes	Yes	—	P/S
(V.)[n].MPA.NPULSES[g].Xn	Number of encoder pulses	Yes	Yes	Yes	S
(V.)[n].MPA.NPULSES2[g].Xn	Number of encoder pulses (2nd feedback)	Yes	Yes	Yes	S
(V.)[n].MPA.INPUTREV[g].Xn	Turns of the motor shaft	Yes	Yes	Yes	P/S
(V.)[n].MPA.INPUTREV2[g].Xn	Turns of the motor shaft (2nd feedback)	Yes	Yes	—	P/S
(V.)[n].MPA.OUTPUTREV[g].Xn	Turns of the machine axis	Yes	Yes	Yes	P/S
(V.)[n].MPA.OUTPUTREV2[g].Xn	Turns of the machine axis (2nd feedback)	Yes	Yes	—	P/S
(V.)[n].MPA.SINMAGNI[g].Xn	Sinusoidal multiplying factor	Yes	Yes	Yes	—
(V.)[n].MPA.ABSFEEDBACK[g].Xn	Absolute feedback system "0" = No "1"= Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.FBACKAL[g]	Feedback alarm activation "0" = No "1"= Yes	Yes	Yes	Yes	—

LOOP SETTING		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.LOOPCH[g].Xn	Analog voltage sign change "0" = No "1"= Yes	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.AXISCH[g].Xn	Feedback sign change "0" = No "1"= Yes	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.INPOSW[g].Xn	In-position zone	Yes	Yes	Yes	Yes	P/S

BACKLASH IN MOVEMENT REVERSAL		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.BACKLASH[g].Xn	Backlash	Yes	Yes	Yes	Yes	P/S

BACKLASH. ADDITIONAL VELOCITY COMMAND PULSE		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.BAKANOUT[g].Xn	Additional velocity command pulse	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.BAKTIME[g].Xn	Duration of the additional velocity command pulse	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.ACTBAKAN[g].Xn	Application of the additional velocity command pulse "0" = G2/G3 "1" = Always	Yes	Yes	Yes	Yes	P/S

FEEDRATE SETTING		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.G00FEED[g].Xn	Feedrate in G00	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.MAXVOLT[g].Xn	Analog voltage for G00FEED	Yes	Yes	Yes	Yes	S

GAIN SETTING		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.PROGAIN[g].Xn	Proportional gain	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.FFWTYPE[g].Xn	Pre-control (feed-forward) type "0" = Off "1" = Feed-forward "2" = Ac-forward "3" = Feed-forward + Ac-forward	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.FFGAIN[g].Xn	Percentage of Feed-Forward in automatic	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.MANFFGAIN[g].Xn	Percentage of Feed-Forward in JOG	Yes	Yes	—	Yes	P/S
(V.)[n].MPA.ACFWFACTOR[g].Xn	Acceleration time constant	Yes	Yes	Yes	Yes	S
(V.)[n].MPA.ACFGAIN[g].Xn	Percentage AC-Forward in automatic	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.MANACFGAIN[g].Xn	Percentage of AC-Forward in JOG	Yes	Yes	—	Yes	P/S

Although in the machine parameter table they may be read with up to four decimals, the following variables will only be read with one or two decimals whichever the case may be.

- *In variables ACFGAIN and MANACFGAIN , only the first decimal is relevant.*
- *In variables FFGAIN and MANFFGAIN only the first two decimals are relevant.*

The PLC reading of *ACFGAIN* and *MANACFGAIN* comes in tenths (x10) The PLC reading of *FFGAIN* and *MANFFGAIN* comes in hundredths (x100) Ver ***"Access to numeric values from the PLC"*** en la página 358.

LINEAR ACCELERATION		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.LACC1[g].Xn	Acceleration of the first section	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.LACC2[g].Xn	Acceleration of the second section	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.LFEED[g].Xn	Change speed	Yes	Yes	Yes	Yes	P/S

TRAPEZOIDAL AND SQUARE SINE ACCELERATION		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.ACCEL[g].Xn	Acceleration	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.DECEL[g].Xn	Deceleration	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.ACCJERK[g].Xn	Acceleration Jerk	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.DECJERK[g].Xn	Deceleration Jerk	Yes	Yes	Yes	Yes	P/S

HOME SEARCH		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.I0TYPE[g].Xn	Reference mark (I0) type "0" = Normal "1" = Increasing distance coded "2" = Decreasing distance coded	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.REFVALUE[g].Xn	Home position	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.REFSHIFT[g].Xn	Offset of the reference point (home)	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.REFFEED1[g].Xn	Fast home searching feedrate	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.REFFEED2[g].Xn	Slow home searching feedrate	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.REFPULSE[g].Xn	Type of I0 pulse "0" = Negative "1" = Positive	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.ABSOFF[g].Xn	Offset with respect to coded ref. mark	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.EXTMULT[g].Xn	External factor for distance-coded mark	Yes	Yes	Yes	Yes	—
(V.)[n].MPA.I0CODDI1[g].Xn	Pitch between 2 fixed coded marks	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.I0CODDI2[g].Xn	Pitch between 2 variable coded marks	Yes	Yes	Yes	Yes	P/S

FOLLOWING ERROR		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.FLWEMONITOR[g].Xn	Monitoring type "0" = Off "1" = Standard "2" = Linear	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.MINFLWE[g].Xn	Maximum following error when stopped	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.MAXFLWE[g].Xn	Maximum following error when moving	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.FEDYNFAC[g].Xn	% of following error deviation	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.ESTDELAY[g].Xn	Following error delay	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.INPOMAX[g].Xn	Time to get in position	Yes	Yes	Yes	Yes	P/S
(V.)[n].MPA.INPOTIME[g].Xn	Minimum time to stay in position	Yes	Yes	Yes	Yes	P/S

AXIS LUBRICATION		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.DISTLUBRI[g].Xn	Distance for lubrication pulse	Yes	Yes	Yes	Yes	P/S

ROTARY AXES AND SPINDLE		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.MODUPLIM[g].Xn	Module's upper limit	—	Yes	Yes	Yes	S
(V.)[n].MPA.MODLOWLIM[g].Xn	Module's lower limit	—	Yes	Yes	Yes	S
(V.)[n].MPA.MODNROT[g].Xn	Module error. Number of turns	—	Yes	Yes	Yes	S
(V.)[n].MPA.MODERR[g].Xn	Module error. Number of increments	—	Yes	Yes	Yes	S

SPINDLE		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.SZERO[g].Xn	Speed considered "0 rpm"	—	—	Yes	—	P/S
(V.)[n].MPA.POLARM3[g].Xn	Analog voltage sign M3 "0" = Negative "1" = Positive	—	—	Yes	—	S
(V.)[n].MPA.POLARM4[g].Xn	Analog voltage sign M4 "0" = Negative "1" = Positive	—	—	Yes	—	S

ANALOG VOLTAGE		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.SERVOOFF[g].Xn	Offset compensation	Yes	Yes	Yes	Yes	—
(V.)[n].MPA.MINANOUT[g].Xn	Minimum analog output	Yes	Yes	Yes	Yes	—

ANALOG OUTPUT / FEEDBACK INPUT		Lin	Rot	Spd	Ana	Ser
(V.)[n].MPA.ANAOUTID[g].Xn	Analog output of the axis	Yes	Yes	Yes	Yes	—
(V.)[n].MPA.COUNTERID[g].Xn	Feedback input for the axis	Yes	Yes	Yes	Yes	—

14.4 Related to jog mode parameters

These variables are read-only (R) synchronous and are evaluated execution time.

They have generic names.

- Replace the letter "i" with the number keeping the brackets.

(V.)MPMAN.NMPG	V.MPMAN.NMPG	
(V.)MPMAN.MPGAXIS[i]	V.MPMAN.MPGAXIS[1]	V.MPMAN.MPGAXIS[2]

HANDWHEELS		PRG	PLC	INT
(V.)MPMAN.NMPG	Number of handwheels	R	R	R
(V.)MPMAN.COUNTERID[i]	Feedback input for the handwheel [i]	R	R	R
(V.)MPMAN.MPGAXIS[i]	Axis associated with handwheel [i]	R	R	R

JOG KEYS		PRG	PLC	INT
(V.)MPMAN.JOGKEYDEF[i]	Axis and moving direction of the JOG [i] key	R	R	R
(V.)MPMAN.JOGTYPE	JOG behavior	R	R	R

This variable may have the following values:

"1", "2"..."16" = Machine parameter set to "+1", "+2"..."+16". (Key for the axis and positive direction)

"-1", "-2"... "-16" = Machine parameter set to "-1", "-2"..."-16". (Key for the axis and negative direction)

"101", "102"..."116" = Machine parameter set to "1", "2"..."16". (Axis key)

"300" = Machine parameter set to "R". (Rapid key)

"301" = Machine parameter set to "+". (Key for positive direction)

"302" = Machine parameter set to "-". (Key for negative direction)

14.5 Related to "M" function parameters

These variables are read-only (R) synchronous and are evaluated execution time.

They have generic names.

- Replace the letter "i" with the number keeping the brackets.

(V.)MPM.MNUM[i]	V.MPM.MNUM[3]
(V.)MPM.MTABLESIZE	V.MPM.MTABLESIZE

"M" FUNCTION TABLE		PRG	PLC	INT
(V.)MPM.MTABLESIZE	Number of elements of the "M" function table	R	R	R
(V.)MPM.MNUM[i]	"M" function number	R	R	R
(V.)MPM.SYNCHTYPE[i]	Type of synchronism of the "M" function "0" = Without synchronism "2" = Before-before "4" = Before-after "8" = after-after	R	R	R
(V.)MPM.MTIME[i]	Estimated time for an "M" function	R	R	R
(V.)MPM.MPROGNAME[i]	Name of the subroutine associated with the "M" function	—	—	R

14.6 Related to kinematic parameters

These variables are read-only (R) synchronous and are evaluated execution time.

They have generic names.

- Replace the "n" letter with the kinematics number.
- Replace the "m" letter with the offset number.

(V.)MPK.KINn[m]	V.MPK.KIN1[1]	V.MPK.KIN6[42]

KINEMATICS		PRG	PLC	INT
(V.)MPK.NKIN	Kinematics table	R	R	R
(V.)MPK.TYPE	Kinetics type	R	R	R
(V.)MPK.KINn[m]	[m] offset of "n" kinematics	R	R	R

14.7 Related to magazine parameters

These variables are read-only (R) synchronous and are evaluated execution time.

They have generic names.

- Replace the "z" character with the magazine number, maintaining the brackets.

(V.)TM.MZSIZE[z]	V.TM.MZSIZE[1]

MAGAZINE		PRG	PLC	INT
(V.)TM.NTOOLMZ	Number of tool magazines	R	R	R
(V.)TM.MZGROUND[z]	Ground tools allowed "0" = No "1"= Yes	R	R	R
(V.)TM.MZSIZE[z]	Magazine size	R	R	R
(V.)TM.MZRANDOM[z]	Random magazine "0" = No "1"= Yes	R	R	R
(V.)TM.MZTYPE[z]	Type of magazine "1" = Asynchronous "2" = Synchronous "3" = Turret "4" = Synchronous with 1 arm "5" = Synchronous with 2 arms	R	R	R
(V.)TM.MZCYCLIC[z]	Cyclic tool changer "0" = No "1"= Yes	R	R	R
(V.)TM.MZOPTIMIZED[z]	Tool management "0" = No "1"= Yes	R	R	R
(V.)TM.MZM6ALONE[z]	Action when executing an M6 without a tool "0" = Nothing "1" = Warning "2" = Error	R	R	R

14.8 Related to OEM parameters

These variables are read-only (R) synchronous and are evaluated execution time.

They have generic names.

- Replace the letter "i" with the parameter number keeping the brackets. This number corresponds with the parameter number in the machine parameter table. For example, the parameter that appears in the MTBPAR table as P0 will be accessed as (V.)MTB.P[0].

(V.)MTB.P[i]	V.MTB.P[3]

SHARED MEMORY		PRG	PLC	INT
(V.)MTB.PLCDATASIZE	Size of the PLC's shared data area	R	R	R

OEM PARAMETER		PRG	PLC	INT
(V.)MTB.SIZE	Number of OEM parameters	R	R	R
(V.)MTB.P[i]	Value of the OEM parameter [i]	R	R	R
(V.)MTB.PF[i]	Value of the OEM parameter [i] Value per 10000	R	R	R

When reading the (V.)MTB.P[i] variable from the PLC, it truncates the decimal portion. The (V.)MTB.PF[i] variables return the parameter value multiplied by 10000.

```
DATA = 54.9876
(V.)MTB.P[10] = 54
(V.)MTB.PF[10] = 549876
```

READING DRIVE VARIABLES		PRG	PLC	INT
(V.)DRV.SIZE	Number of variables to be consulted at the drive	R	R	R
(V.)DRV.name	Value of the variable	R/W	R/W	R/W

The access to drive variables may be either to read or write depending on how it has been set in the machine parameter table. Likewise, the type of access to these variables from the PLC, synchronous or asynchronous, is also defined in the machine parameter table.

14.9 User tables related

These variables are read/write (R/W) synchronous and are evaluated during execution.

They have generic names.

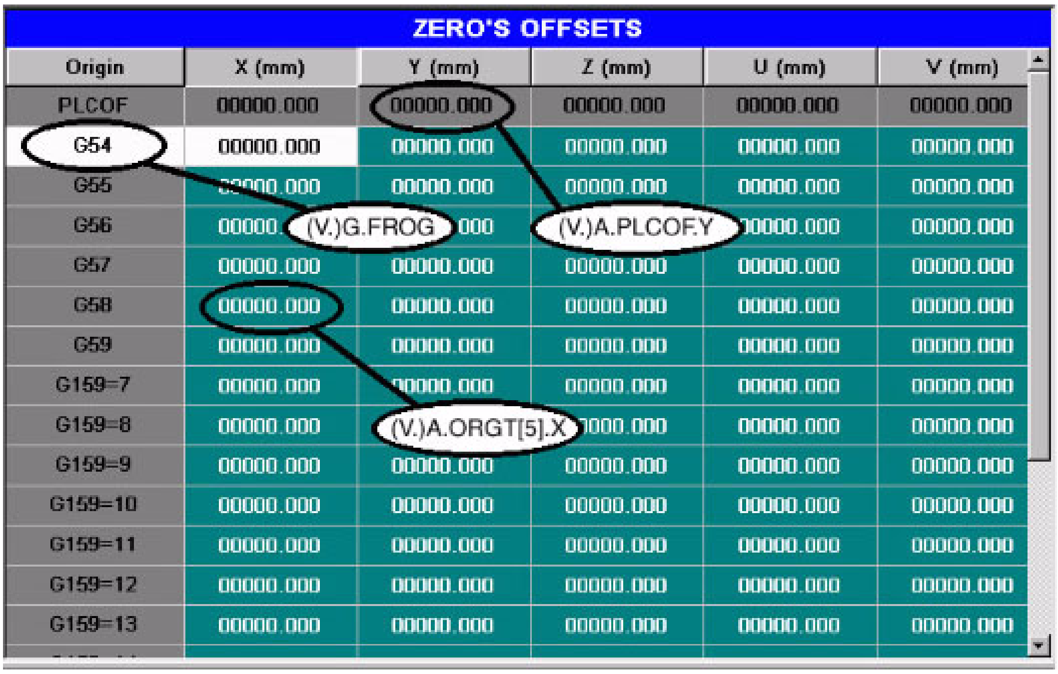
- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the letters "m" and "i" with a number, keeping the brackets.
- Replace the "Xn" character by the name, logic number or index in the channel of the axis.

(V.)A.ORGT[i].Xn	V.A.ORGT[1].X	V.A.ORGT[1].1
(V.)A.FIX.Xn	V.A.FIX.X	V.A.FIX.2
(V.)G.LUPm[n]	V.G.LUP2[12]	

ZERO OFFSET TABLE		**Lin Rot**	**Spd**	**PRG**	**PLC**	**INT**	**Exec**
(V.)G.FORG	First zero offset in the table	—	—	R	R	R	Yes
(V.)G.NUMORG	Number of zero offsets in the table	—	—	R	R	R	Yes
(V.)[n].A.ORG.Xn	Offset of current origin for the Xn axis	Yes	No	R	R	R	No
(V.)[n].A.ORGT[i].Xn	Offset of [i] origin for the Xn axis	Yes	No	R/W	R/W	R/W	Yes
(V.)[n].A.PLCOF.Xn	Offset of PLC origin for the Xn axis	Yes	No	R/W	R/W	R	Yes

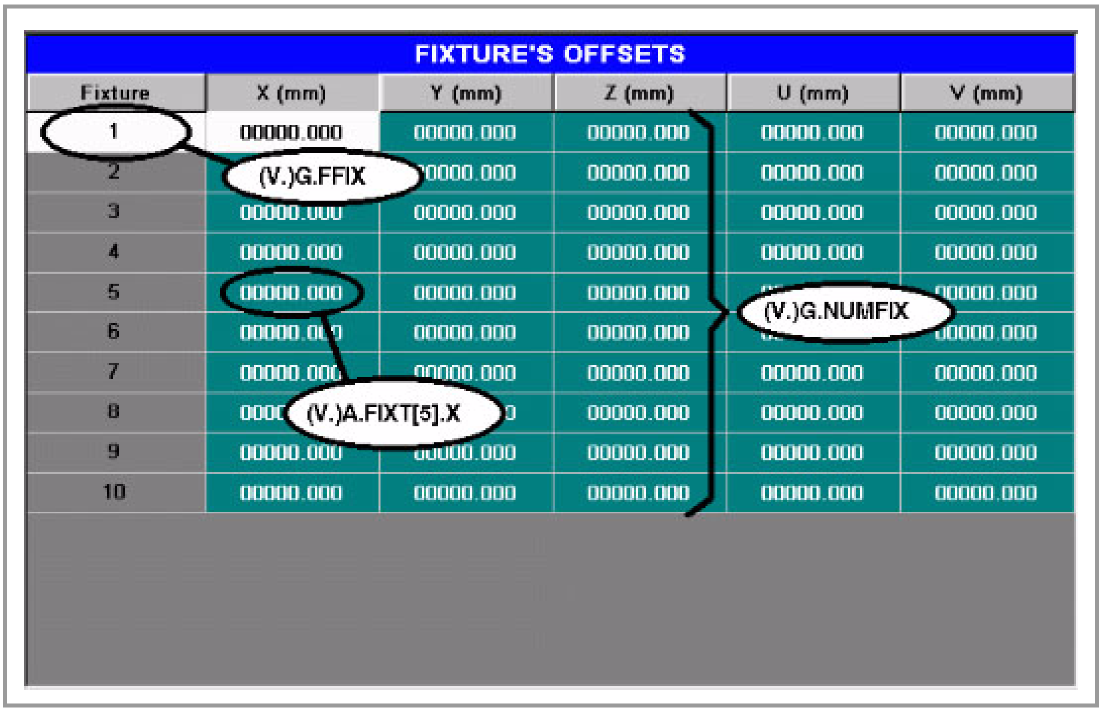
The numbering of zero offsets G54 through G59 is always the same:

G54=1, G55=2, G56=3, G57=4, G58=5, G59=6



Origin	X (mm)	Y (mm)	Z (mm)	U (mm)	V (mm)
PLCOF	00000.000	00000.000	00000.000	00000.000	00000.000
G54	00000.000	00000.000	00000.000	00000.000	00000.000
G55	00000.000	00000.000	00000.000	00000.000	00000.000
G56	00000	000		0000.000	00000.000
G57	00000.000	00000.000	00000.000	00000.000	00000.000
G58	00000.000	00000.000	00000.000	00000.000	00000.000
G59	00000.000	00000.000	00000.000	00000.000	00000.000
G159=7	00000.000	00000.000	00000.000	00000.000	00000.000
G159=8	00000.000		000.000	00000.000	00000.000
G159=9	00000.000	00000.000	00000.000	00000.000	00000.000
G159=10	00000.000	00000.000	00000.000	00000.000	00000.000
G159=11	00000.000	00000.000	00000.000	00000.000	00000.000
G159=12	00000.000	00000.000	00000.000	00000.000	00000.000
G159=13	00000.000	00000.000	00000.000	00000.000	00000.000

FIXTURE TABLE		Lin Rot	Spd	PRG	PLC	INT	Exec
(V.)G.FFIX	First fixture of the table	—	—	R	R	R	Yes
(V.)G.NUMFIX	Number of fixtures in the table	—	—	R	R	R	Yes
(V.)[n].G.FIX	Number of current fixture	—	—	R/W	R	R	No
(V.)[n].A.FIX.Xn	Offset of current fixture for Xn axis	Yes	No	R	R	R	No
(V.)[n].A.FIXT[i].Xn	Offset of [i] fixture for the Xn axis	Yes	No	R/W	R/W	R/W	Yes



ARITHMETIC PARAMETER TABLES		PRG	PLC	INT	Exec
(V.)G.CUP[i]	Value of the common arithmetic parameter [i]	—	R/W	R/W	Yes
(V.)G.CUPF[i]	Value of the common arithmetic parameter [i]. Value per 10000	—	R/W	R/W	Yes
(V.)[n].G.GUP[i]	Value of the global arithmetic parameter [i]	—	R/W	R/W	Yes
(V.)[n].G.GUPF[i]	Value of the global arithmetic parameter [i]. Value per 10000	—	R/W	R/W	Yes
(V.)[n].G.LUPACT[i]	Value of local arithmetic parameter [i] active level	—	R/W	R/W	Yes
(V.)[n].G.LUPm[i]	Value of local arithmetic parameter [i] of m level	—	R/W	R/W	Yes
(V.)[n].G.LUPmF[i]	Value of local arithmetic parameter [i] of m level. Value per 1000	—	R/W	R/W	Yes

When reading variables G.CUP, G.GUP and G.LUP1[i] through G.LUP7[i] from the PLC, it truncates the decimal portion. Variables G.CUPF, G.GUPF and G.LUP1F[i] through G.LUP7F[i] return the parameter value multiplied by 10000.

```
P100 = 23.1234
G.GUP[100] = 23
G.GUPF[100] = 231234
```

14.10 Tool related

For all the tool variables, those referred to the active tool (e.g. TM.TOR) are always for synchronous reading. The variables referred to a tool other than the active one (e.g. TM.TORT[i][j]) are for synchronous reading if the tool is in the magazine and for asynchronous reading if otherwise. The writing of these variables is always asynchronous, be it for the active tool or not.

The reading of the manager's variables is also asynchronous.

These variables are evaluated during block execution. They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the letters "m", "j" and "i" with a number, keeping the brackets.
- Replace the "Xn" character by the name, logic number or index in the channel of the axis.

(V.)[n].TM.TOOL	V.[1].TM.TOOL	V.[4].TM.TOOL
(V.)TM.TORT[m][i]	V.TM.TORT[3][1]	V.TM.TORT[21][2]
(V.)TM.TOFLWT[m][i].Xn	(V.)TM.TOFLWT[4][1].X	(V.)TM.TOFLWT[4][1].1

TOOL AND OFFSETS		PRG	PLC	INT
(V.)TM.T[z][j]	Tool in the [j] position of the [z] magazine	R	R	R
(V.)TM.P[z][m]	Position of the [m] tool in the [z] magazine	R	R	R
(V.)[n].TM.TOOL	Number of the active tool	R	R	R
(V.)[n].TM.TOD	Number of the active tool offset	R	R	R
(V.)[n].TM.NXTOOL	Number of the next tool	R	R	R
(V.)[n].TM.NXTOD	Number of the next tool offset	R	R	R

If in variables (V.)TM.T[z][j] and (V.)TM.P[z][m], the number of the *[z]* magazine is left out, the variables will refer to the former.

The "next tool" is the one already selected but waiting to be activated by executing an M06.

MONITORING		PRG	PLC	INT
(V.)[n].TM.TOMON[i]	Monitoring type of the [i] offset of the active tool	R	R	R
(V.)TM.TOMONT[m][i]	Monitoring type of the [i] offset of the [m] tool	R/W	R/W	R/W
(V.)[n].TM.TLFN[i]	Maximum life of the [i] offset of the active tool	R	R	R
(V.)TM.TLFNT[m][i]	Maximum life of the [i] offset of the [m] tool	R/W	R/W	R/W
(V.)[n].TM.TLFR[i]	Real life of the [i] offset of the active tool	R	R	R
(V.)TM.TLFRT[m][i]	Real life of the [i] offset of the [m] tool	R/W	R/W	R/W
(V.)[n].TM.REMLIFE	Remaining life of the active tool	—	R	R

If in the tool variables, the offset number is left out, the variable will then refer to the active offset.

MAGAZINE		PRG	PLC	INT
(V.)[n].TM.TSTATUS	Status of the active tool	R	R	R
(V.)TM.TSTATUST[m]	Status of the [m] tool	—	R	R
(V.)[n].TM.TLFF	Family of the active tool	R	R	R
(V.)TM.TLFFT[m]	Family of the [m] tool	R/W	R/W	R/W
(V.)[n].TM.ACTUALMZ	Tool Magazine being used by each channel	R	R	R
(V.)TM.MZRESPECTSIZE[z]	In a random magazine [z], the tool always in the same position.	R	R	R
(V.)TM.MZACTUALCH[z]	Channel being used by the tool magazine [z]	R	R	R

The following variables may be accessed from the program (PRG), PLC and interface (INT) are read-write (R/W).

GEOMETRY		Rot Lin	Spd
(V.)[n].TM.TOR[i]	Radius of the tool offset [i] of the active tool	—	—
(V.)TM.TORT[m][i]	Radius of the tool offset [i] of the [m] tool	—	—
(V.)[n].TM.TOI[i]	R wear of the [i] offset of the active tool	—	—
(V.)TM.TOIT[m][i]	R wear of the [i] offset of the [m] tool	—	—
(V.)[n].TM.TOL[i]	Length offset [i] of the active tool	—	—
(V.)TM.TOLT[m][i]	Length of the tool offset [i] of the [m] tool	—	—
(V.)[n].TM.TOK[i]	L wear of the [i] offset of the active tool	—	—
(V.)TM.TOKT[m][i]	L wear of the [i] offset of the [m] tool	—	—
(V.)[n].TM.TOTIPR[i]	Tool tip radius of the [i] offset of the active tool	—	—
(V.)TM.TOTIPRT[m][i]	Tool tip radius of the [i] offset of the [m] tool	—	—
(V.)[n].TM.TOWTIPR[i]	Tool tip radius wear of the [i] offset of the active tool	—	—
(V.)TM.TOWTIPRT[m][i]	Tool tip radius wear of the [i] offset of the [m] tool	—	—
(V.)[n].TM.TOCUTL[i]	Cutting length of the [i] offset of the active tool	—	—
(V.)TM.TOCUTLT[m][i]	Cutting length of the [i] offset of the [m] tool	—	—
(V.)[n].TM.TOAN[i]	Penetration angle of the [i] offset of the active tool	—	—
(V.)TM.TOANT[m][i]	Penetration angle of the [i] offset of the [m] tool	—	—
(V.)[n].TM.TOFL[i].Xn	Xn axis deviation of the [i] offset of the active tool	Yes	No
(V.)[n].TM.TOFL1	Offset of the tool in the first axis of the channel	Yes	No
(V.)[n].TM.TOFL2	Offset of the tool in the second axis of the channel	Yes	No
(V.)[n].TM.TOFL3	Offset of the tool in the third axis of the channel	Yes	No
(V.)TM.TOFLT[m][i].Xn	Xn axis deviation of the [i] offset of the [m] tool	Yes	No
(V.)[n].TM.TOFLW[i].Xn	Xn axis deviation of the [i] offset of the active tool	Yes	No
(V.)[n].TM.TOFLW1	Wear offset of the tool in the first axis of the channel	Yes	No
(V.)[n].TM.TOFLW2	Wear offset of the tool in the second axis of the channel	Yes	No
(V.)[n].TM.TOFLW3	Wear offset of the tool in the third axis of the channel	Yes	No
(V.)TM.TOFLWT[m][i].Xn	Xn axis deviation wear of the [i] offset of the [m] tool	Yes	No

If in the tool variables, the offset number is left out, the variable will then refer to the active offset.

(V.)TM.TOR[i] Radius of active tool, offset [i].

(V.)TM.TOR Radius of active tool, active offset.

(V.)TM.TORT[m][i] Tool radius [m], offset [i].

(V.)TM.TORT[m] Tool radius [m], active offset in the channel.

"CUSTOM" DATA		PRG	PLC	INT
(V.)[n].TM.TOTP1	Additional parameter 1 of the active tool	R/W	R/W	R/W
(V.)[n].TM.TOTP2	Additional parameter 2 of the active tool	R/W	R/W	R/W
(V.)[n].TM.TOTP3	Additional parameter 3 of the active tool	R/W	R/W	R/W
(V.)[n].TM.TOTP4	Additional parameter 4 of the active tool	R/W	R/W	R/W
(V.)TM.TOTP1T[i]	Additional parameter 1 of the [i] tool	R/W	R/W	R/W
(V.)TM.TOTP2T[i]	Additional parameter 2 of the [i] tool	R/W	R/W	R/W
(V.)TM.TOTP3T[i]	Additional parameter 3 of the [i] tool	R/W	R/W	R/W
(V.)TM.TOTP4T[i]	Additional parameter 4 of the [i] tool	R/W	R/W	R/W

TOOL MANAGER		PRG	PLC	INT
(V.)[n].TM.MZSTATUS	Status of the tool manager	—	R	R
(V.)[n].TM.MZRUN	Tool manager running	—	R	R
(V.)[n].TM.MZMODE	Operating mode of the tool manager	R/W	R	R/W
(V.)[n].TM.MZWAIT	Tool manager executing a maneuver	R	R	R

(V.)TM.MZWAIT There is no need to program it in the subroutine associated with `M06`. The subroutine itself waits for the manager's maneuvers to finish. This way, block preparation is not interrupted.

