CNC 8070

REF. 0504

(SOFT V02.0x)

INSTALLATION 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

9. CNC variables

Appendix

ABOUT THIS MANUAL

Title

Installation Manual.

Type of documentation

It describes how to install and start up the CNC.

Internal code

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

CNC 8070 OEM (IN) STAN	Code 03753601
CNC 8070 OEM (IN) AVANZ	Code 03753621

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.

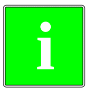
It indicates actions or operations that may hurt people or damage products.

Warning symbol.

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

Obligation symbol.

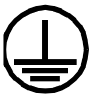
It indicates actions and operations that must be carried out.



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.

SOFTWARE INSTALLATION

1

1.1 Software installation

The CNC installation CD contains all that is necessary to install the software and the documentation needed to install, set up and operate the CNC.

The CNC may be installed in the specific hardware that will later be mounted onto the machine or at a table-top PC that will be used as a simulator for training purposes.

The CNC installed at a PC offers all the features and functions but it can only be used in simulator mode and cannot be connected to any type of machine.

Software installation at the CNC

The CNC is supplied with software properly installed. It is up to manufacturer to set it up and adapt it to his machine.

He can also customize the CNC's look using the screen customizing program FGUIM. Before using this tool, read the relevant documentation carefully.

The CNC software must not be re-installed or modified in any way without the express consent from Fagor Automation.

Fagor Automation shall not be held responsible for any personal injuries, physical or material damage suffered or caused by the CNC due to software manipulation.

Software installation at the PC

The CNC software must be installed in the hard disk of the PC; it cannot be executed directly from the CD. Once the software has been installed, in order to use the CNC, the hardware key supplied with the CD must be connected to the CNC's parallel port.

The installation will start automatically when inserting the CD in the CD drive; if not, double-click on the *Vxx* file, where Vxx indicates the version to be installed. Then, follow the instructions displayed on the screen.

Once the installation is completed, restart the PC.

Minimum PC requirements

In order for the CNC to run properly, its hardware must meet certain requirements.

- Windows® XP operating system.
- Internet Explorer 5.5 or newer.
- Pentium III microprocessor at 800 MHz.
- 256Mb. of RAM memory.
- 6x CD-ROM unit.
- 800x600 screen resolution.

Changing the language of the help files

The help files are installed in English. The product CD contains the help files in different languages. You can change the help files installed by default with the ones provided on the CD.

Locate the folder *Help files* inside the CD, select one of the available languages and copy all the files to its CNC location. The help files installed at the CNC (or at the PC, if it is a simulator) are located in the following folder.

```
C:\Cnc8070\Fagor\MMC\Help
```

The help files can only be in one language at a time.

1.2 Updating the software version

The updates must be carried out using the software supplied by Fagor Automation.

Updating the software maintains the set up of the machine parameters, PLC program, tool table and tool magazine data.

Before updating the sofware

It is recommended to always have a backup copy of the full configuration (ASCII files) such as machine parameter tables, tool tables, active-tools table and tool magazine tables as well as the PLC program.

Should any anomaly occur during the installation, these file will help restore the CNC configuration.

Software update

To update the software, close all the programs that may be running, including the CNC.

The installation will start automatically when inserting the CD in the CD drive; if not, double-click on the *Vxx* file, where Vxx indicates the version to be installed. Then, follow the instructions displayed on the screen.

Updating remote CAN nodes

Every time the CNC is powered up, it verifies the versions of the remote nodes detected in the CAN bus and automatically updates all these devices if necessary. When done loading, it goes on with the usual start-up process.

If the loading is not successful, and, consequently, the software coherence between all the CAN elements cannot be guaranteed, the CNC will display the corresponding error message every time RESET is pressed.

Updating from a version V1.1x or older

Tool and tool magazine table

Due to the improvements made to the tool tables and magazine data in version V2.00, these tables must be updated manually.

Before updating the software, save all the data of these tables in ASCII format and once the installation is completed, load this data into the tables. Both operations are carried out from the tool table and magazine table using the SAVE and LOAD softkeys.

Validation code

After activating the software from a version V1.1x or older, the validation code of this version is no longer valid and a new validation code must be entered.

1.3 Software configuration

The necessary files for the CNC are located in the directory C:\CNC8070 and its relevant subdirectories.

FAGOR **Version directory**

This directory contains the software corresponding to the CNC version installed.

Do not change the contents of this directory. Only authorized personnel from Fagor Automation may modify the contents of this directory.

Fagor Automation shall not be held responsible of the performance of this CNC if the contents of this directory have been changed.

Software updates are carried out in this directory and they do not affect the contents of directories MTB and USERS,

MTB **OEM directory**

This is especially directed at machine manufacturers.

This directory contains the modifications made by the machine manufacturer at the CNC like, for example, the PLC program, machine parameters, custom settings, new screens, integrating external applications, etc.

TMP **Temporary files**

This directory is used by the CNC to save the temporary files generated while operating.

The directory contents are erased every time the CNC is turned on.

USERS **User directory**

It is especially directed at users.

The purpose of this directory is to provide the user with a memory space for storing part-programs, profiles, etc. as they are generated.

UNINST **Uninstall directory**

This directory contains the necessary files to uninstall the software of the 8070 CNC.

To uninstall, double-click on the *fimain.exe* file and follow the instructions displayed on the screen.

1.3.1 MTB (Machine Tool Builder) directory

This is especially directed at machine manufacturers.

This directory contains the modifications made by the machine manufacturer at the CNC like, for example, the PLC program, machine parameters, custom settings, new screens, integrating external applications, etc.

DATA

This directory contains:

- The databases for machine parameters, tables, etc. and the safety backup (in ASCII) of those tables.
- (*.dat) files related to the machining canned cycles (cycle editor).
- The copies made for storing data after turning the CNC off (coordinates, zero offsets, etc.)

DRIVE

This directory contains the information regarding the DDSSETUP.

MMC

This directory contains the CNC custom setting made by the machine builder:

- The directory "...\MMC\CONFIG", the configuration files (ini) and the files that may be modified using the screen customizing tool (Fguim.exe).
- In the directory "...\MMC\IMAGES", the machine builder may include all the files of the application regarding bitmaps, videos, icons, etc.

PLC

This directory keeps the information regarding the PLC integrated by the machine builder:

- The directory "...\PLC\LANG" contains the PLC messages and error messages in the different languages
- The directory "...\PLC\PROJECT" contains the files that make up the PLC project and the object file.
- The directory "...\PLC\WATCH" contains the sets saved from monitoring and the logic analyzer.

RELEASE

When the machine builder integrates his/her own application into the CNC, the components that have been created (*.OCX files) will have to be located in this directory.

SUB

When the machine builder integrates his/her own subroutines (tool change, home search, etc.), they will have to located in this directory.

TUNING

This directory stores the configurations saved by the user in the setup assistance.

1.3.2 USERS directory

It is especially directed at users.

The purpose of this directory is to provide the user with a memory space for storing part-programs, profiles, etc. as they are generated.

Although the user may store these programs in any directory, they should be saved in the directories created for this purpose in order to make it easier and faster to find them and make safety backup copies.

POCKET This directory saves the profiles that have been created using the profile editor and are related to the pockets of the conversational canned cycles.

PRG This directory saves the part-programs created by the user. The user may create new subdirectories and store the programs in a more orderly fashion.

PROFILE This directory saves the profiles that have been created using the profile editor and are related to conversational canned cycles.

REPORTS This directory saves the BMP files generated when printing a graphic image to a file.

MACHINE PARAMETERS

2

In order for the machine tool to be able to properly execute the programmed instructions and interpret the elements connected to it, the CNC must know the specific machine data such as: number of axes, feedrates, accelerations, feedbacks, type of tool magazine, tool changer, etc.

This data is set by the machine manufacturer and is entered as machine parameters. They may be divided into the following groups:

General machine parameters

They set axis and spindle nomenclature, power-up conditions, subroutines associated with specific functions, etc.

Some of these parameters must be defined first because they configure the axis parameter tables. For example, the number and name of the axes and spindles, etc.

Machine parameters for axes and spindles

They indicate the type of axis (linear, rotary, spindle), travel limits, moving conditions, related handwheels, probing, compensations, etc.

Each axis may have up to 4 work ranges. The following must be set for each one: feedrates and gains, home searches, accelerations, etc.

Machine parameters for JOG mode

They set the handwheels and the JOG keys.

Machine parameters for the M function table

They indicate the type of synchronization and the subroutine associated with each M function.

Machine parameters for the Kinetics table

They indicate the type and characteristics of each kinematics.

Machine parameters for the tool magazine

They indicate the number of tool magazines and number of tool pockets (positions), etc.

Machine parameter HMI

They are used for setting the communications environment between the operator and the CNC.

OEM machine parameters

To configure the reading/writing of drive variables, editing cams, defining a group of generic parameters so the manufacturer can use them like machine parameters, etc.

Icons associated with machine parameters

The following icon may appear next to the parameter name

	Restarting the CNC required.

The icons appearing next to the value permit accessing the list of preestablished values, a data table, a set of parameters or to refer to a file.

	The parameter has a list of options.
	To access a data table.
	To access a group of parameters.
	The parameter refers to a file.

Abbreviations used in this chapter

(g.m.p.)	General machine parameter.
(a.m.p.)	Machine parameter for Axes and Spindles

2.1 Parameter matching between the CNC and the Sercos drive

While initializing Sercos, on CNC power-up and when validating the machine parameters of the axes, the CNC updates the following parameters at the drives.

Parameters NP121, NP122, NP131 and NP133 of each set of the CNC will be sent to the corresponding set at the drive. The parameters of the CNC's default set are saved in the rest of the sets of the drive.

Understanding the table

CNC

List of CNC machine parameters.

DRIVE

List of drive parameters that are equivalent to each CNC parameter.

Pos/Vel

It indicates writing the parameter at the drive depends or not on the type Sercos configuration, position (pos) or velocity (vel).

Feedback

It indicates whether or not writing the parameter at the drive depends on the type of axis feedback, internal or external.

CNC	DRIVE	pos/vel	Feedback	Remarks
AXISTYPE + AXISMODE	PP76			PP76=65; Linear axis. PP76=66; Rotary without module. PP76=194; Rotary with module.
PROGAIN	PP104			
I0TYPE	PP115 (bit 1,5)		External	B1=0 B5=0; If normal I0. B1=1 B5=0; if increasing distance-coded I0. B1=1 B5=1; if decreasing distance-coded I0.
NPULSES2	PP115 (bit 0)		External	B0=0; Second rotary feedback (NPULSES<>0). B0=1; Second linear feedback (NPULSES==0).
AXISCH + LOOPCH	PP115 (bit 3)	pos	External	B3=0; Reading of second feedback AXISCH==LOOPCH. B3=1; Reading of second feedback AXISCH<>LOOPCH.
AXISCH	PP55 (bit 0,2,3)	pos		B1=0 B2=0 B3=0; It does not change the sign of the feedback reading (AXISCH==NO) B1=1 B2=1 B3=1; It does not change the sign of the feedback reading (AXISCH==YES)
REFDIREC + DECINPUT + FBACKSRC	PP147 (bit 0) PP147 (bit 5) PP147 (bit 3) PP147 (bit 1)			B0=0; Positive homing direction. B0=1; Negative homing direction. B0=5; Home switch being used. B5=1; No home switch is used. B0=3; Internal feedback. B5=3; External feedback. B1=0; The DECEL signal of the CNC always uses positive logic.
REFEED1	PP41			
REFEED2	PP1			
REFVALUE	PP52 PP54	pos pos	Internal External	
REFSHIFT	PP150 PP151	pos pos	Internal External	It always writes PP150=0 at the drive. It always writes PP150=0 at the drive.

CNC	DRIVE	pos/vel	Feedback	Remarks
ABSOFF	PP177 PP178		Internal External	Only when using distance-coded I0's. Only when using distance-coded I0's.
I0CODDI1 I0CODDI2	PP166 PP165			Only when using distance-coded I0's. Only when using distance-coded I0's.
BACKLASH	PP58	pos		
BACKANOUT	PP2	pos		
BACKTIME	PP3	pos		Only if BACKANOUT<>0
INPUTREV	NP121.x			It affects all the gears.
OUTPUTREV	NP122.x			It affects all the gears.
PITCH	NP123			
INPUTREV2	NP131.x		External	It affects all the gears. Only when using rotary feedback (NPULSES2<>0).
OUTPUTREV2	NP132.x		External	It affects all the gears. Only when using rotary feedback (NPULSES2<>0).
PITCH2	NP133		External	Only when using rotary feedback (NPULSES2<>0).
NPARSETS	GP6			
Limit activation	PP55 (bit 4)			B4=1; Check the limits. B4=0; Ignore the limits (for spindles, rotary axes with module and when both parameters LIMIT+ and LIMIT- are set to 0).
MODULE (360)	PP103			PP103=360; Only if it is a spindle or a rotary axis with module. It always writes 360.
SZERO	SP42			Only if it is a spindle.
INPOSW	PP57			
MAXFLWE	PP159			Only if following error monitoring is active.

CNC	DRIVE	pos/vel	Feedback	Remarks
Type of 2nd feedback	GP10			GP10=0; Second feedback is not being used. GP10=1; TTL signal (SINMAGNI==0). GP10=2; Vpp signal (SINMAGNI<>0).
NPULSES PITCH2 Resolution of the second feedback.	NP117 NP117		External External	Only if it is rotary encoder (NPULSES<>0). Only if it is a linear encoder (NPULSES==0).
PITCH2 Resolution of linear second feedback.	NP118		External	Only if it is a linear encoder (NPULSES==0).

2.2 General machine parameters

Channel configuration

NCHANNEL **Number of channels**

Number of system's channels.

Possible values:	from 1 to 4.
By default:	1

Axis configuration

NAXIS **Number of axes governed by the CNC**

Number of the system's axes without including spindles. All the axes must be taken into account whether they are servo-controlled or not.

Possible values:	from 1 to 28.
By default:	3

Bear in mind that the number of axes does not depend on the number of channels. A channel may have one, several or no axes associated with it. See *"Configuring the axes of the channel"* on page 28.

AXISNAME **Name of each axis**

It shows the table to define the names of the axes. Parameter `NAXIS` sets the number of axes of the system.

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. For example X, X1, Y3, Z9, W, W7, C...

By default:	Starting from AXISNAME1: X, Y, Z...

When defining the axes, bear in mind that the order in which they are defined determines their logic number. The first axis of the table will be logic axis -1- and so on. As with the axis name, the logic number permits identifying the axis in PLC variables, marks, etc.

TANDEM Tandem axis

There may be up to 8 pairs of Tandem axes. Each pair has the following machine parameters to configure it.

TMASTERAXIS	TSLAVEAXIS	TORQDIST
PRELOAD	PRELFITI	TPROGAIN
TINTIME	TCOMPLIM	

Requirements of the tandem axes

Each pair of axes (master and slave) must meet the following requirements:

- Each master tandem axis admits one single slave tandem axis.
- The axes must be sercos in velocity.
- A preload may be applied between the two motors.
- Each motor may have a different rated torque.
- The turning direction of each motor may be different from the other's.
- The torque distribution between moth motors may be different from 1:1 ratio. For example, on motors whose rated torque is different.

TMASTERAXIS Tandem. Master or main axis
TSLAVEAXIS Tandem. Slave axis

In either case, any axis defined by parameter AXISNAME.

TORQDIST Tandem. Torque distribution

It sets the torque supplied by each motor to obtain the total necessary torque on the tandem axis.

This parameter refers to the master axis. It is defined as the percentage of the total torque required from the master axis. The difference between the value of this parameter and 100% is the percentage applied to the slave axis.

If the motors are identical and they're both supposed to output the same torque, this parameter should be set to 50%.

Possible values:	From 0 to 100% (both included).
By default:	50%

PRELOAD — Tandem. Preload between both motors

It is the torque difference to be applied between the master axis and the slave axis. This sets a traction between them in order to eliminate the rack-and-pinion backlash when it is in rest position.

This parameter refers to the master axis. It is defined as the percentage of the rated torque to be applied as preload.

In order for the two axes to supply opposite torques, the preload value must be greater than the maximum torque needed at all times, including accelerations.

Possible values:	from -100% to 100%.
By default:	0 (it disables the preload).

Applying the preload necessarily implies mechanically joining the master and slave axes that make up the tandem axis. Otherwise, the motors will move even without the control velocity command.

PRELFITI — Tandem. Filter time to apply the preload

It sets the time during which preload is applied gradually. It eliminates the torque steps at the input of the tandem compensator when setting a preload value. This avoids a step in the velocity commands of the master and slave axes of the tandem.

Setting it to zero disables the filter.

Possible values:	from 0 to 65535 milliseconds.
By default:	1000ms

TPROGAIN — Tandem. Proportional gain (Kp) for the tandem axis

The proportional controller generates an output proportional to the torque error between the two motors.

$$k_p = \left(\frac{S_{max}}{T_{nom}}\right) \times TPROGAIN$$

$$T_{error} = (-T_{master} + T_{slave} + Preload)$$

$$Speed = k_p \bullet T_{error}$$

Possible values:	from 0 to 100%.
By default:	0 (no proportional gain is applied).

This parameter may be modified from the oscilloscope.

Example: A tandem axis has a maximum speed of 2000 rpm and a rated torque of 20 Nm. TPROGAIN has been set to 10%.

Kp = (2000 rpm / 20 Nm) · 0.1= 10 rpm/Nm.

TINTTIME

Tandem. Integral gain (Kp) for the tandem axis

The integral controller generates an output proportional to the integral of the torque error between the two motors.

$$k_i = \frac{\text{ControlTime}}{\text{IntegralTime}} \times k_p$$

$$T_{error} = (-T_{master} + T_{slave} + \text{Preload})$$

$$\text{Speed} = k_i \cdot \sum T_{error}$$

Possible values:	from 0 to 65535 milliseconds.
By default:	0 (no integral gain is applied).

TCOMPLIM

Tandem. Compensation limit

This parameter limits the maximum compensation applied by the tandem axis. This limit is also applied to the integral.

This parameter refers to the master axis. It is defined as percentage of the maximum speed of the master motor. If programmed with a "0" value, the output of the tandem control will be zero, thus disabling the tandem.

Possible values:	from 0 to 100%.
By default:	0

GANTRY

Gantry axes

There may be up to 8 pairs of Gantry axes. Each pair has the following machine parameters to configure it.

MASTERAXIS	SLAVEAXIS	MAXCOUPE
DIFFCOMP	WARNCOUPE	

Requirements of the gantry axes

Each pair of axes (master and slave) must meet the following requirements:

- The master axis must be defined in the AXISNAME table before the slave axis.
- Both axes must belong to the same channel. The first three axes of the channel cannot be slaves.
- Both axes and drives must be of the same type (same AXISTYPE and DRIVETYPE parameters for both axes).
- Neither the Hirth axes nor the rotary axes that only turn in one direction (parametersHIRTH = NO and UNIDIR = NO) may be gantry.
- Both axes and drives must have the same software limits (same LIMIT+ and LIMIT- parameters for both axes).
- The I0 type (IOTYPE) must be the same for both axes either non-distance-coded or distance-coded (increasing or decreasing).

  When not using distance-coded reference marks (I0), either both axes or just the master axis may have a home switch (parameter DECINPUT).
- When not using absolute feedback (parameter ABSFEEDBACK) parameter REFSHIFT must be set to zero.

MASTERAXIS **Gantry. Master or main axis**
SLAVEAXIS **Gantry. Slave axis**

In either case, any axis defined by parameter `AXISNAME`.

WARNCOUPE **Gantry. Maximum difference allowed to issue a warning**

Maximum difference allowed between the following errors of both axes before issuing a warning. This lets the user act upon the machine before the error is issued.

Its value must be lower that parameter `MAXCOUPE`.

Possible values:	from 0 to 99999.9999 mm or degrees.
	from 0 to 3937.00787 inch.
By default:	0.5000 mm or degrees / 0.01969 inch.

MAXCOUPE **Gantry. Maximum difference allowed**

Maximum difference allowed between the following errors of both axes.

Possible values:	from 0 to 99999.9999 mm or degrees.
	from 0 to 3937.00787 inch.
By default:	1.0000 mm or degrees / 0.03937 inch.

DIFFCOMP **Gantry. Difference compensation after G74**

It corrects the position difference between the master and the slave axes after homing. The slave axis will move until reaching the position of the master axis at the feedrate set by parameter `REFEED2`. This process can only be interrupted with RESET.

Possible values:	Yes / No.
By default:	Yes.

Compensation is applied using the mark `DIFFCOMP(axis)`.

NSPDL **Number of spindles governed by the CNC**

Number of spindles of the system All the spindles must be taken into account whether they are servo-controlled or not.

Possible values:	from 0 to 4.
By default:	1

Bear in mind that the number of spindles does not depend on the number of channels. A channel may have one, several or no spindles associated with it. See *"Configuring the spindles of the channel"* on page 28.

SPDLNAME **Spindle name**

It shows the table to define the names of the spindles. Parameter `NSPDL` sets the number of spindles of the system.

The axis name is defined by 1 or 2 characters. The first character must be the letter -S-. The second character is optional and must be a numerical suffix between 1 and 9. This way, the name of the spindles may be within the range S, S1 ... S9.

By default:	Starting from SPDLNAME1: S, S1...

Time setting

LOOPTIME **CNC cycle (loop) time**

It sets the CNC's loop time.

Possible values:	from 1 to 20 ms.
By default:	4ms.

It greatly depends on the number of inputs, outputs and analog axes of the Bus. Use the following orientative values:

4ms.	Up to 8 analog axes
5ms.	Up to 12 analog axes
6ms.	Up to 16 analog axes
8 ms.	up to 20 analog axes
10 ms.	up to 24 analog axes

PRGFREQ **Frequency of the PRG module (in cycles)**

It indicates how often (every how many CNC cycles) a full scan of the PLC program is executed. This parameter also sets the refreshing frequency of the digital inputs and outputs as well as analog inputs.