14.10.1 Variables only used during block preparation

The CNC reads several blocks ahead of the one being executed in order to calculate in advance the path to follow.

As can be seen in the following example, the block being prepared is calculated with the tool T6; whereas the tool T1 is the one currently selected.

`G1 X100 F200 T1 M6`	(Block in execution)
`Y200`	
`G1 X20 F300 T6 M6`	
`X30 Y60`	(Block being prepared)

There are specific variables for consulting and/or modifying the values being used in the preparation.

They can only be accessed from the program (PRG) and they are evaluated during block preparation.

When writing in any of these variables, the table is not modified; the new value is only assumed for block preparation.

The following table refers to the tool being prepared, unless stated otherwise.

ONLY USED DURING BLOCK PREPARATION		Lin	Rot	Spd	PRG
(V.)[n].G.TOOL	Number of the tool being prepared	—	—	—	R
(V.)[n].G.TOD	Number of tool offset being prepared	—	—	—	R
(V.)[n].G.NXTOOL	Number of next tool being prepared	—	—	—	R
(V.)[n].G.NXTOD	Number of next tool offset being prepared	—	—	—	R
(V.)[n].G.TOR	Radius of the tool offset being prepared	—	—	—	R/W
(V.)[n].G.TOI	Radius wear of the tool offset being prepared	—	—	—	R/W
(V.)[n].G.TOL	Length of the tool offset being prepared	—	—	—	R/W
(V.)[n].G.TOK	Length wear of the tool offset being prepared	—	—	—	R/W
(V.)[n].G.TOTIPR	Tip radius of the offset being prepared	—	—	—	R/W
(V.)[n].G.TOWTIPR	Tip radius wear of the offset being prepared	—	—	—	R/W
(V.)[n].G.TOCUTL	Cutting length of the tool offset being prepared	—	—	—	R/W
(V.)[n].G.TOAN	Penetration angle of the tool offset being prepared	—	—	—	R/W
(V.)[n].A.TOFL.Xn	Deviation of the active offset on the Xn axis	Yes	Yes	No	R/W
(V.)[n].A.TOFLW.Xn	Deviation of the active wear offset on the Xn axis	Yes	Yes	No	R/W
(V.)[n].G.TOFL1	Offset of the tool in the first axis of the channel	Yes	Yes	No	R/W
(V.)[n].G.TOFL2	Offset of the tool in the second axis of the channel	Yes	Yes	No	R/W
(V.)[n].G.TOFL3	Offset of the tool in the third axis of the channel	Yes	Yes	No	R/W
(V.)[n].G.TOFLW1	Wear offset of the tool in the first axis of the channel	Yes	Yes	No	R/W
(V.)[n].G.TOFLW2	Wear offset of the tool in the second axis of the channel	Yes	Yes	No	R/W
(V.)[n].G.TOFLW3	Wear offset of the tool in the third axis of the channel	Yes	Yes	No	R/W
(V.)[n].G.TOMON	Monitoring type of the tool offset being prepared	—	—	—	R/W
(V.)[n].G.TLFN	Nominal life of the tool offset being prepared	—	—	—	R
(V.)[n].G.TLFR	Real life of the tool offset being prepared	—	—	—	R
(V.)[n].G.REMLIFE	Remaining life of the tool offset being prepared	—	—	—	R/W
(V.)[n].G.TSTATUS	Status of the tool being prepared	—	—	—	R
(V.)[n].G.TLFF	Family of the tool offset being prepared	—	—	—	R
(V.)[n].G.TOTP1	Additional parameter 1 of the active tool	—	—	—	R/W
(V.)[n].G.TOTP2	Additional parameter 2 of the active tool	—	—	—	R/W
(V.)[n].G.TOTP3	Additional parameter 3 of the active tool	—	—	—	R/W
(V.)[n].G.TOTP4	Additional parameter 4 of the active tool	—	—	—	R/W

14.11 PLC related

These variables are evaluated when being executed.

They have generic names.

- Replace the letter "i" with a number keeping the brackets.

(V.)PLC.I[n]	V.PLC.I[16]
(V.)PLC.signal	V.PLC.auxend

STATUS		PRG	PLC	INT	R	W
(V.)PLC.STATUS	PLC status "0" = Stopped "1" = Running	R	—	R	—	—

RESOURCES		PRG	PLC	INT	R	W
(V.)PLC.I[i]	Status of PLC input [i]	R/W	—	R/W	—	—
(V.)PLC.O[i]	Status of PLC output [i]	R/W	—	R/W	—	—
(V.)PLC.M[i]	Status of PLC mark [i]	R/W	—	R/W	—	—
(V.)PLC.R[i]	Status of PLC register [i]	R/W	—	R/W	—	—
(V.)PLC.T[i]	Status of PLC timer [i]	R	—	R/W	—	—
(V.)PLC.C[i]	Status of PLC counter [i]	R	—	R/W	—	—
(V.)PLC.signal	Status of exchange signals with CNC (any mark or register)	R/W	—	R/W	—	—

SYMBOLS		PRG	PLC	INT	R	W
(V.)PLC.symbol	Status of the external symbols defined at the PLC	R/W	—	R/W	—	—

This variable may be used to consult only the symbols defined with the PDEF command in the PLC program.

MESSAGES		PRG	PLC	INT	R	W
(V.)PLC.MSG[i]	Status of PLC message [n] "0" = Inactive "1" = Active	R/W	—	R/W	—	—
(V.)PLC.PRIORMSG	Active message with the highest priority (the one with the lowest number among the active ones)	R	—	R	—	—
(V.)PLC.EMERGMSG	Active emerging message (the one shown at full screen)	R	—	R	—	—

ERRORS		PRG	PLC	INT	R	W
(V.)PLC.ERR[i]	Status of PLC error [n] "0" = Inactive "1" = Active	R/W	—	R/W	—	—
(V.)PLC.PRIORERR	Active error with the highest priority (the one with the lowest number among the active ones)	R	—	R	—	—

TIMER		PRG	PLC	INT	R	W
(V.)PLC.TIMER	Value of the timer enabled by PLC	R/W	R/W	R/W	Syn	Syn

The PLC "TIMER" is enabled or disabled with the PLC mark TIMERON. It counts when TIMERON=1

Using the variable (V.)PLC.TIMER, it is possible to consult and/or modify its count. Value in seconds.

14.12 Jog mode related

With the jog selector switch on the operator panel, it is possible to select the "Type of movement", the "Resolution of the handwheel" and the "Incremental jog position".

Those values may also be forced from the PLC. When setting a value from the PLC, the CNC ignores the selector switch.

Example to set the "10" position to the X axis handwheel:

Set variable (V.)A.PLCMMODE.X to "1"
Set variable (V.)PLC.MPGDIX to "2"

For the X axis handwheel to "obey" (not to ignore) the switch:

Set variable (V.)A.PLCMMODE.X to "0"

These variables are synchronous read/write (R/W). All these variables are evaluated when being executed.

TYPE OF MOVEMENT		Lin Rot	Spd	PRG	PLC	INT
(V.)G.MANMODE	Active for all the axes	—	—	R	R	R
(V.)G.CNCMANMODE	At the switch for all of the axes	—	—	R	R	R/W
(V.)PLC.MANMODE	By PLC for all the axes	—	—	R	R/W	R
(V.)[n].A.MANMODE.Xn	Active for the Xn axis	Yes	No	R	R	R
(V.)[n].A.CNCMMODE.Xn	At the switch for the Xn axis	Yes	No	R	R	R/W
(V.)[n].A.PLCMMODE.Xn	By PLC for the Xn axis	Yes	No	R	R/W	R

These variables may have the following values:

"0" = No type is forced from the PLC.

"1" = Handwheel mode.

"2" = Continuous jog mode.

"3" = Incremental jog mode.

The variable "(V.)[n].A.MANMODE.Xn" may also have the following value:

"4" = Handwheel mode without selected axis. The handwheel mode has been selected but the axis to be moved has not been selected.

HANDWHEEL MODE RESOLUTION (POSITION)		PRG	PLC	INT
(V.)G.MPGIDX	Active position for all the handwheels	R	R	R
(V.)G.CNCMPGIDX	Position selected at the switch	R	R	R/W
(V.)PLC.MPGIDX	Position selected by PLC	R	R/W	R

These variables may have the following values:

"1" = Position 1

"2" = Position 10

"3" = Position 100

INCREMENTAL JOG POSITION		PRG	PLC	INT
(V.)G.INCJOGIDX	Active position for all the axes	R	R	R
(V.)G.CNCINCJOGIDX	Position selected by the switch	R	R	R/W
(V.)PLC.INCJOGIDX	Position selected by PLC	R	R/W	R

These variables may have the following values:

"1" = Position 1

"2" = Position 10

"3" = Position 100

"4" = Position 1000

"5" = Position 10000

JOG FEEDRATES		PRG	PLC	INT
(V.)[n].G.FMAN	JOG feedrate in G94	R/W	R	R/W
(V.)[n].G.MANFPR	JOG feedrate in G95	R/W	R	R/W

The variables associated with the jog mode are modified when changing the value of the –F– field from the jog mode screen. These variables are not affected when changing the feedrate from the MDI mode.

14.13 Coordinate related

Remember that a spindle working in closed loop (M19 or G63) behaves like an axis.

These variables are for synchronous reading (R).

They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the "Xn" character by the name, logic number or index in the channel of the axis.
- Replace the "Sn" character by the name, logic number or index in the channel of the spindle.

(V.)[n].A.PPOS.Xn	V.[1].A.PPOS.X	V.[1].A.PPOS.1
(V.)[n].A.POS.Sn	V.[2].A.POS.S	V.[2].A.POS.2

There are real and theoretical coordinates corresponding to the tool base and tool tip. All of them may be referred to Machine Zero or to the current Part Zero.

A theoretical coordinate is the position that the axis must occupy at all times, a real coordinate is the one it actually occupies and the difference between these two is called "following error".

RELATED TO LINEAR AND ROTARY AXES		PRG	PLC	INT	Exec
(V.)[n].A.PPOS.Xn	Programmed coordinates (of the tool tip)	R	R	R	No
(V.)[n].G.PLPPOS1	Programmed coordinate (of the tool tip) First axis of the channel	R	R	R	No
(V.)[n].G.PLPPOS2	Programmed coordinate (of the tool tip) Second axis of the channel	R	R	R	No
(V.)[n].G.PLPPOS3	Programmed coordinate (of the tool tip) Third axis of the channel	R	R	R	No
(V.)[n].A.FLWE.Xn	Following error of the axis	R	R	R	Yes
(V.)[n].A.APOS.Xn	Part coordinates. Real of the tool base	R	R	R	Yes
(V.)[n].A.ATPOS.Xn	Part coordinates. Theoretical of the tool base	R	R	R	Yes
(V.)[n].A.ATIPPOS.Xn	Part coordinates. Real of the tool tip	R	R	R	Yes
(V.)[n].A.ATIPTPOS.Xn	Part coordinates. Theoretical of the tool tip	R	R	R	Yes
(V.)[n].A.POS.Xn	Machine coordinates. Real of the tool base	R	R	R	Yes
(V.)[n].A.TPOS.Xn	Machine coordinates. Theoretical of the tool base	R	R	R	Yes
(V.)[n].A.TIPPOS.Xn	Machine coordinates. Real of the tool tip	R	R	R	Yes
(V.)[n].A.TIPTPOS.Xn	Machine coordinates. Theoretical of the tool tip	R	R	R	Yes

The PPOSS variable returns the target coordinate, in part coordinates and referred to the tool tip, in the current reference system; i.e. taking into consideration the coordinate rotation, scaling factor, active incline plane, etc.

`G1 X10`	V.A.PPOS.X=10
`#SCALE [2]`	(Scaling factor of ·2·)
`G1 X10`	V.A.PPOS.X=20
`G73 Q90`	[Coordinate system rotation (pattern rotation)]
`X10`	V.A.PPOS.Y=20 (since the Y axis is the one that moves)

The values of the PPOS variables read from a program or from the PLC and the interface will be different when the coordinate is affected by tool compensation or when machining in round corner mode. The value read by program will be the programmed coordinate whereas the value read from the PLC or interface will be the real (actual) coordinate considering tool radius compensation and corner rounding.

SPINDLE RELATED		PRG	PLC	INT	Exec
(V.)[n].A.POS.Sn	Real spindle position	R	R	R	No
(V.)[n].A.TPOS.Sn	Theoretical spindle position	R	R	R	Yes
(V.)[n].A.PPOS.Sn	Programmed spindle position	R	R	R	Yes
(V.)[n].A.FLWE.Sn	Spindle following error	R	R	R	Yes

14.14 Feedrate related

These variables are synchronous read/write (R/W).

They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.

FEEDRATES		PRG	PLC	INT	Exec
(V.)[n].G.FREAL	Real CNC feedrate	R	R	R	Yes
(V.)[n].G.FEED	Active feedrate in G94	R	R	R	Yes
(V.)[n].PLC.F	Feedrate by PLC in G94	R	R/W	R	Yes
(V.)[n].G.PRGF	Feedrate by program in G94	R	R	R	No
(V.)[n].G.FPREV	Active feedrate in G95	R	R	R	Yes
(V.)[n].PLC.FPR	Feedrate by PLC in G95	R	R/W	R	Yes
(V.)[n].G.PRGFPR	Feedrate by program in G95	R	R	R	No

The (V.)G.FREAL variable takes into account the accelerations and decelerations of the machine. When the axes are stopped, it returns a value of "0" and when moving it returns the value corresponding to the feedrate type G94/G95. On laser cutting machines, it is recommended to use this variable so the laser power is proportional to the feedrate.

The feedrate in G94 (mm/min) or G95 (mm/rev) may be set by program or by PLC; the one set by PLC has the highest priority. When selecting a new feedrate in MDI mode, the CNC updates the following variables:

- (V.)G.FEED and (V.)G.PRGF with G94 active.
- (V.)G.FPREV and (V.)G.PRGFPR with G95 active.

MACHINING TIME		PRG	PLC	INT	Exec
(V.)G.FTIME	Machining time in G93	R	R	R	No
The machining time is given in seconds.					

FEED-RATE OVERRIDE		PRG	PLC	INT	Exec
(V.)[n].G.FRO	% F active at the CNC	R	R	R	Yes
(V.)[n].A.FRO.Xn	% F active by axis	R/W	R/W	R/W	Yes
(V.)[n].G.PRGFRO	% F by program	R/W	R	R	No
(V.)[n].PLC.FRO	% F by PLC	R	R/W	R	Yes
(V.)[n].G.CNCFRO	% F at the selector switch	R	R	R/W	Yes

(V.)[n].A.FRO.Xn Valid for rotary and linear axes. Also for the independent axes.

The Feedrate override % may be set by program, by PLC or with the selector switch; the one set by program has the highest priority and the one selected with the switch the lowest.

14.15 Related to the spindle speed

These variables are synchronous read/write (R/W).

They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the "Sn" character by the name, logic number or index in the channel of the spindle.

V.A.SREAL.Sn	V.A.SREAL.S

TURNING SPEED		PRG	PLC	INT	Exec
(V.)[n].A.SREAL.Sn	Real spindle speed	R	R	R	Yes

It takes the spindle speed override into account.

When the spindle is stopped, it returns a value of 0. When working in G96 and G97, the speed is in rpm and when working with M19, in º/min.

SPINDLE SPEED IN G97		PRG	PLC	INT	Exec
(V.)[n].A.SPEED.Sn	S active in rpm (G97)	R	R	R	Yes
(V.)[n].PLC.S.Sn	S by PLC in rpm	R	R/W	R	Yes
(V.)[n].A.PRGS.Sn	S by program in rpm	R	R	R	No

The speed may be set by program or by PLC; the one set by PLC has the highest priority.

SPINDLE SPEED IN CSS		PRG	PLC	INT	Exec
(V.)[n].A.CSS.Sn	Active CSS	R	R	R	Yes
(V.)[n].PLC.CSS.Sn	CSS by PLC	R	R/W	R	Yes
(V.)[n].A.PRGCSS.Sn	CSS by program	R	R	R	No

The speed may be set by program or by PLC; the one set by PLC has the highest priority.

MAXIMUM CONSTANT SURFACE SPEED		PRG	PLC	INT	Exec
(V.)[n].A.SLIMIT.Sn	S limit active in Constant Surface Speed mode	R	R	R	Yes
(V.)[n].PLC.SL.Sn	S limit via PLC in Constant Surface Speed mode	R	R/W	R	Yes
(V.)[n].A.PRGSL.Sn	S limit via program in Constant Surface Speed mode	R	R	R	No

These variables only limit the spindle turning speed (rpm) when constant surface speed is active. The maximum Constant Surface Speed may be set by program or by PLC; the one set by PLC has the highest priority.

SPINDLE SPEED OVERRIDE		PRG	PLC	INT	Exec
(V.)[n].A.SSO.Sn	% S active at the CNC	R	R	R	Yes
(V.)[n].A.PRGSSO.Sn	% S by program	R/W	R	R	No
(V.)[n].PLC.SSO.Sn	% S by PLC	R	R/W	R	Yes
(V.)[n].A.CNCSSO.Sn	% S at the switch	R	R	R/W	Yes

The spindle speed override may be set by program, by PLC or with the selector switch; the one set by program has the highest priority and the one set with the selector switch has the lowest.

SPEED IN M19		PRG	PLC	INT	Exec
(V.)[n].A.SPOS.Sn	Active speed in M19	R	R	R	Yes
(V.)[n].PLC.SPOS.Sn	Speed in M19 set by PLC	R	R/W	R	Yes
(V.)[n].A.PRGSPOS.Sn	Speed in M19 by program	R	R	R	No

14.16 Related to the programmed functions

They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the "Xn" character by the name, logic number or index in the channel of the axis.
- Replace the letters "i" and "x" with a number keeping the brackets.

These variables are for synchronous reading (R).

"G" AND "M" FUNCTIONS		PRG	PLC	INT	Exec
(V.)[n].G.GS[i]	Status of the requested "G" function	R	R	R	No
(V.)[n].G.MS[i]	Status of the requested "M" function	R	R	R	No
(V.)[n].G.HGS1..10	Status of the requested "G" (32 bit) functions	R	R	R	No
(V.)[n].G.HGS	History of "G" functions to be displayed	—	—	R	Yes
(V.)[n].G.HMS	History of "M" functions of the master spindle to be displayed	—	—	R	Yes
(V.)[n].G.HMSi	History of "M" functions of the "i" spindle to be displayed	—	—	R	Yes

Variables GS and MS returned a coded value. Each function has a bit that indicates whether the relevant function is active (=1) or not (=0). Examples for status consultation:

(V.)G.GS[1] indicates whether G1 is active (=1) or not (=0)

(V.)G.MS[6] indicates whether M6 is active (=1) or not (=0)

The HGS1..10 variable returns the 32-bit coded status; 1 bit per function. The HGS1 variable corresponds to functions G0 through G31, HGS2 corresponds to G32 through G63 and so on.

The variables HGS and HMS return a coded value. Each function has a bit that indicates whether the relevant variable will be displayed (=1) or not (=0). Bit 0, the least significant bit, corresponds to the G0 or M0 function, bit 1 to G1 or M1 and so on.

These variables are read/write (R/W) and are evaluated during block preparation.

PARAMETERS AND VARIABLES		PRG	PLC	INT
(V.)P.name	Local user variables of the program	R/W	—	—
(V.)S.name	Global user variables of the program	R/W	—	—
(V.)C.(A-Z)	Value of the canned cycle calling parameter	R/W	—	—
(V.)C.CALLP_(A-Z)	Parameter programmed in the call to a canned cycle "0" = It has not been programmed "1" = It has been programmed	R	—	—
(V.)C.P_(A-Z)	Value of the positioning cycle calling parameter	R/W	—	—
(V.)C.P_CALLP_(A-Z)	Parameter programmed in the call to a positioning cycle "0" = It has not been programmed "1" = It has been programmed	R	—	—
(V.)C.PCALLP_(A-Z)	Parameter programmed in a call to a subroutine G18x, #PCALL or #MCALL "0" = It has not been programmed "1" = It has been programmed	R	—	—

The "(V.)P.name" variables maintain their value in local and global subroutines called upon from the program.

The "(V.)S.name" variables maintain their value between programs and after a reset. To initialize these variables, use the instruction #DELETE.

```
G90 G81 Z0 I-15
    V.C.CALLP_Z = 1
    V.C.CALLP_I = 1
    V.C.CALLP_K = 0
    V.C.Z = 0
    V.C.Z = -15
```

```
G160 A30 X100 K10 P6
    V.C.P_CALLP_A = 1
    V.C.P_CALLP_K = 1
    V.C.P_CALLP_R = 0
    V.C.P_A = 30
    V.C.P_X = 100
```

```
#PCALL sub.nc A12.56 D3
    V.C.PCALLP_A = 1
    V.C.PCALLP_D = 1
```

These variables are read-only (R) Synchronous and are evaluated during block preparation.

ARC RELATED		PRG	PLC	INT
(V.)[n].G.R	Arc radius	R	R	R
(V.)[n].G.I/J/K	Arc center coordinates (I, J, K)	R	R	R
(V.)[n].G.CIRERR[i]	Arc center correction	R	R	R

Here are some examples where the starting point is X0 Y0.

Being function G265 active, the CNC recalculates the center if the arc is not exact but it is within tolerance.

```
G2 X120 Y120.001 I100 J20
V.G.R = 101.980881
V.G.I = 100.0004
V.G.J = 20.0004
V.G.CIRERR[1] = -0.000417
V.G.CIRERR[2] = -0.000417
```

Being function G264 active, if the arc is not exact, but it is within tolerances, it executes an arc with the radius calculated from the starting point. It keeps its center.

```
G2 X120 Y120.001 I100 J20
V.G.R = 101.981371
V.G.I = 100
V.G.J = 20
V.G.CIRERR[1] = 0
V.G.CIRERR[2] = 0
```

These variables are read-only (R) Synchronous and are evaluated during block preparation.

MIRROR IMAGE		PRG	PLC	INT
(V.)[n].G.MIRROR	Active mirror images	R	R	R
(V.)[n].G.MIRROR1	Mirror image active on the first axis of the channel	R	R	R
(V.)[n].G.MIRROR2	Mirror image active on the second axis of the channel	R	R	R
(V.)[n].G.MIRROR3	Mirror image active on the third axis of the channel	R	R	R

(V.)[n].G.MIRROR The least significant bits are used, one per axis (1= active and 0=not active). The least significant bit is for the first axis, the next one for the second axis and so on.

SCALING FACTOR		PRG	PLC	INT
(V.)[n].G.SCALE	It indicates the active general scaling factor	R	R	R

POLAR ORIGIN		PRG	PLC	INT
(V.)[n].G.PORGF	Position of the polar origin referred to part zero (abscissa)	R	R	R
(V.)[n].G.PORGS	Position of the polar origin referred to part zero (ordinate)	R	R	R

COORDINATE SYSTEM ROTATION (PATTERN ROTATION)		PRG	PLC	INT
(V.)[n].G.ROTPF	Position of the rotation center referred to part zero (abscissa)	R	R	R
(V.)[n].G.ROTPS	Position of the rotation center referred to part zero (ordinate)	R	R	R
(V.)[n].G.ORGROT	Rotation angle of the coordinate system	R	R	R

AXIS SLAVING		PRG	PLC	INT
(V.)[n].G.LINKACTIVE	Slaving status	R	R	R

BLOCK REPETITION		PRG	PLC	INT
(V.)[n].G.PENDRPT	Number of pending repetitions with #RPT	R	R	R
(V.)[n].G.PENDNR	Number of pending repetitions with NR	R	R	R

(V.)[n].G.PENDRPT and (V.)[n].G.PENDNR indicate the number of repetitions pending to execute. In the first execution, its value is the programmed number of repetitions minus one and in the last one, its value is zero.

These variables are read-only (R) synchronous and are evaluated during execution. They correspond to linear and rotary axes.

PROBING (G100, G101, G102)		PRG	PLC	INT
(V.)[n].A.MEAS.Xn	Measured value. Tool base coordinates	R	R	R
(V.)[n].A.ATIPMEAS.Xn	Measured value. Tool tip coordinates	R	—	—
(V.)[n].G.PLMEAS1	Value measured on the first axis of the channel. Tool tip coordinates	R	—	—
(V.)[n].G.PLMEAS2	Value measured on the second axis of the channel. Tool tip coordinates	R	—	—
(V.)[n].G.PLMEAS3	Value measured on the third axis of the channel. Tool tip coordinates	R	—	—
(V.)[n].A.MEASOF.Xn	Difference with respect to programmed point	R	R	R
(V.)[n].A.MEASOK.Xn	Probing finished "0" = No "1"= Yes	R	R	R
(V.)[n].A.MEASIN.Xn	Coordinate that includes measurement offset	R	R	R
(V.)[n].G.PLMEASOKx	Probing on the plane axes completed	R	—	—

Here is an example where the starting point is X0 and G100 X100 F100 has programmed . The value of (V.)A.MEASIN.Xn is updated (refreshed) when probing with G101.

```
V.A.MEAS.X = 95
V.A.MEASOF.X = -5
V.A.MEASOK.X = 1
```

These variables are read-only (R) synchronous and are evaluated during block preparation.

PROBE		PRG	PLC	INT
(V.)[n].G.ACTIVPROBE	Number of the active probe	R	R	R

These variables are read-only (R) synchronous and are evaluated during execution. These variables correspond to linear and rotary axes; not to spindles.

MOVEMENTS IN MANUAL INTERVENTION		PRG	PLC	INT
(V.)[n].A.MANOF.Xn	Distance moved with G200 or inspection	R	R	R
(V.)[n].A.ADDMANOF.Xn	Distance moved with G201	R	R	R

These values are maintained during the execution of the program even when canceling manual intervention.

These variables are read/write (R/W) synchronous and are evaluated during block preparation. These variables correspond to linear and rotary axes.

KINEMATICS (POSITION)		PRG	PLC	INT
(V.)[n].G.POSROTF	Current position of the main rotary axis	R/W	R/W	R/W
(V.)[n].G.POSROTS	Current position of the secondary rotary axis	R/W	R/W	R/W
(V.)[n].G.TOOLORIF1	Target position for the main rotary axis	R	R	R
(V.)[n].G.TOOLORIS1	Target position for the secondary rotary axis	R	R	R
(V.)[n].G.TOOLORIF2	Target position for the main rotary axis	R	R	R
(V.)[n].G.TOOLORIS2	Target position for the secondary rotary axis	R	R	R

They indicate the position occupied by the rotary axes of the spindle head and the one (target) they must occupy in order to position the tool perpendicular to the defined plane. They are very useful when the spindle is not fully motorized (mono-rotary or manual spindles).

On angular (swivel) spindle heads, there are two possible solutions when calculating this target position:

(V.)G.TOOLORIF1 and (V.)G.TOOLORIS1 indicate the shortest way for the main rotary axis with respect to the zero position.

(V.)G.TOOLORIF2 and (V.)G.TOOLORIS2 indicate the longest way for the main rotary axis with respect to the zero position.

These variables are read-only (R) Synchronous and are evaluated during block preparation. These variables correspond to linear and rotary axes.

INCLINE PLANES		PRG	PLC	INT
(V.)[n].G.CS	Number of the active CS function	R	R	R
(V.)[n].G.ACS	Number of the active ACS function	R	R	R
(V.)[n].G.TOOLCOMP	Compensation function active "1" = RTCP "2" = TLC "3" = None	R	R	R

These variables are read-only (R) synchronous and are evaluated execution time.

DIE RESULTING FROM THE INCLINE PLANE		PRG	PLC	INT
(V.)[n].G.CSMAT1	Die resulting from the incline plane. Element row 1 column 1	R	R	R
(V.)[n].G.CSMAT2	Die resulting from the incline plane. Element row 1 column 2	R	R	R
(V.)[n].G.CSMAT3	Die resulting from the incline plane. Element row 1 column 3	R	R	R
(V.)[n].G.CSMAT4	Die resulting from the incline plane. Element row 2 column 1	R	R	R
(V.)[n].G.CSMAT5	Die resulting from the incline plane. Element row 2 column 2	R	R	R
(V.)[n].G.CSMAT6	Die resulting from the incline plane. Element row 2 column 3	R	R	R
(V.)[n].G.CSMAT7	Die resulting from the incline plane. Element row 3 column 1	R	R	R
(V.)[n].G.CSMAT8	Die resulting from the incline plane. Element row 3 column 2	R	R	R
(V.)[n].G.CSMAT9	Die resulting from the incline plane. Element row 3 column 3	R	R	R
(V.)[n].G.CSMAT10	Offset of the current coordinate system referred to machine zero on the first axis	R	R	R
(V.)[n].G.CSMAT11	Offset of the current coordinate system referred to machine zero on the second axis	R	R	R
(V.)[n].G.CSMAT12	Offset of the current coordinate system referred to machine zero on the third axis	R	R	R

These variables correspond to the transformation matrix from theoretical reference system to the real reference system.

These variables are read-only (R) synchronous and are evaluated during execution.

SYNCHRONIZATION OF CHANNELS		PRG	PLC	INT
(V.)[n].G.MEETST[i]	Status of the MEET type [i] mark in the [n] channel	R	R	R
(V.)[n].G.WAITST[i]	Status of the WAIT type [i] mark in the [n] channel	R	R	R
(V.)[n].G.MEETCH[i]	MEET type mark expected by the [n] channel of the [i] channel	R	R	R
(V.)[n].G.WAITCH[i]	WAIT type mark expected by the [n] channel from the [i] channel	R	R	R

These variables are read-only (R) synchronous and are evaluated during execution. These variables correspond to linear and rotary axes and spindles.

FEED-FORWARD AND AC-FORWARD		PRG	PLC	INT
(V.)[n].A.FFGAIN.Xn	Active percentage of feed-forward	R	R	R
(V.)[n].A.ACFGAIN.Xn	Active percentage of AC-forward	R	R	R

The PLC reading of *ACFGAIN* comes in tenths (x10) The PLC reading of *FFGAIN* comes in hundredths (x100) Ver ***"Access to numeric values from the PLC"*** en la página 358.

14.17 Related to the independent axes

They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the "Xn" character by the name, logic number or index in the channel of the axis.

These variables are read/write (R/W) synchronous and are evaluated during execution.

INDEPENDENT AXES		PRG	PLC	INT
(V.)[n].G.IBUSY	An independent axis is in execution	R	R	R

These variables are read/write (R/W) synchronous and are evaluated during execution. These variables correspond to linear and rotary axes.

INDEPENDENT AXES (POSITIONING)		PRG	PLC	INT
(V.)[n].A.IORG.Xn	Offset for the independent axis	R/W	R/W	R/W
(V.)[n].A.IPRGF.Xn	Feedrate programmed in the independent axis	R	R	R
(V.)[n].A.IPPOS.Xn	Coordinate programmed for the independent axis	R	R	R
(V.)[n].A.ITPOS.Xn	Theoretical coordinate of the independent axis	R	R	R

These variables are read/write (R/W) synchronous and are evaluated during execution. These variables correspond to linear and rotary axes and spindles.

INDEPENDENT AXES (SYNCHRONIZATION)		PRG	PLC	INT
(V.)[n].A.SYNCTOUT.Xn	Maximum time to establish synchronism	R/W	R/W	R/W
(V.)[n].A.SYNCVEL.Xn	Synchronization speed	R/W	R/W	R/W
(V.)[n].A.SYNCPOSW.Xn	Maximum position difference to start correcting it	R/W	R/W	R/W
(V.)[n].A.SYNCVELW.Xn	Maximum velocity difference to start correcting it	R/W	R/W	R/W
(V.)[n].A.SYNCPOSOFF.Xn	Position offset for synchronization	R/W	R/W	R/W
(V.)[n].A.SYNCVELOFF.Xn	Velocity offset for synchronization	R/W	R/W	R/W
(V.)[n].A.GEARADJ.Xn	Fine adjustment of the gear ratio for the synchronization movement	R	R	R

The PLC reading of *GEARADJ* comes in hundredths (x100) Ver ***"Access to numeric values from the PLC"*** en la página 358.

14.18 Related to the machine configuration

They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the "Xn" character by the name, logic number or index in the channel of the axis or of the spindle.
- Replace the letters "i" and "x" with a number keeping the brackets.

These variables are read-only (R) synchronous and are evaluated during execution.

MACHINE CONFIGURATION		PRG	PLC	INT
(V.)G.NUMCH	Number of channels	R	R	R
(V.)[n].G.AXISCH	Name the axes of the channel	—	—	R
(V.)[n].A.ACTCH.Xn	Current channel of the axis or of the spindle	R	R	R
(V.)[n].A.ACTIVSET.Xn	Active axis or spindle set	R	R	R
(V.)[n].G.AXIS	Number of axes of the channel	R	R	R
(V.)[n].G.NAXIS	Number of axes of the channel including the empty positions of the yielded axes	R	R	R
(V.)[n].G.AXISNAMEx	Name of the "x" axis of the channel	R	R	R
(V.)G.GAXISNAMEx	Name of the "x" axis of the system	R	R	R
(V.)[n].G.NSPDL	Number of spindles of the channel	R	R	R
(V.)[n].G.SPDLNAMEx	Name of the "x" spindle of the channel	R	R	R
(V.)G.GSPDLNAMEx	Name of the "x" spindle of the system	R	R	R
(V.)[n].G.MASTERSP	Master spindle of the channel	R	R	R

When parking an axis, it is a good idea to know which axes are available. Variables (V.)[n].G.AXISNAME and (V.).G.GAXISNAME indicate which axes are available. If an axis is not available, this variable returns the "?".

These variables are synchronous read/write (R/W). The variables correspond to linear and rotary axes.

LINEAR AND ROTARY AXIS TRAVEL LIMITS		PRG	PLC	INT	Exec
(V.)[n].A.POSLIMIT.Xn	Positive software limit	R/W	R	R	No
(V.)[n].A.NEGLIMIT.Xn	Negative software limit	R/W	R	R	No
(V.)[n].A.RTPOSLIMIT.Xn	Second positive software travel limit	R/W	R/W	R/W	Yes
(V.)[n].A.RTNEGLIMIT.Xn	Second negative software travel limit	R/W	R/W	R/W	Yes
(V.)[n].G.SOFTLIMIT	Software limits reached	R	R	R	Yes

There are 2 software limits. The CNC applies the most restrictive one.

Variables POSLIMIT and NEGLIMIT correspond to the limits set by machine parameters. When modifying these variables, the CNC assumes those values as the new limits from then on.

They keep their value after a Reset, but are reset when validating the machine parameters and when turning the CNC on. Variables POSLIMIT and NEGLIMINT assume the values of the machine parameters and RTPOSLIMIT and RTNEGLIMIT assume the maximum values.

These variables are read-only (R) synchronous and are evaluated during execution. These variables correspond to linear and rotary axes.

KINEMATICS (DIMENSIONS)		PRG	PLC	INT
(V.)[n].A.HEADOF.Xn	Dimension of the kinematics	R	R	R

It returns the resulting measurement of the active kinematics on that axis. It may be a particular value of DATA (kinematics table) or the combination of several of them depending on the type of kinematics.

These variables are for synchronous reading (R).

WORK PLANE AND AXES		PRG	PLC	INT	Exec
(V.)[n].G.PLANE	Axes making up the work plane	R	R	R	No
(V.)[n].G.PLANE1	First main axis of the channel (abscissa)	R	R	R	No
(V.)[n].G.PLANE2	2nd main axis of the channel (ordinate)	R	R	R	No
(V.)[n].G.PLANE3	Third main axis of the channel	R	R	R	No
(V.)[n].G.PLANELONG	Longitudinal axis of the channel	R	R	R	No
(V.)[n].G.LONGAX	Longitudinal axis	R	R	R	No
(V.)[n].G.PLAXNAME1	Main axes (abscissa)	—	—	R	Yes
(V.)[n].G.PLAXNAME2	Main axes (ordinate)	—	—	R	Yes
(V.)[n].G.PLAXNAME3	Main axes (longitudinal)	—	—	R	Yes

The values returned by (V.)[n].G.PLANE and (V.)[n].G.LONGAX are coded as follows.

X=10 X1=11 X2=12 X3=13 ... X9=19

Y=20 Y1=21 Y2=22 Y3=23 ... Y9=29

Z=30 Z1=31 Z2=32 Z3=33 ... Z9=39

U=40 U1=41 U2=42 U3=43 ... U9=49

V=50 V1=51 V2=52 V3=53 ... V9=59

W=60 W1=61 W2=62 W3=63 ... W9=69

A=70 A1=71 A2=72 A3=73 ... A9=79

B=80 B1=81 B2=82 B3=83 ... B9=89

C=90 C1=91 C2=92 C3=93 ... C9=99

Thus, if the G17 plane is selected, you will obtain:

```
V.G.PLANE = 1020          XY axes (work plane)
V.G.LONGAX = 30           Z axis (longitudinal)
G.PLAXNAME1 = X           (Abscissa axis)
G.PLAXNAME2 = Y           (Ordinate axis)
G.PLAXNAME3 = Z           (Longitudinal axis)
```

These variables are read/write (R/W) synchronous and are evaluated during execution.

ANALOG INPUTS AND OUTPUTS		PRG	PLC	INT
(V.)G.ANAI[i]	[n] input voltage (in volts)	R	R	R
(V.)G.ANAO[i]	[n] output voltage (in volts)	R/W	R/W	R

These variables are read-only (R) synchronous and are evaluated during execution. These variables correspond to linear and rotary axes and spindles.

FEEDBACK INPUTS		PRG	PLC	INT
(V.)[n].A.COUNTER.Xn	Feedback pulses (integer + fraction)	R	R	R
(V.)[n].A.COUNTERST.Xn	Counter status	R	R	R
(V.)[n].A.ASINUS.Xn	Fraction of the A signal	R	R	R
(V.)[n].A.BSINUS.Xn	Fraction of the B signal	R	R	R

For a counter to be active, it must have an analog axis associated with it.

These variables are read/write (R/W) synchronous and are evaluated during execution. They correspond to linear and rotary axes and to the spindle.

RELATED TO THE TANDEM AXIS		PRG	PLC	INT
(V.)[n].A.TPIIN.Xn	Input of the PI of the master axis of the tandem (in rpm)	R	R	R
(V.)[n].A.TPIOUT.Xn	Output of the PI of the master axis of the tandem (in rpm)	R	R	R
(V.)[n].A.TFILTOUT.Xn	Output of the pre-load filter	R	R	R
(V.)[n].A.PRELOAD.Xn	Preload	R/W	R/W	R/W
(V.)[n].A.FTEO.Xn	Velocity command for Sercos	R	R	R
(V.)[n].A.TORQUE.Xn	Current torque in Sercos	R	R	R

The PLC reading of *TORQUE* comes in tenths (x10) Ver ***"Access to numeric values from the PLC"*** en la página 358.

These variables are read/write (R/W) synchronous and are evaluated during block execution. They are valid for linear and rotary axes and for the spindle.

VARIABLES TO BE SET VIA PLC		PRG	PLC	INT
(V.)[n].A.PLCFFGAIN.Xn	% of feed-forward programmed from the PLC	R	R/W	R
(V.)[n].A.PLCACFGAIN.Xn	% of AC-forward programmed from the PLC	R	R/W	R
(V.)[n].A.PLCPROGAIN.Xn	Proportional gain programmed from the PLC	R	R/W	R

In order for the feed-forward and the AC-forward defined this way to be taken into account, they must be active by machine parameter; i.e. by means of machine parameter `FFWTYPE` if it is an analog drive or a simulated drive or parameter `OPMODEP` if it is a Sercos drive.

The values defined by these variables prevail over the ones defined by machine parameters or by program. Setting the variables with a negative value cancels their effect ("0" is a valid value). These variables are initialized neither by a reset nor when validating the parameters.

The PLC will read them in the following units. Ver ***"Access to numeric values from the PLC"*** en la página 358.

The PLC reading of *PLCACFGAIN* comes in tenths (x10)

To set the Z axis variable to ·99.1· from the PLC:

```
()=MOV 991 R1
()=CNCWR(R1,A.PLCACFGAIN.Z,M1000)
```

The PLC reading of *PLCFFGAIN* comes in hundredths (x100)

To set the X axis variable to ·99.12· from the PLC:

```
()=MOV 9912 R1
()=CNCWR(R1,A.PLCFFGAIN.X,M1000)
```

These variables are read-only (R) synchronous and are evaluated in the execution.

VARIABLES FOR ADJUSTING THE POSITION		PRG	PLC	INT
(V.)[n].A.POSINC.Xn	Real position increment of the current sampling period	R	R	R
(V.)[n].A.TPOSINC.Xn	Theoretical position increment of the current sampling period	R	R	R
(V.)[n].A.PREVPOSINC.Xn	Real position increment of the previous sampling period	R	R	R

FINE ADJUSTMENT VARIABLES		PRG	PLC	INT
(V.)[n].A.FEED.Xn	Real instantaneous feedrate value	R	R	R
(V.)[n].A.TFEED.Xn	Theoretical instantaneous feedrate value	R	R	R
(V.)[n].A.ACCEL.Xn	Real instantaneous acceleration value	R	R	R
(V.)[n].A.TACCEL.Xn	Theoretical instantaneous acceleration value	R	R	R
(V.)[n].A.JERK.Xn	Real instantaneous jerk value	R	R	R
(V.)[n].A.TJERK.Xn	Theoretical instantaneous jerk value	R	R	R

14.19 Other variables

They have generic names.

- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the "Xn" character by the name, logic number or index in the channel of the axis or of the spindle.
- Replace the letter "i" with a number keeping the brackets.

These variables are read-only (R) synchronous and are evaluated during execution.

SOFTWARE VERSION		PRG	PLC	INT
(V.)G.VERSION	CNC version and release number	R	R	R

CNC STATUS		PRG	PLC	INT
(V.)[n].G.STATUS	CNC status (brief)	R	R	R
(V.)[n].G.FULLSTATUS	CNC status (detailed)	R	R	R
(V.)G.CNCERR	CNC error number	R	R	R

The information of the CNC status is Binary coded as follows.

STATUS

0000	(0H)	No Ready
0001	(1H)	Ready
0010	(2H)	In execution
0100	(4H)	Interrupted
1000	(8H)	In error

FULLSTATUS

The high portion contains the information of the STATUS variable and its low portion provides further coded information. FULLSTATUS = 0000 (STATUS) 0000 (code).

The list of codes for the low portion of FULLSTATUS is:

0000	(0H)	In Reset
0001	(1H)	In JOG
0010	(2H)	In MDI
0011	(3H)	In program
0100	(4H)	Stopped by M0
0101	(5H)	Stopped by STOP
0110	(6H)	Stopped in Single block mode
1001	(9H)	Checking syntax
1010	(AH)	Block search (without moving the axes)
1011	(BH)	Block search finished. In stand by
1100	(CH)	Calculating execution times
1101	(DH)	In simulation

Example:

In RESET, the low portion of FULLSTATUS is "0" (0000) In JOG mode its value is "1" (0001). In SIMULATION mode is 13 (1101) and so on.

FULLSTATUS=514 (202H) means in execution (0010) + MDI (0010).

These variables are read-only (R) synchronous and are evaluated during execution.

TIMES		PRG	PLC	INT
(V.)G.DATE	Date in year-month-day format (April 25th, 1999 => 990425)	R	R	R
(V.)G.TIME	Time in hours-minutes-seconds format (at 18h 22min 34seg => 182234	R	R	R
(V.)G.CLOCK	Seconds since the CNC was turned on	R	R	R
(V.)[n].G.CYTIME	Part-program execution time (in hundredths of a second)	R	R	R

(V.)[n].G.CYTIME is set to 0 at every new execution even of the same program. It does not measure the time that execution has been stopped.

These variables are read/write (R/W) synchronous and are evaluated during execution.

PARTS COUNTER		PRG	PLC	INT
(V.)[n].G.PARTC	Parts counter	R/W	R/W	R/W
(V.)[n].G.FIRST	First time a program is executed	R	R	R/W

(V.)[n].G.PARTC Is initialized when executing a new program and every time an M30 or an M02 is executed

(V.)[n].G.FIRST Is considered first time execution (=1) every time a new program is selected.

It must be borne in mind that both variables are initialized when changing the program being executed in the channel, even with the #EXEC instruction. For example, when selecting and executing the following program, both variables are initialized. When executing the #EXEC instruction, both variables are re-initialized because the program in execution has changed. If then, this program is executed again, the program in execution changes again and both variables are updated.

```
G0 X100
#EXEC ["program2.nc", 1]
M30
```

In this case, to keep track of how many times the program has been executed, it is recommended to use an arithmetic parameter at the end of the program like a counter.

These variables are read/write (R/W) synchronous and are evaluated during execution.

SINGLE BLOCK, RAPID FUNCTIONS, ETC.		PRG	PLC	INT
(V.)[n].G.SBOUT	Single block function activated	R	R	R
(V.)[n].G.SBLOCK	Single block function requested via keyboard	R	R	R/W
(V.)[n].G.BLKSKIP	Block skip function (\ ) activated	R	R	R/W
(V.)[n].G.M01STOP	Conditional stop function (M01) activated	R	R	R/W
(V.)[n].G.RAPID	Rapid function activated	R	R	R/W

The single block function may be activated or canceled from the keyboard (V.)G.SBLOCK or from the PLC (SBLOCK mark). To activate it, just set one of them high (=1), but to cancel it both must be low (=0).

The conditional stop, block skip and rapid functions are selected via PLC (marks M01STOP, BLKSKIP1 and MANRAPID respectively).

These variables are synchronous read-only (R).

PROGRAM RELATED		PRG	PLC	INT	Exec
(V.)[n].G.FILENAME	Name of the program in execution	—	—	R	Yes
(V.)[n].G.PRGPATH	Path of the program in execution	—	—	R	Yes
(V.)[n].G.FILEOFFSET	Position occupied by the line in execution	R	R	R	Yes
(V.)[n].G.BLKN	Last block executed (number) (If none, value of -1)	R	R	R	No

(V.)[n].G.FILEOFFSET indicates the number of characters existing between the first character of the program and the line being executed. It may be used to highlight the line being executed.

These variables are read/write (R/W) synchronous and are evaluated during execution.

RELATED TO AXES AND SPINDLES		Lin Rot	Spd	PRG PLC	INT
(V.)[n].A.INPOS.Xn	Axis or spindle in position	Yes	Yes	R	R
(V.)[n].A.DIST.Xn	Distance traveled by the axis or spindle	Yes	Yes	R/W	R/W
(V.)G.ENDREP	All the axes are repositioned	—	—	R	R
(V.)[n].G.SPDLREP	M function to be used to reposition the spindle after a tool inspection	—	—	R	R

These variables are read/write (R/W) synchronous and are evaluated during execution.

SIMULATION OF KEYS		PRG	PLC	INT
(V.)G.KEY	Code of the last key accepted by the CNC.	R	R/W	R

(V.)G.KEY To read the last key that has been accepted by the CNC or simulate the keyboard from the PLC by writing in it the code of the desired key.

Keyboard simulation from the PLC.

```
;R110=0 and R111=1
··· = CNCRD(G.KEY, R100, M102)
```

It assigns to register R100 the code of the key pressed last.

```
··· = CNCWR(R101, G.KEY, M101)
```

It indicates to the CNC that a key has been pressed whose code is indicated in register R101.

These variables are synchronous read/write (R/W).

CHANNEL		PRG	PLC	INT	Exec
(V.)[n].G.CNCHANNEL	Channel number	R	R	R	No
(V.)G.FOCUSCHANNEL	Channel with active focus	R	R/W	R/W	Yes

These variables are read-only (R) synchronous and are evaluated during execution.

JOG MOVEMENTS		PRG	PLC	INT
(V.)[n].G.INTMAN	Movements in jog mode are allowed	R	R	R

Jog movements are allowed when the jog mode or the TEACH-IN mode is active, during tool inspection and when functions G200 and G201 are active.

14.20 Alphabetical listing of variables

STATEMENTS AND INSTRUCTIONS 15

There are two types of high level language commands, programming instructions and flow controlling instructions.

Programming instructions

They are defined with the "#" sign followed by the name of the instruction and its associated parameters.

They are used for various operations such as.

- Displaying errors, messages, etc.
- Programming movements referred to machine reference zero (home).
- Executing subroutines, blocks and programs.
- Synchronizing channels.
- Coupling, parking and swapping axes.
- Swapping spindles,
- Machining with the assistance of the C axis.
- Activating collision detection.
- Activating manual intervention.
- Activating high speed machining.
- Etc.

Flow controlling instructions

They are defined with the "$" sign followed by the name of the instruction and its related data.

They are used to make loops and program jumps.

15.1 Programming statements

15.1.1 Display instructions

Error display

It interrupts program execution and displays the indicated error message.

It is programmed using the instruction #ERROR, selecting either the number of the error to be displayed or the error text.

#ERROR — Display an error by selecting its number

It displays the indicated error number and its associated text according to the CNC's error listing. If the indicated error number does not exist in the CNC's error listing, it does not display any text.

The programming format is:

```
#ERROR [<number>]
```

Parameter	Meaning
<number>	Error number.

The error number, that must be an integer, may be defined with a numerical constant, a parameter or an arithmetic expression. When using local parameters, they must be programmed as P0-P25.

```
#ERROR [100000]
#ERROR [P100]
#ERROR [P10+34]
```

#ERROR — Display an error by selecting its text

It displays the indicated error text. If no text is defined, it shows an empty error window.

The programming format is:

```
#ERROR [<text>]
```

Parameter	Meaning
<number>	Error text.

The error text must be defined between quote marks. Certain special characters are defined as follows.

\"	Inserts quote marks in the text.
%%	Inserts the % character.

```
#ERROR ["Message"]
#ERROR ["Parameter \"P100\" is wrong"]
#ERROR ["Difference between P12 and P14 > 40%%"]
```

Including external values in the error text

The identifier %D or %d may be used to insert external values (parameters or variables) into the text. The data whose value is to be displayed must be defined after the text.

```
#ERROR ["Wrong %d value",120]
#ERROR ["Tool %D expired",V.G.TOOL]
#ERROR ["Wrong %D - %D values",18,P21]
```

Up to 5 identifiers %D or %d may be defined, but there must be as many data values as identifiers.

Display warnings

It displays on the screen the indicated warning without interrupting the execution of the program.

It is programmed using the instruction #WARNING, selecting either the number of the warning to be displayed or the text.

#WARNING

Display a warning by selecting its number

It displays the indicated warning number and its associated text according to the CNC's error listing. If the indicated warning number does not exist in the CNC's error listing, it does not display any text.

The programming format is:

```
#WARNING [<number>]
```

Parameter	Meaning
<number>	Warning number.

The warning number, that must be an integer, may be defined with a numerical constant, a parameter or an arithmetic expression. When using local parameters, they must be programmed as P0-P25.

```
#WARNING [100000]
#WARNING [P100]
#WARNING [P10+34]
```

#WARNING

Display a warning by selecting its text

It displays the indicated warning text. If no text is defined, it shows an empty warning window.

The programming format is:

```
#WARNING [<text>]
```

Parameter	Meaning
<number>	Warning text.

The warning text must be defined between quote marks. Certain special characters are defined as follows.

\" Inserts quote marks in the text.

%% Inserts the % character.

```
#WARNING ["Message"]
#WARNING ["Parameter \"P100\" is wrong"]
#WARNING ["Difference between P12 and P14 > 40%%"]
```

Including external values in the error text

The identifier %D or %d may be used to insert external values (parameters or variables) into the text. The data whose value is to be displayed must be defined after the text.

```
#WARNING ["Wrong %d value",120]
#WARNING ["Tool %D expired",V.G.TOOL]
#WARNING ["Wrong %D - %D values",18,P21]
```

Up to 5 identifiers %D or %d may be defined, but there must be as many data values as identifiers.

Message display

The indicated message appears at the top of the screen and it does not interrupt the execution of the program. The message will stay active until a new one is activated (it is not canceled when executing the end-of-program function "M02" or "M30").

The text to be displayed is programmed using the #MSG instruction

#MSG

Display a message

The programming format is:

```
#MSG ["<text>"]
```

Parameter	Meaning
<text>	Message text.

The text of the message must be defined between quote marks. Certain special characters are defined as follows.

\"	Inserts quote marks in the text.
%%	Inserts the % character.

If no text is defined, the message is erased from the screen.

```
#MSG ["User message"]
#MSG ["The tool \"T1\" is a finishing tool"]
#MSG ["80%% of feedrate is being used"]
#MSG [""]
```

Including external values in the error text

The identifier %D or %d may be used to insert external values (parameters or variables) into the message. The data whose value is to be displayed must be defined after the text.

```
#MSG ["Part number %D", P2]
#MSG ["The current tool is %D", V.G.TOOL]
#MSG ["Finishing F=%D mm/min. and S=%D RPM", P21, 1200]
```

Up to 5 identifiers %D or %d may be defined, but there must be as many data values as identifiers.

Graphic area

#DGWZ Defines the graphics area

The graphic area can be defined with the instruction #DGWZ (Define Graphics Work Zone).

The programming format is:

```
#DGWZ [<Xmin>,<Xmax>,<Ymin>,<Ymax>,<Zmin>,<Zmax>]
```

Each of these parameters of this instruction corresponds to each limit of the axes.

Parameter	Meaning
<Xmin>	Lower X axis limit.
<Xmax>	Upper X axis limit.
<Ymin>	Lower Y axis limit.
<Ymax>	Upper Y axis limit.
<Zmin>	Lower Z axis limit.
<Zmax>	Upper Z axis limit.

Both limits may be positive or negative, but the lower limits of an axis must always be smaller than the upper limits for that axis.

The new graphic area defined is kept until another one is defined, modified at the graphics window or the CNC is turned off. On power-up, the CNC assumes the graphic area defined by default.

15.1.2 Enabling and disabling instructions

#ESBLK — **Beginning of the single-block treatment**

#DSBLK — **Beginning of the single-block treatment**

The #ESBLK and #DSBLK instructions activate and deactivate the single block treatment.

When executing the #ESBLK instruction, the CNC executes the following blocks as if they were a single block. This single block treatment remains active until canceled by executing the #DSBLK instruction.

```
G01 X20 Y0 F850
G01 X20 Y20
#ESBLK
   (Beginning of single block)
G01 X30 Y30
G02 X20 Y40 I-5 J5
G01 X10 Y30
G01 X20 Y20
#DSBLK
   (End of single block)
G01 X20 Y0
M30
```

This way, when executing a program in "SINGLE BLOCK" mode, the group of blocks located between #ESBLK and #DSBLK will be executed in a row. In other words, the execution will not be interrupted after each block; it will go on until reaching the #DSBLK instruction.

#ESTOP — **Enable the CYCLE STOP signal**

#DSTOP — **Disable the CYCLE STOP signal**

The #ESTOP and #DSTOP instructions enable and disable the CYCLE STOP signal whether it comes from the operator panel or from the PLC.

When executing the #DSTOP statement, the CNC disables the CYCLE STOP key of the operator panel and the CYCLE STOP signal coming from the PLC. It is kept disabled until canceled by the #ESTOP instruction.

#EFHOLD — **Enable the feed-hold signal**

#DFHOLD — **Disable the feed-hold signal**

The #EFHOLD and #DFHOLD instructions enable and disable the FEED-HOLD coming from the PLC.

When executing the #DFHOLD instruction, the CNC disables the FEED-HOLD input coming from the PLC. It is kept disabled until canceled by the #EFHOLD instruction.

15.1.3 Programming referred to machine reference zero (home)

With this CNC, the movements may be referred to home, temporarily canceling the active zero offsets and tool radius and length compensation.

When moving with respect to machine reference zero, function G70 or G71 programmed by the user is ignored. The movements are carried out in the units (millimeters or inches) set by the OEM (units assumed by the CNC on power-up).

The programmed movements do not admit polar coordinates, nor other kinds of transformations such as mirror image, coordinate (pattern) rotation or scaling factor. While the #MCS function is active, functions for setting a new origin such as G92, G54-G59, G158, G30, etc. are not admitted either.

#MCS **Movement referred to machine zero.**

This instruction may be added to any block containing a movement so it is executed in the machine reference system.

```
G92 X0 Y0
G01 X30 Y30 F850
    (Origin : Part zero)
#MCS X30 Y30
    (Origin : Machine zero (home))
G01 X40 Y40
    (Origin : Part zero)
M30
```

#MCS ON **It activates the machine coordinate system**

#MCS OFF **It cancels the machine coordinate system**

The #MCS ON and #MCS OFF instructions activate and deactivates the machine coordinate system so the movements programmed between both instructions are executed according to the machine reference system.

```
G92 X0 Y0
G01 X50 Y50
#MCS ON
    (Origin : Machine zero (home))
G01 ···
G02 ···
G00 ···
#MCS OFF
    (Origin : Part zero)
G01 X70 Y70
M30
```

Both instructions must be programmed alone in the block.

15.1.4 Subroutine instructions

A subroutine is a set of blocks that, properly identified, may be called upon and executed once or several times from any program position.

There are two types of subroutines, local and global.

- The global subroutine is stored in the CNC memory as an independent program and may be called upon from any other program being executed.
- The local subroutine is defined as part of a program and may only be called upon from the program that contains it.

Since a subroutine may be called upon from the main program (or a subroutine) and another subroutine from this one and so on, the CNC limits the number of these calls to a maximum of 20 nesting levels.

Local subroutine

It must be defined before the body of the program. Several local subroutines may be defined in the same program.

The beginning of a subroutine is defined by "%L<name>", where <name> may be up to 14 characters long and consist of uppercase and lowercase letters as well as numbers (no blank spaces are allowed).

The end of the subroutine is defined with M17, M29 or #RET.

Global subroutine

It is defined as separate program. The name used to store it in the CNC will be the name of the subroutine. The name of a global subroutine does not admit parenthesis because these characters have a special meaning within the part-program.

As opposed to a program that ends with an M30, a global subroutine must end with an M17, M29 or #RET.

Define the path of the subroutines

#PATH **Define the location of the subroutines**

The #PATH instruction may be used to define a predetermined location for searching global subroutines as follows.

```
#PATH ["<text>"]
```

If no path is defined in the subroutine call, the CNC will first look for the subroutine in the path defined using this instruction.

```
#PATH ["C:\Cnc8070\Users\Prg\"]
#PATH ["C:\Cnc8070\Users\"]
```

Subroutine execution

When calling a global subroutine indicating the full path, the search is carried out only in the indicated directory. If the path is not indicated, the search is carried out in this order and in these directories:

1. Directory selected with the #PATH instruction.
2. Directory of the program being executed.
3. Directory defined by machine parameter SUBPATH.

LL

Call to a local subroutine

It calls a local subroutine.