Possible values:	from 1 to 100.
By default:	2

Thus, with a sampling period `LOOPTIME` = 4ms of and a frequency of `PRGFREQ` = 2, the PLC program will be executed every 4 x 2 = 8 ms.

CAN and Sercos bus configuration

SERBRATE Sercos transmission rate

It indicates the Sercos transmission speed used when communicating with the drives. Set it with the same value used by the drives.

Possible values:	2/4/8/16 Mbps (Megabits per second).
By default:	4 Mbps.

Speeds of 8 and 16 Mbps require a Sercos board that can work at these speeds. Otherwise the speed will be limited to 2 and 4 Mbps.

Sercos communication at 8 and 16 Mhz requires a drive version V6.05.

SERPOWSE Sercos optical power

Defines the Sercos power or the intensity of the light going through the optic fiber. Its value depends on the total length of the cable used. Set it with the same value used by the drives.

Possible values:	from 1 to 6 ("Sercos I" board). from 1 to 8 ("Sercos II" board).
By default:	4 ("Sercos I" board). 2 ("Sercos II" board).

Assigning other values, e.g. a value of 4 for a length of 3 m, can cause communication errors due to fiber optic signal distortion.

Recommended values ("Sercos I" board).

2	For lengths under 7 meters.
4	For lengths between 7 and 15 meters.
6	For lengths over 15 meters.

Recommended values ("Sercos II" board).

1 to 4	For lengths under 15 meters.
5 to 6	For lengths between 15 and 30 meters.
7	For lengths between 30 and 45 meters.
8	For lengths over 45 meters.

CANLENGTH CAN Bus cable length

The speeds that may be programmed in the CAN line depend on the total length of the CAN bus.

20	30	40	50	60	70	80	90	100	m
1000	888	800	727	666	615	571	533	500	KHz

Possible values:	up to 20, 30, 40, 50, 60, 70, 80, 90, 100 and more than 100 meters.
By default:	Up to 20 meters.

Default conditions

They indicate the conditions assumed by the CNC on power-up, after executing an M02 or M30 or after a Reset.

INCHES **Default work units (mm, inch)**

It indicates the work units assumed by the CNC by default. To change them from the part-program, use function `G70` or `G71`.

Possible values:	MM / INCH
By default:	MM

Related to arithmetic parameters

MAXLOCP **Maximum local arithmetic parameter**
MINLOCP **Minimum local arithmetic parameter**

They define the group of local arithmetic parameters to be used. 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.

Possible values:	from 0 to 99.
By default:	MAXLOCP=25 and MINLOCP=0.

MAXGLBP **Maximum global arithmetic parameter**
MINGLBP **Minimum global arithmetic parameter**

They define the group of global arithmetic parameters to be used. Global parameters may be accessed from any program or subroutine called upon from the channel. There is a group of global parameters in each channel.

Possible values:	from 100 to 9999.
By default:	MAXGlbP=299 and MINGlbP=100.

ROPARMAX **Maximum global read-only arithmetic parameter**
ROPARMIN **Minimum global read-only arithmetic parameter**

They are used to protect a group of global arithmetic parameters so they cannot be modified.

Possible values:	from 100 to 9999.
By default:	ROPARMAX=0 and ROPARMIN=0 (none is protected).

MAXCOMP **Maximum arithmetic parameter common to all the channels**
MINCOMP **Minimum arithmetic parameter common to all the channels**

They define the group of local arithmetic parameters common to all the channels to be used. The common parameters may be accessed from any channel. The value of these parameters is shared by all the channels.

Possible values:	from 10000 to 19999.
By default:	MAXCOMP=10025 and MINCOMP=10000.

Cross compensation

CROSSCOMP **Cross compensation tables**

Up to 9 cross compensation tables are possible. They are used when an axis suffers position variations due to the movement of another axis.

The CNC calculates the compensation to be applied to each axis considering the order in which the tables are defined.

Define the tables in the order used when measuring; otherwise, the result will be different.

Each table has the following machine parameters to configure it.

MOVAXIS	COMPAXIS	NPCROS
BIDIR	REFNEED	TYPCROSS

MOVAXIS **Axis whose movement affects another axis (master)**
COMPAXIS **Axis suffering the effects of the movement (compensated)**

In either case, any axis defined by parameter `AXISNAME`.

NPCROSS **Number of points in the table**

Each cross compensation table can have up to 1000 points.

Possible values:	from 0 to 1000.
By default:	0 (there is no table).

TYPCROSS **Type of compensation**

Determines whether the cross compensation will be applied on to theoretical or real coordinates.

Possible values:	Real / Theoretical.
By default:	Real

BIDIR **Bi-directional compensation**

It indicates whether the compensation is different for each direction

Possible values:	Yes / No.
By default:	No.

REFNEED **Mandatory home search**

It indicates whether the home search is necessary before applying compensation or not.

Possible values:	Yes / No.
By default:	No.

DATA **Table defining the compensation at each point**

At each point (CROSSCOMP) of the table, parameters POSITION, POSERROR and NEGERROR must be defined.

Parameter NEGERROR must be defined only with BIDIR = Yes.

POSITION **Position of the master axis**
POSERROR **Error in the positive direction**
NEGERROR **Error in the negative direction**

The table must indicate the amount of error to be compensated in specific positions of the moving axis.

Possible values:	within ±99999.9999 mm or degrees. within ±3937.00787 inch.
By default:	0

When defining the various profile points in the table, the following requirements must be met.

- The points of the table must be ordered by their position on the axis and the table must begin by the most negative point (or least positive) to be compensated.
- For axis positioning outside this area, the CNC will apply the compensation that was defined for the nearest end.
- The machine reference point must have "0" error.

Execution time

MINAENDW **Minimum duration of the AUX END signal**

AUX END is the synchronization signal that the PLC sends to the CNC to indicate that the M, S, T function have been executed. The value assigned to this parameter must be equal to or greater than the PLC's input frequency.

This parameter has the following meanings.

- It sets the time that the AUX END signal must stay active for the CNC to consider it a valid signal.
- For M functions (which do not need synchronization), it indicates the duration of the MSTROBE signal.
- For H functions (which do not need synchronization), it indicates the duration of the HSTROBE signal.

Possible values:	from 0 to 65535 ms.
By default:	10 ms.

Refer to the chapter on "CNC-PLC communications", section: "Transferring M, H, S functions".

REFTIME **Estimated home searching time**
HTIME **Estimated time for an H function**
DTIME **Estimated time for a D function**
TTIME **Estimated time for a T function**

In Editing - Simulation mode, there is an option that allows calculating the time required to execute a part with the machining conditions established in the program. To fine tune that calculation, one may define these parameters that indicate the estimated time for processing particular functions.

The values are generic, for any H, D, T or for homing one or several axes at a time.

Possible values:	from 0 to 1000000 ms.
By default:	0 ms.

Spindle machine parameter SPDLTIME indicates the estimated time to execute an S function. See *"Spindle"* on page 44.

The machine parameter MTIME of the M function table indicates the estimated time to execute an M function. See *"M function table"* on page 84.

Numbering of the digital inputs and outputs

NDIMOD — Total of digital input modules

It indicates the number of these modules connected to the same Bus CAN. After defining this value, it is possible to set the numbers of digital inputs corresponding to each module.

If not defined, the CNC numbers the digital inputs sequentially according to the order of the modules in the Bus.

Possible values:	From 0 to 64.
By default:	0 (no numbering is defined).

DIMODADDR — Table of digital input modules

It shows the list of digital input modules connected to the same Bus CAN.

When inserting a new module, the first modules will be assigned the numbering of the table and the last one will be assigned the next valid base index after the highest one assigned until then.

DIMOD 1..64 — Base index of the digital input modules

Base index from which the digital inputs of the module are numbered.

The values of the base index must be multiple of 16, plus 1 (i.e. 1, 17, 33, etc.). If an invalid base index is entered, it assumes the nearest previous valid one. The base indexes may follow any order and they do not have to be sequential.

Possible values:	From 0 to 1009. Only the values will be multiple of 16, plus 1 (1, 17, 33, 49 ...).
By default:	The first valid value.

NDOMOD — Total of digital output modules

It indicates the number of these modules connected to the same Bus CAN. After defining this value, it is possible to set the numbers of digital outputs corresponding to each module.

If not defined, the CNC numbers the digital outputs sequentially according to the order of the modules in the Bus.

Possible values:	From 0 to 64.
By default:	0 (no numbering is defined).

DOMODADDR — Table of digital output modules

It shows the list of digital output modules connected to the same Bus CAN.

When inserting a new module, the first modules will be assigned the numbering of the table and the last one will be assigned the next valid base index after the highest one assigned until then.

DOMOD 1..64 **Base index of the digital output modules**

Base index from which the digital outputs of the module are numbered.

The values of the base index must be multiple of 16, plus 1 (i.e. 1, 17, 33, etc.). If an invalid base index is entered, it assumes the nearest previous valid one. The base indexes may follow any order and they do not have to be sequential.

Possible values:	From 0 to 1009. Only the values will be multiple of 16, plus 1 (1, 17, 33, 49 ...).
By default:	The first valid value.

Probe setting

PROBE **A table-top probe is being used**

Possible values:	Yes / No.
By default:	No.

PROBEDATA **Probe parameters**

It shows the parameters needed to setup the probe.

PRBDI1	PRBPULSE1
PRBDI2	PRBPULSE2

When using a tabletop probe, besides these parameters it is necessary to define the probe position. See *"Tabletop probe position"* on page 38.

PRBDI1 **Digital input associated with probe 1**
PRBDI2 **Digital input associated with probe 2**

It indicates the number of the digital input associated with each probe.

Possible values:	from 1 to 1024.
By default:	0 (there is no digital input associated with the probe).

PRBPULSE1 **Type of pulse of probe 1**
PRBPULSE2 **Type of pulse of probe 2**

It indicates whether the probe functions of the CNC act on an up flank (positive pulse "24V or 5V") or with the down flank (negative pulse "0V") of the signal provided by the probe.

In any case, the probe pulse must be at least 20 ms long for the CNC to consider it valid.

Possible values:	Positive / Negative.
By default:	Positive.

Shared memory

PLCDATASIZE **Size of the PLC's shared data area**

It may be used to define a memory area to exchange data between a PLC program written in C language and an external application.

Possible values:	0 to 500,000 bytes.
By default:	0.

2.2.1 General machine parameters. Channels

Channel configuration

GROUPID

Group the channel belongs to

A group is a set of two or more channels with the following characteristics:

- All the channels are in the same work mode (JOG or automatic).
- A reset in any of the channels of the group affects all of them.
- Any error in any of the channels of the group interrupts the execution in all of them.

Possible values:	From 0 to 2.
By default:	0 (it does not belong to any group)

CHTYPE

Type of channel

A channel may be governed from the CNC, from the PLC or from both.

Possible values:	CNC / PLC / CNC+PLC.
By default:	CNC

Channels governed by the PLC are displayed in automatic, jog and edit/simulation modes. The tables can be accessed. If this type of channels must be display during setup, it must be defined as a channel governed from the CNC+PLC. Once the setup is completed, define it as a PLC channel.

HIDDENCH

Hidden channel

Hidden channels are not displayed and cannot be selected.

Possible values:	Yes / No.
By default:	No.

A hidden channel is not affected by RESET. To reset it, either group it with another one or reset it from the PLC mark RESETIN.

Configuring the axes of the channel

CHNAXIS **Number of axes of the channel**

Number of the channel axes without including spindles. All the axes must be taken into account whether they are servo-controlled or not.

A channel may have initially one, several or no axes associated with it. In any case, the number of axes assigned to the channel cannot be higher than the number of axes of the system, defined by parameter `NAXIS`.

Possible values:	from 0 to 28.
By default:	3

It is possible to define the configuration of the axes again from the part-program, add or remove axes, using the `#SET AX`, `#FREE AX` and `#CALL AX` instructions.

CHAXISNAME **Name of the axes of the channel**

It shows the table to define the names of the axes of the channel. Any axis defined by parameter `AXISNAME` may belong to the channel.

Configuring the spindles of the channel

CHNSPDL **Number of spindles of the channel**

Number of spindles of the channel. All the spindles must be taken into account whether they are servo-controlled or not.

A channel may have initially one, several or no spindles associated with it. In any case, the number of spindles assigned to the channel cannot be higher than the number of spindles of the system, defined by parameter `NSPDL`.

Possible values:	from 0 to 4.
By default:	1

It is possible to define the configuration of the spindles again from the part-program, add or remove spindles, using the `#SET SP`, `#FREE SP` and `#CALL SP` instructions.

CHSPDLNAME **Name of each spindle of the channel**

It shows the table to define the names of the spindles of the channel. Any spindle defined by parameter `SPDLNAME` may belong to the channel.

CAXNAME **Axis working as "C" axis (by default)**

It must be defined whenever there is a C axis. Any spindle may be set to work as a C axis. See *"Rotary axes and spindle"* on page 44.

Possible values:	Any spindle of the channel that has been set as C axis.
By default:	C

When setting more than one C axis, use the program instruction #CAX to indicate the one that is active. Only one C axis may be active in each channel.

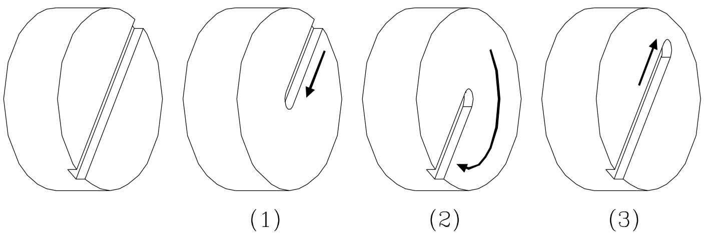
When programming the #CAX instruction without indicating the name of the axis, the CNC assumes the one mentioned in this parameter.

ALIGNC **"C" axis alignment for diametrical machining**

It indicates whether the tool can machine the whole surface diametrically in a single run (ALIGNC=No) or the "C" axis must be aligned (ALIGNC=Yes).

Possible values:	Yes / No.
By default:	Yes.

ALINGC = Yes



(1) Machining up to the center.
(2) Rotate the "C" axis 180°.
(3) Resume machining by moving the tool back.

Time setting (channel)

PREPFREQ **Number of blocks to prepare per cycle**

Possible values:	from 1 to 8
By default:	1

Before using other values, check with the service department.

ANTIME **Anticipation time**

It is used on punch presses that have an eccentric cam as a punching system. It indicates how long before the axes reach position, the anticipation logic signal `ADVINPOS` of the channel is activated .

This signal may be used to start the movement of the punch before the axes reach the position. This reduces idle time, thus increasing the number of punches per minute.

Possible values:	From 0 to 10000000 ms.
By default:	0

If the total duration of the movement is lower than the value in the parameter, the anticipation signal `ADVINPOS` will be activated immediately.

If set to zero, the anticipation signal `ADVINPOS` is always active.

Channel's default conditions

They indicate the conditions assumed by the channel on power-up, after executing an `M02`, `M30` or after a Reset.

KINID **Default kinematics number**

It indicates the kinetics number (not type) active by default. To select another one from the part-program, use the `#KIN ID` instruction.

Possible values:	from 0 to 6.
By default:	0

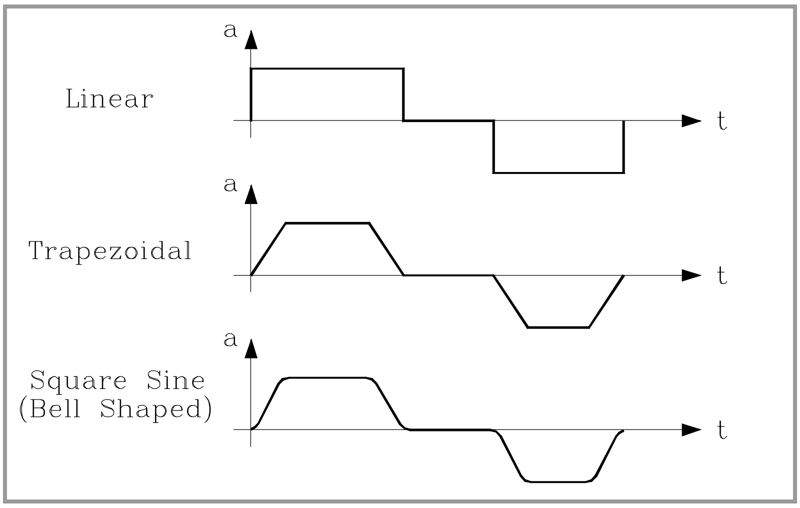
The CNC may have up to 6 different kinematics. See *"2.6 Machine parameters for kinetics"* on page 86.

SLOPETYPE

Default acceleration type

It indicates the type of acceleration applied by default in automatic movements. To select another one from the part-program, use the #SLOPE instruction.

There are three types of acceleration, namely: linear, trapezoidal and square sine (bell shaped). It is recommended to use square-sine type acceleration.



Possible values:	Linear.
	Trapezoidal.
	Square sine (bell shaped).
By default:	Square sine (bell shaped).

When working in manual (JOG) mode, the CNC always applies linear acceleration.

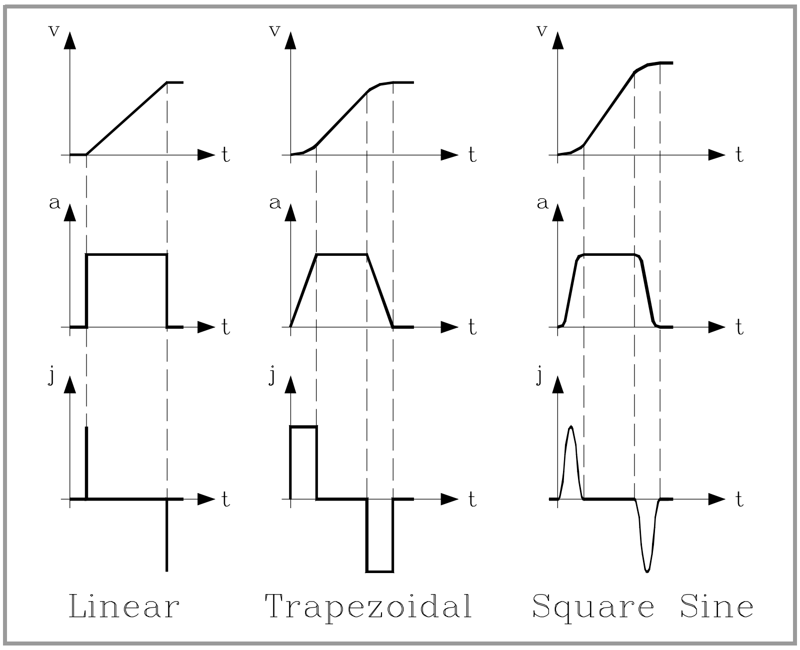
Depending on the type of acceleration selected, the machine parameters will show the ones needed to configure the acceleration. See ***"Linear acceleration"*** on page 66. See ***"Trapezoidal and square sine accelerations"*** on page 67.

Description of the types of acceleration

Square-sine acceleration provides the system's best response. The movements are smoother and the axis mechanics does not suffer as much. Linear acceleration provides the poorest response.

However, the smoother the system's response, the slower the movements. Linear acceleration provides the fastest movements and the square sine the slowest.

The figure below shows the graphs for velocity (v), acceleration graph (a) and jerk (j) for each case. As acceleration represents the velocity change per time unit, the jerk represents the acceleration change per time unit.



IPLANE

Main plane (G17/G18) by default

It indicates the main plane assumed by the CNC by default. To change them from the part-program, use function G17, G18 or G19.

Possible values:	G17 / G18.
By default:	G17.

The axes that form the work plane depend on machine parameter CHAXISNAME. See ***"Configuring the axes of the channel"*** on page 28.

	ABSCISSA AXIS	ORDINATE AXIS
G17	CHASIXNAME1	CHASIXNAME2
G18	CHASIXNAME3	CHASIXNAME1
G19	CHASIXNAME2	CHASIXNAME3

ISYSTEM

Type of programming (G90/G91) by default.

It indicates the type of coordinates assumed by the CNC by default. To change them from the part-program, use function G90 or G91.

The coordinate of a point may be defined either in absolute coordinates (G90) referred to part zero or in incremental coordinates (G91) referred to the current position.

Possible values:	G90 / G91.
By default:	G90.

IMOVE

Type of movement (G0/G1) by default.

It indicates the type of movement assumed by the CNC by default. To change them from the part-program, use function `G0` or `G1`.

Movements in G0 are carried out in rapid as set by parameter `G00FEED`. Movements in G1 are carried out at the feedrate active at the CNC.

Possible values:	G0 / G1.
By default:	G1.

IFEED

Type of feedrate (G94/G95) by default.

It indicates the type of feedrate assumed by the CNC by default. To change them from the part-program, use function `G94` or `G95`.

- With `G94`, the feedrate is assumed in mm/min or degrees/min or inches/min.
- With `G95`, the feedrate is assumed in mm/rev or degrees/rev or inches/rev.

The typical configuration for the mill model will be G94. The typical configuration for the lathe model will be G95.

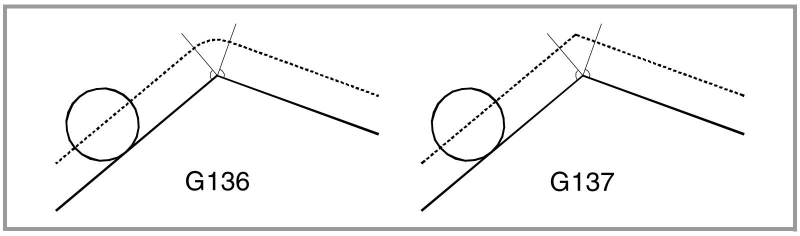
Possible values:	G94 / G95.
By default:	G94.

IRCOMP

Tool radius compensation mode (G136/G137) by default

It indicates the compensation type assumed by the CNC by default. To change them from the part-program, use function `G136` or `G137`.

Being the radius compensation active, the compensated paths may be blended together using circular paths (G136) or linear paths (G137).