The programming format is:

```
LL <sub>
```

Parameter	Meaning
<sub>	Name of the subroutine

```
LL sub2.nc
```

L

Call to a global subroutine

It calls a global subroutine whose full path may be defined.

The programming format is:

```
L <path><sub>
```

Parameter	Meaning
<path>	Subroutine location.
<sub>	Name of the subroutine

```
L C:\Cnc8070\Users\Prg\sub1.nc
L C:\Cnc8070\Users\sub2.nc
L Sub3.nc
```

#CALL

Call to a local or global subroutine

It calls a subroutine (local or global) whose full path may be defined.

The programming format is:

```
#CALL <path><sub>
```

Parameter	Meaning
<path>	Subroutine location.
<sub>	Name of the subroutine

When there are two subroutines, one local and the other one global, with the same name, the following criteria is applied. If the path has been defined in the call, the CNC will execute the global subroutine, otherwise, it will execute the local one.

```
#CALL C:\Cnc8070\Users\Prg\sub1.nc
#CALL C:\Cnc8070\Users\sub2.nc
#CALL Sub3.nc
```

#PCALL Call to a local or global subroutine initializing parameters

It calls a subroutine (local or global) whose full path may be defined. This type of call allows initializing local parameters of the subroutine.

The programming format is:

```
#PCALL <path><sub> P0 P1 P2...
```

Parameter	Meaning
<path>	Subroutine location.
<sub>	Name of the subroutine

When there are two subroutines, one local and the other one global, with the same name, the following criteria is applied. If the path has been defined in the call, the CNC will execute the global subroutine, otherwise, it will execute the local one.

The values of the call parameters may be defined in two ways. With the parameter name P0, P1, P2, etc. or with the letters A-Z (except the Ñ) in such a way that "A" is the same as P0 and "Z" is the same as P25.

```
#PCALL C:\Cnc8070\Users\Prg\sub1.nc
#PCALL C:\Cnc8070\Users\sub2.nc A12.3 P10=6
#PCALL Sub3.nc A12.3 F45.3 P10=6
```

When using local parameters in the subroutine calls, besides generating a new nesting level, it will also generate a new nesting level for the local parameters; there may be up to 7 nesting levels of local parameters within the 20 nesting levels of the subroutines.

#MCALL Call to a local or global subroutine, being modal, initializing parameters

It calls a subroutine (local or global) whose full path may be defined. This type of call allows initializing local parameters of the subroutine.

With this type of call, the subroutine becomes modal; i.e. the subroutine remains active in successive movements and it is repeated at the end of each move. The modal subroutine is canceled with the instruction #MDOFF.

The programming format is:

```
#MCALL <path><sub> P0 P1 P2...
```

Parameter	Meaning
<path>	Subroutine location.
<sub>	Name of the subroutine

When there are two subroutines, one local and the other one global, with the same name, the following criteria is applied. If the path has been defined in the call, the CNC will execute the global subroutine, otherwise, it will execute the local one.

The values of the call parameters may be defined in two ways. With the parameter name P0, P1, P2, etc. or with the letters A-Z (except the Ñ) in such a way that "A" is the same as P0 and "Z" is the same as P25.

```
#MCALL C:\Cnc8070\Users\Prg\sub1.nc
#MCALL C:\Cnc8070\Users\sub2.nc A12.3 P10=6
#MCALL Sub3.nc A12.3 F45.3 P10=6
```

When using local parameters in the subroutine calls, besides generating a new nesting level, it will also generate a new nesting level for the local parameters; there may be up to 7 nesting levels of local parameters within the 20 nesting levels of the subroutines.

Turning the function into non-modal

The modal subroutine is canceled with the instruction `#MDOFF` and in the following cases:

- After executing an M02 or an M30 and after a RESET.
- When changing the work plane.
- When programming a probing move (G100).
- When modifying the configuration of the axes (#FREE AX, #CALL AX and #SET AX).
- Call to a another subroutine (#PCALL, #CALL, L, LL, G180-189).
- Activating a canned cycle

Considerations about the modal character of the subroutine

The modal subroutine will not be executed in the motion blocks programmed inside the subroutine itself or in the subroutines associated with T or M6. It will not be executed either, when programming a number of block repetitions using a NR value of ·0·.

If a motion block contains a number of repetitions NR other than 0 while a modal subroutine is active, both the movement and the subroutine will be repeated NR times.

If while a subroutine is selected as modal, a block containing the instruction `#MCALL` is executed, the current subroutine will stop being modal and the new selected subroutine will become modal.

#MDOFF

Turning the function into non-modal

The instruction `#MDOFF` means that the subroutine that became modal with the instruction `#MCALL` stops being modal in this block.

15.1.5 Program instructions

It is possible to execute blocks and even programs in a channel from a program being executed in another channel.

#EXEC

Executes a program in the indicated channel

With this instruction, it is possible to execute a program in the indicated channel. The execution of the program starts in the indicated channel in parallel (at the same time) with the block following the #EXEC instruction.

If the channel where it is to be executed is busy, it will issue the relevant error message.

The programming format is:

```
#EXEC [<path><prg>,<channel>]
```

Parameter	Meaning
<path>	File location
<prg>	Program to be executed.
<channel>	Channel where the block is to be executed.

```
#EXEC [PRG1.NC,2]
```
(It executes in channel 2 the indicated program)

```
#EXEC [C:\CNC8070\USERS\PRG\EXAMPLE.NC,3]
```
(It executes in channel 3 the indicated program)

Program location

The program to be executed may be defined by either writing the full path or without it. When a call indicates the full path, it will only look for it in the indicated directory. If the path is not indicated, the search is carried out in this order and in these directories:

1. Directory selected with the #PATH instruction.
2. Directory of the program that executes the #EXEC instruction.
3. Directory defined by machine parameter SUBPATH.

Considerations

If the channel is not indicated or it coincides with the channel where the #EXEC instruction is executed, the indicated program will be executed as a subroutine. In this case, functions M02 and M30 will carry out all the associated actions (initialization, sending to the PLC, etc.) except the one for finishing the program. After executing function M02 or M30, it goes on executing the blocks programmed after the #EXEC instruction.

A program containing the #EXEC instruction may be executed, simulated, syntax checked or searched for a particular block. In all the cases, programs called upon using the #EXEC instruction are executed in the same conditions as the original program

#EXBLK Executes a block in the indicated channel

With this instruction, it is possible to execute a block in the indicated channel. If the channel where it is to be executed is busy, it will issue the relevant error message. After executing the block, the channel goes back to the previous work mode.

The programming format is:

```
#EXBLK [<block>,<channel>]
```

Parameter	Meaning
<block>	Block to be executed.
<channel>	Optional. Channel where the block is to be executed.

```
#EXBLK [G01 X100 F550, 2]
```
(The block is executed in channel 2)

```
#EXBLK [T1 M6]
```
(The block is executed in the current channel)

If the channel is not indicated and the instruction is executed from the program, the block is executed in its own channel. If the channel is not indicated and the instruction is executed in MDI, the block is executed in the active channel.

15.1.6 Electronic axis slaving

Two axes may be slaved to each other so the movement of one of them (slave) depends on the movement of the other one (master).

It is possible to have several axis couplings (slaving) at the same time.

Axis coupling is activated with the `#LINK` instruction and canceled with the `#UNLINK` instruction. When reaching the end of program with a coupled pair of axes, this slaving is canceled after executing an M02 or M30.

Considerations about axis coupling

Although the `#LINK` instruction admits several sets (pairs) of axes, the following limitations must be taken into account:

- The main axes (the first three axes of the channel) cannot be slaves.
- Both axes of the master-slave pair must be of the same type (linear or rotary).
- The master axis of a pair cannot be the slave of another pair.
- An axis cannot be slaved to more than one master axis.

Likewise, a new slaving (coupling) cannot be activated without deactivating the pairs previously slaved.

#LINK

Activate the electronic coupling (slaving) of axes

This instruction defines and activates the electronic coupling of axes. Several couplings may be activated at the same time. When executing this instruction, all the axes defined as slaves depend on their relevant masters. On these slave axes, no movement may be programmed while they stay coupled.

This instruction may also be used to define the maximum following error difference allowed between the master axis and its slave.

The programming format is:

```
#LINK [<master>,<slave>,<error>][...]
```

Parameter	Meaning
<master>	Master axis.
<slave>	Slave axis.
<error>	Optional. Maximum difference allowed between the following errors of both axes.

Programming the amount of error is optional; if not programmed, this test is not carried out. The maximum error will be defined in millimeters or inches for linear axes and in degrees for rotary axes.

```
#LINK [X,U][Y,V,0.5]
#LINK [X,U,0.5][Z,W]
#LINK [X,U][Y,V][Z,W]
```

#UNLINK

Cancel the electronic coupling (slaving) of axes

This instruction deactivates the active axis slaving.

```
#LINK [X,U][Y,V,0.5]
    (Defines and activates axis coupling)
#UNLINK
    (Cancels axis coupling)
```

When reaching the end of program with a coupled pair of axes, this slaving is canceled after executing an M02 or M30.

15.1.7 Axis parking

Some machines, depending on the type of machining, may have two different configurations (axes and spindles). In order to prevent the elements not present in one of the configurations from causing an error message (drives, feedback systems, etc.) the CNC allows parking them.

For example, a machine that swaps a normal spindle with a rectangular one may have the following axes configurations:

- With a normal spindle, X Y Z axes configuration.
- With an orthogonal spindle, X Y Z A B axes configuration.

In this case, when working with the normal spindle, the A and B axes may be parked to ignore their signals.

Several axes and spindles may stay parked at the same time, but they must always be parked (and unparked) one by one.

Use the #PARK instruction to park the axes and spindles and #UNPARK to cancel (unpark) them. The axes and spindles stay parked after executing an M02 or M30, after a RESET and even after turning the CNC off and back on.

Considerations about axis parking

The CNC does not allow parking an axis if it belongs to the main plane, if it is part of the active transformation or is the master/slave of a gantry pair or slaved.

Considerations about spindle parking

The CNC will not allow parking a spindle in the following cases.

- If the spindle is not stopped.
- If the spindle is working as a C axis.
- If G96 or G63 is active and it is the master spindle of the channel.
- If G33 or G95 is active and it is the master spindle of the channel or the spindle is used to synchronize the feedrate.
- If it belongs to a pair of synchronized spindles, be it the master or the slave.

If after parking the spindles, there is only one spindle left in the channel, it will become the new master. If a spindle is unparked and it is the only spindle of the channel, it is also assumed as the new master spindle.

#PARK — Parks an axis

This instruction is used to park the selected axis or spindle. When any of them is parked, the CNC interprets that it no longer belongs to the machine configuration and no longer controls it (ignoring the signals from the drive and from the feedback systems, etc.).

Once an axis or spindle has been parked, the part-program cannot mention it (movements, speed, M functions, etc.).

The programming format is:

```
#PARK <axis/spindle>
```

Each element (axis or spindle) must be parked separately. However, a second element can be parked without having to unpark the first one.

When trying to park an axis or spindle that is already parked, the programming is ignored.

```
#PARK A
```
(It parks the "A" axis)

```
#PARK S2
```
(It parks spindle "S2")

#UNPARK — Unparks an axis

This instruction is used to unpark the selected axis or spindle. When unparking one of them, the CNC interprets that it belongs to the machine configuration and starts controlling it.

The programming format is:

```
#UNPARK <axis/spindle>
```

The axes must be unparked one by one.

When trying to unpark an axis or spindle that is already parked, the programming is ignored.

```
#UNPARK A
```
(It unparks the "A" axis)

```
#UNPARK S
```
(It unparks the "S" spindle)

15.1.8 Axis swapping

Initially, each channel has some axes assigned to it as set by the machine parameters. While executing a program, a channel may release its axes or request new axes. This possibility is determined by machine parameter AXISEXCH, which establishes whether an axis can change channels or whether this change is permanent or not.

A permanent change is maintained after the end of the program, after a reset and on power-up. The original configuration may be restored either by validating the general parameters and restarting or by a part-program that undoes the changes.

It also restore the machine parameter settings if a checksum error occurs when powering up the CNC. .

Knowing if an axis can change channels

Machine parameter AXISEXCH may be consulted using the following variable.

```
V.MPA.AXISEXCH.Xn
```

Replace "Xn" with the name or logic number of the axis.

Value	Meaning
0	It cannot change channels.
1	The change is temporary.
2	The change is permanent.

Knowing in which channel the axis is

It is possible to know in which channel the axis is by using the following variable.

```
V.[n].A.ACTCH.Xn
```

Replace "Xn" with the name or logic number of the axis.

Replace the "n" letter with the channel number.

Value	Meaning
0	It is not in any channel.
1-4	Channel number.

Commands for modifying the axis configuration via program

The following instructions are used to modify the configuration of the axes. It is possible to add or remove axes, change their names and even redefine the main axes of the channel by swapping their names.

Changing the configuration of the axes cancels the active polar origin, the pattern rotation, the mirror image and the scaling factor.

In the configuration of the axes (if G17 is active), the axis that occupies the first position must be the abscissa axis, the second will be the ordinate axis, the third will be the axis perpendicular to the work plane, the forth will be the first auxiliary axis and so on.

#SET AX

Sets the axis configuration

Defines a new axis configuration in the channel. The channel axes not programmed in the instruction and the nonexistent programmed ones will be added. The axes are placed in the channel in the positions as they are programmed in the instruction #SET AX. Optionally, one or several offsets may be applied to the defined axes.

It is the same as programming a #FREE AX of all the axes and then a #CALL AX of all the new axes.

The instruction #SET AX may also be used only to order the existing axes in the channel differently.

The programming format is:

```
#SET AX [<Xn>,...] <offset> <...>
```

Parameter	Meaning
<Xn>	Axes that make up the new configuration. If instead of defining an axis, a zero is written, an empty space (without an axis) appears in this position.
<offset>	Optional. It sets which offset is applied to the axes. Several offsets may be applied.

```
#SET AX [X,Y,Z]
#SET AX [X,Y,V1,0,A]
```

Offset setting

The offsets that may be applied to the axes are identified with the following commands. To apply several offsets, program the relevant commands separated by a blank space.

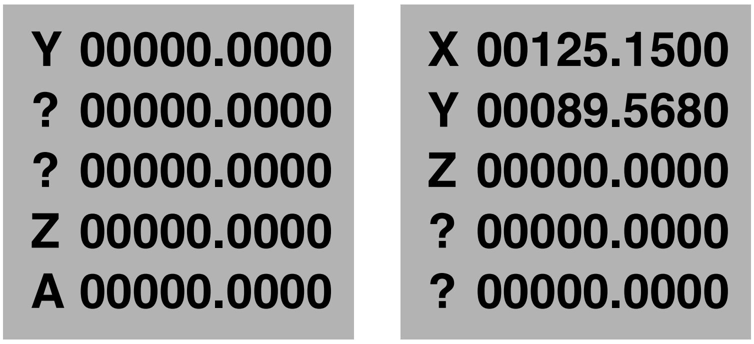
Command	Meaning
ALL	Include all the offsets.
LOCOF	Include the offset of the reference search.
FIXOF	Include the fixture offset.
TOOLOF	Include the tool offset.
ORGOF	Include zero offset.
MEASOF	Include measurement offset.
MANOF	Include the offset of the manual operations.

```
#SET AX [X,Y,Z] ALL
#SET AX [X,Y,V1,0,A] ORGOF TOOLOF
```

If when defining a new configuration only the order of the axes in the channel is swapped, the offsets are ignored.

Screen display

At first, the axes appear ordered as they have been defined in the general machine parameter table (by channels) and then as the swapping is defined.



#SET AX [Y, 0, 0, Z, A] #SET AX [X, Y, Z] FIXOF ORGOF

Screen display of the different configurations. Let us suppose a machine with 5 axes X-Y-Z-A-W.

#CALL AX

Adds an axis to the configuration

it adds one or more axes to the preset configuration and it also allows defining its position. If the axis already exists in the configuration, it is placed in the new position. Optionally, one or several offsets may be applied to the defined axes.

The programming format is:

```
#CALL AX [<Xn>,<pos>...] <offset> <...>
```

Parameter	Meaning
<Xn>	Axes to be added to the configuration. If the axis already exists, it is placed in the new position.
<pos>	Optional. Position of the axis in the new configuration. If not programmed, the axis is placed after the last one. If the position is occupied, the relevant error message will be issued.
<offset>	Optional. It sets which offset is applied to the axes. Several offsets may be applied.

```
#CALL AX [X,A]
```
(It adds the X and A axes to the configuration, after the last existing axis)
```
#CALL AX [V,4,C]
```
(It adds the V axis to position 4 and the C axis after the last one)

Offset setting

The offsets that may be applied to the axes are identified with the following commands. To apply several offsets, program the relevant commands separated by a blank space.

Command	Meaning
ALL	Include all the offsets.
LOCOF	Include the offset of the reference search.
FIXOF	Include the fixture offset.
TOOLOF	Include the tool offset.
ORGOF	Include zero offset.
MEASOF	Include measurement offset.
MANOF	Include the offset of the manual operations.

```
#CALL AX [X] ALL
#CALL AX [V1,4,Y] ORGOF TOOLOF
```

Screen display

At first, the axes appear ordered as they have been defined in the general machine parameter table (by channels) and then as the swapping is defined.

```
Y 00000.0000
X 00000.0000
W 00000.0000
Z 00000.0000
? 00000.0000
```

Axis configuration

#SET AX [Y, 0, 0, Z]
- Y: Abscissa axis.
- Z: First auxiliary axis.

#CALL AX [X,2, W, 3]
- Y: Abscissa axis.
- X: Ordinate axis.
- W: Axis perpendicular to the plane.
- Z: First auxiliary axis.

#FREE AX

Frees an axis from the configuration

Removes the programmed axes from the current configuration. After removing an axis, the position is free, but the order of the axes that remain in the channel does not change.

The programming format is:

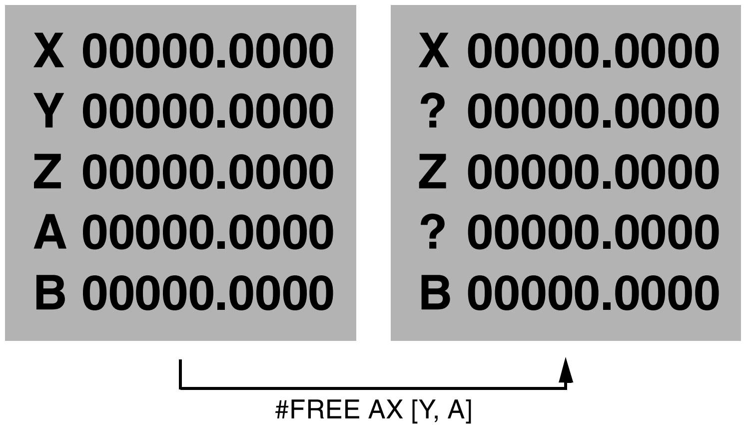
```
#FREE AX [<Xn>,...]
```

Parameter	Meaning
<Xn>	Axis to be removed from the configuration

```
#FREE AX [X,A]
    (It removes the X and A axes from the configuration)
#FREE AX ALL
    (Removes all the axes from the channel)
```

Screen display

At first, the axes appear ordered as they have been defined in the general machine parameter table (by channels) and then as the swapping is defined.



Screen display of the different configurations. Let us suppose a machine with 5 axes X-Y-Z-A-W.

#RENAME AX

Renames the axes

It changes the name of the axes. For each programmed axis pair, the first axis takes the name of the second one. If the second axis is present in the configuration, it takes the name of the first one.

The change of the name of the axes only remains during the execution of the program. The original names of the axes are restored when starting the next program.

The programming format is:

```
#RENAME AX [<Xn1>,<Xn2>][...]
```

Parameter	Meaning
<Xn1>	Axis whose name is to be changed
<Xn2>	new axis name.

```
#RENAME AX [X,X1]
```
(The X axis is now called X1. If X1 already exists in the channel, it is called X)
```
#RENAME AX [X1,Y][Z,V2]
```

15.1.9 Spindle swapping

The CNC can have up to four spindles distributed between the various channels of the system. A channel may have one, several or no spindles associated with it.

Initially, each channel has some spindles assigned to it as set by the machine parameters. While executing a program, a channel may release its spindles or request new spindles. This possibility is determined by machine parameter AXISEXCH, which establishes whether a spindle can change channels or whether this change is permanent or not.

A permanent change is maintained after the end of the program, after a reset and on power-up. The original configuration may be restored either by validating the general parameters and restarting or by a part-program that undoes the changes.

It also restore the machine parameter settings if a checksum error occurs when powering up the CNC. .

Knowing if a spindle can change channels

Machine parameter AXISEXCH may be consulted using the following variable.

```
V.MPA.AXISEXCH.Sn
```

Replace "Sn" with the spindle name.

Value	Meaning
0	It cannot change channels.
1	The change is temporary.
2	The change is permanent.

Knowing in which channel the spindle is

It is possible to know in which channel the spindle is by using the following variable.

```
V.[n].A.ACTCH.Sn
```

Replace "Sn" with the spindle name.

Replace the "n" letter with the channel number.

Value	Meaning
0	It is not in any channel.
1-4	Channel number.

Commands for modifying the spindle configuration via program

The following instructions are used to modify the configuration of the spindles of the channel. It is possible to add or remove spindles, change the name of the spindles and define which one is the master spindle of the channel.

#FREE SP

Frees a spindle from the configuration

Removes the defined spindles from the current configuration.

The programming format is:

```
#FREE SP [<Sn>,...]
#FREE SP ALL
```

Parameter	Meaning
<Sn>	Spindle name.
ALL	Frees all the spindles of the channel.

```
#FREE SP [S]
```
(It removes the spindle S from the configuration)
```
#FREE SP [S1,S4]
```
(It removes spindles S1 and S4 from the configuration)
```
#FREE SP ALL
```
(It removes all the spindles from the configuration)

#CALL SP

Add a spindle to the configuration

It adds one or several spindles to the current configuration. The position of the spindles in the channel is not relevant. To add a spindle to the channel, the spindle must be free; it must not be in another channel.

The programming format is:

```
#CALL SP [<Sn>,...]
```

Parameter	Meaning
<Sn>	Spindle name.

```
#CALL SP [S1]
```
(It adds spindle S1 to the configuration)
```
#CALL SP [S,S2]
```
(It adds spindles S and S2 to the configuration)

#SET SP

Sets the spindle configuration

Defines a new spindle configuration. The spindles existing in the channel and not programmed in #SET SP are removed and those programmed that are not already in the channel will be added.

It is the same as programming a #FREE SP of all the spindles and then a #CALL SP of all the new spindles. The programming format is:

```
#SET SP [<Sn>,...]
```

Parameter	Meaning
<Sn>	Spindle name.

```
#SET SP [S]
```
(Configuring one spindle)
```
#SET SP [S1,S2]
```
(Configuring two spindles)

#RENAME SP — Rename the spindles

It changes the name of the spindles. For each programmed spindle pair, the first spindle takes the name of the second one. If the second spindle is present in the configuration, it takes the name of the first one.

The change of the name of the spindles only remains during the execution of the program. The original names of the spindles are restored when starting the next program.

The programming format is:

```
#RENAME SP [<Sn>,<Sn>][...]
```

Parameter	Meaning
<Sn>	Spindle name.

```
#RENAME SP [S,S1]
#RENAME SP [S1,S2][S3,S]
```

15.1.10 Selecting the master spindle of a channel

#MASTER

Establishes the master spindle of a channel

The master spindle is the main spindle of the channel. It is the spindle that receives the commands when no specific spindle is mentioned.

The programming format is:

```
#MASTER <Sn>
```

Parameter	Meaning
<Sn>	Spindle name.

```
#MASTER S
#MASTER S2
```

If no master spindle is indicated, it assumes one according to the following criteria. In general, whenever a channel has a single spindle, it will be its master spindle.

- If the whole system only has one spindle, it will be the master spindle of the current channel.
- If a spindle is added to a channel that does not have one, it will be the master spindle.
- If a channel releases its master spindle and it has only one spindle left, this one will be its new master spindle.
- If a channel having two spindles but no master spindle releases one of them, the remaining one will be its master spindle.
- At first, in a channel with several spindles, the master spindle will be the one configured by machine parameters.
- If two or more spindles remain in a channel and none of the previous rules may be applied, the master spindle must be defined using the #MASTER instruction.

The same treatment described for adding or removing spindles is applied for parking and unparking spindles.

On startup, it follows the same criteria to decide which is the master spindle of the channel. If this spindle is parked, it will assume the next spindle, if there is one, as master spindle of the channel.

15.1.11 Longitudinal tool axis selection

The longitudinal axis of the tool may be selected using the instruction #TOOL AX.

#TOOL AX

Longitudinal axis selection

This instruction allows to select any machine axis as the new longitudinal axis.

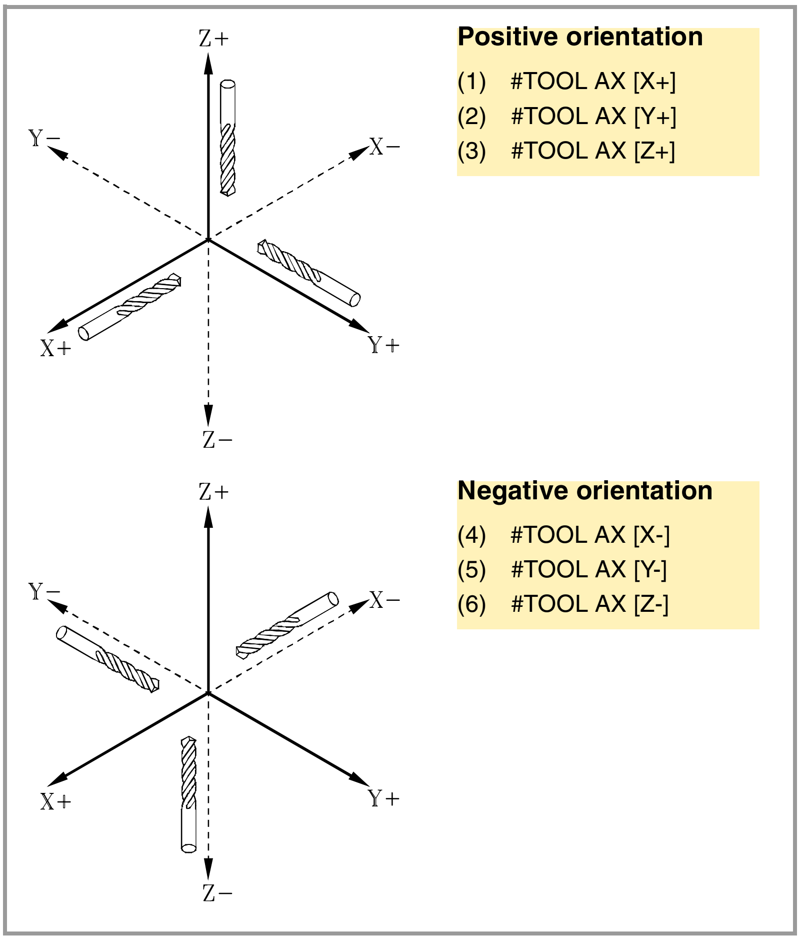
The programming format is:

```
#TOOL AX [<axis><+/->]
```

Parameter	Meaning
<axis>	Longitudinal axis of the tool.
<+/->	Tool orientation.

Tool orientation is established as follows.

+	The tool positions in the positive direction of the axis.
-	The tool positions in the negative direction of the axis.



.

15.1.12 "C" axis: Activate the spindle as "C" axis

The spindle may be activated or deactivated as a "C" axis using the instructions #CAX and #CAX OFF.

To activate an axis or spindle as "C" axis, it must have been defined as such by the machine manufacturer (CAXIS)..

Although the machine may have several spindles defined as "C" axis, only one of them may be active.

#CAX

Activate the spindle as "C" axis

It activates a spindle as "C" axis. The "C" axis will be programmed as if it were a rotary axis (in degrees).

The programming format is:

```
#CAX [<Sn>,<name>]
```

Parameter	Meaning
<Sn>	Optional. Spindle to be activated as C axis.
<name>	Optional. Name of the C axis.

The spindle needs only be indicated when a spindle other than the master is to be activated as a C axis. Otherwise, there is no need to program it.

The <name> parameter sets the name that will identify the C axis. This name will be used in the part program to define the movements. If not programmed, there is a default name in the machine parameters to name it (CAXISNAME).

To activate the master spindle as "C" axis.

```
#CAX
G01 Z50 C100 F100
G01 X20 C20 A50
#CAX OFF
```

If several spindles may be activated as C axis.

```
#CAX [S1,C1]
```

(The spindle "S1" is activated as "C" axis under the name of "C1")

```
G01 Z50 C1=100 F100
G01 X20 C1=20 A50 S1000
#CAX OFF
```

Considerations about working with the C axis

Activating a running spindle as C axis stops the spindle.

While being a spindle active as "C" axis, no speed may be programmed for it.

When activating the spindle as "C" axis, the CNC carries out a home search of the "C" axis.

#CAX OFF

Cancels the C axis

It cancels the C axis and the spindle goes back to working as a normal spindle.

The programming format is:

```
#CAX OFF
```

15.1.13 "C" axis: Machining of the face of the part

Machining on the face is activated and deactivated with the instructions #FACE and #FACE OFF. For this type of machining either a rotary axis or a spindle may be used as "C" axis.

- When using an axis, it is activated as "C" axis after defining the plane with the #FACE instruction.
- When using a spindle, it must be activated as "C" axis in advance using the #CAX instruction.

To activate an axis or spindle as "C" axis, it must have been defined as such by the machine manufacturer (CAXIS). Depending on the machine configuration, it may be necessary to define the relevant kinematics (TYPE 41/42).