Possible values:	G136 / G137
By default:	G136

ICORNER **Type of corner (G5/G7/G50) by default.**

It indicates the type of corner assumed by the CNC by default. To change them from the part-program, use function G5, G7 or G50.

There are three types of corners, namely: square (G7), rounded (G5) and semi-rounded (G50).

- When working in square corner, the CNC starts executing the next movement when the axis gets into the in-position zone defined by parameter INPOSW.
- When working in round corner, it is possible to control the corner of the programmed profile.
- When working in semi-rounded corner, the CNC starts executing the next movement once the theoretical interpolation of the current move is completed.

Possible values:	G50 / G5 / G7
By default:	G50

If G05 is selected, parameter ROUNTYPE must be set.

ROUNDTYPE **Rounding type in G5 (by default)**

It indicates the type of rounding applied by default when working in round corner. To change it from the program, use the #ROUNDPAR instruction.

The rounding may be executed by limiting the chordal error or the feedrate. The chordal error (#ROUNDPAR [1]) defines the maximum deviation allowed between the programmed point and the resulting profile. The feedrate (#ROUNDPAR [2]) defines the percentage of the active feedrate to be used for machining.

Possible values:	Chordal error / % Feedrate
By default:	Chordal error

Depending on the option selected, either parameter MAXROUND or ROUNDFEED will have to be set.

MAXROUND **Maximum rounding error in G5**

It sets the maximum deviation allowed between the programmed point and the profile resulting from rounding the corner.

The CNC takes it into account if ROUNDTYPE = Chordal error.

Possible values:	from 0 to 99999.9999 mm or degrees. from 0 to 3937.00787 inch.
By default:	1.0000 mm or degrees / 0.03937 inch.

ROUNDFEED **Percentage of feedrate in G5**

It sets the percentage of the active feedrate to be used for machining.

The CNC takes it into account if ROUNDTYPE = % Feedrate.

Possible values:	from 0 to 100.
By default:	100

Arc center correction

CIRINERR **Absolute radius error**
CIRINFACT **Percentage radius error**

They set the conditions for correcting the center position in circular interpolations. This function is controlled via program using functions G264 and G265.

On circular interpolations, the CNC calculates the radius of the starting point and end point of the tool path. Theoretically, they should be the same; but these parameters may be used to set the maximum difference allowed between both radius.

Parameter CIRINERR indicates the maximum absolute error allowed.

Possible values:	from 0 to 99999.9999 mm or degrees.
	from 0 to 3937.00787 inch.
By default:	0.0100 mm or degrees / 0.00039 inch.

Parameter CIRINFACT indicates the maximum relative error allowed (% of the radius).

Possible values:	from 0 to 100.0 %.
By default:	0.1 %.

Both parameters are taken into account. The CNC will show the relevant error message when this difference between them is greater than CIRINERR and greater than (CIRINFACT x Radius).

Feedrate override

MAXOVR **Maximum axis override (%)**

It indicates the maximum percentage to be applied to the programmed axis feedrate (feedrate override).

Possible values:	from 0 to 255.
By default:	200

The percentage applied to the programmed feedrate may be set by program, via PLC or by the %FEED switch of the panel. The one set by program has the highest priority and the one set by the switch has the lowest priority.

Different values may be set for each axis via PLC and by program. The one selected at the switch is common to all of them.

RAPIDOVR **Override acts in G00 (from 0 to 100%)**

It indicates whether the feedrate % may be modified (between 0% and 100%) or not when working in G0. If not allowed, the percentage will stay fixed at 100%.

Possible values:	Yes / No.
By default:	Yes

Regardless of the value assigned to this parameter, the override always attends to the 0% position and never acts over 100%. It is always possible to change the % of feedrate when moving in jog mode.

FEEDND **Apply the programmed feedrate to all the axes of the channel**

It indicates whether the programmed feedrate is applied to all the axes of the channel or only to the main axes.

Possible values:	Yes / No.
By default:	No.

FEEDND = YES

The programmed feedrate will be the result of composing the movements onto all the axes of the channel.

FEEDND = NO

If a movement has been programmed on any of the main axes, the programmed feedrate will be the result of composing the feedrate onto these axes. The rest of the axes move at their corresponding feedrate to end the movement of them all at the same time.

The programmed feedrate is limited only if an axis could exceed its MAXFEED. If none of the main axes are programmed, the programmed feedrate will be reached on the axis moving the farthest so they can all reach their destination at the same time.

IMOVEMACH **Movement of the independent axis referred to machine coordinates**

It indicates whether the movements of the independent axes are referred to machine coordinates (IMOVEMACH = YES) or to part coordinates (IMOVEMACH = NO), before the coordinate transformations.

Possible values:	Yes / No.
By default:	No.

Related to subroutines

OEM subroutines

OEM subroutines are those associated with T, G74 and G180-G189 functions

These subroutines must always be in the directory C:\CNC8070\MTB \SUB. If they are not in this directory, an error message will be issued:

User subroutines

Are those subroutines associated with the programs.

These subroutines may be located anywhere. When calling any of these subroutines using the instructions: #PCALL, #CALL, etc without indicating the path, it will look for them in this order and in the following directories:

1. Directory selected using the #PATH instruction.
2. Directory of the program being executed.
3. Directory indicated in the machine parameter SUBPATH.

When the call indicates the whole path, it will only look for it in the indicated directory.

SUBTABLE — OEM-subroutines table

It shows the parameters that define the OEM subroutines associated with functions T, G74 and G180-G189. These subroutines must always be in the directory C:\CNC8070 \MTB \SUB.

TOOLSUB REFPSUB OEMSUB

TOOLSUB — Name of the subroutine associated with T

This subroutine is executed automatically every time a T function (tool selection) is executed.

Possible values:	any text with up to 64 characters.

REFPSUB (G74) — Subroutine associated with function G74

Function G74 (home search) may be programmed in two ways: indicating the axes and the order they will be homed or by programming G74 alone (without axes).

When executing a block containing the G74 function alone (without axes), the CNC calls the subroutine indicated in this parameter. This subroutine must contain the home searches and the desired order.

This subroutine is also called when homing the axis in JOG mode without selecting the axes.

Possible values:	any text with up to 64 characters.

OEMSUB (G18x) **Subroutines associated with functions G180 through G189**

They indicate the number of the subroutines associated with functions `G180` through `G189`. Every time one of these functions is executed, its associated subroutine is called upon.

Possible values:	any text with up to 64 characters.

SUBPATH **Path of program subroutines**

It indicates the directory, by default, containing the user subroutines.

Tabletop probe position

PROBEDATA **Channel related probe parameters**

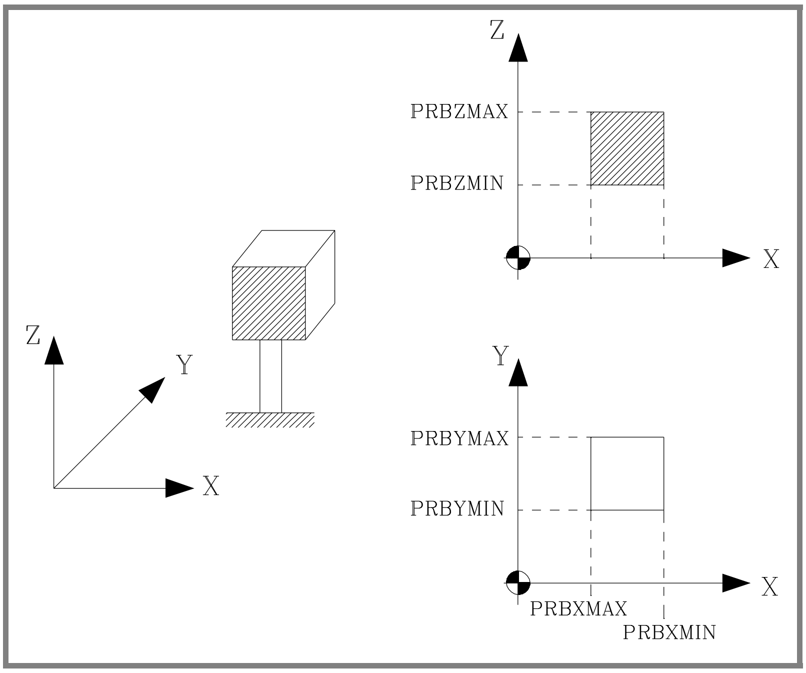
It shows the parameters needed to define the position of the tabletop probe.

PRB1MAX	PRB2MAX	PRB3MAX
PRB1MIN	PRB2MIN	PRB3MIN

Besides these parameters, it is also necessary to configure the probe signals. See *"Probe setting"* on page 25.

PRB1MAX **Maximum probe coordinate (abscissa axis)**
PRB1MIN **Minimum probe coordinate (abscissa axis)**
PRB2MAX **Maximum probe coordinate (ordinate axis)**
PRB2MIN **Minimum probe coordinate (ordinate axis)**
PRB3MAX **Maximum probe coordinate (axis perpendicular to the plane)**
PRB3MIN **Minimum probe coordinate (axis perpendicular to the plane)**

They define the position of the tabletop probe used for tool calibration.



They must be defined in absolute coordinates referred to machine reference zero. For a LATHE model CNC, the coordinates must be given in radius.

Possible values:	within ±99999.9999 mm. within ±3937.00787 inch.
By default:	0

2.3 Machine parameters for the axes

The CNC only shows the parameters for the selected type of axis and drive. That's why it displays some characters next to each parameter indicating the relevant type of axis and drive.

L, R, S	Linear (L), Rotary (R), Spindle (S)
A, S, X	Analog (A), Sercos (S), Simulated (X)

Belonging to the channel

AXISEXCH **Channel changing permission** **(L R S) (A S X)**

It determines whether it is possible for the axis or spindle to change channels via part-program and, if so, whether the change is temporary or permanent. In other words, whether the change is maintained after `M02`, `M30` or a reset.

Possible values:	No / Temporary / Maintained.
By default:	No.

Type of axis and drive

AXISTYPE **Type of axis** **(L R S) (A S X)**

Possible values:	Linear, Rotary, Spindle.
By default:	Linear

The axes defined here may be configured as gantry or tandem axes. See *"Axis configuration"* on page 13.

DRIVETYPE **Drive type** **(L R S) (A S X)**

Possible values:	Analog, Sercos or Simulated.
By default:	Simulated

The simulated axis option must be used when there is no physical axis. The CNC simulates all the movements, it assumes the theoretical coordinate as real and does not output velocity commands.

The simulated axes are not activated with the validation code. As many simulated axes as you wish are possible as long as the sum of simulated axes and physical axes does not exceed the maximum number of axes allowed (maximum value of parameter `NAXIS`).

SERCOSDATA **SERCOS drive data** **(L R S) (S)**

There are the following machine parameters to configure it.

DRIVEID OPMODEP FBACKSRC

DRIVEID **Sercos drive address** **(L R S) (S)**

It indica the position (node) the drive occupies in the Sercos connection.

Possible values:	from 1 to 16.
By default:	1

OPMODEP **Sercos drive operation mode** **(L R S) (S)**

It indicates the Sercos drive's operating mode. Velocity or feedback command.

The axes (except tandem axes) should work in position-Sercos mode and the spindles should work in velocity-sercos mode. However, tandem axes must necessarily work in velocity-Sercos mode.

Possible values:	Position / Speed.
By default:	Position.

Refer to the drive manual for further detail.

FBACKSRC **Type of feedback** **(L R S) (S)**

Type of feedback used to close the position loop. When using internal feedback, the position value is taken from motor feedback whereas when using external feedback, it is taken from direct feedback.

Possible values:	Internal / External.
By default:	Internal.

Refer to the drive manual for further detail.

Hirth axis

HIRTH **Hirth axis** **(L R) (A S X)**

A Hirth axis is the one that can only be position at positions multiple of a given value.

Possible values:	Yes / No.
By default:	No.

HPITCH **Hirth axis pitch** **(L R) (A S X)**

Possible values:	from 0 to 99999.9999 mm or degrees. within 0 and 3937.00787 inch.
By default:	0

Axis configuration for lathe type machines

FACEAXIS **Face axis (lathe)** **(L) (A S X)**

LONGAXIS **Longitudinal axis (lathe)** **(L) (A S X)**

On turning machines, you must indicate which one is the longitudinal axis and which one is the cross axis.

Possible values:	Yes / No.
By default:	No.

Typical lathe setting:

X axis	FACEAXIS = Yes	LONGAXIS = No
Z axis	FACEAXIS = No	LONGAXIS = Yes
Rest of axes	FACEAXIS = No	LONGAXIS = No

Typical Mill setting:

All the axes	FACEAXIS = No	LONGAXIS = No

Rotary axes

AXISMODE **Operating mode of the rotary axis** **(R) (A S X)**

It indicates how the axis will behave in relation to the number of turns and position display.

Possible values:	Linearlike, Module
By default:	Module

Behavior when AXISMODE = Module

It behaves like a rotary axis. The limits of the module must be positive or zero, e.g. 0º to 360º, 0º to 400º or 95º -230º; not, for example, -100 to -230º

Movements in G0/G1 and G90/G91 may be programmed.

- For movements in G90, it is possible to program more than one turn or values outside the module; but the whole travel must always be less than a full turn.

  If the axis is neither SHORTESTWAY nor UNIDIR , the programmed sign will indicate the turning direction whereas the absolute value of the coordinate will indicate the target position.

- For movements in G91, the programmed sign will indicate the turning direction whereas the absolute value of the coordinate will indicate the distance to move.

The coordinates are always displayed between the values set by the set parameters MODUPLIM and MODLOWLIM, by default 0 and 360º. See *"Module definition in rotary axes and spindle"* on page 75.

Parameters SHORTESWAY and UNIDIRmust be set. Parameters LIMIT+ and LIMIT- have no meaning

Behavior when AXISMODE = Linearlike

It behaves like a linear axis. Movements in G0/G1 and G90/G91 may be programmed.

The reading is free and in degrees (not affected by mm/inch). There are travel limits set by LIMIT+ and LIMIT-. See *"Software axis limits"* on page 46.

Parameters SHORTESWAY, UNIDIR and those for set MODUPLIM and MODLOWLIM do not apply.

UNIDIR

Unidirectional rotation (R) (A S X)

It indicates whether the movements (G00/G01) in G90 of the rotary axes may be made in either direction or they must always rotate in the same direction (positive or negative). If the axis is not UNIDIR , the programmed sign will indicate the turning direction whereas the absolute value of the coordinate will indicate the target position.

The G91 movements are carried out in the programmed direction. If the axis is UNIDIR, the programmed direction must coincide with the one preset for the axis; otherwise, the relevant error message will be issued because the axis cannot turn in the opposite direction. Likewise, the error will also come up when programming a mirror image on these axes.

The CNC takes it into account if AXISMODE = Module.

Possible values:	No (both directions), Positive, Negative.
By default:	No (both directions).

SHORTESTWAY

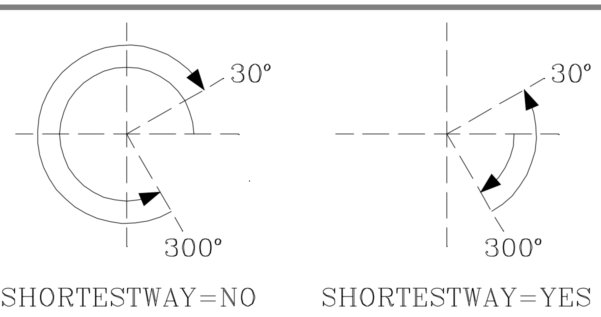
Via shortest way (R) (A S X)

It indicates whether the linear axis movements in G00/G01 in G90 of the rotary axes are carried out via the shortest way or not. Otherwise, the programmed sign will indicate the turning direction whereas the absolute value of the coordinate will indicate the target position.

The G91 movements are carried out in the programmed direction.

The CNC takes it into account if AXISMODE = Module.

Possible values:	Yes / No.
By default:	No.



If starting at B0, a movement to B300 is programmed and then another one to B30.

Rotary axes and spindle

MODCOMP **Module compensation** **(R S) (A Ss X)**

It must be activated when the axis resolution is not exact. Range parameters MODNROT and MODERR set the compensation to be applied to obtain the exact reading. See *"Module definition in rotary axes and spindle"* on page 75.

The CNC takes it into account if AXISMODE = Module.

Possible values:	Yes / No.
By default:	No (without compensation).

CAXIS **Works as a "C" axis** **(R S) (A S X)**

It indicates whether the axis or spindle can work as a C axis or not.

Possible values:	Yes / No.
By default:	No.

CAXSET **Work set for "C" axis** **(R S) (A S X)**

It indicates which work set NPARSETS the axis uses when working as "C" axis.

The CNC takes it into account if CAXIS = Yes.

Possible values:	1 to 4.
By default:	1

Spindle

AUTOGEAR **Automatic gear change** **(S) (A S X)**

It indicates whether the gear change is automatically generated by activating (if necessary) the auxiliary functions M41, M42, M43 and M44 when programming the speed.

Possible values:	Yes / No.
By default:	No.

LOSPDLIM **lower percentage for rpm OK** **(S) (A S X)**
UPSPDLIM **Upper percentage for rpm OK** **(S) (A S X)**

When working with M3 and M4, the REVOK signal is set to high (=1) when the actual spindle rpm are between these percentages.

The REVOK signal may be used to handle the Feedhold signal and avoid machining at lower or higher rpm than the ones programmed.

Possible values:	from 0 to 255.
By default:	UPSPDLIM=150 LOSPDLIM=50

SPDLTIME **Estimated time for an S function** **(S) (A S X)**

In Editing - Simulation mode, there is an option that allows calculating the time required to execute a part with the machining conditions established in the program.

To fine tune that calculation, one may define this parameter that indicates the estimated time for processing the S function.

Possible values:	from 0 to 1000000 ms.
By default:	0 ms.

When assigning a value other than "0", the CNC interprets that the S value must be passed on to the PLC using signals SSTROBE + SFUN1.

SPDLSTOP **M2, M30 and Reset stop the spindle** **(S) (A S X)**

They indicate whether the execution of functions M2, M30or a reset stops the spindle or not. Otherwise, it will be necessary to program function M5.

Possible values:	Yes / No.
By default:	Yes.

SREVM05 **G84. Reversal stops the spindle** **(S) (A S X)**

It indicates whether the spindle must be stopped (with M5) when reversing the spindle in tapping cycle.

Possible values:	Yes / No.
By default:	No.

STEPOVR	**Spindle Override step**	**(S) (A S X)**
MINOVR	**Minimum spindle override**	**(S) (A S X)**
MAXOVR	**Maximum spindle override**	**(S) (A S X)**

They set the incremental step used to override the spindle speed programmed with the Spindle Override keys of the operator panel. It also sets the maximum and minimum values of the spindle override.

Possible values:	from 0 to 255.
By default:	STEPOVR = 5
	MINOVR = 50
	MAXOVR = 150

Software axis limits

LIMIT+	**Positive software limit**	**(L R) (A S X)**
LIMIT-	**Negative software limit**	**(L R) (A S X)**

On linear and rotary axes, they set the axis travel limits.

On rotary axes, they are only taken into account if `AXISMODE` = Linearlike.

Possible values:	within ±99999.9999 mm or degrees.
	within ±3937.00787 inch.
By default:	Maximum values.

If both are set to "0", the limits will be ignored, the axis may move indefinitely in either direction.

SWLIMITTOL	**Software limits tolerance**	**(L R) (A S X)**

It indicates the maximum variation or oscillation allowed for an axis located at the limit.

Possible values:	from 0 to 99999.9999 mm or degrees.
	within 0 and 3937.00787 inch
By default:	0.1000 mm or degrees (0.00394 inch).

Runaway protection

TENDENCY	**Activation of tendency test**	**(L R S) (A S)**

It detects axis runaway due to positive feedback. It should be activated during machine setup.

Possible values:	Yes / No.
By default:	No.

PLC Offset

PLCOINC **PLC offset increment per cycle** **(L R S) (A S X)**

The CNC applies at all times the offset set by PLC. A typical utility is to correct the axis dilatations due to temperature.

This parameter indicates whether the PLC offset variations are assumed instantaneously or in steps.

Possible values:	from 0 to 99999.9999 mm or degrees.
	within 0 and 3937.00787 inch.
By default:	0 (they are assumed instantaneously).

Example:

It is set `PLCOINC` = 0.001mm (one micron per CNC cycle).

If the PLC Offset had an initial value of 0.25 mm and the new value is 0.30 mm, the PLC Offset applied per cycle will be:

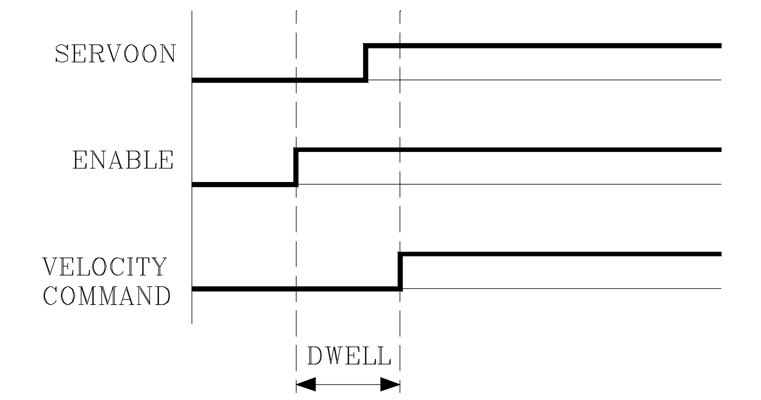
0.250 0.251 0.252 0.253 · · · · · 0.297 0.298 0.299 0.300

Dwell for dead axes

DWELL **Dwell for dead axes** **(L R S) (A S X)**

When an axis has a brake, for example very heavy vertical axes, it is only governed while it is moving. When it is governed by the CNC (movements), it is referred to as being alive and, when not moving (brake on), it is referred to as being "dead".

To bring it to "life", release the brake and close the position loop. The time required for this operation must be defined by the `DWELL` parameter.



The `ENABLE` signal indicates to the PLC to let the axis move and the `SERVOON` signal that the drive is ready.

Possible values:	within 0 and 1000000 ms.
By default:	0 (no dwell).

Radius / diameter

DIAMPROG **Programming in diameters** **(L) (A S X)**

On turning machines, the coordinates of the cross axis may be programmed in either radius or diameter. To change the type of coordinates via program, use function G151 or G152.

The CNC takes it into account on axes if FACEAXIS = Yes.

Possible values:	Yes / No.
By default:	No

Home search

REFDIREC **Homing direction** **(L R S) (A S X)**

Possible values:	Negative / Positive.
By default:	Positive.

DECINPUT **Availability of a home switch** **(L R S) (A S)**

Possible values:	Yes / No.
By default:	Yes.

Probe

PROBEAXIS **Probing axis** **(L R) (A S X)**

It indicates whether the axis is involved in a probing move (G100).

Possible values:	Yes / No.
By default:	No.

PROBERANGE **Maximum braking distance** **(L R) (A S X)**

It sets the maximum braking distance for the probe after probing to avoid breaking it (ceramic, etc). The CNC issues an error messages when this distance is exceeded.

Possible values:	from 0 to 99999.9999 mm or degrees. within 0 and 3937.00787 inch.
By default:	1.0000 mm or degrees (0.03937 inch)

PROBEFEED **Maximum probing feedrate** **(L R) (A S X)**

Possible values:	from 0 to 36000000.0000 mm/min or º/min. from 0 to 1417322.83465 inch/min.
By default:	100.0000 mm/min o degrees/min. 3.93701 inch/min.

It must be smaller than the feedrate needed to brake within the distance set by PROBERANGE with the acceleration and jerk values of the axis. Otherwise, it will show a warning when validating the axis parameters indicating the maximum feedrate that may be reached.

PROBEDELAY **Delay for the probe 1 signal** **(L R) (A S X)**
PROBEDELAY2 **Delay for the probe 2 signal** **(L R) (A S X)**

Parameter PROBEDELAY corresponds to the probe set by PRBID1 y PROBEDELAY2 to the probe set by PRBID2. See ***"Probe setting"*** on page 25.

In some types of probes, there is a short delay of a few milliseconds from the probing instant to when the CNC actually receives the signals (infrared communication, etc.). In these cases, it must indicate the time elapsed from when the probing takes place till the CNC receives the signal.

Possible values:	within 0 and 65535 ms.
By default:	0 (no delay).

Probe calibration cycle #PROBE 2 may be used to set this parameter. After it is executed, the cycles returns, in arithmetic parameters P298 and P299, the best value to be assigned to parameter PROBEDELAY for the abscissa and ordinate axes.

Repositioning of the axes in tool inspection

REPOSFEED **Maximum repositioning feedrate** **(L R) (A S X)**

Repositioning feedrate after a tool inspection. If not defined, the CNC assumes asr repositioning feedrate the one defined for the jog mode (JOGFEED).

Possible values:	from 0 to 200000.0000 mm/min or º/min. from 0 to 7873.992 inch/min.
By default:	0

The value of parameter REPOSFEED must always be smaller than G00FEED, MAXMANFEED and JOGRAPFEED.

Independent axis

POSFEED **Positioning feedrate (independent axis)** **(L R S) (A S X)**

Positioning feedrate of the independent axis.

Possible values:	from 0 to 36000000.0000 mm/min or º/min. from 0 to 1417322.83465 inch/min.
By default:	1000

DSYNCVELW **Velocity synchronization window** **(L R S) (A S X)**

Maximum velocity difference allowed. When exceeded, it starts correcting it.

Possible values:	from 0 to 36000000.0000 mm/min or º/min. from 0 to 1417322.83465 inch/min.
By default:	100

DSYNCPOSW **Position synchronization window** **(L R S) (A S X)**

Maximum position difference allowed. When exceeded, it starts correcting it.

Possible values:	from 0 to 36000000.0000 mm/min or º/min. from 0 to 1417322.83465 inch/min.
By default:	0.0100 mm or degrees (0.00039 inch)

Manual operating mode

MANUAL **Manual (jog) operating mode parameters** **(L R) (A S X)**

It shows the parameters for the Manual operating mode.

It is only available for axes, NOT for the spindle.

MANPOSSW **Maximum positive travel with G201** **(L R) (A S X)**
MANNEGSW **Maximum negative travel with G201** **(L R) (A S X)**

When using function G201, Manual mode laid over the Automatic mode, they indicate how far the axis may be moved in both directions.

Possible values:	within ±99999.9999 mm or degrees. within ±3937.00787 inch.
By default:	For MANPOSSW, the maximum positive value. For MANNEGSW, the maximum negative value.

JOGFEED **Continuous JOG mode feedrate** **(L R) (A S X)**

Possible values:	from 0 to 200000.0000 mm/min or º/min. from 0 to 7873.992 inch/min.
By default:	1000.0000 mm/min o degrees/min. 39.37008 inch/min.

JOGRAPFEED **Continuous rapid JOG mode feedrate** **(L R) (A S X)**
MAXMANFEED **Continuous maximum JOG mode feedrate** **(L R) (A S X)**

Possible values:	from 0 to 200000.0000 mm/min or º/min. from 0 to 7873.992 inch/min.
By default:	10000.0000 mm/min o degrees/min. 393.70079 inch/min.

MAXMANACC **Maximum acceleration in JOG mode** **(L R) (A S X)**

Possible values:	from 1.0000 to 1000000.0000 mm/s^2 or degrees/s^2. from 0.03937 to 39370.07874 inch/s^2.
By default:	1000.0000 mm/s^2 or degrees/s^2. 39.37008 inch/s^2.

MANFEEDP **Maximum % of jogging feedrate in G201** **(L R) (A S X)**
IPOFEEDP **Maximum % of execution feedrate in G201** **(L R) (A S X)**
MANACCP **Maximum % of jogging acceleration in G201** **(L R) (A S X)**
IPOACCP **Maximum % of execution acceleration in G201** **(L R) (A S X)**

When using function G201, Manual mode laid over the Automatic mode, they indicate the maximum feedrate and acceleration used in each mode.

MANFEEDP % of MAXMANFEED as jogging feedrate limit.

IPOFEEDP % of G00FEED as execution feedrate limit.

MANACCP % of MAXMANACC as jogging acceleration limit.

IPOACCP % of ACCEL as execution acceleration limit.

Possible values:	from 0 to 100.
By default:	20 (manual) and 80 (execution).

The sum of both should not exceed 100 in order so as not to exceed the dynamic limits of the machine under certain conditions.

It must be born in mind that when applying G201 while moving the axes, the feedrate and the acceleration instantaneously assume the values set by IPOFEEDP and IPOACCP.

Considering the following values for the Y axis:

G00FEED: 1000mm/min.

JOGFEED: 100mm/min.

MAXMANFEED: 120mm/min.

IPOFEEDP: 50%

MANFEEDP: 50%

When executing the following blocks:

```
N10 G201 #AXIS [Y]
N20 G1 Y100 F1000
```

At block N20, the maximum execution feedrate of the Y axis is not 1000 mm/min (G00FEED), but 500 mm/min due to the 50% limitation of IPOFEED over G00FEED. Therefore, in spite of the programmed feedrate "F1000", the axis will move at 500 mm/min due to the limitation in G201.

If while the execution, the Y axis is moved via JOG panel, a 100 mm/min feedrate (JOGFEED) should be added. However, the maximum jogging feedrate will be 60 min/min because it has been limited to 50% of MANFEEDP of MAXMANFEED.

Therefore, the Y axis will move at 560 mm/min when combining the automatic and jog modes.

Manual operating mode. Handwheels

MPGRESOL **Handwheel resolution** **(L R) (A S X)**

It shows 3 parameters, one per each switch position. They indicate how much the axis must move at each switch position (1, 10, 100) for each handwheel pulse.

The most typical values are those set by default.

- For MPGRESOL1, 0.0010 mm or degrees.
- For MPGRESOL10, 0.0100 mm or degrees.
- For MPGRESOL100, 0.1000 mm or degrees.

Possible values:	from 0.0001 to 99999.9999 mm or degrees. within 0.00001 and 3937.00787 inch.

Example

We have a graduated disk with 100 lines and we would like a feed of 0.001 mm per line at position 1.

- With 100 line/turn handwheel, we have 1 pulse/line; thus MPGRESOL1 = 0.0010 mm.
- With 200 line/turn handwheel, we have 2 pulses/line; thus MPGRESOL1 = 0.0005 mm.
- With 25 line/turn handwheel, we have 1 pulse per 4 lines; thus MPGRESOL1 = 0.0040 mm.

MPGFILTER **Filter time for the handwheel** (L R) (A S X)

It smooths the handwheel movements avoiding sudden variations. It indicates the number of CNC cycles used to are the handwheel pulses read.

Possible values:	from 1 to 1000.
By default:	10

Manual operating mode. Incremetal JOG

INCJOGDIST **Incremental jog distances** (L R) (A S X)

It shows 5 parameters, one per each switch position. One must define the distance traveled by each axis at each switch position (1, 10, 100, 1000, 10000)

The most typical values are those set by default.

- For `INCJOGDIST1`, 0.0010 mm or degrees.
- For `INCJOGDIST10`, 0.0100 mm or degrees.
- For `INCJOGDIST100`, 0.1000 mm or degrees.
- For `INCJOGDIST1000`, 1.0000 mm or degrees.
- For `INCJOGDIST10000`, 10.0000 mm or degrees.

Possible values:	from 0.0001 to 99999.9999 mm or degrees. within 0.00001 and 3937.00787 inch.

INCJOGFEED **Incremental jog feedrates** (L R) (A S X)

It shows 5 parameters, one per each switch position. One must define the feedrate for each axis at each switch position (1, 10, 100, 1000, 10000).

Possible values:	from 0 to 200000.0000 mm/min or º/min. from 0 to 7873.992 inch/min.
By default:	1000.0000 mm/min o degrees/min. 39.37008 inch/min.

Leadscrew error compensation

LSCRWCOMP **Leadscrew error compensation** (L R S) (A S X)

Possible values:	Yes / No.
By default:	No.

LSCRWDATA **Leadscrew compensation table** (L R S) (A S X)

Each table has the following machine parameters to configure it.

NPOINTS	BIDIR	REFNEED
DATA		

NPOINTS

Number of points of the table **(L R S) (A S X)**

The leadscrew error compensation table can have up to 1000 points.

Possible values:	from 0 to 1000.
By default:	0 (there is no table).

TYPLSCRW

Type of compensation **(L R S) (A S X)**

Determines whether the leadscrew error compensation will be applied on to theoretical or real coordinates.

Possible values:	Real / Theoretical.
By default:	Real

BIDIR

Bi-directional compensation **(L R S) (A S X)**

It indicates whether the compensation is different for each direction

Possible values:	Yes / No.
By default:	No.

REFNEED

Mandatory home search **(L R S) (A S X)**

It indicates whether the home search is necessary before applying compensation or not.

Possible values:	Yes / No.
By default:	No.

DATA

Leadscrew error compensation at each point (L R S) (A S X)

At each point (`LSCRWDATA`) of the table, parameters `POSITION`, `POSERROR` and `NEGERROR` must be defined.

Parameter `NEGERROR` must be defined only with `BIDIR` = Yes.

POSITION	**Position of each point**	**(L R S) (A S X)**
POSERROR	**Error in the positive direction**	**(L R S) (A S X)**
NEGERROR	**Error in the negative direction**	**(L R S) (A S X)**

Each parameter of the table represents a profile point to be compensated. The position that occupies the point in the profile will be referred to machine zero.

Possible values:	within ±99999.9999 mm or degrees. within ±3937.00787 inch.
By default:	0

When defining the various profile points in the table, the following requirements must be met.

- The points of the table must be ordered by their position on the axis and the table must begin by the most negative point (or least positive) to be compensated.
- For axis positioning outside this area, the CNC will apply the compensation that was defined for the nearest end.
- The machine reference point must have "0" error.

Filters to eliminate resonance frequency

FILTER	**Filter table**	**(L R S) (A S X)**

Up to 3 different filters may be defined for each axis or spindle. The filters defined for the spindle will only be applied when it works as a C axis or it is doing a rigid tapping.

Using 3 different filters allows removing more than one resonance frequency. Usually, a Low Passing filter or an "antiresonance" filter is used. Both filters may also be applied to the same axis or spindle when the resonance frequency is within the Low Passing bandwidth.

i

In order to obtain a good part finish, it is recommended to set all the axes that interpolate with each other with the same filter type and with the same frequency.

Each table has the following machine parameters to configure it.

ORDER TYPE FREQUENCY

When defining "antiresonance" type filters, parameters NORBWIDTH and SHARE must also be defined.

ORDER **Filter order** **(L R S) (A S X)**

The down slop will be softened: the greater the number the steeper the slope.

Possible values:	from 0 to 10
By default:	0 (No filter is applied).

It is recommended to always define it with a value of 3 when a filter is to be applied. Before assigning another value, check with the Service Department of Fagor Automation.

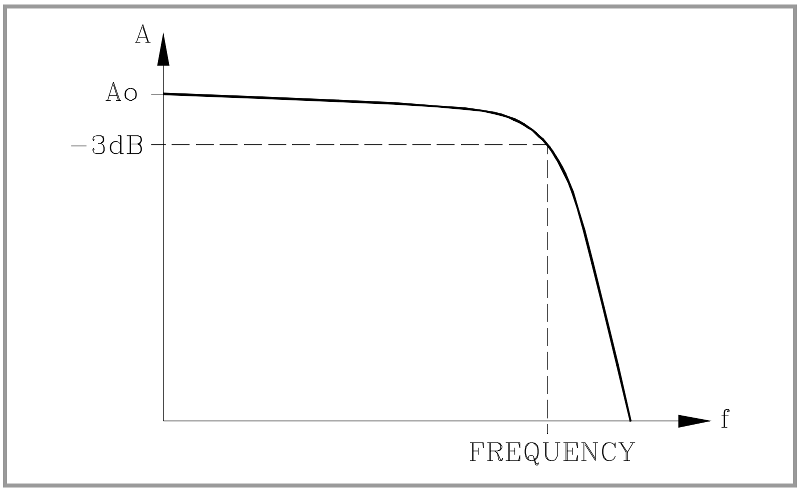
TYPE **Type of filter** **(L R S) (A S X)**

There are two types of filters, namely, "low passing" and "antiresonance" (band-rejection, notch filter).

Possible values:	Low Passing and band-rejection (notch filter).
By default:	Low Passing.

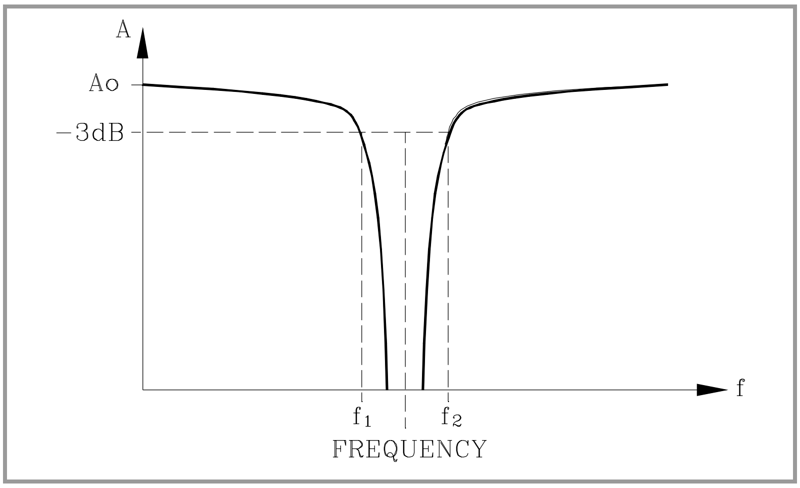
"Low passing" filter

The "low passing" filter is used to eliminate the Jerk smoothing movements out although it has the drawback of slightly rounding the corners off.



Antiresonance filter (band-rejection, notch filter)

The band-rejection filter (notch filter) must be used when the machine has a resonance frequency to be removed.



FREQUENCY **Break or mid frequency** **(L R S) (A S X)**

On a "Low Passing" filter, it indicates the break point frequency or frequency where the amplitude drops 3 dB or it reaches 70% of the nominal amplitude.

-3dB = 20 log (A/Ao) ==> A = 0,707 Ao

On the band-rejection filter (notch filter), it indicates the center frequency or the frequency where the resonance reaches its maximum value.

Possible values:	from 0 to 500.0
By default:	30.0

NORBWIDTH **Standard bandwidth** **(L R S) (A S X)**

It is calculated with the following formula. f1 and f2 correspond to the break frequency where the amplitude drops 3 dB or it reaches 70% of the nominal amplitude.

-3dB = 20 log (A/Ao) ==> A = 0,707 Ao

$$\mathrm{NORBWIDTH} = \frac{\mathrm{FREQUENCY}}{(f_2 - f_1)}$$

The CNC takes it into account if TYPE = Antiresonance (notch filter).

Possible values:	from 0 to 100.0
By default:	1.0

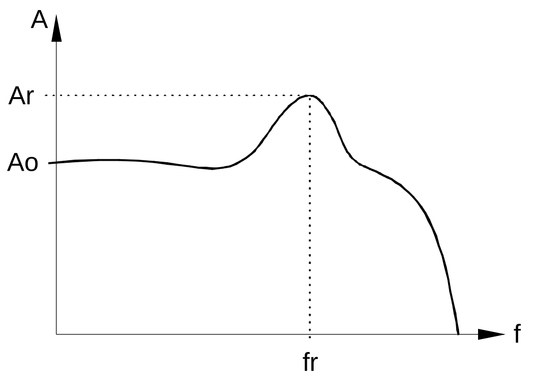
SHARE **% of signal going through the filter** **(L R S) (A S X)**

It indicates the percentage of signal going through the filter. This value must be equivalent to the percentual overshooting of the resonance because it must compensate for it.

The CNC takes it into account if TYPE = Antiresonance (notch filter).

Possible values:	from 0 to 100.
By default:	100.

Example of a calculation for a particular response of the machine.



SHARE=100(Ar-Ao)/Ar

Work sets

NPARSETS **Number of parameter sets** **(L R S) (A S X)**

Up to 4 different ranges may be defined to indicate the dynamics of the axis in each one of them (feedrates, gains, accelerations, etc.).

Possible values:	1 to 4.
By default:	1

DEFAULTSET **Default work set** **(L R S) (A S X)**

It indicates the set assumed by the CNC on power-up, after executing an M02 or M30 or after a Reset.

When defined with a "0" value, the set is always maintained.

Possible values:	0 to 4.
By default:	1

To select a set via part-program, use function `G112`.

On spindles, function `G112` selects the set but it does not carry out a gear change. To select a set and make a gear change, use functions `M41` through `M44`.

2.3.1 Machine parameters for the axes. Work sets

Up to 4 different ranges may be defined to indicate the dynamics of the axis in each one of them (feedrates, gains, accelerations, etc.). See *"Work sets"* on page 58.

Each set only shows the parameters for the selected type of axis and drive. That's why, there are some characters next to them indicating their relevant type of axis and drive.

L, R, S	Linear (L), Rotary (R), Spindle (S)
A, S, X	Analog (A), Sercos (S), Speed Sercos (Ss), Simulated (X)

For Sercos drive, parameters

Resolution

PITCH **Leadscrew pitch** **(L R S) (A S X)**

PITCH2 **Leadscrew pitch (2nd feedback)** **(L R S) (S)**

- On a linear axis with a rotary encoder and leadscrew, it defines the leadscrew pitch.
- On a linear axis with a linear encoder (scale), it defines the pitch of the scale.
- On a rotary axis, it sets the number of degrees per turn of the encoder.

Possible values:	from 0 to 99999.9999 mm or degrees. within 0 and 3937.00787 inch.
By default:	5 mm or degrees (0.19685 inch).

Example:

Axis with a 5 mm pitch leadscrew	`PITCH` = 5 mm
Axis with a 20 µm pitch Fagor scale	`PITCH` = 0.020 mm
Rotary axis with a 1/10 gear ratio	`PITCH` = 36º

INPUTREV **Turns of motor shaft** **(L R S) (A S X)**

OUTPUTREV **Turns of machine axis** **(L R S) (A S X)**

It sets the gear ratio between the motor shaft and the final axis that moves the machine.

Possible values:	from 1 to 32767.
By default:	1.

The possible gear ratio existing between the motor and the encoder can also be entered directly through parameter `PITCH`. In this case, parameters `INPUTREV` and `OUTPUTREV` must be set to ·1·.

INPUTREV2 **Turns of motor shaft (2nd feedback)** **(L R S) (S)**

OUTPUTREV2 **Turns of machine axis (2nd feedback)** **(L R S) (S)**

It sets the gear ration when not using second feedback.

Possible values:	from 1 to 32767.
By default:	1.

NPULSES **Number of encoder pulses** **(L R S) (A Ss X)**

NPULSES2 **Number of encoder (2nd feedback) pulses** **(L R S) (A Ss X)**

Number of pulses per turn of the encoder. With linear encoder (scale) set `NPULSES` = 0 and `NPULSES2` = 0.

When using a gear reduction on the axis, the whole assembly must be taken into account when defining the number of pulses per turn.

Possible values:	from 0 to 65535.
By default:	1250

SINMAGNI **Sinusoidal multiplying factor** **(L R S) (A X)**

It indicates the multiplying factor applied to the sinewave feedback of the axis.

Possible values:	from 0 to 255.
By default:	0

For square feedback signals, set `SINMAGNI` = 0. The CNC applies a x4 factor.

ABSFEEDBACK **Absolute feedback system** **(L R S) (A S X)**

Possible values:	Yes / No.
By default:	No.

FBACKAL **Feedback alarm activation** **(L R S) (A)**

Possible values:	Yes / No.
By default:	No.

Loop setting

LOOPCH **Analog voltage sign change** **(L R S) (A S X)**
AXISCH **Feedback sign change** **(L R S) (A S X)**

If the axis runs away, the CNC issues a following error message. Change the value of parameter LOOPCHG

If it does not run away, but the counting direction is not the desired one, change the values of both parameters AXISCHG and LOOPCHG.