Although the machine may have several axes defined as "C" axis, only one of them may be active.

#FACE

It activates machining of the turning side of the part

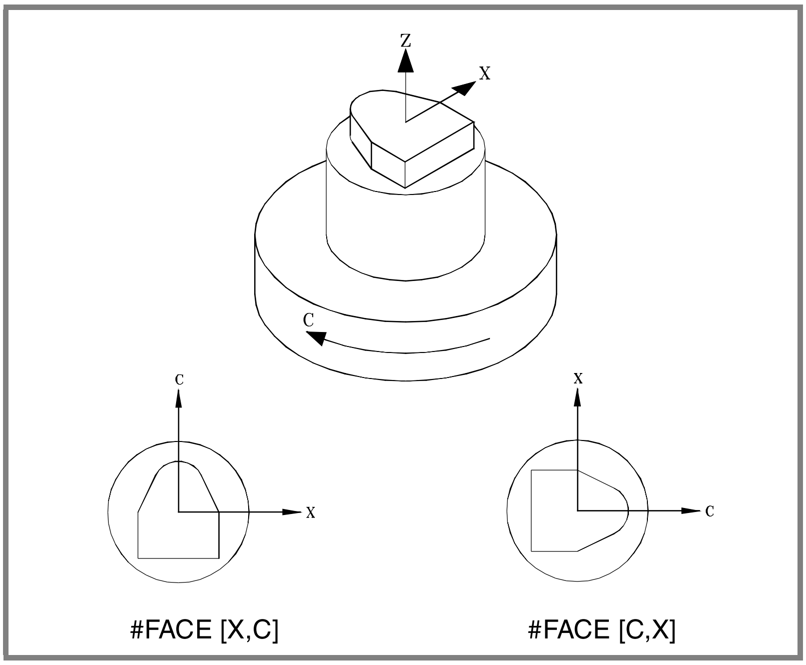
It activates the machining on the face of the part and it defines the work plane. The axis to be activated as "C" axis will be determined by the work plane defined.

The programming format is:

```
#FACE [<abs>,<ord>,<long>]
```

Parameter	Meaning
<abs>	Abscissa axis of the work plane.
<ord>	Ordinate axis of the work plane.
<long>	Optional. Longitudinal axis of the tool.

The "C" axis will be programmed as if it were a linear axis (in millimeters or inches) and the CNC will calculate the corresponding angular movement depending on the selected radius.



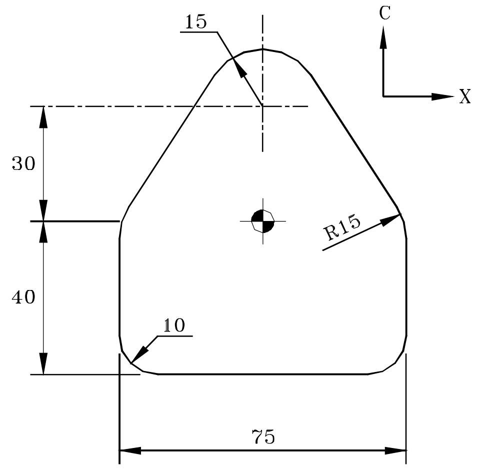
#FACE OFF

It cancels machining on the face of the part

It cancels the machining of the face of the part.

The programming format is:

```
#FACE OFF
```




```
#FACE [X,C]
G90 X0 C-90
G01 G42 C-40 F600
G37 I10
X37.5
G36 I10
C0
G36 I15
X12.56 C38.2
G03 X-12.58 C38.2 R15
G01 X-37.5 C0
G36 I15
C-40
G36 I10
X0
G38 I10
G40 C-90
#FACE OFF
M30
```

15.1.14 "C" axis: Machining of the turning side of the part

Machining on the turning side is activated and deactivated with the instructions #CYL and #CYL OFF. For this type of machining either a rotary axis or a spindle may be used as "C" axis.

- When using an axis, it is activated as "C" axis after defining the plane with the #CYL instruction.
- When using a spindle, it must be activated as "C" axis in advance using the #CAX instruction.

To activate an axis or spindle as "C" axis, it must have been defined as such by the machine manufacturer (CAXIS). Depending on the machine configuration, it may be necessary to define the relevant kinematics (TYPE 43).

Although the machine may have several axes defined as "C" axis, only one of them may be active.

#CYL

It activates machining of the turning side of the part.

It activates the machining of the turning side and it defines the work plane. The axis to be activated as "C" axis will be determined by the work plane defined.

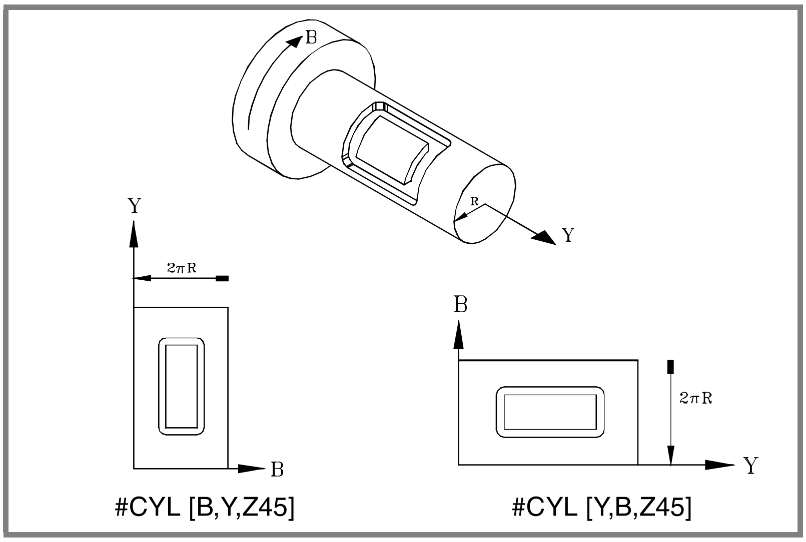
The programming format is:

```
#CYL [<abs>,<ord>,<long><radius>]
```

Parameter	Meaning
<abs>	Abscissa axis of the work plane.
<ord>	Ordinate axis of the work plane.
<long>	Longitudinal axis of the tool.
<radius>	Optional. Radius of the cylinder that will be machined.

Programming the radius is optional. If not programmed, it assumes as the cylinder radius the distance between the rotation center and the tool tip. This makes it possible to develop the surface on cylinders with variable radius without having to indicate the radius.

The "C" axis will be programmed as if it were a linear axis (in millimeters or inches) and the CNC will calculate the corresponding angular movement depending on the selected radius.



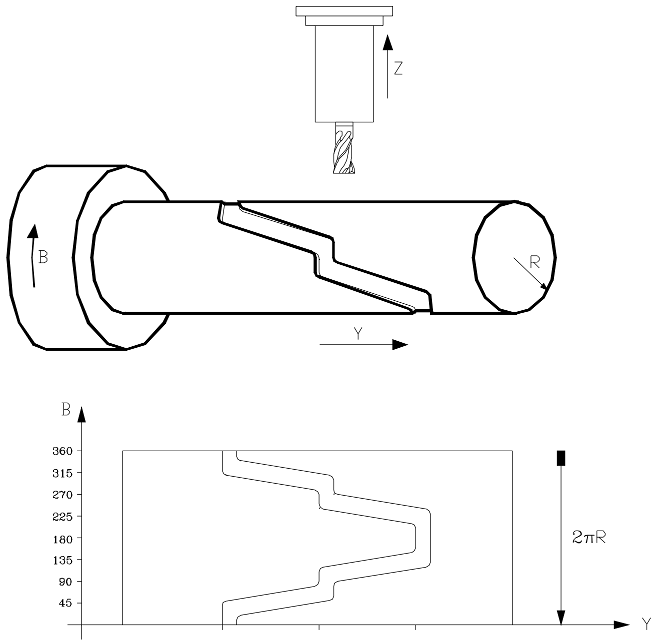
#CYL OFF

It cancels machining of the turning side of the part

It cancels machining of the turning side of the part.

The programming format is:

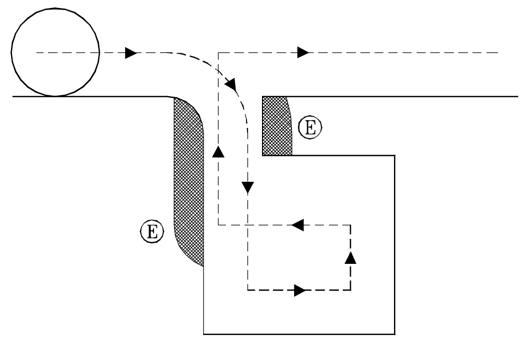
```
#CYL OFF
```




```
#CYL [Y,B,Z20]
G90 G42 G01 Y70 B0
G91 Z-4
G90 B15,708
G36 I3
Y130 B31.416
G36 I3
B39,270
G36 I3
Y190 B54.978
G36 I3
B70,686
G36 I3
Y130 B86.394
G36 I3
B94,248
G36 I3
Y70 B109.956
G36 I3
B125,664
G91 Z4
#CYL OFF
M30
```

15.1.15 Collision detection

With this option, the CNC analyzes in advance the blocks to be executed in order to detect loops (intersections of the profile with itself) or collisions in the programmed profile. The operator may define up to 200 blocks to be analyzed.



The example shows machining mistakes (E) due to a collision in the programmed profile. This type of mistakes may be avoided by using collision detection.

When detecting a loop or a collision, the CNC will not execute the blocks that cause it and the screen will display a warning to let the operator know that the programmed profile has been modified. It will display a warning for each loop or collision eliminated.

The information contained in the blocks removed if it is not a movement in the active plane will be executed (including the movements of the other axes).

Considerations for the collision detecting process.

- Collision detection may be applied even when tool radius compensation is not active.
- Being collision detection active, it is possible to apply zero offsets, coordinate presetting and tool changes. However, home searches and measurements are NOT possible.
- Changing the work plane will interrupt the collision detecting process. The CNC checks for collisions in the blocks stored so far and resumes the process with the new plane starting with the new motion blocks.
- The collision detecting process will be interrupted when programming a instruction (explicit or implicit) that involves synchronizing block preparation and execution (e.g. #FLUSH). The process will resume after executing that instruction.
- Collision detection cannot be activated if a Hirth axis is active and it is part of the main plane. Likewise, while collision detection is active, an axis cannot be activated as a Hirth axis and the work plane cannot be changed if one of the axis is a Hirth axis.

#CD ON

Activate collision detection

It activates the collision detecting process. Being collision detection already active, it lets modify the number of blocks to be analyze.

The programming format is:

```
#CD ON [<blocks>]
```

Parameter	Meaning
<blocks>	Optional. Number of blocks to analyze.

Defining the number of blocks to be analyzed is optional. If not defined, the CNC assumes the maximum (200 blocks). The horizon of blocks may be changed at any time, even while collision detection is active.

#CD OFF

Cancels collision detection

It cancels the collision detecting process.

The process will also be canceled automatically after executing an M02 or M30 and after an error or a reset.

Example of a profile with a loop.

```
#CD ON [50]
G01 X0 Y0 Z0 F750
X100 Y0
Y -50
X90
Y20
X40
Y -50
X0
Y0
#CD OFF
```

Example of profile collision.

```
#CD ON
G01 G41 X0 Y0 Z0 F750
X50
Y -50
X100
Y -10
X60
Y0
X150
Y -100
X0
G40 X0 Y0
#CD OFF
M30
```

15.1.16 Related to manual intervention

With these instructions, it is possible to set the feedrate and the movements in jog mode when manual intervention is active. Manual intervention is activated from the program using functions G200, G201 and G202.

The following may be defined with these instructions:

- The axis feedrate for manual intervention in each work mode (continuous or incremental JOG) and handwheel resolution.

  These values may be defined before or after activating manual intervention and stay active until the end of the program or a reset.

- The limits for the movements made with additive manual intervention. These limits are ignored when executing the movements by program.

  The limits may be defined after activating manual intervention and stay active until it is deactivated.

#CONTJOG

Continuous JOG

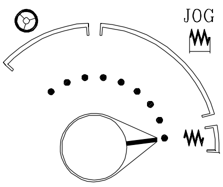
This instruction defines the indicated axis feedrate for continuous JOG.

The programming format is:

```
#CONTJOG [<F>] <Xn>
```

Parameter	Meaning
<F>	Feedrate.
<Xn>	Axis.

The feedrate will be programmed in mm/min. or inches/min. according to the active units.




```
...
N100 #CONTJOG [400] X      Feedrate in continuous JOG. X axis.
N110 #CONTJOG [600] Y      Feedrate in continuous JOG. Eje Y.
N120 G201 #AXIS [X,Y]
...
```

#INCJOG

Incremental JOG

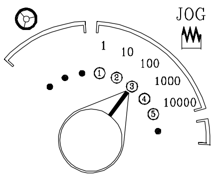
This instruction defines the indicated incremental movement and axis feedrate for each incremental JOG position of the selector switch.

The programming format is:

```
#INCJOG [<inc1>,<F>]...[<inc10000>,<F>] <Xn>
```

Parameter	Meaning
<inc>	Increment in each position of the incremental jog.
<F>	Feedrate in each position of the incremental jog.
<Xn>	Axis.

The feedrate will be programmed in mm/min. or inches/min. and the movement in mm. or inches according to the active units.



...
N100 #INCJOG [[0.1,100][0.5,200][1,300][5,400][10,500]] X
N110 G201 #AXIS [X]
...
The movements and feedrates of the X axis in each position are:
(1) 0.1mm a 100mm/min.
(2) 0.5mm a 200mm/min.
(3) 1mm a 300mm/min.
(4) 5mm a 400mm/min.
(5) 10mm a 500mm/min.

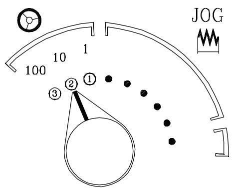
#MPGRESOL Handwheels

This instruction defines the distance per handwheel pulse for the indicated axis at each position of the selector switch.

The programming format is:

```
#MPGRESOL [<pos1>,<pos2>,<pos3>] <Xn>
```

Parameter	Meaning
<pos>	Resolution in each handwheel position.
<Xn>	Axis.



...
N100 #MPGRESOL [0.1,1,10] X
N110 G201 #AXIS [X]
N120 #MPGRESOL [0.5] Y
...
The distance per X axis handwheel pulse in each position is:
(1) 0.1 mm/turn of the handwheel.
(2) 1mm/turn of the handwheel.
(3) 10mm/turn of the handwheel.

This instruction sets the distance per handwheel pulse in a time period equal to the CNC's cycle time. If the feedrate required for this movement exceeds the maximum set by the machine manufacturer, the feedrate will be limited to this value and the axis moving distance will be less than what has been programmed in the instruction.

Example: If a 5 mm move is programmed and the cycle time is 4 msec, the resulting feedrate is 1250 mm/sec. If the maximum feedrate is limited to 1000 mm/sec., the actual distance moved will be 4 mm.

#SET OFFSET

Limits

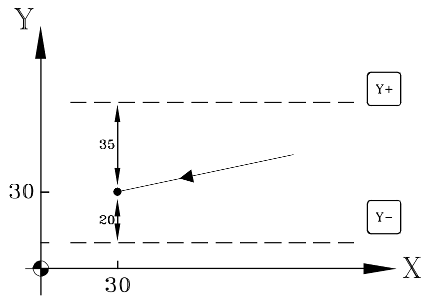
This instruction defines the upper and lower limits of the indicated axis, within which the axes can be jogged during additive manual intervention.

The programming format is:

```
#SET OFFSET [<lower>, <upper>] <Xn>
```

Parameter	Meaning
<lower>	Lower limit.
<upper>	Upper limit
<axis>	Axis.

The limits are referred to the axis position. The lower limit must be less than or equal to zero and the upper limit must be zero or greater than zero.



...

G01 X30 Y30 F550

N100 G201 #AXIS [Y] (Y axis manual interpolation)

N110 #SET OFFSET [-20,35] Y (Y axis limits)

N120 G01 X100 Y45 F400

...

#SYNC POS

Synchronization

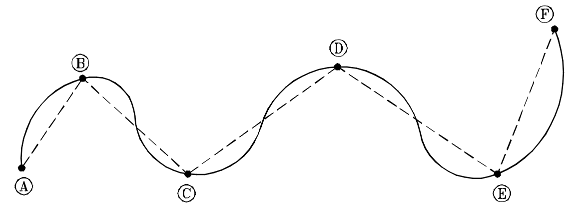
This instruction synchronizes the preparation coordinate with the execution one and assumes the additive manual offset.

The programming format is:

```
#SYNC POS
```

15.1.17 Splines (Akima)

This type of machining adapts the programmed contour to a spline type curve that goes through all the programmed points.



The dashed line shows the programmed profile.

The solid line shows the spline.

The contour to be splined is defined with straight paths (G00/G01). When defining an arc (G02/G03), the spline is interrupted while machining it and it resumes on the next straight path. The transitions between the arc and the spline is done tangentially.

#SPLINE ON **Activate spline adaptation.**

When executing this instruction, the CNC interprets that the points programmed next are part of the spline and begins making the curve.

The programming format is:

```
#SPLINE ON
```

The machining of splines cannot be activated if tool radius compensation (G41/G42) with linear transition between blocks (G136) or viceversa.

#SPLINE OFF **Cancel spline adaptation.**

When executing this instruction, the CNC ends the spline and goes on machining as the path were programmed.

The programming format is:

```
#SPLINE OFF
```

The spline can only be canceled if at least 3 points have been programmed. When defining the initial and final tangents of the spline, 2 points will be enough.

#ASPLINE MODE Select type of tangent.

This instruction sets the type of initial and final tangents of the spline that determines the transition from the previous and to the next path. It is optional; if not defined, the tangent is calculated automatically.

The programming format is:

```
#ASPLINE MODE [<initial>,<final>]
```

Parameter	Meaning
<initial>	Initial tangent.
<final>	Final tangent

The initial and final tangent of the spline may take one of the following values. If not programmed, it assumes a value of 1.

Value	Meaning
1	The tangent is calculated automatically.
2	Tangent to the previous /next block.
3	Tangent as specified.

If defined with a value of ·3·, the initial tangent is defined using the `#ASPLINE STARTTANG` instruction and the final tangent using the `#ASPLINE ENDTANG` instruction If not defined, it applies the values used last.

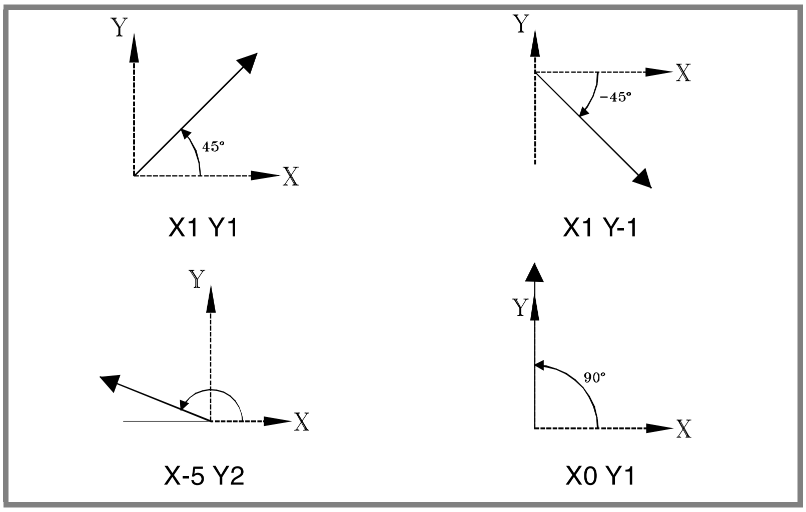
#ASPLINE STARTTANGInitial tangent

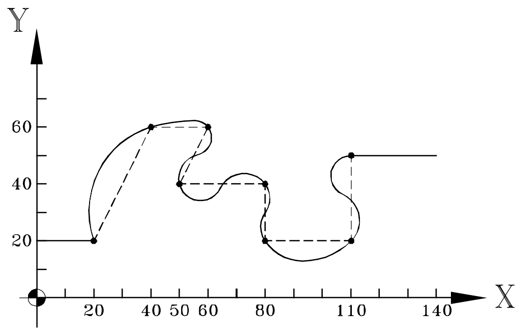
#ASPLINE ENDTANGFinal tangent

These instructions define the initial and final tangents of the spline. The tangent is determined by giving its vectorial direction along the different axes.

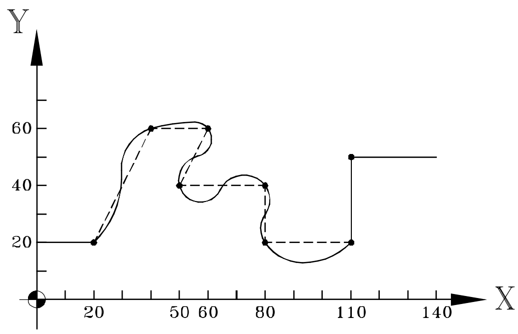
The programming format is:

```
#ASPLINE STARTTANG <axes>
#ASPLINE ENDTANG <axes>
```





N10 G00 X0 Y20	
N20 G01 X20 Y20 F750	(Starting point of the spline)
N30 #ASPLINE MODE [1,2]	(Type of initial and final tangent)
N40 #SPLINE ON	(Activation of the spline)
N50 X40 Y60	
N60 X60	
N70 X50 Y40	
N80 X80	
N90 Y20	
N100 X110	
N110 Y50	(Last point of the spline)
N120 #SPLINE OFF	(Cancellation of the spline)
N130 X140	
N140 M30	



N10 G00 X0 Y20	
N20 G01 X20 Y20 F750	(Starting point of the spline)
N30 #ASPLINE MODE [3.3]	(Type of initial and final tangent)
N31 #ASPLINE STARTTANG X1 Y1	
N32 #ASPLINE ENDTANG X0 Y1	
N40 #SPLINE ON	(Activation of the spline)
. . .	
N120 #SPLINE OFF	(Cancellation of the spline)
N130 X140	
N140 M30	

15.1.18 Polynomial interpolation

The CNC permits interpolating straight lines and arcs and the #POLY instruction may be used to interpolate complex curves, like a parabola.

#POLY

Polynomial interpolation

This type of interpolation lets machining a curve given by a polynomial of up to a 4th degree where the interpolation parameter is the length of the arc.

The programming format is:

```
#POLY [<eje>[a,b,c,d,e]...SP<sp> EP<ep>
```

Parameter	Meaning
<axis>	Axis to interpolate.
a,b,c,d,e	Coefficients of the polynomial.
<sp>	Initial parameter of the interpolation.
<ep>	Final parameter of the interpolation.

One must define all the axes to be interpolated and their corresponding coefficients next to them.

$$a + b\cdot\langle axis\rangle + c\cdot\langle axis\rangle^2 + d\cdot\langle axis\rangle^3 + e\cdot\langle axis\rangle^4$$

Programming a parabola. The polynomial may be represented as follows:

Coefficients of the X axis: [0,60,0,0,0]

Coefficients of the Y axis: [1,0,3,0,0]

Starting parameter: 0

End parameter: 60

```
G0 X0 Y0 Z1 F1000
G1
#POLY [X[0,60,0,0,0] Y[1,0,3,0,0] SP0 EP60]
M30
```

15.1.19 High speed machining

Nowadays, lots of parts are designed using CAD-CAM systems. This type of information is later post-processed to generate a CNC program, usually made up of a large number of very short blocks of several mm or just a few tenths of a micron.

In this type of parts, the CNC must be capable to analyze a large number of points in advance so it can generate a continuous path that goes through (or near) the points of the program while keeping (the best way possible) the programmed feedrate and the restrictions of maximum acceleration, jerk, etc of each axis and of the path.

The command to execute programs made up of lots of small blocks, typical of high speed machining, is carried out with a single instruction `#HSC ON`. The parameter of this instruction is the maximum contour error permitted. From this instruction on, the CNC modifies the geometry through intelligent algorithms for eliminating unnecessary points and automatically generating splines and polynomial transitions between blocks. This way, the contour is traveled at a variable feedrate according to the curvature and the parameters (programmed acceleration and feedrate) but without going beyond the set error limits.

Chordal error setting

As mentioned earlier, the error caused by the CNC between the programmed part and the resulting part is never greater than the programmed value. On the other hand, the CAM system also generates an error when processing the original part and converting the paths into a CNC program. The resulting error may be the sum of the two; therefore, the desired maximum error must be spread between both processes.

Selecting a large chordal error when generating the program and a small chordal error when executing it, the execution is slower and of lower quality. In this case ridges will appear because the CNC perfectly follows the CAM generated polyhedron. It is recommended to post-process at the CAM with a smaller error than the one used for high speed cutting HSC (between 10% and 20%). For example, for a maximum error of 50 microns, we could post-process with an error of 5 or 10 microns and program the rest in the HSC command. This larger margin for the CNC lets you modify the profile while respecting the dynamics of each axis without causing undesired effects like ridges.

Finally, since the CNC works with an accuracy of nanometers, better results may be obtained if the coordinate have 4 or 5 decimals than if they only have 2 or 3. This has no negative effect because the block processing time does not change noticeably. The slight increase in the size of the programs does not represent a problem for storing them thanks to the high capacity hard disk or for transmitting them thanks to Ethernet.

#HCS ON

It activates high speed machining

It activates the HSC mode that lets execute programs made up of lots of small blocks, typical of high speed machining.

The programming format is:

```
#HCS ON [CONTERROR <error>]
```

Parameter	Meaning
<error>	Optional. Maximum contouring error allowed.

The parameter of this instruction is the maximum contour error permitted between the programmed path and the resulting path. Programming it is optional; if not defined, it assumes as maximum contouring error the value set in machine parameter MAXROUND.

#HCS OFF

It cancels high speed machining (cutting)

It cancels the high speed machining mode.

The programming format is:

```
#HCS OFF
```

HSC is also canceled when programming any of the functions G05, G07 or G50.

15.1.20 Acceleration control

The acceleration and the jerk (variation of acceleration) applied on the movements are set by machine parameters. However, those values may be changed from the program using functions G130, G131, G132 and G133.

The #SLOPE instruction sets the influence of the values defined by these functions on how the acceleration will behave.

#SLOPE

It sets the behavior of the acceleration

This instruction sets the influence of the values defined with functions G130-G133 in the behavior of the acceleration.

The programming format is:

```
#SLOPE [<type>,<jerk>,<accel>,<move>]
```

Parameter	Meaning
<type>	Acceleration type.
<jerk>	Optional. It sets the influence of the jerk.
<accel>	Optional. It sets the influence of the acceleration.
<mov>	Optional. It affects the movements in G00.

```
#SLOPE [1,1,0,0]
#SLOPE [1]
#SLOPE [2,,,1]
```

It is not necessary to program all the parameters. The values that each parameter may take are the following.

- The <type> parameter determines the type of acceleration.

Value	Meaning
0	Linear acceleration.
1	Trapezoidal acceleration.
2	Square sine (bell shaped) acceleration.

By default, it assumes a value of ·0·.

The optional <jerk> parameter sets the influence of the Jerk defined with functions G132 and G133. It will only be taken into account in trapezoidal and square-sine type acceleration.

Value	Meaning
0	It modifies the jerk of the acceleration and deceleration stage.
1	It modifies the jerk of the acceleration stage.
2	It modifies the jerk of the deceleration stage.

By default, it assumes a value of ·0·.

- The optional <accel> parameter sets the influence of the acceleration set with functions G130 and G131.

Value	Meaning
0	Always applied.
1	Only applied in the acceleration stage.
2	Only applied in the deceleration stage.

By default, it assumes a value of ·0·.

- The optional <move> parameter determines whether functions G130, G131, G132 and G133 affect the G00 movements or not.

Value	Meaning
0	They affect the G00 movements.
1	They do NOT affect the G00 movements.

By default, it assumes a value of ·0·.

15.1.21 Coordinate transformation

This section describes the instructions related to coordinate transformation. The chapter on *"13 Coordinate transformation"* in this manual describes in further detail how to program these instructions and how they work.

#KIN ID

Kinematics selection

It is used to select the kinematics of the spindle, defining the type of spindle being used, its characteristics and dimensions.

The programming format is:

```
#KIN ID [<num>]
```

Parameter	Meaning
<num>	Optional. Number of kinematics to activate.

If not programmed, the CNC will assume the default kinematics set by the machine manufacturer.

#CS

Define and select the machining coordinate system in an incline plane

#ACS

Define and select the coordinate system of the fixture in an incline plane

With the #CS instruction, up to 5 coordinate systems may be defined, stored, activated an deactivated. With the #ACS instruction, up to 5 fixture coordinate systems may be defined, stored, activated and deactivated.

Both instructions use the same programming format and may be used together or separately. The parameters associated with the instructions have the following meaning.

Parameter	Meaning
[n]	Coordinate system number (1..5).
MODE m	Definition mode used (1..6).
V1...V3	Components of the translation vector.
φ1...φ3	Rotation angles.
0/1	Axis to be aligned in modes 3, 4 and 5.