Possible values:	Yes / No.
By default:	No.

INPOSW **In position zone** **(L R S) (A S X)**

The in-position zone is defined as the zone before and after the programmed position where the axis is considered to be in position. Parameter INPOSW defines the width of both zones.

Possible values:	from 0.0001 to 99999.9999 mm or degrees. within 0.00000 and 3937.00787 inch.
By default:	0.0100 mm or degrees (0.00039 inch).

Backlash compensation in movement reversal

BACKLASH **Backlash** **(L R S) (A S X)**

When an axis has backlash and reverses its moving direction, there is a delay from the instant the motor starts turning to the moment when the axis actually moves. This usually happens on axes with encoder and on old machines whose leadscrew assembly is defective (worn out).

Use a dial indicator to measure this backlash. Move the axis in one direction and set the dial indicator to 0. Move the axis in the opposite direction in incremental mode until detecting that the axis moves. The amount of backlash is the difference between the commanded distance and what it actually moved.

Possible values:	within ±3.2768 mm or degrees (±0.12901 inch).
By default:	0

With linear encoders (scales), set BACKLASH = 0

Backlash compensation with additional command pulse

BAKANOUT **Additional command pulse** **(L R S) (A S)**

Additional velocity command pulse to make up for the possible leadscrew backlash when reversing the moving direction. Every time the movement is inverted, the CNC will apply to that axis the velocity command corresponding to the movement plus the additional velocity command pulse set in this parameter.

This additional command will be applied for the time period indicated by (a.m.p.) BAKTIME.

Possible values:	With analog drive, between 0 and 32767. With Sercos drive, within ±1000 rpm.
By default:	0 (Not applied).

With an analog drive, the additional velocity command is given in units of the D/A converter with an integer between 0 and 32767. 10 V correspond to the value of 32767.

BAKANOUT	1	3277	32767
Analog voltage	0.3 mV	1 V	10 V

When setting an additional command pulse, parameters BAKTIME and ACTBAKANmust also be set.

BAKTIME **Duration of the additional command pulse** **(L R S) (A S)**

It indicates the duration of the additional velocity command pulse to make up for backlash in movement reversals.

Possible values:	from 0 to 65535.
By default:	0

ACTBAKAN **Application of the additional command pulse(L R S) (A S)**

It determines when the additional command pulse is applied to compensate for backlash peaks.

Possible values:	Always. G2 / G3.
By default:	Always.

Feedrate setting

G00FEED **Feedrate in G00** **(L R S) (A S X)**

Rapid positioning (traverse, G00) are always carried out at the maximum speed possible. The one indicated by G00FEED.

Possible values:	from 0 to 200000.0000 mm/min, degrees/min.
	from 0 to 7873.992 inch/min.
	from 0 to 100000.0000 rpm.
By default:	10000.0000 mm/min, degrees/min or rpm.
	393.70079 inch/min.
	from 0 to 3000.0000 rpm.

MAXVOLT **Analog voltage to reach G00FEED** **(L R S) (A Ss)**

This is the analog voltage the CNC must output so the axis can reach its maximum rapid traverse feedrate G00FEED.

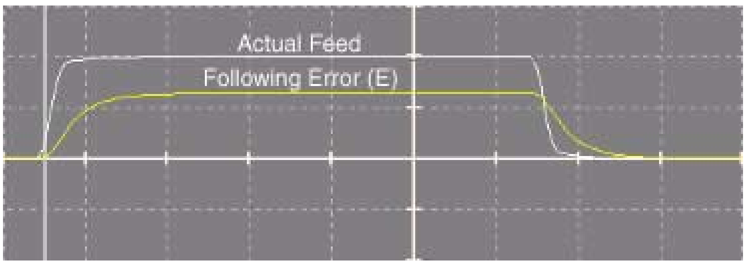
Possible values:	from 0 to 10000.0000 mV.
By default:	9500 mV (9.5 V).

Gain setting

PROGAIN **Proportional gain** **(L R S) (A S X)**

It sets the following error "ε" (difference between the theoretical instantaneous position and the actual - real - axis position) for a particular feedrate.

Possible values:	de 0.0 a 100.0 (1000/min).
By default:	1



Each square of this graph represents a following error "ε" of 800 μm.

Example:

To obtain a following error (axis lag) of 1 mm for a feedrate of 1000 mm/min. (a gain of 1):

Feedrate = Following error x PROGAIN

Feedrate / "ε" = 1000 (mm/min) / 1 (mm) = 1000 / min

PROGAIN = 1

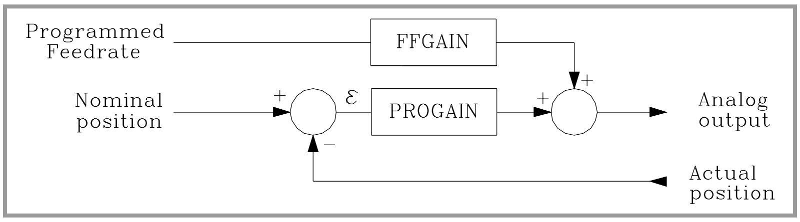
FFWTYPE **Pre-control type** **(L R S) (A S X)**

Possible values:	OFF
	Feed Forward
	AC-Forward
	Feed Forward + AC-Forward
By default:	OFF

FFGAIN **Percentage of Feed-Forward in automatic** **(L R S) (A S X)**

It helps improve the position loop minimizing the amount of following error "ε". It should only be used when working with non-linear acceleration and deceleration.

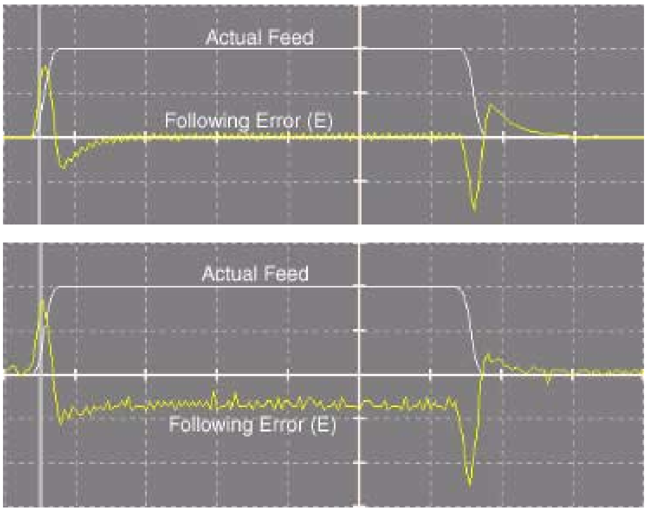
It sets the part of the analog output that is proportional to the programmed feedrate. The rest will be proportional to the following error "ε".



The CNC considers it when working with Feed Forward. Parameter FFWTYPE if it is an analog drive or simulated, and parameter OPMODEP if it is Sercos.

Possible values:	from 0 to 120 (%).
By default:	0

The best adjustment is achieved when minimizing the following error as much as possible but without reaching negative values.



***(A)** Proper adjustment with Feed-forward.*
***(B)** Wrong adjustment with Feed-forward.*
Each square of this graph represents a following error "ε" of 10 µm.

MANFFGAIN

Percentage of Feed-Forward in manual (L R S) (A X)

Although there are three types of accelerations, only linear acceleration is used in JOG mode. Sometimes, the Feed Forward selected for the automatic mode may be too high for the Jog mode.

In those cases, this parameter allows adapting the Feed Forward, applied to the JOG mode.

The CNC considers it when working with Feed Forward. Parameter `FFWTYPE` if it is an analog drive or simulated, and parameter `OPMODEP` if it is Sercos.

Possible values:	from 0 to 120 (%).
By default:	0

ACFWFACTOR

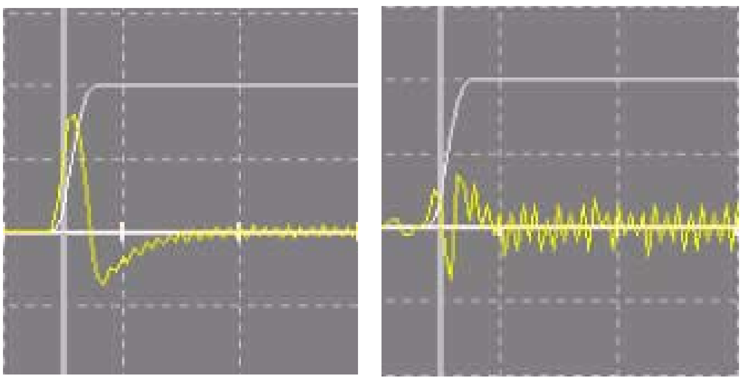
Acceleration time constant (L R S) (A Ss X)

It is recommended to assign to this parameter a value of the order of the system response time. Since the system response time is usually an unknown value that depends on the inertia of the machine and on the drive adjustment, it is recommend to try with several values.

The CNC considers it when working with AC-Forward. Parameter `FFWTYPE` if it is an analog drive or simulated, and parameter `OPMODEP` if it is Sercos.

Possible values:	within 0.001 and 1000000.0000 ms.
By default:	1000.0000 ms.

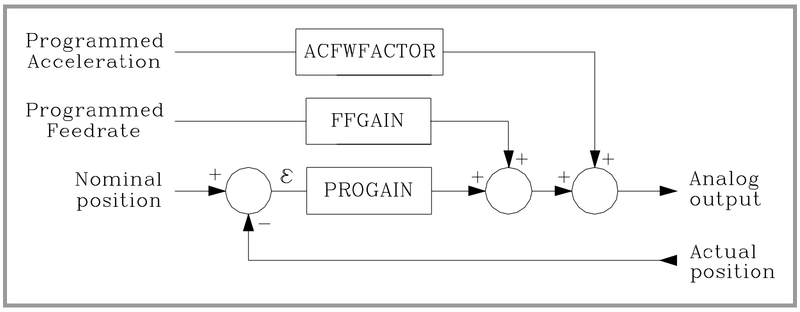
The best adjustment is achieved when minimizing the following error as much as possible but without inverting the peaks. The peaks of the right graph are inverted. Bad adjustment.



The graph on the left shows the system response without AC-Forward gain (10 μm per square) and the one on the right with AC-Forward (1 μm).

ACFGAIN **Percentage of AC-Forward in automatic** **(L R S) (A S X)**
MANACFGAIN **Percentage of AC-Forward in JOG mode** **(L R S) (A X)**

They are similar to parameters FFGAIN and MANFFGAIN; but they affect the AC-Forward. They improve the system response to acceleration changes. They minimize the amount of following error "ε" on start-ups, braking and direction reversals.

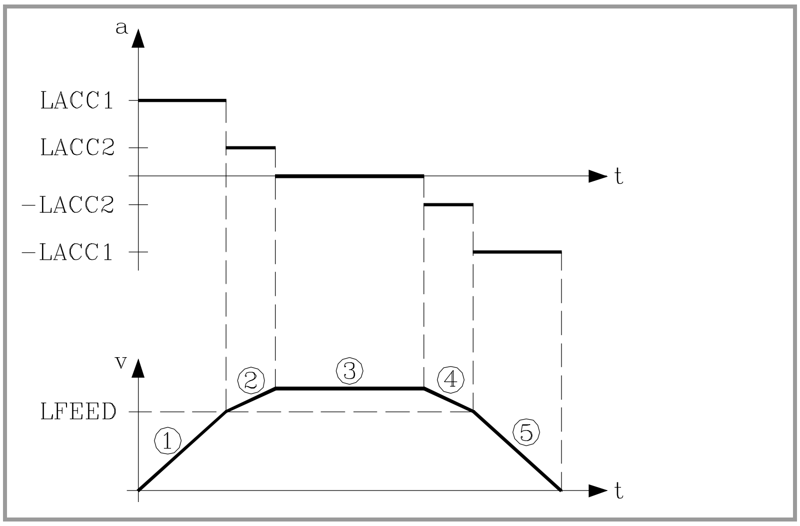


The CNC considers it when working with AC-Forward. Parameter ACFWFACTOR.

Possible values:	from 0 to 120 (%)
By default:	0

Linear acceleration

The type of acceleration is defined with parameter SLOPETYPE. See *"Channel's default conditions"* on page 30.



1. The axis starts moving with the acceleration indicated in LACC1 and maintains that acceleration until reaching the feedrate indicated in LFEED.
2. From that instant on, it keeps moving with the acceleration indicated in LACC2.
3. When reaching the programmed feedrate, the acceleration goes back to "0".
4. When braking, it decelerates with the value indicated in LACC2.
5. When the feedrate is lower than the one indicated in LFEED, it decelerates with the value of LACC1.

2. MACHINE PARAMETERS
Machine parameters for the axes

CNC 8070

(SOFT V02.0x)

LACC1 **Acceleration of the first section** (L R S) (A S X)

LACC2 **Acceleration of the second section** (L R S) (A S X)

Possible values:	from 1.0000 to 1000000.0000 mm/s^2 or degrees/s^2.
	from 0.03937 to 39370.07874 inch/s^2.
By default:	1000.0000 mm/s^2 or degrees/s^2.
	39.37008 inch/s^2.

LFEED **Change speed** (L R S) (A S X)

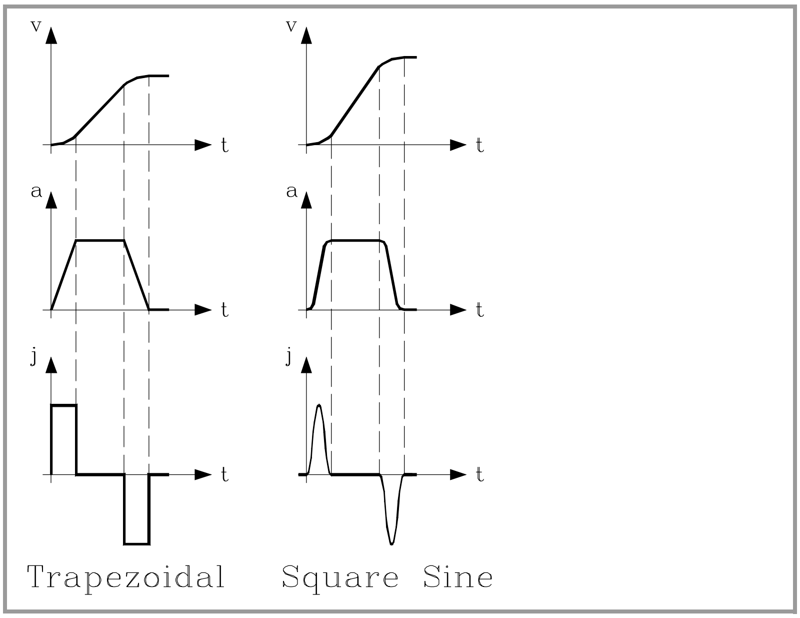
Possible values:	from 0 to 200000.0000 mm/min, degrees/min.
	from 0 to 7873.992 inch/min.
	from 0 to 100000.0000 rpm.
By default:	1000.0000 mm/min, degrees/min or rpm.
	39.37008 inch/min.
	from 0 to 10000.0000 rpm.

Trapezoidal and square sine accelerations

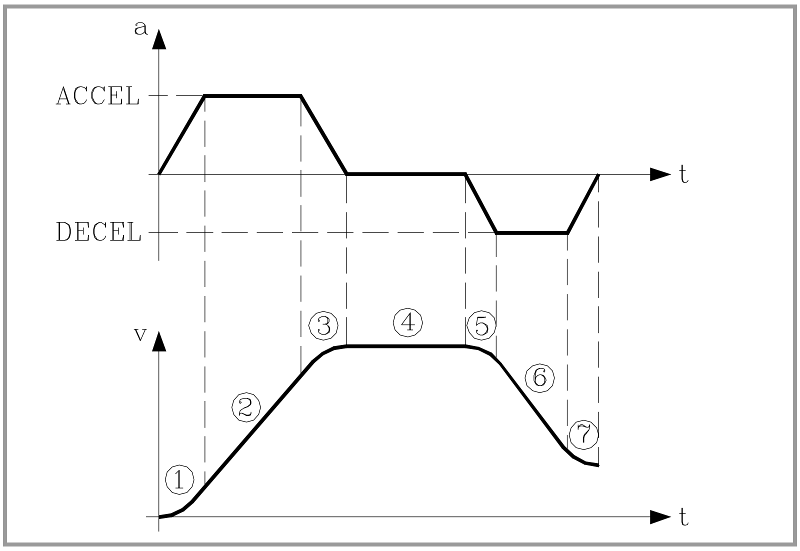
Trapezoidal acceleration may be used to program ramps in order to smooth out the acc/dec changes.

Square-sine acceleration is an improved trapezoidal acceleration. It smooths out the jerk so the movements are softer and the axis mechanics suffers less.

The figure below shows the graphs for velocity (v), acceleration graph (a) and jerk (j) for each case.



The dynamics of both accelerations are similar and they are defined using the same parameters. We now show the dynamics of the trapezoidal acceleration.



1. The axis starts moving with a uniformly increasing acceleration, with a slope limited by ACCJERK, until reaching the acceleration indicated in ACCEL.
2. The acceleration becomes constant at the ACCEL value.
3. Before reaching the programmed feedrate, there is a deceleration limited by ACCJERK.
4. It goes on at the programmed feedrate and with no acceleration.
5. To slow down or stop the axis, a uniformly decreasing deceleration is applied with a slope limited by DECJERK.
6. The deceleration becomes constant at the DECEL value.
7. Before reaching the programmed feedrate, or stopping, there is a deceleration limited by DECJERK.

ACCEL **Acceleration** **(L R S) (A S X)**
DECEL **Deceleration** **(L R S) (A S X)**

Possible values:	from 1.0000 to 1000000.0000 mm/s² or degrees/s².
	from 0.03937 to 39370.07874 inch/s².
By default:	1000.0000 mm/s² or degrees/s².
	39.37008 inch/s².

ACCJERK **Acceleration Jerk** **(L R S) (A S X)**
DECJERK **Acceleration Jerk** **(L R S) (A S X)**

Possible values:	from 1.0000 to 1000000000.0000 mm/s³ or degrees/s³.
	from 0.03937 to 39370.078.74010 inch/s³.
By default:	10000.000 mm/s³ or degrees/s³.
	393.70087 inch/s³.

Home search

Feedback system with distance-coded reference marks (I0)

Axes whose feedback system offers distance-coded reference marks (coded I0) may be referenced (homed) at any point of the machine.

The axis moves the minimum distance possible, less than 200 mm, in the direction set by general axis parameter "REFDIREC" and at the feedrate indicated in "REFEED2".

Feedback system without distance-coded reference marks (I0)

When the feedback system does not have distance-coded reference marks, the axis must always be homed at a specific point of the machine which is referred to as Machine Reference Point or Home.

The installer must place a home switch for each axis at the Machine Reference Point.

The axis moves in the direction set by axis parameter "REFDIREC" and at the feedrate indicated by "REFEED1" until reaching the home switch.

When pressing the home switch, it reverses and moves back at the feedrate indicated by "REFEED2". It keeps on moving after releasing the home switch until the CNC detects a marker pulse (I0) from the feedback device.

The Machine Reference Point must also be set when the feedback system has distance-coded reference marks (I0) and leadscrew error compensation is being applied on to that axis.

The home search may be carried out anywhere on the machine; but the leadscrew error at the machine reference point (home) point must be "0".

I0TYPE **Type of reference mark (I0)** **(L R S) (A S X)**

Possible values:	Incremental (Not distance-coded)
	Increasing distance-coded
	Decreasing distance-coded
By default:	Incremental (Not distance-coded)

Fagor linear encoders with increasing distance-coded I0

MOVX, MOVY, MOVP, FOX, FOP

Fagor linear encoders with decreasing distance-coded I0

COVX, COVP.

REFVALUE

Position of the reference point **(L R S) (A S X)**

The Machine Reference Point (home) must be defined when:

- The feedback system does not have distance-coded marks (I_0)
- The feedback system has distance-coded marks (I_0) and leadscrew error compensation is being applied on that axis.

Set the home position referred to Machine Reference Zero.

Possible values:	within ±99999.9999 mm or degrees.
	within ±3937.00787 inch.
By default:	0

REFSHIFT

Offset of the reference point **(L R S) (A S X)**

Sometimes, to readjust the machine, it is necessary to take down the feedback device, thus when putting back up, the new home point might no coincide with the previous one.

Since the home point must still be the same, the difference between the new point and the old point must be assigned to parameter REFSHIFT .

Possible values:	within ±99999.9999 mm or degrees.
	within ±3937.00787 inch.
By default:	0

REFFEED1 **Fast home searching feedrate** **(L R S) (A S X)**

REFFEED2 **Slow home searching feedrate** **(L R S) (A S X)**

When the feedback system does not have distance-coded reference marks (I0), the home search is carried out at the feedrate indicated by "REFEED1" until the home switch is reached. It then reverses its movement at the feedrate indicated by REFEED2 and it goes on until the CNC receives the reference marker pulse from the feedback device.

Possible values:	from 0 to 200000.0000 mm/min, degrees/min.
	from 0 to 7873.992 inch/min.
	from 0 to 100000.0000 rpm.
By default:	

REFFEED1: 1000.0000 mm/min or degrees/min (39.37001 inch/min) or 100.000 rpm.

REFFEED2: 100.0000 mm/min or degrees/min (3.93700 inch/min) or 10.000 rpm.

REFPULSE

Type of I0 pulse **(L R S) (A S X)**

It indicates the type of flank of the I0 signal that is used for home search.

Possible values:	Positive / negative.
By default:	Positive.