The programming format is:

- Defines and stores a new #CS or #ACS.

```
#CS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
#ACS DEF [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
```

- Defines, stores and activates a new #CS or #ACS.

```
#CS ON [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
#ACS ON [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
```

- Defines and activates (without storing) a new #CS or #ACS. Only one of them may be defined; to define another one, the previous one must be canceled.

```
#CS ON [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
#ACS ON [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
```

- Cancels and deletes all current #CS or #ACS and defines, stores and activates a new one.

```
#CS NEW [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
#ACS NEW [n] [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
```

- Cancels and deletes all current #CS or #ACS and defines and activates a new one (without storing).

```
#CS NEW [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
#ACS NEW [MODE m, V1, V2, V3, φ1, φ2, φ3, 0/1]
```

- Assumes and stores the current coordinate system as a new #CS or #ACS.

```
#CS DEF ACT [n]
#ACS DEF ACT [n]
```

- Activates the #CS or #ACS stored last.

```
#CS ON
#ACS ON
```

- Activates a stored #CS or #ACS.

```
#CS ON [n]
#ACS ON [n]
```

- Cancels the #CS or #ACS activated last.

```
#CS OFF
#ACS OFF
```

- Cancels all the activated #CS or #ACS.

```
#CS OFF ALL
#ACS OFF ALL
```

#RTCP ON **Activate RTCP (Rotation Tool Center Point) transformation**

#RTCP OFF **Cancel RTCP (Rotation Tool Center Point) transformation**

With RTCP transformation, the tool may be oriented without changing its tip's position on the part.

The programming format is:

```
#RTCP ON
#RTCP OFF
```

The RTCP function cannot be selected while the TLC function is active.

#TOOL ORI **Tool perpendicular to the work plane**

It positions the tool perpendicular to the work plane. This positioning takes place in the first motion block programmed next.

The programming format is:

```
#TOOL ORI
```

#TLC ON **Activate tool length compensation**

#TLC OFF **Activate tool length compensation**

CAD-CAM programs take the tool length into consideration and generate the coordinates for the tool base.

When not having the same size tool for machining, the #TLC function compensates for the difference between the actual (real) tool length and the theoretical one (the one calculated).

The programming format is:

```
#TLC ON [n]
#TLC OFF
```

Parameter	Meaning
[n]	Tool length difference (real - theoretical)

The TLC function cannot be selected while the RTCP function is active.

15.1.22 Definition of macros

Macros may be used to define a program block or part of it with their own names in the format "MacroName" = "CNCblock". Once the macro has been defined, programming "MacroName" will be the same as programming "CNCblock". When executing a macro from the program (or MDI), the CNC will execute its associated program block.

The macros defined via program (or MDI) are stored in a CNC table; this way, they are available for the rest of the programs without having to define them again. This table is initialized on CNC power-up and it can also be initialized from the part-program using the `#INIT MACROTAB` instruction, thus deleting the macros saved.

#DEF: **Macro definition**

Up to 50 different macros may be defined at the CNC. The defined macros may be accessed from any program. When trying to define too many macros, the CNC issues the relevant error message. The macro table may be initialized (erasing all the macros) using the instruction `#INIT MACROTAB`.

The definition of the macro must be programmed alone in the block.

The programming format is:

```
#DEF "MacroName" = "BloqueCNC"
```

Parameter	Meaning
MacroName	Name used to identify the macro in the program. It may have up to 30 characters and consist of letters and numbers.
CNCBlock	Program block. It may be up to 140 characters long.

Several macros may be defined in a block as follows.

```
#DEF "Macro1"="Block1" "Macro2"="Block2" ...
```

```
(Definition of macros)
#DEF "READY"="G0 X0 Y0 Z10"
#DEF "START"="SP1 M3 M41" "STOP"="M05"
(Execution of macros)
"READY" (same as programming G0 X0 Y0 Z10)
P1=800 "START" F450 (same as programming S800 M3 M41)
G01 Z0
X40 Y40
"STOP" (same as programming M05)
```

Definition of arithmetic operations in the macros.

When including arithmetic operations in the definition of a macro, the whole arithmetic operation must be included.

Correct definition of a macro.

```
#DEF "MACRO1"="P1*3"
#DEF "MACRO2"="SIN [\"MACRO1\"]"
```

The following macros are defined wrong.

```
#DEF "MACRO1"="56+"
#DEF "MACRO2"="12"
#DEF "MACRO3="\"MACRO1\"\"MACRO2\""

#DEF "MACRO4"="SIN["
#DEF "MACRO5"="45]"
#DEF "MACRO6="\"MACRO4\"\"MACRO5\""
```

Concatenating of macros. Including macros in the definition of other macros.

The definition of a macro can include other macros. In this case, one of the macros included in the definition must be delimited with the \" characters (`\"macro\"`).

Example1

```
#DEF "MACRO1"="X20 Y35"
#DEF "MACRO2"="S1000 M03"
#DEF "MACRO3"="G01 \"MA1\" F100 \"MA2\""
```

Example 2

```
#DEF "POS"="G1 X0 Y0 Z0"
#DEF "START"="S750 F450 M03"
#DEF "MACRO"="\"POS\" \"START\""
```

#INIT MACROTAB — Resetting the table of macros

When defining a macro from a program (or MDI), it is stored in a CNC table so it is available for all the rest of the programs. This instruction resets the table of macros erasing the ones stored in it.

15.1.23 Block repetition

This instruction may be used to execute a portion of the program defined between two blocks which will be identified with labels. The label of the last block must be programmed alone.

Optionally, it is possible to define the number of repetitions of the execution; if not defined, it will be repeated once.

The number of blocks to be repeated must be defined in the same program or subroutine from which this instruction is executed. They may also be after the program (after function M30)

Up to 20 nesting levels are allowed.

#RPT

Block repetition

The programming format is:

```
#RPT [<blk1>,<blk2>,<n>]
```

Parameter	Meaning
<blk1>	First block.
<blk2>	Last block.
<n>	Optional. Number of repetitions.

Since the labels to identify the blocks may be of two types (number and name), the #RPT instruction may be programmed as follows:

- The label is the block number.

  In the blocks containing the first and last labels, program the ":" character after the block number. This is required in every label that is the target of a jump.

```
N10 #RPT [N50,N70]

N50: G01 G91 X15 F800          (first block)
X-10 Y-10
X20
X-10 Y10
N70:                           (last block)
```

- The label is the block name.

```
N10 #RPT [[BEGIN],[END]]

[BEGIN] G01 G91 F800           (first block)
X-10 Y-10
X20
X-10 Y10
G90
[END]                          (last block)
```

Once the repetition is done, the execution resumes at the block after the one containing the #RPT instruction.

Considerations

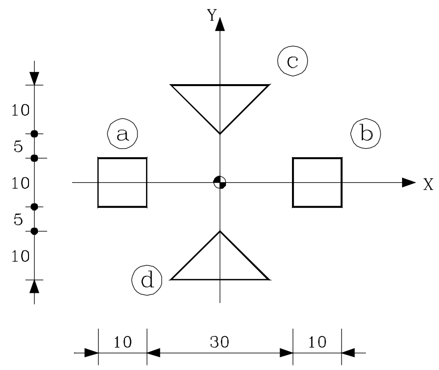
The labels of the first and last blocks must be different. To repeat the execution of a single block, program as follows:

```
N10 #RPT [N10,N20,4]

N10: G01 G91 F800                 (first block)
N20:                              (last block)
```

The execution of a block can also be repeated with the "NR" command. Ver *"Programming in ISO code"* en la página 5.

It is not possible to repeat a group of blocks that close a control loop if the opening of the control loop is not within the instructions being repeated.

```
N10 #RPT [N10,N20]

N10: $FOR P1=1,10,1
G0 XP1
$ENDFOR
G01 G91 F800
N20:
```




```
%PROGRAM
G00 X-25 Y-5
N10: G91 G01 F800          (Definition of profile "a")
X10
Y10
X -10
Y -10
G90
N20:
G00 X15
#RPT [N10, N20]            (Block repetition. Profile "b")
#RPT [[INIT], [END], 2]    (Block repetition. Profiles "c" and "d")
M30

[INIT]
G1 G90 X0 Y10
G1 G91 X10 Y10
```

```
X -20
X10 Y-10
G73 Q180
[END]
```

15.1.24 Communication and synchronization between channels

Each channel may execute its own program simultaneously and independently from other channels. But, besides this, it can also communicate with other channels, transfer information or synchronize in specific points.

The communication takes place on the basis of a number of marks managed by the part-programs of each channel. These marks establish whether the channel is waiting to be synchronized or it may be synchronized, etc.

There are two different ways to synchronize, each offers a different solution.

- Using the `#MEET` instruction.

  The easiest way to synchronize. It stops the execution in all the channels involved in the synchronization.

  The set of marks being used are initialized after executing an M02 or an M30, after a reset or on power-up.

- Using the instructions `#WAIT` - `#SIGNAL` - `#CLEAR`.

  This method is somewhat more complicated than the previous one, but more versatile. It does not stop the execution in all the channels in order to synchronize.

  The set of marks being used are maintained after executing an M02 or an M30, after a reset or on power-up.

The synchronism marks of the two methods are independent from each other. The marks managed by the `#MEET` instruction neither affect nor are affected by the rest of the instructions.

Other ways to synchronize channels

The common arithmetic parameters can also be used to communicate and synchronize channels. By writing a certain value from a channel and later reading it from another channel, it is possible to set the condition to follow up on the execution of a program.

Accessing the variables of a channel from another channel can also be used as a way to communicate.

Swapping axes between channels also makes it possible to synchronize processes, because a channel cannot grab an axis until it has been released by another one.

CHANNEL 1	CHANNEL 2	CHANNEL 3
G1 F1000	X1=0 Y1=0 Z1=0	G1 F1000
S3000 M3	G1 F1000	X2=20 Z2=10
#FREE AX [Z] (Frees the Z axis)	#FREE AX[Z1] (Frees the Z1 axis)	#FREE AX[Z2] (Frees the Z2 axis)
X30 Y0	G2 X1=-50 Y1=0 I-25	X2=100 Y2=50
#CALL AX [Z1,Z2] (It adds the Z1 and Z2 axes)	#CALL AX [Z] (Adds the Z axis)	#CALL AX[Z2] (Recovers the Z2 axis)
X90 Y70 Z1=-30 Z2=-50	G1 X1=50 Z20	G0 X2=0 Y2=0 Z2=0
#FREE AX [Z1,Z2] (Frees the Z1 and Z2 axes)	#FREE AX[Z] (Frees the Z axis)	M30
X0	X1=20	
#CALL AX [Z] (Recovers the Z axis)	#CALL AX [Z1] (Recovers the Z1 axis)	
G0 X0 Y0 Z0	G0 X1=0 Y1=0 Z1=0	
M30	M30	

Consultation variables

The information about the status of the synchronization marks may be consulted using the following variables.

- MEET or WAIT type mark expected by the "n" channel from the "m" channel

  ```
  V.[n].G.MEETCH[m]
  V.[n].G.WAITCH[m]
  ```

  Replace the letters "n" and "m" with the channel number.

- Status of the MEET or WAIT type "m" mark in the "n" channel

  ```
  V.[n].G.MEETST[m]
  V.[n].G.WAITST[m]
  ```

#MEET

It activates the mark indicated in the channel and waits for it to be activated in the rest of the programmed channels.

This instruction, after activating the mark in its own channel, waits for it to also be active in the programmed channels before resuming the execution. Each channel has 10 marks that are numbered from 1 to 10.

Programming the same instruction in several channels, all of them stop and wait for the rest to reach the indicated point before they all resume the execution at the same time from that point on.

The programming format is:

```
#MEET [<mark>, <channel>,...]
```

Parameter	Meaning
<mark>	Synchronization mark that is activated in the channel itself and must be activated in the rest of the channels before going on.
<channel>	Channel or channels where the same mark must be activated.

There is no need to include the number of its own channel in each instruction because the mark is activated when executing the #MEET instruction. However, it is recommended to program it in order to make the program more understandable.

Operation

Programming the same instruction in each channel, all of them are synchronized at that point and the execution resumes from there on. It works as follows.

1. It activates the mark selected in its own channel.
2. It waits for the mark to be activated in all the indicated channels.
3. After synchronizing the channels, it deletes the mark from its own channel and goes on executing the program.

Each channel stops on its #MEET. When the last one of them reaches the command and checks that all the marks are active, the process unlocks for all of them at the same time.

In the following example, it waits for mark ·5· to be active in channels ·1·, ·2· and ·3· to synchronize the channels and resume the execution.

CHANNEL 1	CHANNEL 2	CHANNEL 3
%PRG_1	%PRG_2	%PRG_3
...	...	...
...	#MEET [5,1,2,3]	...
#MEET [5,1,2,3]	...	...
...	...	...
...	...	#MEET [5,1,2,3]
M30	M30	M30

#WAIT

It waits for the mark to be activated in the indicated channel

The #WAIT instruction waits for the indicated mark to be active in the specified channels. If the mark is already active when executing the command, the execution is not interrupted and the program keeps running.

Each channel has 10 marks that are numbered from 1 to 10.

The programming format is:

```
#WAIT [<mark>, <channel>,...]
```

Parameter	Meaning
<mark>	Synchronization mark waited for to be activated.
<channel>	Channel or channels that must activate the mark.

As opposed to the #MEET instruction, it does not activate the indicated mark of its own channel. The marks of the channel are activated using the instruction #SIGNAL.

#SIGNAL

It activates the mark in its own channel

The #SIGNAL instruction activates the indicated marks in its own channel. Each channel has 10 marks that are numbered from 1 to 10. These marks correspond to the #WAIT instructions.

This instruction does not perform any wait; it goes on executing. Once synchronized, the marks are deactivated, if so wished, using the #CLEAR instruction.

The programming format is:

```
#SIGNAL [<mark>,...]
```

Parameter	Meaning
<mark>	Synchronization marks that is activated in the channel.

#CLEAR

It clears the synchronism marks of the channel

This instruction activates the indicated marks in its own channel. If no marks are programmed, it deletes all of them.

The programming format is:

```
#CLEAR
#CLEAR [<mark>,...]
```

Parameter	Meaning
<mark>	Synchronization marks that is deleted in the channel.

In the following example, channels ·1· and ·2· wait for mark ·5· to be active in channel ·3· to synchronize. When mark ·5· is activated in channel ·3·, it resumes the execution in all three channels.

CHANNEL 1	CHANNEL 2	CHANNEL 3
%PRG_1	%PRG_2	%PRG_3
...	...	...
...	#WAIT [5,3]	...
#WAIT [5,3]	...	...
...	...	#SIGNAL [5]
...	...	...
...	...	#CLEAR [5]
M30	M30	M30

15.1.25 Movements of independent axes

This function has a specific manual.

This manual that you are reading now only offers some information about this function. Refer to the specific documentation to obtain further information regarding the requirements and operation of the independent axes.

The CNC has the possibility of executing independent positioning and synchronization. For this type of movements, each CNC axis has an independent interpolator that keeps track of the current position on its own without depending on the tracking of the general interpolator of the CNC.

It is possible to execute an independent movement and general movement simultaneously. The result will be the sum of the two interpolators.

The CNC stores up to a maximum of two independent-motion instructions per axis. The rest of instructions sent when there are two pending execution imply a wait from the part-program.

Restrictions for the independent axes

Any axis of the channel may be moved independently using the associated instructions. However, this function presents the following restrictions.

- A spindle can only move independently when set in axis mode with the instruction `#CAX` . However, it can always be the master of a synchronization.
- A rotary axis may be of any module, but the lower limit must always be zero.
- A Hirth axis cannot move independently.

Synchronizing the interpolators

In order for the incremental movements to take the real coordinate of the machine into account, each interpolator must be synchronized with that real coordinate. The synchronization is done from the part-program using the instruction `#SYNC POS`.

Resetting the CNC synchronizes the theoretical coordinates of both interpolators with the real coordinate. These synchronizations will only be necessary when inserting instructions of both types of interpolators.

Every time the program is initiated or an MDI block is executed, the coordinate of the general interpolator of the CNC is synchronized and every new independent instruction (without any one pending) also synchronizes the coordinate of the independent interpolator.

Influence of the movements in block preparation

None of these blocks interrupt block preparation, but they do interrupt the interpolation. Therefore, it will not blend two blocks, there will be an intermediate one.

Positioning move (#MOVE)

The various types of positioning are programmed with the following instructions.

`#MOVE`	- Absolute positioning move.
`#MOVE ADD`	- Incremental positioning move.
`#MOVE INF`	- Infinite (endless) positioning move.

The programming format for each of them is the following. Optional parameters are indicated between the <> characters.

```
#MOVE <ABS> [Xpos <,Fn> <,blend>]
#MOVE ADD [Xpos <,Fn> <,blend>]
#MOVE INF [X+/- <,Fn> <,blend>]
```

[Xpos]Axis and position to reach

Axis and position to reach. With `#MOVE ABS` it will be defined in absolute coordinates whereas with `#MOVE ADD` it will be defined in incremental coordinates.

The moving direction is determined by the coordinate or the increment programmed. For rotary axes, the moving direction is determined by the type of axis. If normal, via the shortest path; if unidirectional, in the preset direction.

[X+/-]Axis and moving direction

Axis (without coordinate) to position. The sign indicates the moving direction.

It is used with `#MOVE INF` to execute an endless (infinite) movement until the axis limit is reached or until the movement is interrupted.

[Fn]Positioning speed

Positioning feedrate.

Feedrate given in mm/min, inches/min or degrees/min.

Optional parameter. If not defined, it assumes the feedrate set by machine parameter `POSFEED`.

[blend] Dynamic blend with the next block

Optional parameter. The feedrate used to reach the position (dynamic blend with the next block) is defined by an optional parameter.

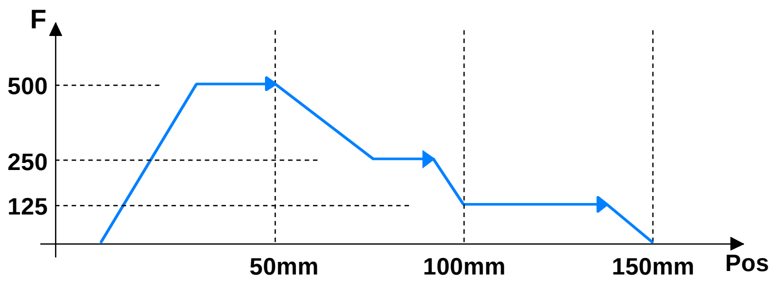
The feedrate used to reach the position is given by one of these elements:

[blend]	Type of dynamic blend
PRESENT	It reaches the indicated position at the positioning feedrate specified for the block itself.
NEXT	It reaches the indicated position at the positioning feedrate specified in the next block.
NULL	The indicated position is reached at zero feedrate.
WAITINPOS	The indicated position is reached at zero feedrate and it waits to be in position before executing the next block.

Programming this parameter is optional. If not programmed, the dynamic blend is carried out according to machine parameter ICORNER as follows.

ICORNER	Type of dynamic blend
G5	According to the setting for the PRESENT value.
G50	According to the setting for the NULL value.
G7	According to the setting for the WAITINPOS value.

```
P100 = 500 (feedrate)
#MOVE [X50, FP100, PRESENT]
#MOVE [X100, F[P100/2], NEXT]
#MOVE [X150, F[P100/4], NULL]
```



Synchronization move (#FOLLOW ON)

The activation and cancellation of the different types of synchronization are programmed with the following instructions.

#FOLLOW ON	- Activates the synchronization movement.
#FOLLOW OFF	- Cancels the synchronization movement.

The programming format for each of them is the following. Optional parameters are indicated between the <> characters.

```
#FOLLOW ON [master, slave, Nratio, Dratio <,synctype>]
#FOLLOW OFF [slave]
```

Executing the #FOLLOW OFF instruction involves eliminating the synchronization speed of the slave. The axis will take some time to brake and the instruction will stay in execution during that time.

[master]Master axis

Name of the master axis.

[slave]Slave axis

Name of the slave axis.

[Nratio]Gear ratio (slave axis)

Numerator of the gear ratio. Turns of the slave axis.

[Dratio]Gear ratio (master axis)

Denominator of the gear ratio. Rotations of the master axis.

[synctype]Type of synchronization

Optional parameter. Indicator that determines whether it is a velocity or position synchronization type.

[synctype]	Type of synchronization
POS	It is a position synchronization.
VEL	It is a velocity synchronization.

Programming it is an option. If not programmed, it executes a velocity synchronization.

```
#FOLLOW ON [X, Y, N1, D1]
#FOLLOW ON [A1, U, N2, D1, POS]
#FOLLOW OFF [Y]
```

15.1.26 Additional programming instructions

#COMMENT BEGIN **Beginning of comment**

#COMMENT END **End of comment**

The instructions #COMMENT BEGIN and #COMMENT END indicate the beginning and end of a comment.

The programming format is:

```
#COMMENT BEGIN
#COMMENT END
```

The blocks programmed between them are considered by the CNC as a single comment and are ignored when executing the program.

```
#COMMENT BEGIN
   P1 : Machining width
   P2 : Machining length.
   P3 : Machining depth
#COMMENT END
```

#FLUSH **Interruption of block preparation**

The CNC reads several blocks ahead (preparation) of the one being executed in order to calculate in advance the path to follow.

The #FLUSH instruction interrupts this block preparation in advance, executes the last prepared blocs, synchronizes the preparation and execution of blocks and then goes on with the program. When resuming, it begins preparing blocks again in advance.

The programming format is:

```
#FLUSH
```

The blocks have data that is analyzed when it is read; to analyze it when it is executed, then use the #FLUSH instruction.

This instruction is useful to evaluate a "block skip" condition at the time of the execution.

```
...
N110 #FLUSH
/N120 G01 X100
...
```

It must be borne in mind that interrupting block preparation may result in compensated paths different from the one programmed, undesired joints when working with very short moves, jerky axis movements, etc.

#WAIT FOR

Wait for an event

This instruction interrupts program execution until the condition is met.

The programming format is:

```
#WAIT FOR [<condition>]
```

```
#WAIT FOR [V.PLC.O[1] == 1]
```

It is possible to compare numbers, parameters or arithmetic expressions whose result is a number.

#SELECT PROBE

Probe selection

It may be used to select the probe.

The programming format is:

```
#SELECT PROBE [<probe>]
```

Value	Meaning
1	Probe 1
2	Probe 2

This instruction is only necessary when having several probes installed on the machine.

#TANGFEED RMIN

Constant tangential feedrate

When applying constant tangential feedrate (G196), with this instruction it is possible to set a minimum radius so this type of feedrate is only applied on arcs whose radius is larger than this minimum.

The programming format is:

```
#TANGFEED RMIN [<radius>]
```

If it is not programmed or it is set to zero, the CNC applies constant tangential feedrate on all the arcs.

The chapter on ***"5 Technological functions"*** in this manual describes in further detail how to operate with constant tangential feedrate.

#ROUNDPAR

Corner rounding

With this instruction, it is possible to select and define the type of corner rounding to be applied. There are 5 types of corner rounding.

This instruction may have up to 6 parameters associated with it and their meanings depend on the type of corner selected.

The chapter on ***"7 Geometry assistance"*** in this manual describes in further detail the types of corner rounding available and how to define them.

#TIME

Timing

It interrupts the execution of the program for the indicated time period (in seconds).

The programming format is:

```
#TIME [&lt;time&gt;]
```

The brackets may be omitted when the time is programmed with a constant or a parameter.

```
P1=20
#TIME [P1+2]
    (22 second dwel)
#TIME 5
    (5 second dwel)
```

The dwell can also be programmed with the G04 function as described in chapter *"8 Additional preparatory functions"* of this manual.

#SCALE

Scaling factor

It may be used to enlarge or shrink the programmed parts. This way, it is possible to make part batches with similar shapes but with different dimensions with a single program. It is the same as function G72.

The programming format is:

```
#SCALE [&lt;scale&gt;]
```

After activating the scaling factor, all the programmed coordinates are multiplied by the defined scaling factor until a new scaling factor is defined or it is canceled (by programming a scaling factor of "1").

The chapter *"7 Geometry assistance"* of this manual offers a more detailed description on how to program the scaling factor.

15.2 Flow controlling instructions

15.2.1 Jump to a block ($GOTO)

$GOTO N<expression>
$GOTO [<label>]

One of the following parameters is defined in this instruction:

<expression>	It may be a number, parameter or arithmetic expression whose result is a number.
<label>	It may be a sequence of up to 14 characters consisting of uppercase and lowercase letters and numbers (neither blank spaces nor quote marks allowed).

This instruction provokes a jump to the block defined with "N<expression>" or "[<label>]", that may be defined at a point before or after the $GOTO instruction. Program execution continues after the jump at the indicated block.

The $GOTO instruction may be programmed in two ways:

- With a block number.

  In these blocks that are the target of a jump, the label must be programmed followed by ":".

Destination	N<number>:
Call	$GOTO N<number> or N<number>:

- With a label.

Destination	[<label>]
Call	$GOTO [<label>]

The call instruction and the destination block must be in the same program or subroutine. There cannot be a jump to a subroutine or between subroutines.

N10 $GOTO N60	N40:	N10 $GOTO [LABEL]
...	...	...
N60: ...	N90 $GOTO N40:	N40 [LABEL]

There cannot be jumps to blocks contained in another instruction ($IF, $FOR, $WHILE, etc.)

Although the flow controlling instructions must be programmed alone in the block, the $GOTO instruction may added to an $IF instruction in the same block. This way, it is possible to exit the blocks contained in an instruction ($IF, $FOR, $WHILE, etc.) without having to end the loop.

```
N10 P0=10
N20 $WHILE P0<=10
N30  G01 X[P0*10] F400
N40  P0=P0-1
N50  $IF P0==1 $GOTO N100
N60  $ENDWHILE
N100: G00 Y30
M30
```

15.2.2 Conditional execution ($IF)

$IF <condition>... $ENDIF

The following parameter is defined in this instruction:

<condition>	It may be a comparison between two numbers, parameters or arithmetic expressions whose result is a number.

This instruction analyzes the programmed condition.

- If the condition is true, it executes the blocks contained between the $IF and $ENDIF instruction.
- If the condition is false, the execution continues at the block after $ENDIF.

```
...
N20 $IF P1==1
 N30...
 N40...
N50 $ENDIF
N60 ...
```

If P1 is equal to 1, it will execute blocks N30 through N40.

If P1 is other than 1, the execution continues at N60.

The $IF instruction always ends with a $ENDIF, except when adding a $GOTO instruction, in which case, it must NOT be programmed.

```
...
N20 $IF P1==1 $GOTO N40
N30...
N40: ...
N50...
```

If P1 is equal to 1, the execution continues at block N40.

If P1 is other than 1, the execution continues at N30.

As an option, the $ELSE and $ELSEIF instructions may be inserted between $IF and $ENDIF.

$IF <condition> ... $ELSE ... $ENDIF

This instruction analyzes the programmed condition.

- If the condition is true, it executes the blocks contained between $IF and $ELSE and the execution continues at the block after $ENDIF.
- If the condition is false, it executes the blocks contained between $ELSE and $ENDIF.

```
N20 $IF P1==1
 N30...
 N40...
N50 $ELSE
 N60...
 N70...
N80 $ENDIF
N90 ...
```

If P1 is equal to 1, it will execute blocks N30 through N40. The execution continues at N90.

If P1 other than 1, the execution continues at N50.

$IF <condition1>... $ELSEIF<condition2>... $ENDIF

This instruction analyzes the following programmed conditions.

- If <condition1> is true, it executes the blocks contained between $IF and $ELSEIF.
- If <condition1> is false, it analyzes <condition2>. If true, it executes the blocks contained between $ELSEIF and $ENDIF (or the next $ELSEIF if any).
- If all the conditions are false, the execution continues at the block after $ENDIF.

As many $ELSEIF instructions as necessary may be programmed.

```
N20 $IF P1==1
N30...
N40...
N50 $ELSEIF P2==[-5]
N60...
N70 $ELSE
N80...
N90 $ENDIF
N100 ...
```

- *If P1 is equal to 1, it will execute blocks N30 through N40. The execution continues at N100.*
- *If P1 is other than 1 and P2 is equal to -5, it executes block N60. The execution continues at N100.*
- *If P1 is other than 1 and P2 is other than -5, it executes block N80 and the execution continues at N100.*

An $ELSE instruction may also be included. In this case, if all the conditions are false, it will execute the blocks contained between $ELSE and $ENDIF.

15.2.3 Conditional execution ($SWITCH)

$SWITCH <expression1>... $CASE<expression2>... $ENDSWITCH

The following parameters are defined in this instruction:

<expression>	It may be a number, parameter or arithmetic expressing whose result is a number.

This instruction calculates the result of <expression1> and executes the blocks contained between the $CASE instruction, whose <expression2> has the same value as the calculated result and the corresponding $BREAK instruction.

The $SWITCH instruction always ends with a $ENDSWITCH.

The $CASE instruction always ends with a $BREAK. As many $CASE instructions as necessary may be programmed.

As an option, a $DEFAULT instruction may be inserted in such a way that if the result of <expression1> does not coincide with the value of any <expression2>, it executes the blocks contained between $DEFAULT and $ENDSWITCH.

```
N20 $SWITCH [P1+P2/P4]
N30 $CASE 10
N40...
N50...
N60 $BREAK
N70 $CASE [P5+P6]
N80...
N90...
N100 $BREAK
N110 $DEFAULT
N120...
N130...
N140 $ENDSWITCH
N150...
```

If the result of the expression [P1+P2/P4].

- *Is "10", it executes blocks N40 through N50. The execution continues at N150.*
- *Is equal to [P5+P6], it executes blocks N80 through N90. The execution continues at N150.*
- *Is other than "10" and [P5+P6], it executes blocks N120 and N130. The execution continues at N150.*

15.2.4 Block repetition ($FOR)

$FOR <n> = <expr1>,<expr2>,<expr3>... $ENDFOR

The following parameters are defined in this instruction.

<n>	It may be an arithmetic parameter of a write variable.
<expr>	It may be a number, parameter or arithmetic expressing whose result is a number.