ABSOFF

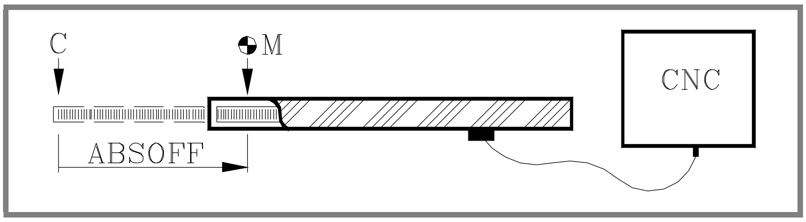
Offset referred to the distance-coded I0 (L R S) (A S X)

The CNC takes it into account if `I0TYPE` = Distance coded.

With linear encoders with distance-coded (I_0) reference marks, it is possible to know the position of the machine by simply moving the axis a distance of 20 or 100 mm.

After reading two consecutive distance-coded (I_0)reference marks (20 or 100 mm away from each other, it is possible to know the axis position with respect to the zero point of the graduated glass (C).

In order for the CNC to show the position with respect to Machine Zero (M), this parameter must be assigned the position of the machine zero (M) with respect to the glass zero point (C).



The zero point of the glass (beginning of the distance code) may be in or out the scale's measuring length.

Possible values:	within ±99999.9999 mm or degrees. within ±3937.00787 inch.
By default:	0

EXTMULT

External factor for distance-coded mark (L R S) (A X)

The CNC takes it into account if `I0TYPE` = Distance coded.

It indicates the relationship between the mechanical period (of the graduation on the glass) and the electrical period (of the feedback signal) being applied to the CNC.

Example with a Fagor linear encoder.

A "FOX" type Fagor linear encoder has a graduation period of 100 µm (gap between lines) and an electrical signal period of 4 µm.

`EXTMULT` = 100 / 4 = 25

Possible values:	from 0 to 256.
By default:	0

Values to be assigned for Fagor encoders with distance-coded I0.

Rotary encoder			EXTMULT
HO	SO	90,000 pulses	5
HO	SO	180,000 pulses	10
HOP	SOP	18,000 pulses	1

Linear encoder					EXTMULT
SOP SVOP	GOP	MOP MOC MOT MOVP	COP COC COT COVP		1
SOX SVOX	GOX	MOX	COX		2
		MOY MOVY	COY		10
				LOP	1
				LOX	10
		MOVX	COVX	FOT	5
				FOX	25
				FOP	1

I0CODDI1 **Gap between two fixed distance-coded I0's(L R S) (A S X)**

I0CODDI2 **Gap between two variable distance-coded I0's(L R S) (A S X)**

The CNC takes it into account if `I0TYPE` = Distance coded.

It is defined in number of waves.

Possible values:	from 0 to 65535.
By default:	I0CODD1=1000 and I0CODD2=1001.

Example with a Fagor linear encoder.

Gap between two fixed distance-coded I_0's	20.000 mm
Gap between two variable distance-coded I_0's	20.020 mm
Period of the sinewave signal	20 mm
Number of waves between fixed I_0's	20000/(20 x EXTMULT) = 1000
Number of waves between variable I_0's	20020/(20 x EXTMULT) = 1001

Values to be assigned for Fagor encoders with distance-coded I0.

Linear encoder				I0CODDI1	I0CODDI2
SOP SVOP	GOP	MOP MOC MOT MOVP	COP COC COT COVP	1000	1001
SOX SVOX	GOX	MOX	COX	1000	1001
		MOY MOVY	COY	1000	1001

Linear encoder				I0CODDI1	I0CODDI2
			LOP	2000	2001
			LOX	2000	2001
MOVX	COVX		FOT	1000	1001
			FOX	1000	1001
			FOP	1000	1001

Rotary encoder			I0CODDI1	I0CODDI2
HO	SO	90,000 pulses	1000	1001
HO	SO	180,000 pulses	1000	1001
HOP	SOP	18,000 pulses	1000	1001

Following error

Following error is the difference between the theoretical position and the actual (real) position of the axis.

The following error decreases when increasing the axis gain.

The more similar (identical) the following errors of interpolating axes are, the better the machining of curved sections will be on circular interpolations.

FLWEMONITOR **Type of monitoring** **(L R S) (A S X)**

Possible values: Off, Standard, Linear.

Off: The following error is not monitored, thus no error message will be issued.

Standard: The following error is monitored all the time and an error message will be issued when it exceeds the value of parameters MAXFLWE and MINFLWE.

Linear: It is a dynamic monitoring that allows for a percentual following error. This percentage is set by parameter FEDYNFACT.

By default: Off (no monitoring)

MINFLWE **Maximum following error when stopped** **(L R S) (A S)**

It indicates the maximum amount of following error allowed when the axis is stopped.

The CNC takes it into account if `FLWEMONITOR` other than "Off".

Possible values: from 0 to 99999.9999 mm or degrees.
within 0 and 3937.00787 inch.

By default: 1.0000 mm or degrees (0.03937 inch).

The `MINFLWE` value cannot be greater than 1/4 of the total axis travel (`LIMITPOS` - `LIMITNEG`).

MAXFLWE — Maximum following error in motion (L R S) (A S)

The CNC takes it into account if FLWEMONITOR other than "Off".

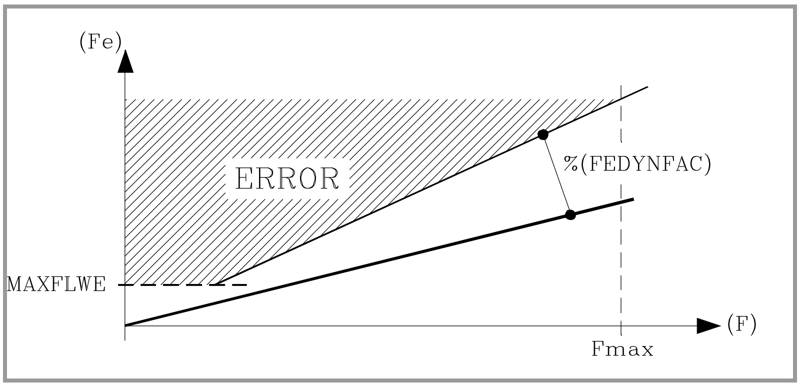
- With FLWEMONITOR = Standard, it indicates the maximum amount of following error allowed when the axis is moving.
- With FLWEMONITOR = Linear, it indicates the value starting at which the following error is dynamically monitored.

Possible values:	from 0 to 99999.9999 mm or degrees.
	within 0 and 3937.00787 inch.
By default:	1.0000 mm or degrees (0.03937 inch).

FEDYNAC — % of following error deviation (L R S) (A S)

The CNC takes it into account if FLWEMONITOR = Linear. It indicates the permitted percentage error, deviation of the real following error with respect to the theoretical one.

The CNC calculates the maximum and minimum following error (Fe) at all times depending on feedrate (F). If is not within the permitted zone (shaded area of the figure) , the CNC will issue the relevant error message.

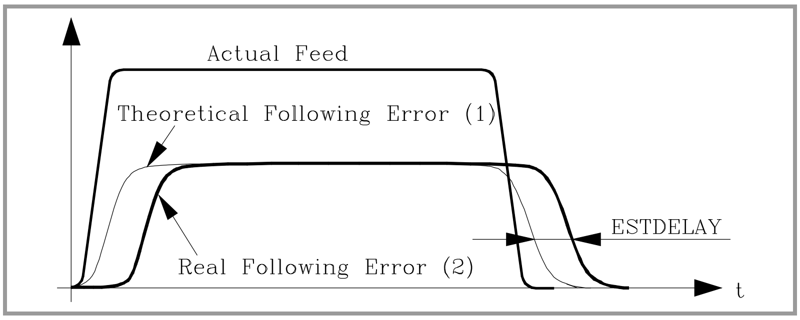


Parameter MAXFLWE indicates the value starting at which the following error will be dynamically monitored.

Possible values:	from 0 to 100 (%).
By default:	50

ESTDELAY — Following error delay (L R S) (A S)

This parameter is used to define a delay applied when estimating the following error so the theoretical value (1) comes closer to the real one (2) thus avoiding undesired following error messages.



Possible values:	within 0 and 1000000 ms.
By default:	0

INPOMAX **Time to get in position** **(L R S) (A S X)**

INPOTIME **Minimum in position time** **(L R S) (A S X)**

Parameter INPOMAX limits (maximum time) the time the axis needs to get in position.

Parameter INPOTIME sets the time the axis must stay in the in-position zone so the CNC considers it to be "in position".

They ensure that when working with dead axes (axes only controlled while moving), the movement will be completed when they are in position.

Possible values:	within 0 and 1000000 ms.
By default:	0

Axis lubrication

DISTLUBRI **Distance for lubrication pulse** **(L R S) (A S X)**

The lubrication signal is activated after travelling the distance indicated in this parameter.

Possible values:	from 0 to 2000000000 mm or degrees. from 0 to 78739920 inch.
By default:	0 (no lubrication).

The PLC reads this parameter in mm instead of doing it in tenths of a micron (0.0001 mm).

The CNC logic inputs and outputs: LUBR(axis), LUBRENA(axis) and LUBROK(axis) must be used in order for the PLC lubricates the axes and gears.

1. The LUBRENA(axis) mark indicates whether this feature is to be used or not.
2. When the axis has traveled the distance indicated by (a.m.p.) DISTLUBRI, the LUBR(axis) mark is set to "1" to "tell" the PLC that the axis must be lubricated.
3. After lubricating the axis, the PLC sets the LUBROK(axis) mark high (=1) to let the CNC know that the axis has been lubricated.
4. The CNC sets the LUBR(axis) mark low (=0) and resets its count to "0".

Module definition in rotary axes and spindle

MODUPLIM **Module's upper limit** **(R S) (A Ss X)**

MODLOWLIM **Module's lower limit** **(R S) (A Ss X)**

The CNC takes them into account if (a.m.p.) AXISMODE = Module.

Possible values:	within ±99999.9999º.
By default:	MODUPLIM = 360º and MODLOWLIM = 0º.

For a reading within ±180°, set MODUPLIM = 180º and MODLOWLIM = -180º.

MODNROT **Module error. Turns** **(R S) (A Ss X)**
MODERR **Module error. Increments** **(R S) (A Ss X)**

The CNC takes them into account if (a.m.p.) `AXISMODE` = Module and `MODCOMP` = Yes.

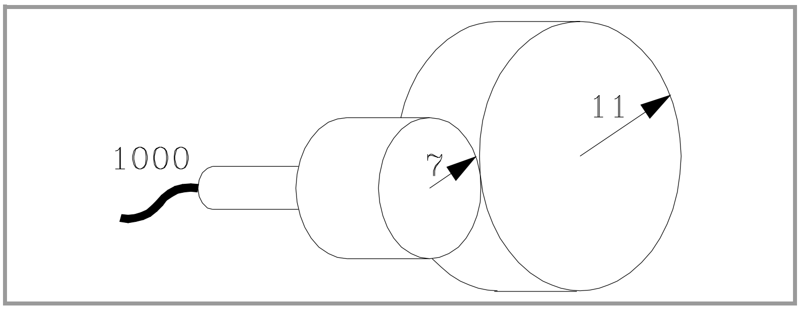
They indicate the compensation to be applied to compensate for an inexact axis resolution.

Possible values:	MODNROT	from 1 to 32767 turns.
	MODERR	within ±32767.
By default:	MODNROT = 1 and MODERR = 0.	

Example:

A rotary axis having a gear ratio of 7/11 and a 1000 line encoder mounted on the motor.

PITCH = 360 x 7/11 = 229.090909 ...



- If NPULSES = 1000 and PITCH = 229.091 Every 11 turns of the motor:

  the CNC will read 11 x 229.091 = 2520.0010º

  and the axis will rotate 7 x 360º = 2520.0000º

  Therefore: MODNROT = 11 MODERR = -10

- If NPULSES = 1000 and PITCH = 229.090 Every 11 turns of the motor:

  the CNC will read 11 x 229.090 = 2519.9900º

  and the axis will rotate 7 x 360º = 2520.0000º

  Therefore: MODNROT = 11 MODERR = 100

Spindle

SZERO **Speed considered "0 rpm"** **(S) (A X)**

It indicates the rpm value below which the spindle is considered to be stopped.

Possible values:	from 0 to 100000 rpm.
By default:	0

POLARM3 **Sign of the analog voltage for M3** **(S) (A Ss X)**
POLARM4 **Sign of the analog voltage for M4** **(S) (A Ss X)**

Possible values:	Positive / Negative.
By default:	POLARM3 = Positive. POLARM4 = Negative.

Analog voltage

SERVOOFF **Offset compensation** **(L R S) (A)**

Analog voltage applied as offset to the drive.

It is given in D/A converter units which could be any integer within ±32767. A value of ±32767 corresponds to ±10V.

SERVOOFF	1	3277	32767
Offset	0.3 mV	1 V	10 V

The default value for this parameter is ·0·.

MINANOUT **Minimum analog output** **(L R S) (A)**

It is given in D/A converter units which could be any integer from 0 to 32767. A value of 32767 corresponds to 10V.

MINANOUT	1	3277	32767
Analog voltage	0.3 mV	1 V	10 V

The default value for this parameter is "0".

Analog output / Feedback input

ANAOUTID **Axis analog output** **(L R S) (A)**
COUNTERID **Feedback input of the axis** **(L R S) (A)**

They are numbered following the order of the remote groups (rotary switch of the Power Supply element).

If there are several Counter modules in each group, the order is from top to bottom and from left to right.

Possible values:	from 0 to 16
By default:	0

2.4 Machine parameters for JOG mode

Handwheel configuration

A handwheel may be configured so it is common to all the axes or just for one (individual).

- An individual handwheel can only move the associated axis.
- When using a common handwheel, the axis to be moved must be selected using the JOG keys. They cannot move axes that have an individual handwheel associated with them.

The possible configurations are:

- A single handwheel to control all the axes.
- One or several individual handwheels. Each associated with one axis.
- One or several individual handwheels and a handwheel common to the rest of the axes.

To set the resolution of each axis, the distance it moves at each switch position, set the (a.m.p.) MPGRESOL.

NMPG **Number of handwheels**

Possible values:	from 0 to 3.
By default:	0 (there are no handwheels).

MANPG **Table of handwheels**

A table is created for each handwheel with all the parameters COUNTERID and MGAXIS

COUNTERID **Feedback input for the handwheel**

The handwheels may be connected through the keyboards (3 per keyboards) and through the counter modules of the remote groups (4 per module).

Possible values:	from -8 to 16.
By default:	0

The handwheels connected via keyboard are numbered from -1 to -8. The order of the keyboards is that of the CAN bus.

- First keyboard -1, -2, -3.
- Second keyboard -4, -5, -6.
- Third keyboard -7, -8.

The ones connected to the remote groups (values from 1 to 16) are numbered according to the order of the remote groups (rotary switch of the Power Supply module). If there are several Counter modules in each group, the order is from top to bottom and from left to right.

MPGAXIS **Axis associated with the handwheel**

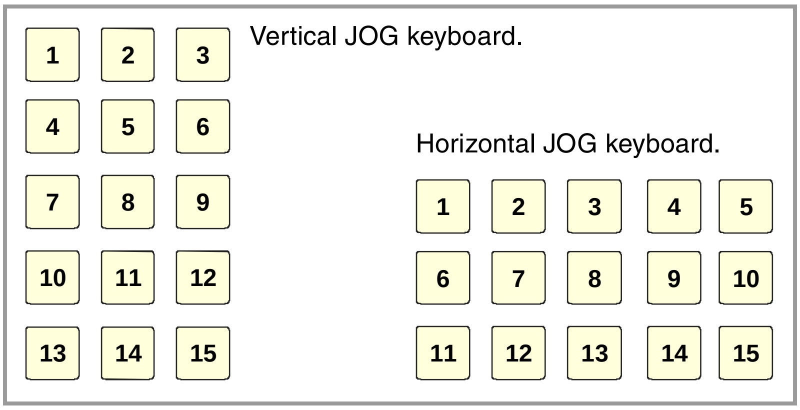
Name of the axis that is associated with the handwheel.

To define a general handwheel, associated with all the axes that do not have their own handwheel, leave this parameter blank.

Configuration of the JOG keys

JOGKEYDEF **Axis and moving direction**

There are 15 parameters to define the function of each JOG key. The first parameter corresponds to the upper left hand key; the rest of the parameters will count from left to right and from the top down.



The JOG keypad may consist of the following keys:

X+	7+	*Keys to define the axis and the jogging direction.*
X	7	*Keys to define the axis to be jogged.*
+	–	*Keys to define the direction of the movement.*
		Rapid key

Both types of keys may be defined at the same JOG keypad.

To define the behavior of each key, assign to them one of the following values:

- They are defined with a value between –1 and +16 (signed) for the keys that define the axis and the direction. The sign indicates the positive direction (+) or the negative direction (-) and the number corresponds to the logic axis (g.m.p.) AXISNAME.
- They are defined with a value between 1 and 16 (unsigned) for the keys that only define the axis.
- They are defined with the "+" and "-" values for the keys that only define the moving direction.
- They are defined with the "R" value for the rapid key.

JOGTYPE JOG behavior

This parameter is applied when the jog keyboard has different keys for selecting the axis and the jogging direction.

In this case, jogging an axis requires activating both the axis key and the moving direction. There are two options, depending on how the jog keyboard has been configured.

- With the "pressed axis" option, the axis will move while both keys are pressed, the axis key and the direction key.
- With the "maintained axis" option, pressing the axis key will select it. The axis will move while the direction key is kept pressed. To de-select the axis, press [ESC] or [STOP].

Possible values:	Pressed axis / Maintained axis.
By default:	Pressed axis.

2.4.1 Example of how to set the handwheels and JOG keys

Handwheel setting

A machine with X + Y + Z + A axes, we would like:

- X axis handwheel

  Feedback input: keyboard (MPG1)

  Disk graduated with 100 lines

  Pulses/ turn: 100

  Resolutions (X): 0.001, 0.01, 0.1

- Y axis handwheel

  Feedback input: keyboard (MPG2)

  Disk graduated with 100 lines

  Pulses/ turn: 200

  Resolutions (Y): 0.001, 0.01, 0.1

- Handwheel rest of the axes (Z, A)

  Feedback input: Counter (X1)

  Disk graduated with 100 lines

  Pulses/ turn: 100

  Resolutions (Z): 0.001, 0.01, 0.1

  Resolutions (A): 0.01, 0.1, 1

Parameter setting:

NMPG	3	3-handwheel system

- X axis handwheel (MANPG 1)

COUNTERID	-1	Keyboard (MPG1)
MPGAXIS	X	
MPGRESOL 1	0.001	resolution 0.001
MPGRESOL 10	0.01	resolution 0.01
MPGRESOL 100	0.1	resolution 0.1

- Y axis handwheel (MANPG 2)

COUNTERID	-2	Keyboard (MPG2)
MPGAXIS	Y	
MPGRESOL 1	0.0005	resolution 0.001
MPGRESOL 10	0.005	resolution 0.01
MPGRESOL 100	0.05	resolution 0.1

- Handwheel associated with the rest of the axes (Z + A)

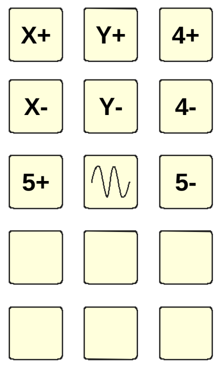
COUNTERID	1	Counter (X1)
MPGAXIS		Associated with rest of axes
Z- MPGRESOL 1	0.001	resolution 0.001
Z- MPGRESOL 10	0.01	resolution 0.01
Z- MPGRESOL 100	0.1	resolution 0.1
A- MPGRESOL 1	0.01	resolution 0.01
A- MPGRESOL 10	0.1	resolution 0.1
A- MPGRESOL 100	1	resolution 1

Setting the JOG keys

Example. On a machine with the X, Y, U, V axes defined as AXISNAME 1, 2, 3, 4, we would like to jog the X and Y axes with the keys that have the same names, the U axis with the keys for the 4th axis and the V axis with those of the 5th axis.

Example 1: Vertical JOG keyboard

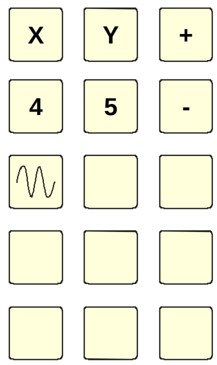
JOG keypad



JOGKEYDEF	Key	Value
1	[X+]	1+
2	[Y+]	2+
3	[4+]	3+
4	[X-]	1-
5	[Y-]	2-
6	[4-]	3-
7	[5+]	4+
8	[R]	R
9	[5-]	4-
10 - 15	- - -	

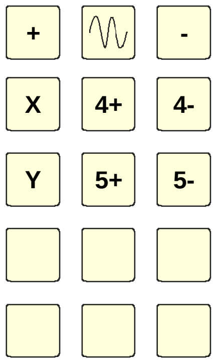
Example 2: Vertical JOG keyboard

JOG keypad



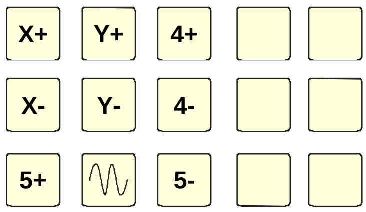
JOGKEYDEF	Key	Value
1	[X]	1
2	[Y]	2
3	[+]	+
4	[4]	3
5	[5]	4
6	[-]	-
7	[R]	R
8	- - -	
9	- - -	
10 - 15	- - -	

Example 3: Vertical JOG keyboard



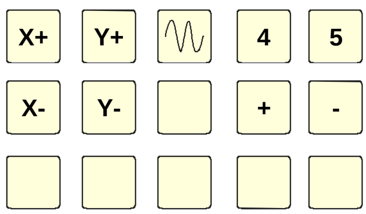
JOG keypad	JOGKEYDEF	Key	Value
	1	[+]	+
	2	[R]	R
	3	[-]	-
	4	[X]	1
	5	[4+]	3+
	6	[4-]	3-
	7	[Y]	2
	8	[5+]	4+
	9	[5-]	4-
	10 - 15	- - -	

Example 4: Horizontal JOG keyboard



JOG keypad	JOGKEYDEF	Key	Value
	1	[X+]	1+
	2	[Y+]	2+
	3	[4+]	3+
	4 - 5	- - -	
	6	[X-]	1-
	7	[Y-]	2-
	8	[4-]	3-
	9 - 10	- - -	
	11	[5+]	4+
	12	[R]	R
	13	[5-]	4-
	14 - 15	- - -	

Example 4: Horizontal JOG keyboard



JOG keypad	JOGKEYDEF	Key	Value
	1	[X+]	1+
	2	[Y+]	2+
	3	[R]	R
	4	[4]	4
	5	[5]	5
	6	[X-]	1-
	7	[Y-]	2-
	8	- - -	
	9	[+]	+
	10	[-]	-
	11 - 15	- - -	

2.5 Machine parameters for the M function table

M function table

MTABLESIZE **Number of table elements**

Up to 200 auxiliary M functions may be defined. Each function may be assigned a subroutine and defined the type of synchronization.