When executing this instruction, <n> takes the value of <expr1> and it changes its value up to the value of <expr2>, in steps indicated by <expr3>. At each step, it executes the blocks contained between $FOR and $ENDFOR.

```
...
N20 $FOR P1=0,10,2
N30...
N40...
N50...
N60 $ENDFOR
N70...
```

It executes blocks N30 through N50 from P1=0 until P1=10, in steps of 2 (thus 6 times).

```
...
N12 $FOR V.P.VAR_NAME=20,15,-1
N22...
N32...
N42 $ENDFOR
N52...
```

It executes blocks N22 through N32 in steps of -1 (thus 5 times) from V.P.VAR_NAME=20 to V.P.VAR_NAME=15.

The $BREAK instruction lets ending block repetition even if the stop condition is not met. The execution of the program will continue at the block after $ENDFOR.

```
...
N20 $FOR P1= 1,10,1
N30...
N40 $IF P2==2
N50 $BREAK
N60 $ENDIF
N70...
N80 $ENDFOR
...
```

Block repetition stops if P1 is greater than 10, or if P2 = 2.

The $CONTINUE instruction starts the next repetition even when the current one has not finished. The blocks programmed after $CONTINUE up to $ENDFOR will be ignored in this repetition.

15.2.5 Conditional block repetition ($WHILE)

$WHILE <condition>... $ENDWHILE

The following parameter is defined in this instruction:

<condition>	It may be a comparison between two numbers, parameters or arithmetic expressions whose result is a number.

While the condition is true, it executes the blocks contained between $WHILE and $ENDWHILE. The condition is analyzed at the beginning of each new repetition.

```
...
N20 $WHILE P1<= 10
N30 P1=P1+1
N40...
N50...
N60  $ENDWHILE
...
```

While P1 is smaller than or equal to 10, it executes blocks N30 through N50.

The $BREAK instruction lets ending block repetition even if the stop condition is not met. The execution of the program will continue at the block after $ENDWHILE.

```
...
N20 $WHILE P1<= 10
N30...
N40 $IF P2==2
N50 $BREAK
N60 $ENDIF
N70...
N80 $ENDWHILE
...
```

Block repetition stops if P1 is greater than 10, or if P2 = 2.

The $CONTINUE instruction starts the next repetition even when the current one has not finished. The blocks programmed after $CONTINUE up to $ENDWHILE will be ignored in this repetition.

```
...
N20 $WHILE P1<= 10
 N30...
 N40 $IF P0==2
  N50 $CONTINUE
 N60 $ENDIF
 N70...
 N80...
N80 $ENDWHILE
...
```

If P0=2, it ignores blocks N70 through N80 and it starts a new repetition at N20.

15.2.6 Conditional block repetition ($DO)

$DO ... $ENDDO <condition>

The following parameter is defined in this instruction:

<condition>	It may be a comparison between two numbers, parameters or arithmetic expressions whose result is a number.

While the condition is true, it repeats the execution of the blocks contained between $DO and $ENDDO. The condition is analyzed at the end of each repetition, therefore the group of blocks is executed at least once.

```
...
N20 $DO
 N30 P1=P1+1
 N40...
 N50...
N60 $ENDDO P1<=10
N70...
```

Blocks N30 through N50 are executed while P1 is smaller than or equal 10.

The $BREAK instruction lets ending block repetition even if the stop condition is not met. The execution of the program continues at the block after $ENDDO.

```
...
N20 $DO
 N30...
 N40 $IF P2==2
  N50 $BREAK
 N60 $ENDIF
 N70...
N80 $ENDDO P1<= 10
...
```

Block repetition stops if P1 is greater than 10, or if P2 = 2.

The $CONTINUE instruction starts the next repetition even when the current one has not finished. The blocks programmed after $CONTINUE up to $ENDDO will be ignored in this repetition.

```
...
N20 $DO
 N30...
 N40 $IF P0==2
  N50 $CONTINUE
 N60 $ENDIF
 N70...
 N80...
N80 $ENDDO P1<= 10
...
```

If P0=2, it ignores blocks N70 through N80 and it starts a new repetition at N20.

PROBING CANNED CYCLES. 16

The CNC offers the following probing canned cycles.

- Tool radius and length calibration canned cycle.
- Probe calibration canned cycle.
- Surface measuring canned cycle.
- Outside corner measuring canned cycle.
- Inside corner measuring canned cycle.
- Angle measuring canned cycle.
- Corner and angle measuring canned cycle.
- Hole measuring canned cycle.
- Boss measuring canned cycle.

Tool and probe calibrations cycles are carried out in the G17, G18 and G19 planes. The rest of the cycles can also be executed in any plane defined with function G20.

Programming

The canned cycles are programmed using the #PROBE instruction.

The #PROBE instruction calls a probing cycle indicated by a number or any expression resulting in a number. Also, it permits initializing the parameters of that cycle with the values to be used to execute it using the assignment instructions.

When using more than one probe, the probe to be used must be selected before executing the canned cycles. The selection is made using the instruction #SELECT PROBE (*"15.1.26 Additional programming instructions"*).

Considerations

Probing canned cycles are not modal; therefore, they must be programmed every time any of them is to be executed.

The probes used when executing these cycles are:

- Probe located in a fixed position of the machine, used to calibrate tools.
- Probe located in the tool holding spindle; it will be treated as a tool and will be used in the various measuring cycles.

Executing a probing canned cycles does not change the history of the previous "G" functions, except the tool radius compensation functions G41 and G42.

16.1 Tool calibration

It is used to calibrate the tool of the spindle in length or in radius. The following operations are possible with this cycle.

- Calibrate the length of a tool.
- Measure the length wear of a tool.
- Calibrate the radius of a tool.
- Measure the radius wear of a tool.
- Calibrate the radius and length of a tool.
- Measure the radius wear and length wear of a tool.

It requires a table-top probe, installed in a fixed position of the machine and with its sides parallel to the X, Y and Z axes.

If it is the first time that the tool is being calibrated, it is recommended to enter its approximate dimensions in the tool offset table. Once the cycle has concluded, the tool table is updated with the data corresponding to the tool offset that is currently selected.

Programming

The programming format for this cycle is:

#PROBE 1 B I J F K L D S M C N X U Y V Z W

Depending on the operation to be carried out, it will not be necessary to define all the parameters.

Parameters X, U, Y, V, Z, W

They define the probe position. They are optional parameters that usually need not be defined.

- Parameters X-Y-Z refer to the minimum coordinates of the probe on the first axis, second axis and on the axis perpendicular to the plane respectively.
- Parameters U-V-W refer to the maximum coordinates of the probe on the first axis, second axis and on the axis perpendicular to the plane respectively.

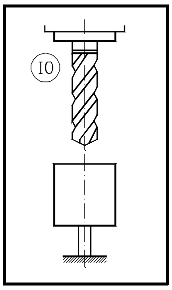
In certain machines, due to lack of repeatability in the mechanical positioning of the probe, the probe must be calibrated again before each calibration.

Instead of re-defining the machine parameters every time the probe is calibrated, those coordinates may be indicated in these parameters. The CNC does not modify the machine parameters and takes into account the coordinates indicated in X, U, Y, V, Z, W only during this calibration.

If any of the X, U, Y, V, Z, W fields is left out, the CNC takes the value assigned to the corresponding machine parameter.

16.1.1 Measure or calibrate the length of a tool.

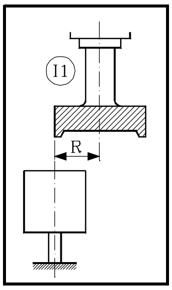
The calibration or measurement may be made on the tool shaft or on its tip. The type of calibration or measurement is selected when calling the canned cycle.



On the tool shaft.

It is useful for drilling tools, spherical mills or tools whose diameter is smaller than the probe surface area.

It is carried out with spindle stopped.



On the tool tip.

It is useful for calibrating tools with several cutting edges (endmills) or tools whose diameter is larger than the probe surface area.

It may be carried out with spindle stopped or climb cutting.

Programming

The programming format depends on the type of operation to be carried out.

- Tool length calibration on its shaft:

  #PROBE 1 B I0 J0 F X U Y V Z W

- Tool length calibration at its end:

  #PROBE 1 B I1 J0 F D S N X U Y V Z W

- Tool length wear measurement on its shaft:

  #PROBE 1 B I0 J1 F L C X U Y V Z W

- Tool length wear measurement on its end:

  #PROBE 1 B I1 J1 F L D S C N X U Y V Z W

B Safety distance. It must be programmed with a positive value greater than 0.

I Type of measurement or calibration.

Value	Meaning
0	Length on the tool shaft.
1	Length on the tool tip.
2	Measure or calibrate the tool radius.
3	Tool length and radius.

If not programmed, the canned cycle will take the value of "I0".

J Operation to be carried out.

Value	Meaning
0	Tool calibration.
1	Wear measurement.

If not programmed, the canned cycle will take the value of "J0".

F Probing feedrate.

L Maximum length wear allowed.

If not programmed, the cycle assumes the value "L0" (the tool will not be rejected due to length wear).

D Radius or distance referred to the tool shaft where it is probed.

If not programmed, the cycle assumes the tool radius value.

S Tool direction and turning speed. The chosen direction must be opposite to the cutting direction (Positive if M3 and negative if M4).

If not programmed, the cycle assumes the value "S0" (calibration with spindle stopped).

C Behavior when exceeding the maximum wear allowed.

Value	Meaning
0	It shows a message of rejected tool and stops the cycle.
1	The cycle replaces the tool with another one of the same family.

If not programmed, the canned cycle will take the value of "C0".

N Number of cutting edges to be measured; The "S" parameter must be defined with a value other than zero.

If not programmed, the cycle assumes the value "N0" (one single measurement).

X, U, Y, V, Z, W

Optional parameters.

After ending the cycle

Once the calibration cycle has ended

It updates global arithmetic parameter P299 and the values assigned to the tool offset selected in the tool table.

P299	(Measured length) - (previous length (L+LW)).
L	Measured length.
LW	0

If the dimension of each edge was requested (parameter "N"), the values will be assigned to global parameters P271 and the following ones.

Once the wear measuring cycle has ended

It compares the measured value with the theoretical length assigned in the table.

- If the maximum wear allowed is exceeded, it sets the "expired tool" indicator and acts as follows:

C0	It issues a "rejected tool" message and interrupts the execution so the user may select another tool.
C1	The cycle replaces the tool with another one of the same family.

- If the measurement difference does not exceed the maximum allowed, it updates global arithmetic parameter P299 and the values assigned to the tool offset selected in the tool table.

P299	Measured length - theoretical length (L).
L	Theoretical length (it maintains the previous value).
LW	Measured length - theoretical length (L).

If the dimension of each edge was requested (parameter "N"), the values will be assigned to global parameters P271 and the following ones.

16.1.2 Measure or calibrate the radius of a tool.

It may be carried out with spindle stopped or climb cutting.

Programming

The programming format depends on the type of operation to be carried out.

- Tool radius calibration:

  #PROBE 1 B I2 J0 F K S N X U Y V Z W

- Measure the radius wear:

  #PROBE 1 B I2 J1 F K S M C N X U Y V Z W

B Safety distance. It must be programmed with a positive value greater than 0.

I Type of measurement or calibration.

Value	Meaning
0	Length on the tool shaft.
1	Length on the tool tip.
2	Measure or calibrate the tool radius.
3	Tool length and radius.

If not programmed, the canned cycle will take the value of "I0".

J Operation to be carried out.

Value	Meaning
0	Tool calibration.
1	Wear measurement.

If not programmed, the canned cycle will take the value of "J0".

F Probing feedrate.

K Probe side used.

Value	Meaning
0	On the X+ side.
1	On the X- side.
2	On the Y+ side.
3	On the Y- side.

If not programmed, the canned cycle will take the value of "K0".

S Tool direction and turning speed. The chosen direction must be opposite to the cutting direction (Positive if M3 and negative if M4).

If not programmed, the cycle assumes the value "S0" (calibration with spindle stopped).

M Maximum radius wear allowed.

If not programmed, the cycle assumes the value "M0" (the tool will not be rejected due to length wear).

C Behavior when exceeding the maximum wear allowed.

Value	Meaning
0	It shows a message of rejected tool and stops the cycle.
1	The cycle replaces the tool with another one of the same family.

If not programmed, the canned cycle will take the value of "C0".

N Number of cutting edges to be measured; The "S" parameter must be defined with a value other than zero.

If not programmed, the cycle assumes the value "N0" (one single measurement).

X, U, Y, V, Z, W

Optional parameters.

After ending the cycle

Once the calibration cycle has ended

It updates global arithmetic parameter P298 and the values assigned to the tool offset selected in the tool table.

P298	(Measured radius) - (previous radius (R+RW)).
R	Measured radius.
RW	0

If the dimension of each edge was requested (parameter "N"), the values will be assigned to global parameters P251 and the following ones.

Once the wear measuring cycle has ended

It compares the measured value with the theoretical radius assigned in the table.

- If the maximum wear allowed is exceeded, it sets the "expired tool" indicator and acts as follows:

C0	It issues a "rejected tool" message and interrupts the execution so the user may select another tool.
C1	The cycle replaces the tool with another one of the same family.

- If the measurement difference does not exceed the maximum allowed, it updates global arithmetic parameter P298 and the values assigned to the tool offset selected in the tool table.

P298	Measured radius - theoretical radius (R).
R	Theoretical radius (it maintains the previous value).
RW	Measured radius - theoretical radius (R).

If the dimension of each edge was requested (parameter "N"), the values will be assigned to global parameters P251 and the following ones.

16.1.3 Measure or calibrate the radius and length of a tool.

It may be carried out with spindle stopped or climb cutting.

Programming

The programming format depends on the type of operation to be carried out.

- Tool radius calibration:

  #PROBE 1 B I3 J0 F K D S N X U Y V Z W

- Measure the radius wear:

  #PROBE 1 B I3 J1 F K L D S M C N X U Y V Z W

B Safety distance. It must be programmed with a positive value greater than 0.

I Type of measurement or calibration.

Value	Meaning
0	Length on the tool shaft.
1	Length on the tool tip.
2	Measure or calibrate the tool radius.
3	Tool length and radius.

If not programmed, the canned cycle will take the value of "I0".

J Operation to be carried out.

Value	Meaning
0	Tool calibration.
1	Wear measurement.

If not programmed, the canned cycle will take the value of "J0".

F Probing feedrate.

K Probe side used.

Value	Meaning
0	On the X+ side.
1	On the X- side.
2	On the Y+ side.
3	On the Y- side.

If not programmed, the canned cycle will take the value of "K0".

L Maximum length wear allowed.

If not programmed, the cycle assumes the value "L0" (the tool will not be rejected due to length wear).

D Radius or distance referred to the tool shaft where it is probed.

If not programmed, the cycle assumes the tool radius value.

S Tool direction and turning speed. The chosen direction must be opposite to the cutting direction (Positive if M3 and negative if M4).

If not programmed, the cycle assumes the value "S0" (calibration with spindle stopped).

M Maximum radius wear allowed.

If not programmed, the cycle assumes the value "M0" (the tool will not be rejected due to length wear).

C Behavior when exceeding the maximum wear allowed.

Value	Meaning
0	It shows a message of rejected tool and stops the cycle.
1	The cycle replaces the tool with another one of the same family.

If not programmed, the canned cycle will take the value of "C0".

N Number of cutting edges to be measured; The "S" parameter must be defined with a value other than zero.

If not programmed, the cycle assumes the value "N0" (one single measurement).

X, U, Y, V, Z, W

Optional parameters.

After ending the cycle

Once the calibration cycle has ended

It updates global arithmetic parameters P298, P299 and the values assigned to the tool offset selected in the tool table.

P298	Measured radius - previous radius (R+RW).
P299	Measured length - previous length (L+LW).
R	Measured radius.
L	Measured length
RW	0
LW	0

If the dimension of each edge was requested (parameter "N"), the lengths will be assigned to global arithmetic parameters P271 and the following ones, and the radii to global arithmetic parameters P251 and the following ones.

Once the wear measuring cycle has ended

It compares the measured length with the theoretical one assigned in the table.

- If the maximum wear allowed is exceeded, it sets the "expired tool" indicator and acts as follows:

C0	It issues a "rejected tool" message and interrupts the execution so the user may select another tool.
C1	The cycle replaces the tool with another one of the same family.

- If the measurement difference does not exceed the maximum allowed, it updates global arithmetic parameter P299 and the values assigned to the tool offset selected in the tool table.

P299	Measured length - theoretical length (L).
L	Theoretical length (it maintains the previous value).
LW	Measured length - theoretical length (L).

It compares the measured value with the theoretical radius assigned in the table.

- If the maximum wear allowed is exceeded, it sets the "expired tool" indicator and acts as follows:

C0	It issues a "rejected tool" message and interrupts the execution so the user may select another tool.
C1	The cycle replaces the tool with another one of the same family.

- If the measurement difference does not exceed the maximum allowed, it updates global arithmetic parameter P298 and the values assigned to the tool offset selected in the tool table.

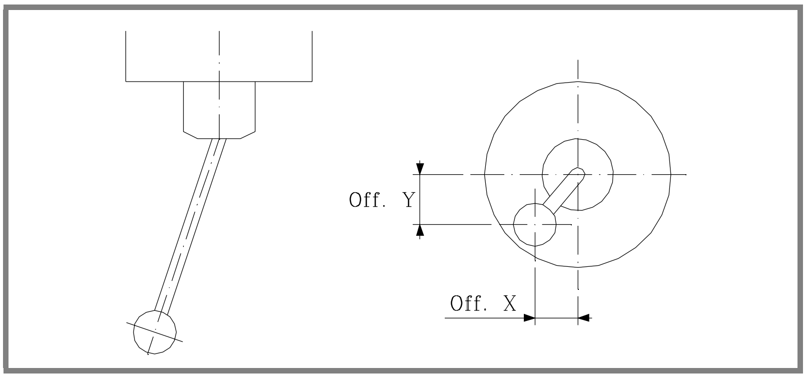
P298	Measured radius - theoretical radius (R).
R	Theoretical radius (it maintains the previous value).
RW	Measured radius - theoretical radius (R).

if it requested the dimension of each edge (parameter "N"), the lengths will be assigned to global arithmetic parameters from P271 on and the radii to global arithmetic parameters from P251 on.

16.2 Probe calibration

It is used to calibrate the probe located in the spindle. This probe must be previously length-calibrated, it will be the one used in measuring-with-probe canned cycles.

The cycle measures the deviation of the probe ball with respect to the shaft of the tool holder using a previously machined hole of known center and dimensions.



Each measuring probe being used will be treated by the CNC like any other tool. The fields of the tool offsets for each probe will have the following meaning:

R	Probe ball radius. This value must be entered manually in the table.
L	Probe length. This value will be assigned by the tool length calibration cycle.
Off. X	Deviation of the probe ball shaft with respect to the tool holder along the abscissa axis. This value will be assigned by this cycle.
Off. Y	Deviation of the probe ball shaft with respect to the tool holder along the ordinate axis. This value will be assigned by this cycle.

Follow these steps to calibrate it:

1. Once the probe characteristics have been checked, manually enter the offset for the ball radius value (R).
2. After selecting the relevant tool number and the offset number, execute the tool length calibration cycle; it will update the "L" value and initialize the "Off. Z" value to 0.
3. Execution of the probe calibration canned cycle updating the values of "Off. X" and "Off. Y".

Programming

The programming format for this cycle is:

#PROBE 2 X Y Z B J E H F

X Real hole center coordinate along the abscissa axis.

Y Real hole center coordinate along the ordinate axis.

Z Real hole center coordinate along the axis perpendicular to the plane.

B Safety distance. It must be programmed with a positive value greater than 0.

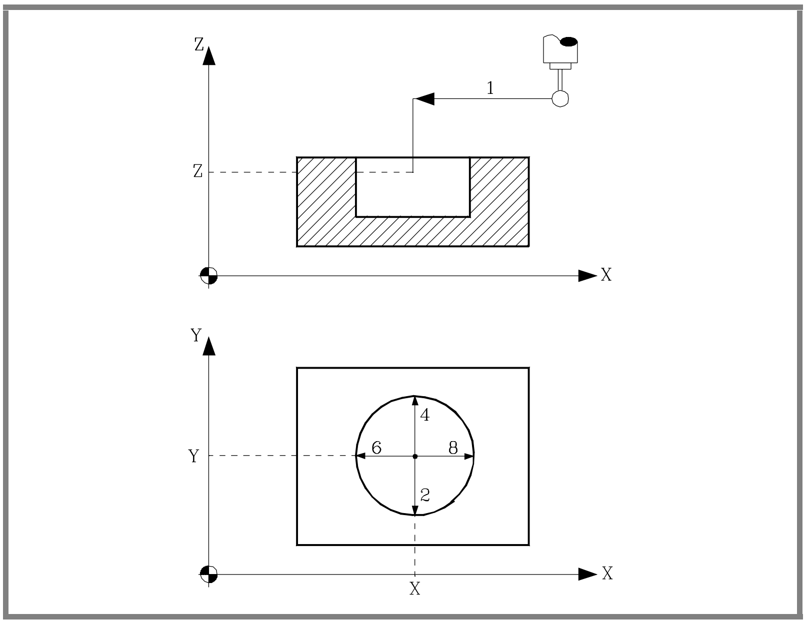
J Real diameter of the hole. It must be programmed with a positive value greater than 0.

E Distance the probe withdrawals after the first probing movement. It must be programmed with a positive value greater than 0.

H Feedrate for the first probing movement.

F Probing feedrate.

Basic operation



1. Approach movement.

   Probe's rapid movement (G00) from the cycle calling point to the center of the hole.

   The approach movement is made in two stages:

   ·1· Movement in the main work plane.

   ·2· Movement along the longitudinal axis.

2. Probing movement.

The probing movement is made in two stages:

·1· Probe movement along the ordinate axis at the feedrate indicated by (H) until the probe signal is received.

The maximum distance for the probing movement is "B+(J/2)", if once this distance has been traveled, the CNC has not yet received the probe signal, it will display the corresponding error message and will stop the movement of the axes.

·2· Probe's rapid withdrawal (G00) the distance indicated by (E).

·3· Probe movement along the ordinate axis at the feedrate indicated by (F) until the probe signal is received.

3. Withdrawal movement.

Probe's rapid movement (G00) from the probed point to the real center of the hole.

4. Second probing movement.

It is similar to the previous one.

5. Withdrawal movement.

Probe's rapid movement (G00) from the probed point to the real center of the hole along the ordinate axis.

6. Third probing movement.

It is similar to the previous ones.

7. Withdrawal movement.

Probe's rapid movement (G00) from the probed point to the real center of the hole.

8. Fourth probing movement.

It is similar to the previous ones.

9. Withdrawal movement.

This movement consists of:

·1· Probe's rapid movement (G00) from the probed point to the real center of the hole.

·2· Movement along the longitudinal axis up to the coordinate of the cycle calling point along that axis.

·3· Movement in the main work plane up to the cycle calling point.

Once the cycle has ended, the CNC will have modified in the tool table the values of "Off X" and "Off. Y" for the tool offset currently selected.

Likewise, in arithmetic parameters P298 and P299, it returns the best value to be assigned to axis machine parameter PROBEDELAY for the abscissa and ordinate axes.

16.3 Surface measuring canned cycle

A probe must be used mounted in the spindle, previously calibrated using canned cycles #PROBE 1 and #PROBE 2.

With this cycle, it is possible to correct the value of the offset of the tool used to machine the surface. This correction only takes place when the measuring error exceeds a programmed value.

Programming

The cycle may be programmed in any work plane. Depending on the work plane, the theoretical coordinates of the cycle may be defined in several ways:

- In the active work plane, except if the plane is formed by any of the axes A-B-C.

#PROBE 3 X50 Y65 Z15 …	Main axes X-Y-Z
#PROBE 3 X1=50 Y2=65 Z1=15 …	Main axes X1-Y2-Z1

- Using parameters X-Y-Z. When the plane is not formed by these axes, these parameters are interpreted as coordinates in the first axis, second axis and axis perpendicular to the work plane respectively.

#PROBE 3 X50 Y65 Z15 …	Main axis X1-B-C

The programming format in the G17, G18 or G19 plane is:

#PROBE 3 X Y Z B K F C D L

X Theoretical X coordinate of the point over which the measurement will be taken.

Y Theoretical Y coordinate of the point over which the measurement will be taken.

Z Theoretical Z coordinate of the point over which the measurement will be taken.

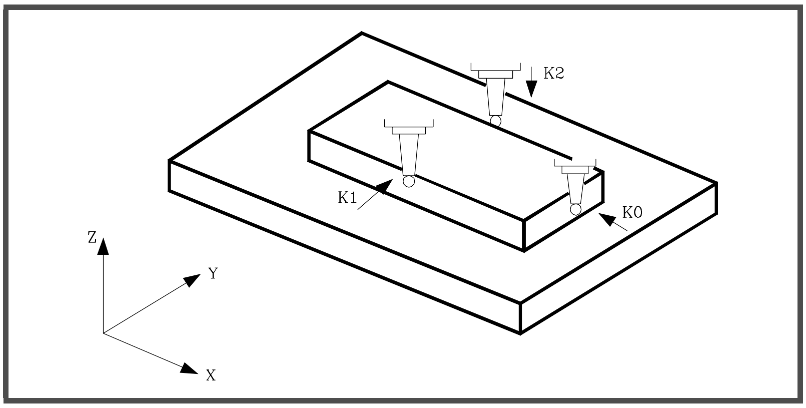
B Safety distance. It must be programmed with a positive value greater than 0.

When calling the cycle, the probe must be located, with respect to the point to be measured, at a greater distance than this value

K Axis used to measure the surface.

Value	Measuring axis
0	With the abscissa axis of the work plane.
1	With the ordinate axis of the work plane.
2	With the longitudinal axis of the work plane.

If not programmed, the canned cycle will take the value of "K0".



F Probing feedrate.

C It indicates where the probing cycle must end.

Value	Meaning
0	The probe returns to the point from where the cycle was called.
1	The cycle ends over the measured point. The longitudinal axis returns to the coordinate corresponding to the point where the cycle was called.

If not programmed, the canned cycle will take the value of "C0".

T Tool whose offset is to be corrected.

If not programmed, the CNC will interpret that it is the tool used for machining.

D Number of the tool offset to be corrected once the measurement is concluded.

If not programmed or programmed with a 0 value, the CNC will interpret that the correction is not wanted.

L Tolerance to be applied to the measured error. It must be programmed with an absolute value and the offset correction will be applied only if the error exceeds that value.

If not programmed, the CNC will set this parameter to "0".

Result of the measurement

Once the cycle has ended, the CNC returns the real values obtained in the measurement to the following general arithmetic parameters:

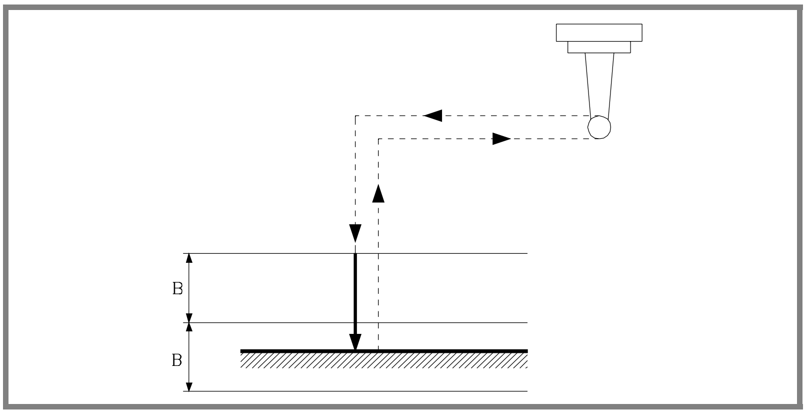
P298	Actual (real) surface coordinate.
P299	Detected error. Difference between the actual surface coordinate and the programmed theoretical coordinate.

If a tool offset number (D) was selected, the CNC will modify its values as long as the measurement error is equal to or greater than the tolerance (L).

Depending on the axis used for measuring (LW), the correction will be applied either on the length value or on the radius value.

- If the measurement is made with the axis perpendicular to the work plane, it will change the length wear (LW) of the indicated offset (D).
- If the measurement is made with one of the axis forming the plane, it will change the radius wear (RW) of the indicated offset (D).

Basic operation



1. Approach movement.

   Rapid probe movement (G00) from the cycle calling point to the approach point.

   This point is located in front of the point to be measured, at a safety distance (B) from it and along the probing axis (K).

   The approach movement is made in two stages:

   ·1· Movement in the main work plane.

   ·2· Movement along the longitudinal axis.

2. Probing movement.

Probing movement along the selected axis (K) at the indicated feedrate (F) until the probe signal is received.

The maximum probing distance is 2B. If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

Once probing is over, the CNC will assume the actual position of the axes when the probe signal is received as their theoretical position.

3. Withdrawal movement.

Rapid probe movement (G00) from the probing point to the cycle calling point.

The withdrawal movement is made in three stages:

·1· Movement to the approach point along the probing axis.

·2· Movement along the longitudinal axis up to the coordinate of the cycle calling point along that axis.

·3· When programming (C0), it makes a movement in the main work plane to the cycle calling point.

16.4 Outside corner measuring canned cycle

A probe must be used mounted in the spindle, previously calibrated using canned cycles #PROBE 1 and #PROBE 2.

Programming

The cycle may be programmed in any work plane. Depending on the work plane, the theoretical coordinates of the cycle may be defined in several ways:

- In the active work plane, except if the plane is formed by any of the axes A-B-C.