Possible values:	from 0 to 200.
By default:	50

It must be borne in mind that some auxiliary functions, besides what is indicated in this table, have a specific meaning when used in a CNC program. These functions are M00, M01, M02, M03, M04, M05, M06, M08, M09, M19, M30, M41, M42, M43 and M44.

DATA **M function table**

For each data (DATA), the following parameters must be defined: MNUM, SYNCHTYPE, MTIME and MPROGNAME.

MNUM **M function number**

Possible values:	from 0 to 65535.

SYNCHTYPE **Type of synchronization**

Since these functions may be programmed together with the movement of the axes, in the same block, it must be indicated when the function is to be sent out to the PLC and when it will be checked that it has been already executed (synchronization).

It may be sent and/or synchronized before or after the movement.

Possible values:	M not synchronized
	Sent before - Synchronized before
	Sent before - Synchronized after
	Sent after - Synchronized after

- If an M function is used to turn on a lamp, it will be set without synchronization because there is no need to check that the lamp has indeed turned on.
- Functions M03 and M04 to start up the spindle, they should be executed and synchronized before the movement.
- Function M5, to stop the spindle, should be executed and synchronized after the movement.

MTIME

Estimated time for an M function

In Editing - Simulation mode, there is an option that allows calculating the time required to execute a part with the machining conditions established in the program.

This parameter may be set to fine tune that calculation.

Possible values:	from 0 to 1000000 ms.
By default:	0 ms.

MPROGNAME

Name of subroutine associated with M function

The subroutines associated with the M functions must be located in the "C:\CNC8070 \MTB \SUB" folder. To send the `M` function to the PLC, it must be programmed in the subroutine.

Possible values:	any text with up to 64 characters.

The type of synchronization of the `M` functions that have an associated subroutine must be: "M without synchronization" or "Sent after - Synchronized after". The CNC executes the associated subroutine after the executing the programmed movement (if any).

i

To have different procedures in the subroutines associated with certain M functions, the code of each channel may be differentiated within the subroutine using the variable `(V.)G.CNCHANNEL`.

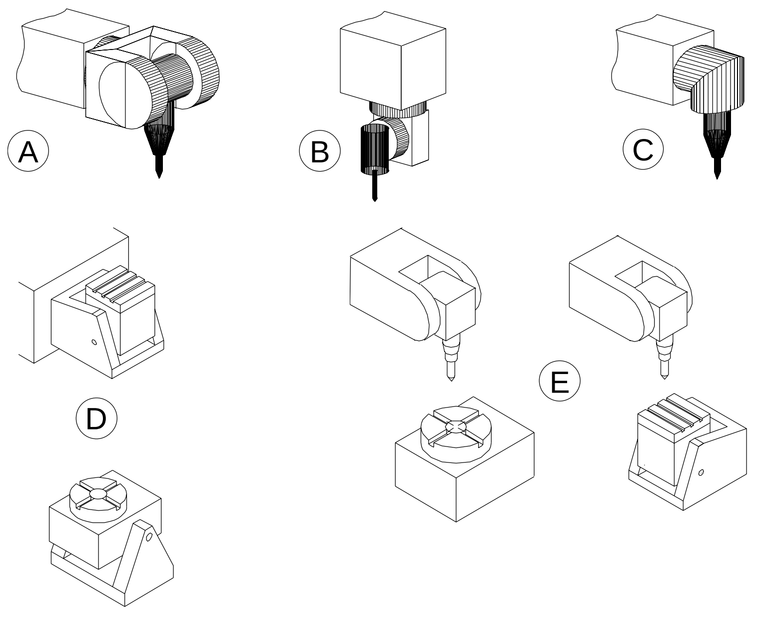
2.6 Machine parameters for kinetics

Up to 6 different kinetics may be set for a machine. The type of kinetics to be applied must be defined for each of them.

(g.m.p.) KINID indicates the kinetics number (not type) assumed by the CNC on power-up. To select another one from the part-program, use the #KIN ID instruction.

The following types of kinematics may be controlled.

- Spherical, orthogonal and angular spindles.
- Rotary tables.
- Spindle and table.
- Kinematics of the –C– axis.



***(A)** Spherical spindle.*
***(B)** Orthogonal spindle.*
***(C)** Swivel (angular) spindle.*
***(D)** Rotary tables.*
***(E)** Spindle - table.*

Distribution of axes

One kinematics may be active per channel. A kinematics may be configured by between 3 and 5 axes. All the axes making up the kinematics must belong to the same channel and must occupy the first positions in the following order.

1st axis	First main axis of the plane (abscissa).
2nd axis	2nd main axis of the plane (ordinate).
3rd axis	Longitudinal axis.
4th axis	Four axes of the kinematics.
5th axis	Fifth axis of the kinematics.
6th axis and the next ones	Rest of the axes

The first 3 axes must be linear. Spindle compensation will be applied on to them. The fourth and fifth axes of the kinematics may be either rotary or linear, depending on the type of kinematics.

Kinematics

NKIN **Number of different kinematics**

Possible values:	from 0 to 6.
By default:	0

KINEMATIC **Kinematics table**

The following parameters must be defined for each kinetics:

TYPE Kinematics type.

DATA1 - DATA42 Data required by each kinematics.

The various types of kinetics offered by Fagor and their associated data shown next. To include the kinetics for your machine, contact Fagor Automation.

TYPE **Kinematics type**

1 = Orthogonal or spherical spindle head YX

2 = Orthogonal or spherical head ZX

3 = Orthogonal or spherical spindle head XY

4 = Orthogonal or spherical spindle head ZY

5 = Angular spindle head XZ

6 = Angular spindle head YZ

7 = Angular spindle head ZX

8 = Angular spindle head ZY

9 = Rotary table AB

10 = Rotary table AC

11 = Rotary table BA

12 = Rotary table BC

13 = Spindle - AB table

14 = Spindle - AC table

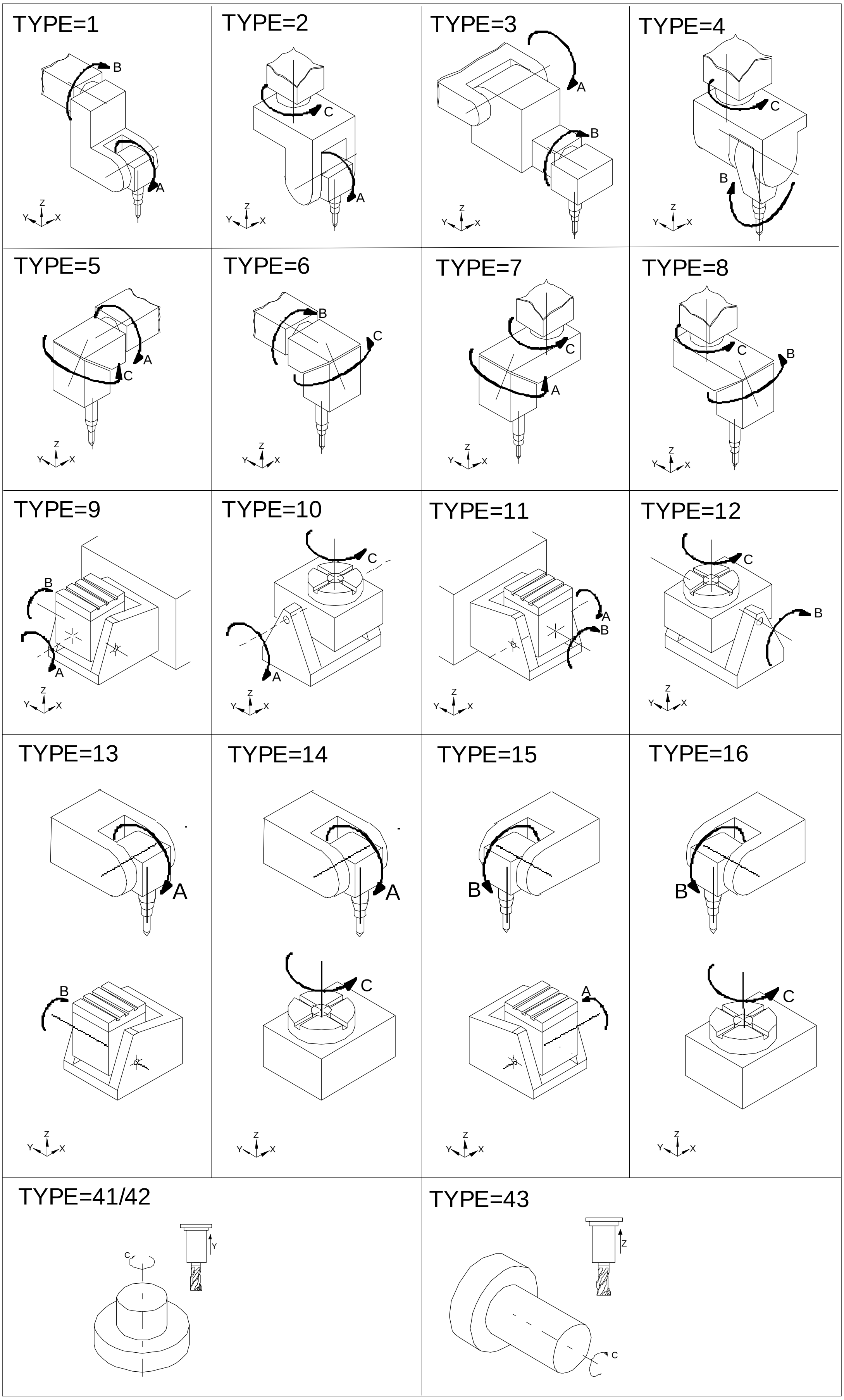
15 = Spindle - BA table

16 = Spindle - BC table

41 = C axis. Machining of the face of the part when ALIGNC=YES.

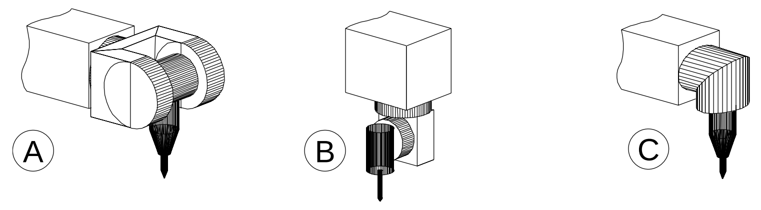
42 = C axis. Machining of the face of the part when ALIGNC=NO.

43 = C axis. Machining of the turning side of the part.


TYPE=1
TYPE=2
TYPE=3
TYPE=4
TYPE=5
TYPE=6
TYPE=7
TYPE=8
TYPE=9
TYPE=10
TYPE=11
TYPE=12
TYPE=13
TYPE=14
TYPE=15
TYPE=16
TYPE=41/42
TYPE=43

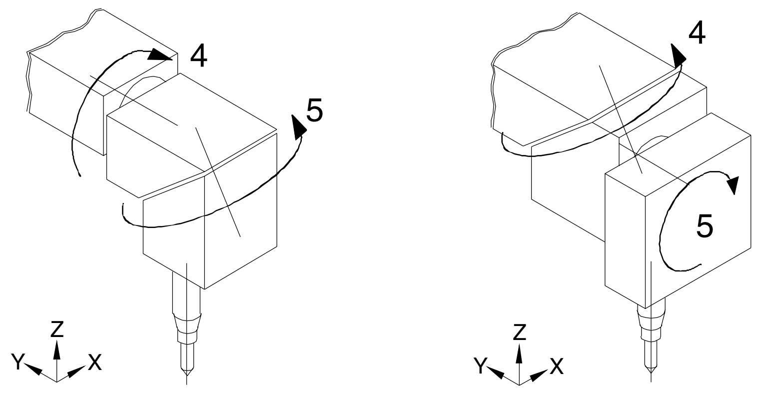
Definition of the spindle kinmatics (Types 1 through 8)

It is possible to control spheric spindle heads orthogonal and angular.

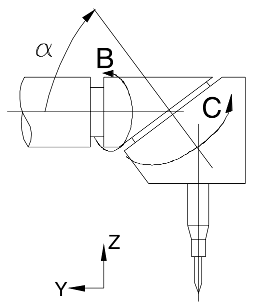


(A) *Spherical spindle.*
(B) *Orthogonal spindle.*
(C) *Swivel (angular) spindle.*

When having an angular spindle head, the main rotary axis (4) must rotate around one of the main axes (X, Y, Z) and the secondary or dragged axis (5) will form a particular angle.



The left figure meets this condition, whereas in the right one the main rotary axis (4) does not rotate around the Y axis (it forms an angle with it).



From now on, all the explanations will assume that the main axes are X Y and Z and their associated rotary axes A, B and C.

DATA1...DATA7 **Spindle dimensions**

All of them need not be defined. The parameters to be defined for each model and their meanings are listed here below.

They may be defined with a positive or negative value. The (+) sign in the illustrations indicates that the direction is assumed as positive.

DATA1 It indicates the distance between the tip of the quill and the secondary rotary axis along the Z axis.

DATA2 It indicates the distance between the secondary rotary axis and the main axis along the X axis.

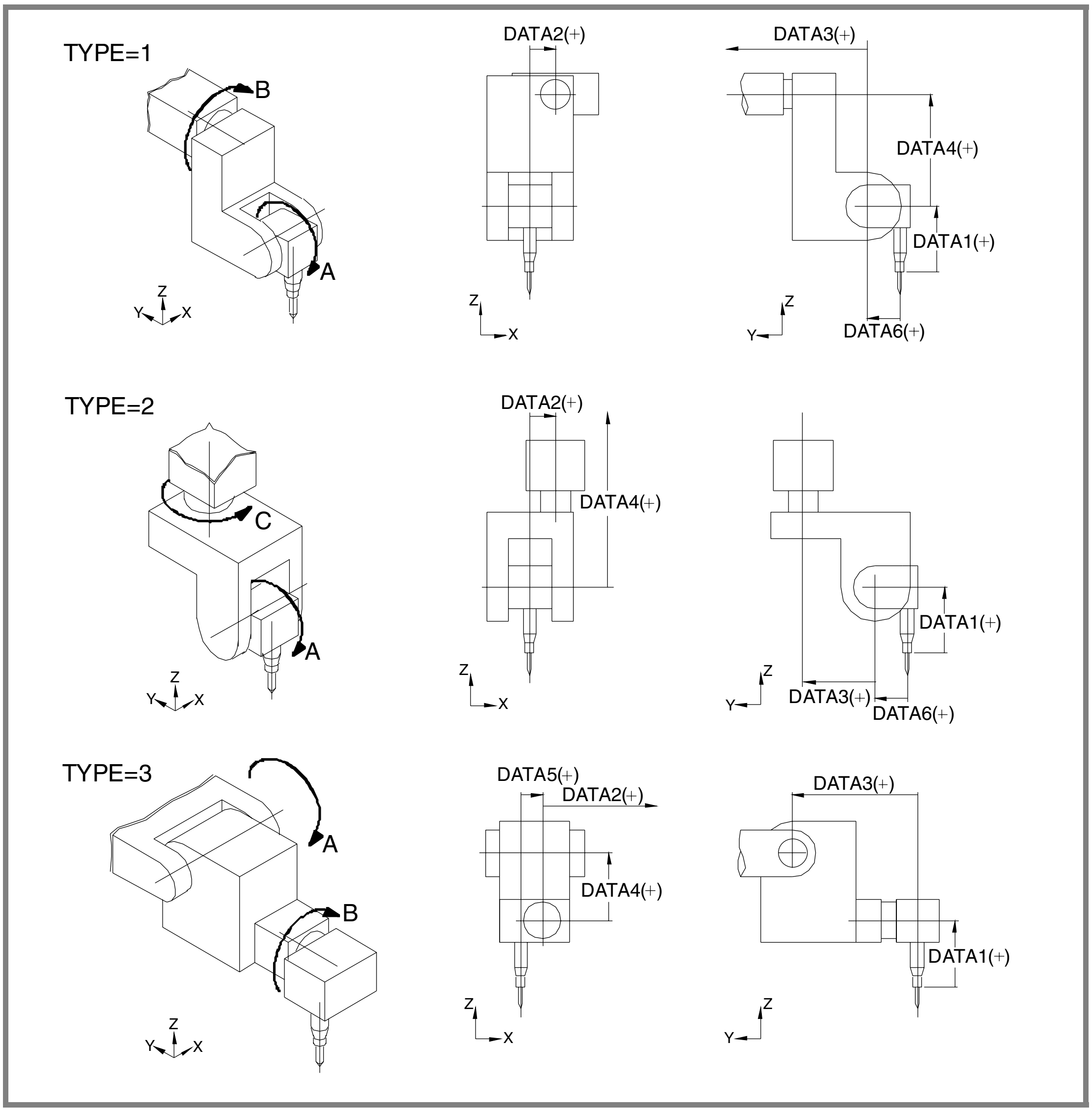
DATA3 It indicates the distance between the secondary rotary axis and the main axis along the Y axis.

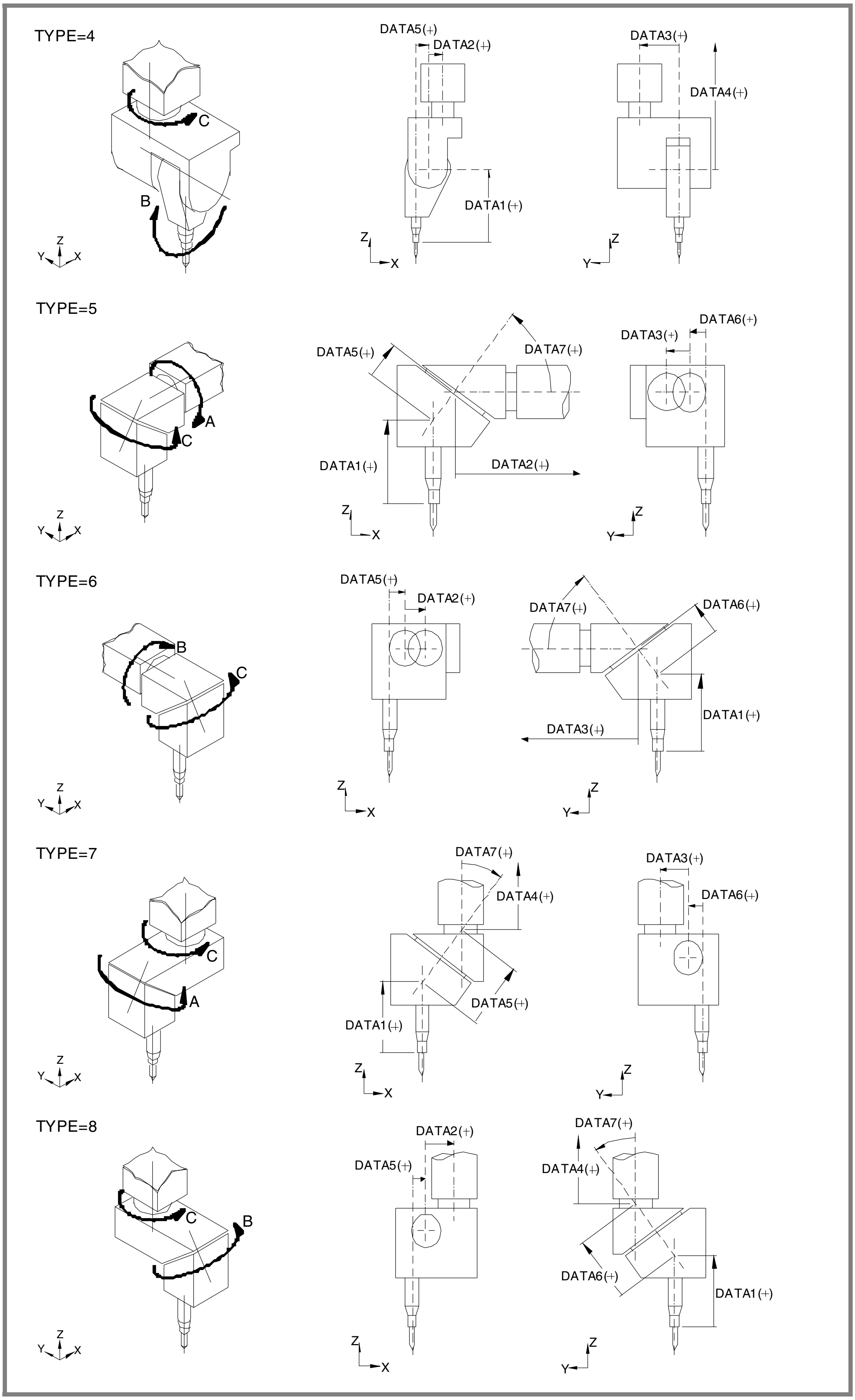
DATA4 It indicates the distance between the secondary rotary axis and the main axis along the Z axis.

DATA5 It indicates the distance between the tool axis and the secondary rotary axis along the X axis.

DATA6 It indicates the distance between the tool axis and the secondary rotary axis along the Y axis.

DATA7 It indicates the angle between the main and secondary rotary axes principal on swivel spindle heads.