#PROBE 4 X50 Y65 Z15 …	Main axes X-Y-Z
#PROBE 4 X1=50 Y2=65 Z1=15 …	Main axes X1-Y2-Z1

- Using parameters X-Y-Z. When the plane is not formed by these axes, these parameters are interpreted as coordinates in the first axis, second axis and axis perpendicular to the work plane respectively.

#PROBE 4 X50 Y65 Z15 …	Main axis X1-B-C

The programming format in the G17, G18 or G19 plane is:

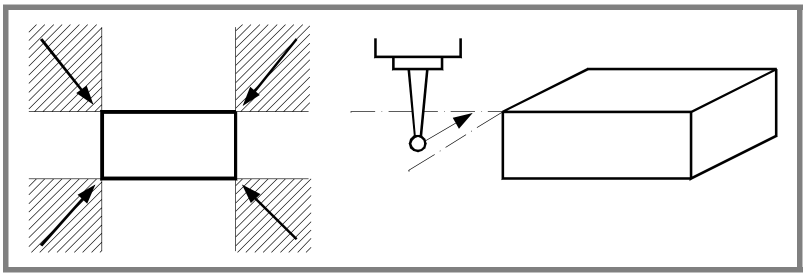
#PROBE 4 X Y Z B F

X Theoretical X coordinate of the corner to be measured.

Y Theoretical Y coordinate of the corner to be measured.

Z Theoretical Z coordinate of the corner to be measured.

Depending on the part corner to be measured, the probe must be placed in the corresponding shaded area (see figure) before calling the cycle.



B Safety distance. It must be programmed with a positive value greater than 0.

When calling the cycle, the probe must be located, with respect to the point to be measured, at a greater distance than this value

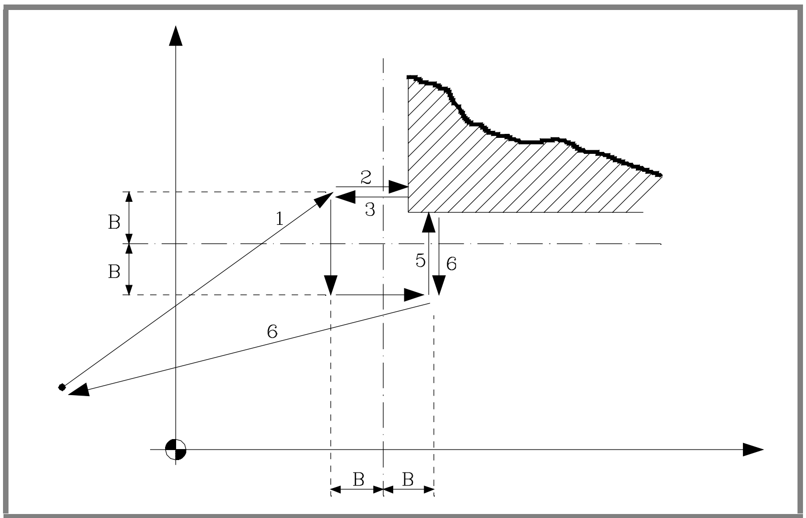
F Probing feedrate.

Result of the measurement

Once the cycle has ended, the CNC returns the real values obtained in the measurement to the following general arithmetic parameters:

P296	Real coordinate of the corner along the abscissa axis
P297	Real coordinate of the corner along the ordinate axis.
P298	Error detected along the abscissa axis. Difference between the actual corner coordinate ant the programmed theoretical coordinate.
P299	Error detected along the ordinate axis. Difference between the actual corner coordinate ant the programmed theoretical coordinate.

Basic operation



1. Approach movement.

Rapid probe movement (G00) from the cycle calling point to the first approach point located at a (B) distance from the side to be probed.

The approach movement is made in two stages:

·1· Movement in the main work plane.

·2· Movement along the longitudinal axis.

2. Probing movement.

Probing movement along the abscissa axis at the indicated feedrate (F) until the probe signal is received.

The maximum probing distance is 2B. If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

3. Withdrawal movement.

Rapid probe movement (G00) from the probing point to the first approach point.

4. Second approach movement.

Rapid probe move (G00) from the first approach point to the second.

This approach movement is made in two stages:

·1· Movement along the ordinate axis.

·2· Movement along the abscissa axis.

5. Second probing movement.

Probing movement along the ordinate axis at the indicated feedrate (F) until the probe signal is received.

The maximum probing distance is 2B. If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

6. Withdrawal movement.

Rapid probe movement (G00) from the second probing point to the cycle calling point.

The withdrawal movement is made in three stages:

·1· Movement to the second approach point along the probing axis.

·2· Movement along the longitudinal axis up to the coordinate of the cycle calling point along that axis.

·3· Movement in the main work plane up to the cycle calling point.

16.5 Inside corner measuring canned cycle

A probe must be used mounted in the spindle, previously calibrated using canned cycles #PROBE 1 and #PROBE 2.

Programming

The cycle may be programmed in any work plane. Depending on the work plane, the theoretical coordinates of the cycle may be defined in several ways:

- In the active work plane, except if the plane is formed by any of the axes A-B-C.

#PROBE 5 X50 Y65 Z15 …	Main axes X-Y-Z
#PROBE 5 X1=50 Y2=65 Z1=15 …	Main axes X1-Y2-Z1

- Using parameters X-Y-Z. When the plane is not formed by these axes, these parameters are interpreted as coordinates in the first axis, second axis and axis perpendicular to the work plane respectively.

#PROBE 5 X50 Y65 Z15 …	Main axis X1-B-C

The programming format in the G17, G18 or G19 plane is:

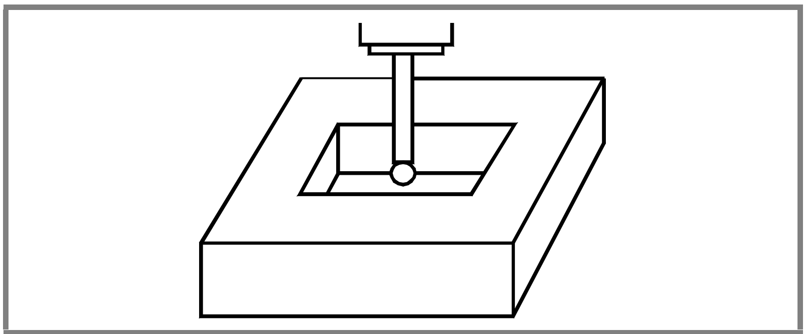
#PROBE 5 X Y Z B F

X Theoretical X coordinate of the corner to be measured.

Y Theoretical Y coordinate of the corner to be measured.

Z Theoretical Z coordinate of the corner to be measured.

The probe must be placed inside the pocket before calling the cycle.



B Safety distance. It must be programmed with a positive value greater than 0.

When calling the cycle, the probe must be located, with respect to the point to be measured, at a greater distance than this value

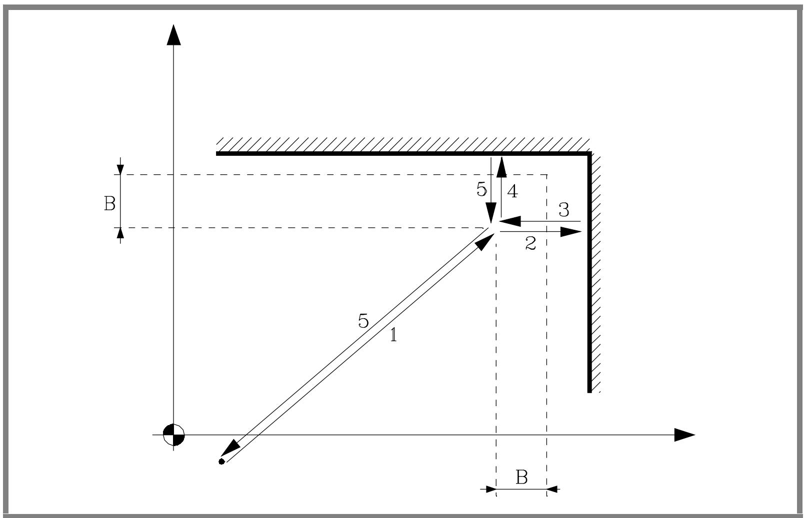
F Probing feedrate.

Result of the measurement

Once the cycle has ended, the CNC returns the real values obtained in the measurement to the following general arithmetic parameters:

P296	Real coordinate of the corner along the abscissa axis
P297	Real coordinate of the corner along the ordinate axis.
P298	Error detected along the abscissa axis. Difference between the actual corner coordinate ant the programmed theoretical coordinate.
P299	Error detected along the ordinate axis. Difference between the actual corner coordinate ant the programmed theoretical coordinate.

Basic operation



1. Approach movement.

Rapid probe movement (G00) from the cycle calling point to the first approach point located at a (B) distance from the two sides to be probed.

The approach movement is made in two stages:

·1· Movement in the main work plane.

·2· Movement along the longitudinal axis.

2. Probing movement.

Probing movement along the abscissa axis at the indicated feedrate (F) until the probe signal is received.

The maximum probing distance is 2B. If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

3. Withdrawal movement.

Rapid probe movement (G00) from the probing point to the approach point.

4. Second probing movement.

Probing movement along the ordinate axis at the indicated feedrate (F) until the probe signal is received.

The maximum probing distance is 2B. If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

5. Withdrawal movement.

Rapid probe movement (G00) from the second probing point to the cycle calling point.

The withdrawal movement is made in three stages:

·1· Movement to the approach point along the probing axis.

·2· Movement along the longitudinal axis up to the coordinate of the cycle calling point along that axis.

·3· Movement in the main work plane up to the cycle calling point.

16.6 Angle measuring canned cycle

A probe must be used mounted in the spindle, previously calibrated using canned cycles #PROBE 1 and #PROBE 2.

This cycle may be used to measure angles between ±45º.

- If the angle to be measured is equal to or greater than 45º, the CNC will display the relevant error.
- If the angle to be measured is equal to or smaller than –45º the probe will collide with the part.

Programming

The cycle may be programmed in any work plane. Depending on the work plane, the theoretical coordinates of the cycle may be defined in several ways:

- In the active work plane, except if the plane is formed by any of the axes A-B-C.

#PROBE 6 X50 Y65 Z15 …	Main axes X-Y-Z
#PROBE 6 X1=50 Y2=65 Z1=15 …	Main axes X1-Y2-Z1

- Using parameters X-Y-Z. When the plane is not formed by these axes, these parameters are interpreted as coordinates in the first axis, second axis and axis perpendicular to the work plane respectively.

#PROBE 6 X50 Y65 Z15 …	Main axis X1-B-C

The programming format in the G17, G18 or G19 plane is:

#PROBE 6 X Y Z B F

X Theoretical X coordinate of the vertex of the angle to be measured.

Y Theoretical Y coordinate of the vertex of the angle to be measured.

Z Theoretical Z coordinate of the vertex of the angle to be measured.

B Safety distance. It must be programmed with a positive value greater than 0.

The probe must be located at a greater distance than twice this value, with respect to the programmed point to be measured, when calling the cycle.

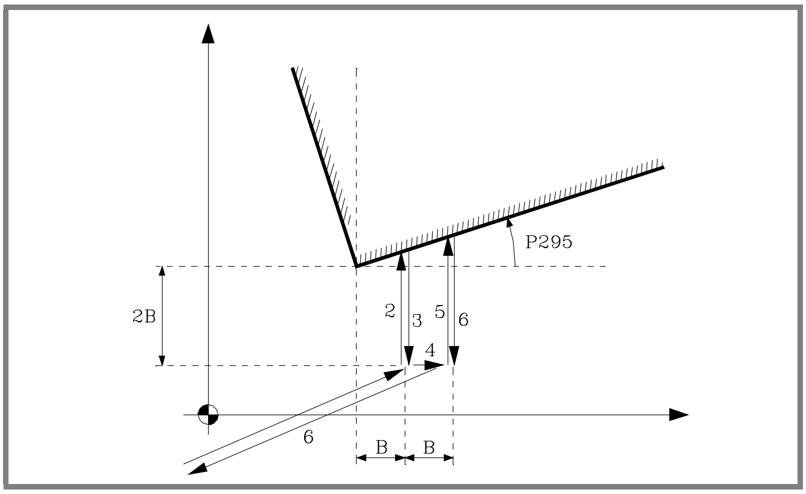
F Probing feedrate.

Result of the measurement

Once the cycle has ended, the CNC returns the real value obtained in the measurement to the following general arithmetic parameter:

P295	Inclination angle of the part with respect to the abscissa axis.

Basic operation



1. Approach movement.

Rapid probe movement (G00) from the cycle calling point to the first approach point located at a (B) distance from the programmed vertex and at (2B) from the side to be probed.

The approach movement is made in two stages:

·1· Movement in the main work plane.

·2· Movement along the longitudinal axis.

2. Probing movement.

Probing movement along the ordinate axis at the indicated feedrate (F) until the probe signal is received.

The maximum probing distance is 3B. If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

3. Withdrawal movement.

Rapid probe movement (G00) from the probing point to the first approach point.

4. Second approach movement.

Rapid probe move (G00) from the first approach point to the second. It is located at a (B) distance from the first one.

5. Second probing movement.

Probing movement along the ordinate axis at the indicated feedrate (F) until the probe signal is received.

The maximum probing distance is 4B. If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

6. Withdrawal movement.

Rapid probe movement (G00) from the second probing point to the cycle calling point.

The withdrawal movement is made in three stages:

·1· Movement to the second approach point along the ordinate axis.

·2· Movement along the longitudinal axis up to the coordinate of the cycle calling point along that axis.

·3· Movement in the main work plane up to the cycle calling point.

16.7 Outside corner and angle measuring canned cycle

A probe must be used mounted in the spindle, previously calibrated using canned cycles #PROBE 1 and #PROBE 2.

This cycle may be used to measure angles between ±45º.

- If the angle to be measured is equal to or greater than 45º, the CNC will display the relevant error.
- If the angle to be measured is equal to or smaller than –45º the probe will collide with the part.

Programming

The cycle may be programmed in any work plane. Depending on the work plane, the theoretical coordinates of the cycle may be defined in several ways:

- In the active work plane, except if the plane is formed by any of the axes A-B-C.

#PROBE 7 X50 Y65 Z15 …	Main axes X-Y-Z
#PROBE 7 X1=50 Y2=65 Z1=15 …	Main axes X1-Y2-Z1

- Using parameters X-Y-Z. When the plane is not formed by these axes, these parameters are interpreted as coordinates in the first axis, second axis and axis perpendicular to the work plane respectively.

#PROBE 7 X50 Y65 Z15 …	Main axis X1-B-C

The programming format in the G17, G18 or G19 plane is:

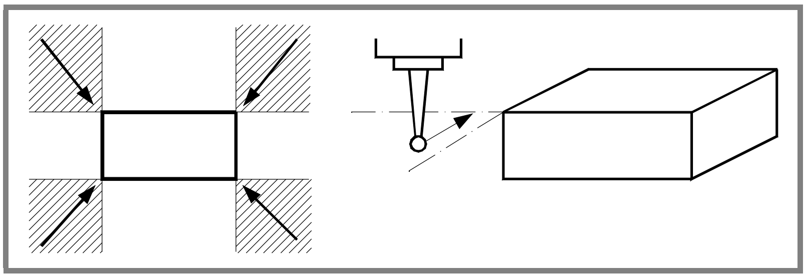
#PROBE 7 X Y Z B F

X Theoretical X coordinate of the corner to be measured.

Y Theoretical Y coordinate of the corner to be measured.

Z Theoretical Z coordinate of the corner to be measured.

Depending on the part corner to be measured, the probe must be placed in the corresponding shaded area (see figure) before calling the cycle.



B Safety distance. It must be programmed with a positive value greater than 0.

The probe must be located at a greater distance than twice this value, with respect to the programmed point to be measured, when calling the cycle.

F Probing feedrate.

Result of the measurement

Once the cycle has ended, the CNC returns the real values obtained in the measurement to the following general arithmetic parameters:

P295 Inclination angle of the part with respect to the abscissa axis.

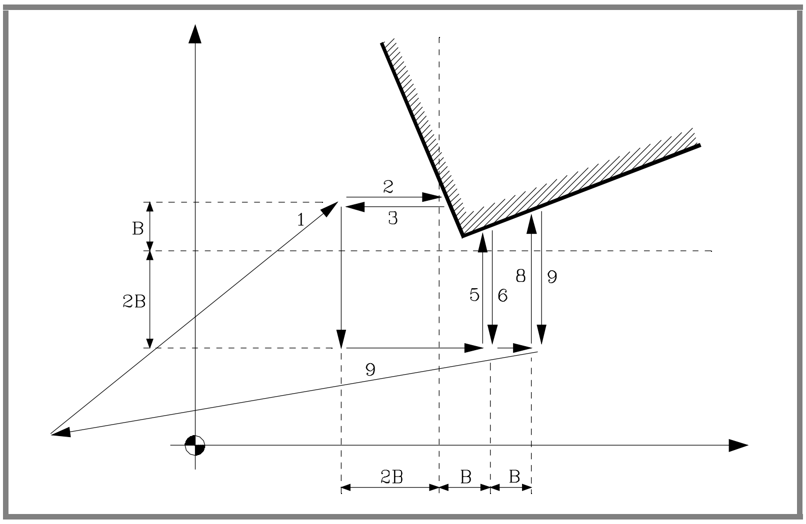
P296 Real coordinate of the corner along the abscissa axis

P297 Real coordinate of the corner along the ordinate axis.

P298 Error detected along the abscissa axis. Difference between the actual corner coordinate ant the programmed theoretical coordinate.

P299 Error detected along the ordinate axis. Difference between the actual corner coordinate ant the programmed theoretical coordinate.

Basic operation



1. Approach movement.

Rapid probe movement (G00) from the cycle calling point to the first approach point located at a (2B) distance from the side to be probed.

The approach movement is made in two stages:

·1· Movement in the main work plane.

·2· Movement along the longitudinal axis.

2. Probing movement.

Probing movement along the abscissa axis at the indicated feedrate (F) until the probe signal is received.

The maximum probing distance is 3B. If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

3. Withdrawal movement.

Rapid probe movement (G00) from the probing point to the first approach point.

4. Second approach movement.

Rapid probe movement (G00) from the first approach point to the second, located at a (2B) distance from the second side to be probed.

This approach movement is made in two stages:

·1· Movement along the ordinate axis.

·2· Movement along the abscissa axis.

5. Second probing movement.

Probing movement along the ordinate axis at the indicated feedrate (F) until the probe signal is received.

The maximum probing distance is 3B. If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

6. Withdrawal movement.

Rapid probe movement (G00) from the probing point to the second approach point.

7. Third approach movement.

Rapid probe move (G00) from the second approach point to the third. It is located at a (B) distance from the previous one.

8. Third probing movement.

Probing movement along the ordinate axis at the indicated feedrate (F) until the probe signal is received.

The maximum probing distance is (4B), if once this distance has been reached the CNC has not yet received the probe signal, it will display the relevant error code and stop the movement of the axes.

9. Withdrawal movement.

Rapid probe movement (G00) from the third probing point to the cycle calling point.

The withdrawal movement is made in three stages:

·1· Movement to the third approach point along the probing axis.

·2· Movement along the longitudinal axis up to the coordinate of the cycle calling point along that axis.

·3· Movement in the main work plane up to the cycle calling point.

16.8 Hole measuring canned cycle

A probe must be used mounted in the spindle, previously calibrated using canned cycles #PROBE 1 and #PROBE 2.

Programming

The cycle may be programmed in any work plane. Depending on the work plane, the theoretical coordinates of the cycle may be defined in several ways:

- In the active work plane, except if the plane is formed by any of the axes A-B-C.

#PROBE 8 X50 Y65 Z15 …	Main axes X-Y-Z
#PROBE 8 X1=50 Y2=65 Z1=15 …	Main axes X1-Y2-Z1

- Using parameters X-Y-Z. When the plane is not formed by these axes, these parameters are interpreted as coordinates in the first axis, second axis and axis perpendicular to the work plane respectively.

#PROBE 8 X50 Y65 Z15 …	Main axis X1-B-C

The programming format in the G17, G18 or G19 plane is:

#PROBE 8 X Y Z B J E C H F

X Theoretical hole center coordinate along the X axis.

Y Theoretical hole center coordinate along the Y axis.

Z Theoretical hole center coordinate along the Z axis.

B Safety distance. It must be programmed with a positive value greater than 0.

J Theoretical hole diameter. It must be programmed with a positive value greater than 0.

This cycle may be used to measure holes whose diameters are no greater than (J+B).

E Distance the probe withdrawals after the first probing movement. It must be programmed with a positive value greater than 0.

C It indicates where the probing cycle must end.

Value	Meaning
0	The probe returns to the point from where the cycle was called.
1	The cycle ends at the real hole center.

If not programmed, the canned cycle will take the value of "C0".

H Feedrate for the first probing movement.

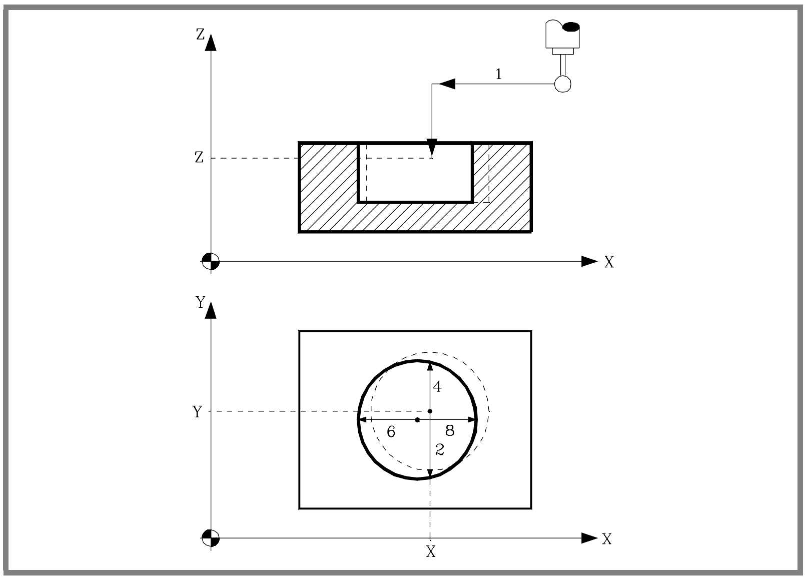
F Probing feedrate.

Result of the measurement

Once the cycle has ended, the CNC returns the real values obtained in the measurement to the following general arithmetic parameters:

P294	Hole diameter.
P295	Hole diameter error. Difference between the real and the programmed diameters.
P296	Real center coordinate along the abscissa axis
P297	Real center coordinate along the ordinate axis.
P298	Error detected along the abscissa axis. Difference between the actual center coordinate and the programmed theoretical coordinate.
P299	Error detected along the ordinate axis. Difference between the actual center coordinate and the programmed theoretical coordinate.

Basic operation



1. Approach movement.

Probe's rapid movement (G00) from the cycle calling point to the center of the hole.

The approach movement is made in two stages:

·1· Movement in the main work plane.

·2· Movement along the longitudinal axis.

2. Probing movement.

   This movement consists of:

   ·1· Probing movement along the ordinate axis at the indicated feedrate (H) until the probe signal is received.

   The maximum probing distance is "B+(J/2)". If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

   ·2· Probe's rapid withdrawal (G00) the distance indicated by (E).

   ·3· Probing movement along the ordinate axis at the indicated feedrate (F) until the probe signal is received.

3. Withdrawal movement.

   Rapid probe movement (G00) from the probing point to the theoretical center of the hole.

4. Second probing movement.

   It is similar to the previous one.

5. Withdrawal movement.

   Rapid probe movement (G00) from the probing point to the real center of the hole (calculated) along the ordinate axis.

6. Third probing movement.

   It is similar to the previous ones.

7. Withdrawal movement.

   Rapid probe movement (G00) from the probing point to the theoretical center of the hole.

8. Fourth probing movement.

   It is similar to the previous ones.

9. Withdrawal movement.

   Rapid probe movement (G00) from the probing point to the real center (calculated) of the hole.

   When programming (C0), the probe moves to the cycle calling point.

   ·1· Movement along the longitudinal axis up to the coordinate of the cycle calling point along that axis.

   ·2· Movement in the main work plane up to the cycle calling point.

16.9 Boss measuring canned cycle

A probe must be used mounted in the spindle, previously calibrated using canned cycles #PROBE 1 and #PROBE 2.

Programming

The cycle may be programmed in any work plane. Depending on the work plane, the theoretical coordinates of the cycle may be defined in several ways:

- In the active work plane, except if the plane is formed by any of the axes A-B-C.

#PROBE 9 X50 Y65 Z15 …	Main axes X-Y-Z
#PROBE 9 X1=50 Y2=65 Z1=15 …	Main axes X1-Y2-Z1

- Using parameters X-Y-Z. When the plane is not formed by these axes, these parameters are interpreted as coordinates in the first axis, second axis and axis perpendicular to the work plane respectively.

#PROBE 9 X50 Y65 Z15 …	Main axis X1-B-C

The programming format in the G17, G18 or G19 plane is:

#PROBE 9 X Y Z B J E C H F

X Theoretical boss center coordinate along the X axis.

Y Theoretical boss center coordinate along the Y axis.

Z Theoretical boss center coordinate along the Z axis.

B Safety distance. It must be programmed with a positive value greater than 0.

J Theoretical boss diameter. It must be programmed with a positive value greater than 0.

This cycle may be used to measure bosses whose diameters are no greater than (J+B).

E Distance the probe withdrawals after the first probing movement. It must be programmed with a positive value greater than 0.

C It indicates where the probing cycle must end.

Value	Meaning
0	The probe returns to the point from where the cycle was called.
1	The cycle will ends by positioning the probe over the center of the boss, at a (B) distance from the programmed theoretical coordinate.

If not programmed, the canned cycle will take the value of "C0".

H Feedrate for the first probing movement.

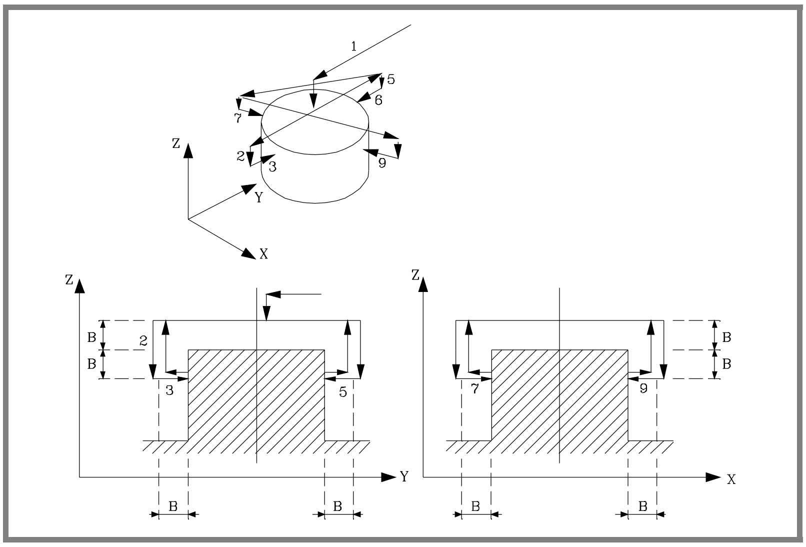
F Probing feedrate.

Result of the measurement

Once the cycle has ended, the CNC returns the real values obtained in the measurement to the following general arithmetic parameters:

P294	Boss diameter.
P295	Boss diameter error. Difference between the real and the programmed diameters.
P296	Real center coordinate along the abscissa axis
P297	Real center coordinate along the ordinate axis.
P298	Error detected along the abscissa axis. Difference between the actual center coordinate and the programmed theoretical coordinate.
P299	Error detected along the ordinate axis. Difference between the actual center coordinate and the programmed theoretical coordinate.

Basic operation



1. Positioning over the boss center.

Rapid probe movement (G00) from the cycle calling point to the center of the boss.

The approach movement is made in two stages:

·1· Movement in the main work plane.

·2· Movement along the longitudinal axis up to a (B) distance from the programmed surface.

2. Movement to the first approach point.

This probe movement is made in rapid (G00) and consists of:

·1· Movement along the ordinate axis.

·2· Movement of the longitudinal axis a (2B) distance.

3. Probing movement.

This probing movement is made in three stages:

·1· Probing movement along the ordinate axis at the indicated feedrate (H) until the probe signal is received.

The maximum probing distance is "B+(J/2)". If once this distance has been reached, the CNC has not yet received the probe signal, it will issue the relevant error code and stop the movement of the axes.

·2· Rapid probe withdrawal (G00) the distance indicated in (E).

·3· Probing movement along the ordinate axis at the indicated feedrate (F) until the probe signal is received.

4. Movement to the second approach point.

Rapid probe movement (G00) from the probing point to the next approach point.

The movement is carried out in two stages.

·1· Withdrawal to the first approach point.

·2· Movement a (B) distance over the boss up to the second approach point.

5. Second probing movement.

Same as the first probing movement.

6. Movement to the third approach point.

It is similar to the previous one.

7. Third probing movement.

It is similar to the previous ones.

8. Movement to the fourth approach point.

It is similar to the previous ones.

9. Fourth probing movement.

It is similar to the previous ones.

10. Withdrawal movement.

The withdrawal movement is made in three stages:

·1· Withdrawal to the fourth approach point.

·2· Rapid probe withdrawal (G00) at a (B) distance over the boss up to the (calculated) real boss center.

·3· When programming (C0), the probe moves to the cycle calling point.

It first moves along the longitudinal axis to the coordinate corresponding to this axis of the cycle calling point and, then it moves in the main work plane to the cycle calling point.