TYPE=4
C
B
DATA5(+)
DATA2(+)
DATA1(+)
DATA3(+)
DATA4(+)
TYPE=5
A
C
DATA5(+)
DATA7(+)
DATA1(+)
DATA2(+)
DATA3(+)
DATA6(+)
TYPE=6
B
C
DATA5(+)
DATA2(+)
DATA7(+)
DATA6(+)
DATA3(+)
DATA1(+)
TYPE=7
C
A
DATA7(+)
DATA4(+)
DATA5(+)
DATA1(+)
DATA3(+)
DATA6(+)
TYPE=8
C
B
DATA2(+)
DATA5(+)
DATA7(+)
DATA4(+)
DATA6(+)
DATA1(+)

DATA8 **Rest position of the main rotary axis**

DATA9 **Rest position of the secondary rotary axis.**

The rest position of the spindle is when the tool is perpendicular to the work plane (parallel to the longitudinal axis).

Possible values:	within ±99999.9999º.
By default:	0

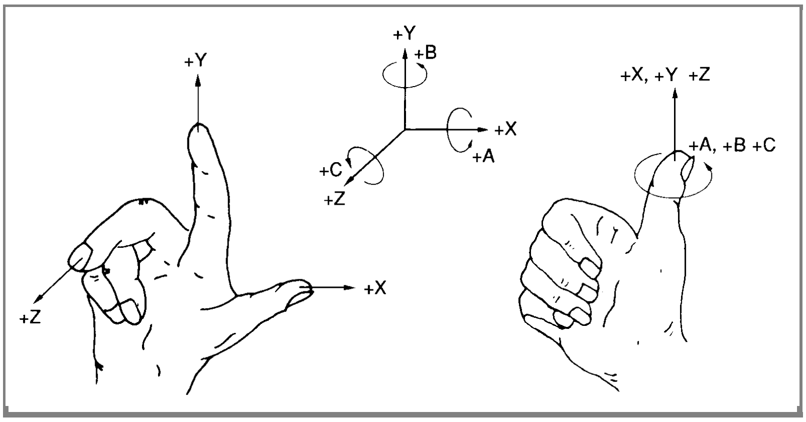
DATA10 **Turning direction of the main rotary axis**

DATA11 **Turning direction of the secondary rotary axis**

Possible values:	0	direction following the standard DIN 66217.
	1	direction contrary to the standard DIN 66217.
By default:	0	

The direction of the XYZ axes, according to the DIN 66217 standard is easy to remember using the rule of the right hand.

On rotary axes, the turning direction is established when bending your fingers (closing your hand) around the associated linear axis while your thumb is pointing in the positive direction of the linear axis.



DATA12 **Manual rotary axes or servo-controlled**

0 =Both axes are servo-controlled

1 = Main axis manual and secondary servo-controlled

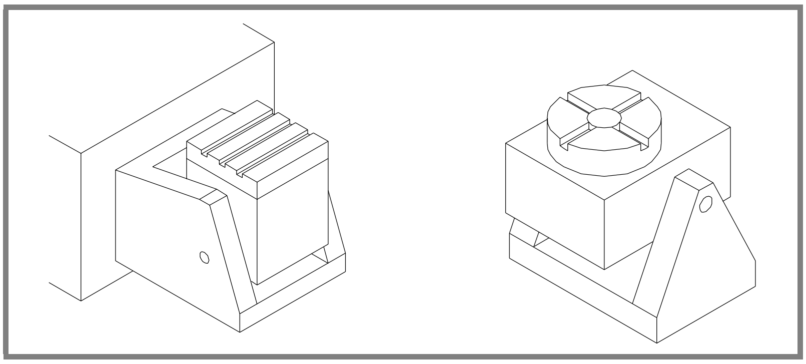
2 = Main axis servo-controlled and secondary manual

3 = Both axes are manual.

DATA 13...DATA42 **(Not being used at this time)**

Definition of the table kinematics (Types 9 through 12)

The following types of rotary tables may be controlled.



DATA1 **(Not being used at this time)**

DATA2...DATA5 **Table dimensions**

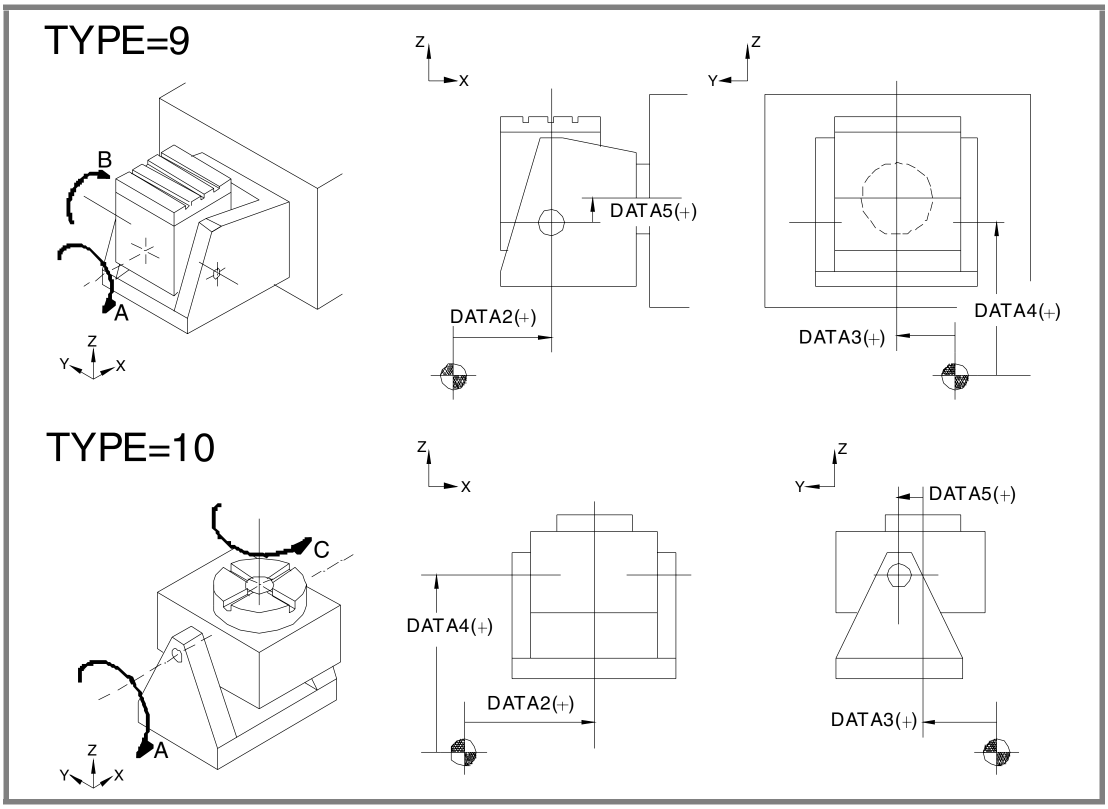
They may be defined with a positive or negative value. The (+) sign in the illustrations indicates that the direction is assumed as positive.

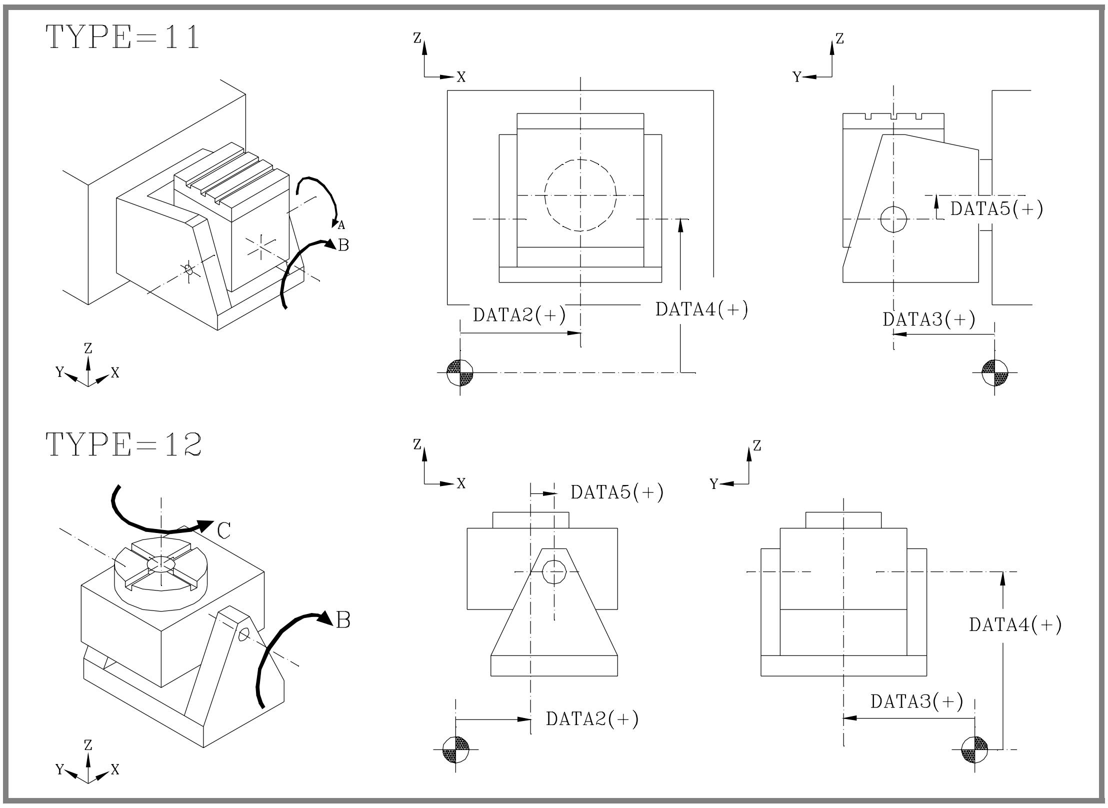
DATA2 It indicates the position of the secondary rotary axis or the intersection with the primary axis along the X axis.

DATA3 It indicates the position of the secondary rotary axis or the intersection with the primary axis along the Y axis.

DATA4 It indicates the position of the secondary rotary axis or the intersection with the primary axis along the Z axis.

DATA5 It indicates the distance between the secondary and the main rotary tables.





DATA6...DATA7 **(Not being used at this time)**

DATA8 **Rest position of the main rotary axis**

DATA9 **Rest position of the secondary rotary axis.**

The rest position of the spindle is when the tool is perpendicular to the work plane (parallel to the longitudinal axis).

Possible values:	within ±99999.9999º.
By default:	0

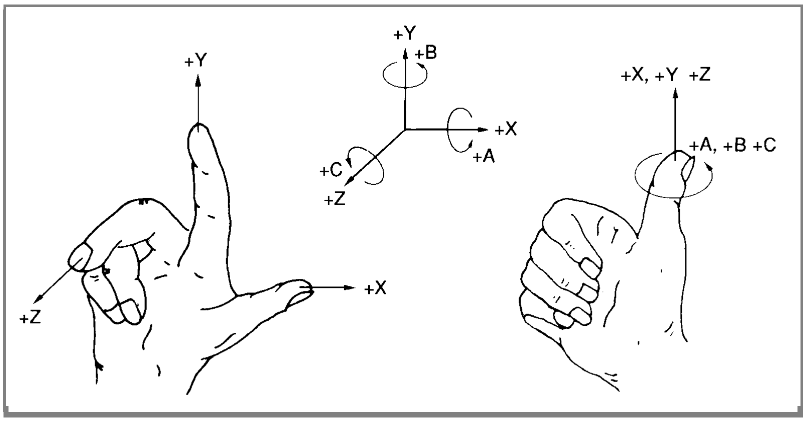
DATA10 **Turning direction of the main rotary axis**

DATA11 **Turning direction of the secondary rotary axis**

Possible values:	0	direction following the standard DIN 66217.
	1	direction contrary to the standard DIN 66217.
By default:	0	

The direction of the XYZ axes, according to the DIN 66217 standard is easy to remember using the rule of the right hand.

On rotary axes, the turning direction is established when bending your fingers (closing your hand) around the associated linear axis while your thumb is pointing in the positive direction of the linear axis.



DATA12 **Manual rotary axes or servo-controlled**

0 =Both axes are servo-controlled

1 = Main axis manual and secondary servo-controlled

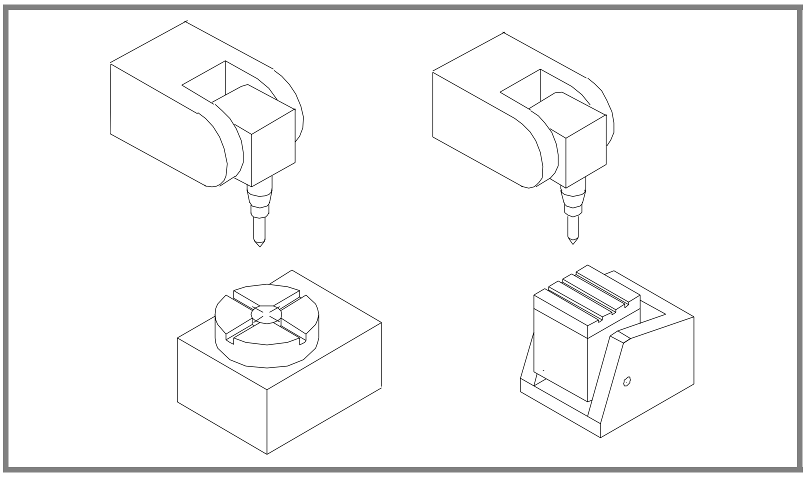
2 = Main axis servo-controlled and secondary manual

3 = Both axes are manual.

DATA 13...DATA42 **(Not being used at this time)**

Definition of the kinematics of the spindle - table (Types 13 through 16)

In this type of kinematics, one rotating axis is at the spindle and the other one at the table. The one at the spindle orients the tool and that of the work table orients the part.



The order of the axes in the channel where the kinematics is applied is:

- The first two axes correspond to the work plane.
- The third axis corresponds to the tool axis.
- The fourth axis corresponds to the rotary axis of the spindle.
- The fifth axis corresponds to the rotary axis of the table.

The type of kinematics is defined being the tool parallel to the third axis of the channel and the work plane perpendicular to the tool.

DATA1...DATA6 **Spindle dimensions and table placement.**

All of them need not be defined. The parameters to be defined for each kinematics and their meanings are listed here below.

They may be defined with a positive or negative value. The (+) sign in the illustrations indicates the direction it assumes as positive.

DATA1 It indicates the distance between the tip of the quill and the spindle rotary axis along the Z axis.

DATA2 It indicates the distance between the tool axis and the spindle rotary axis along the X axis.

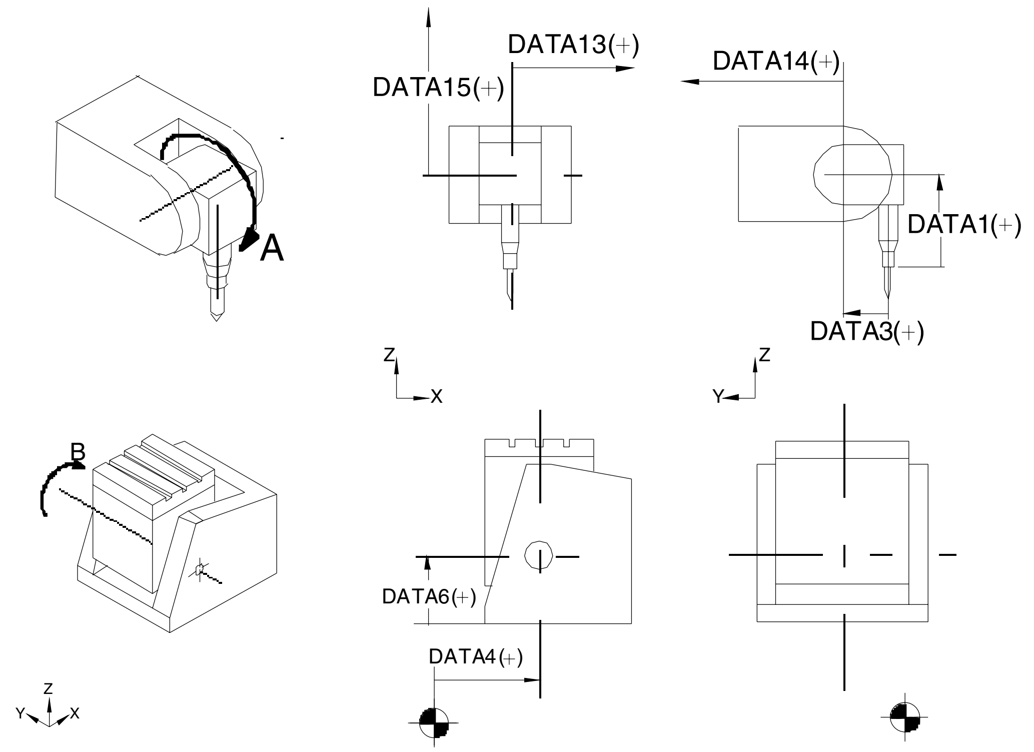
DATA3 It indicates the distance between the tool axis and the spindle rotary axis along the Y axis.

DATA4 It indicates the position of the table's rotary axis along the X axis.

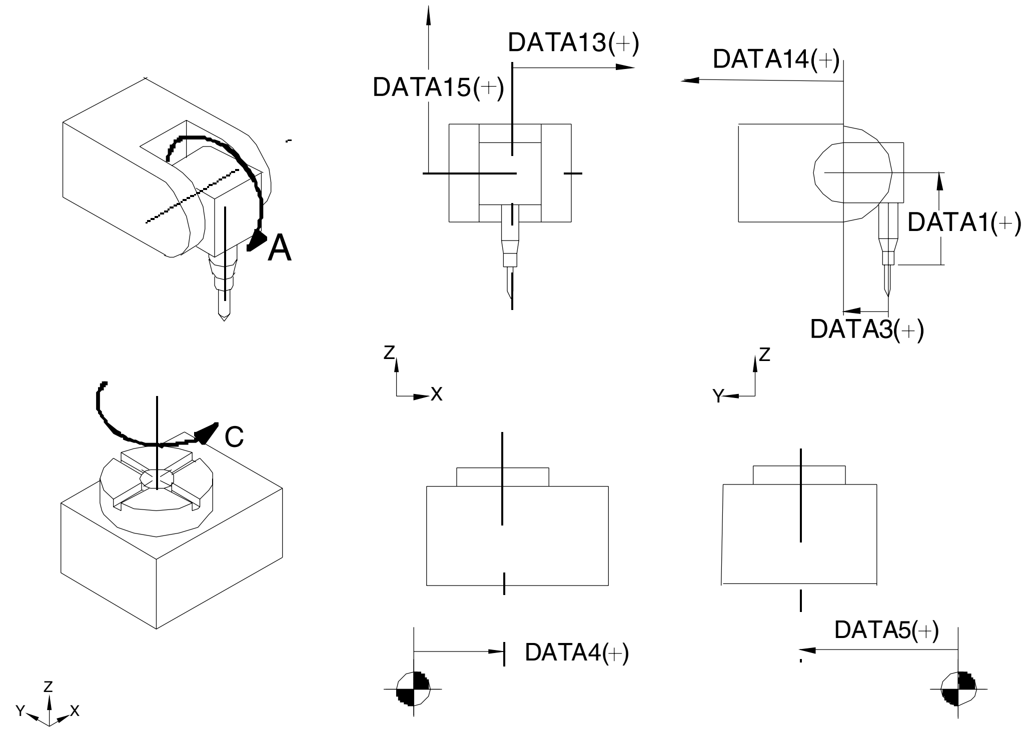
DATA5 It indicates the position of the table's rotary axis along the Y axis.

DATA6 It indicates the position of the table's rotary axis along the Z axis.

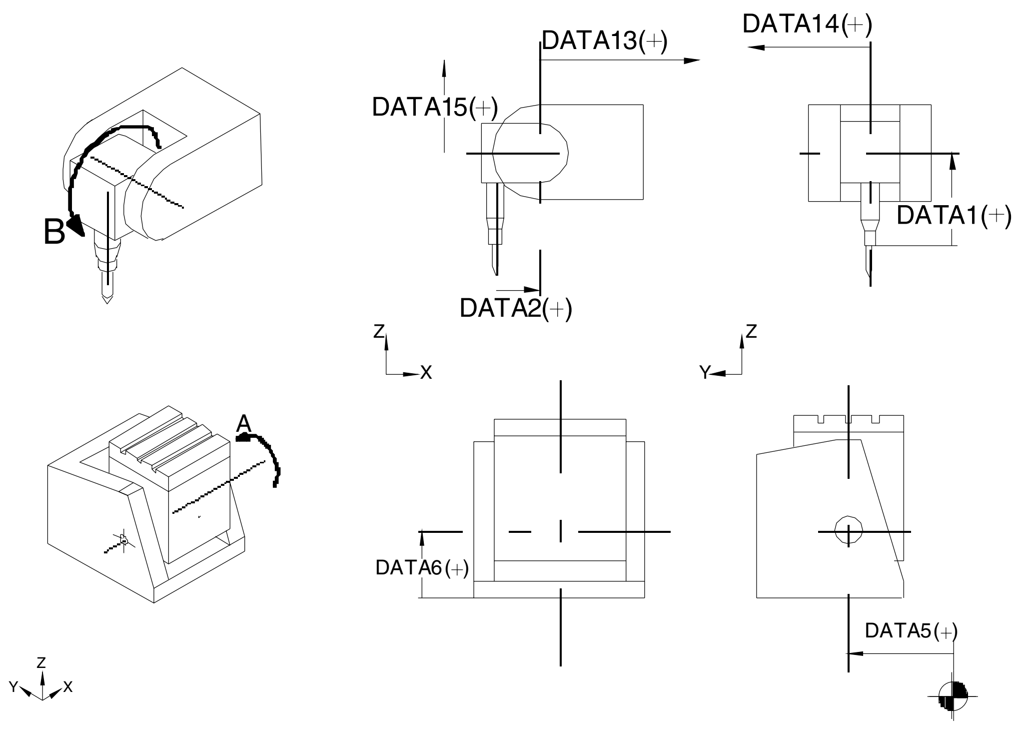
TYPE=13



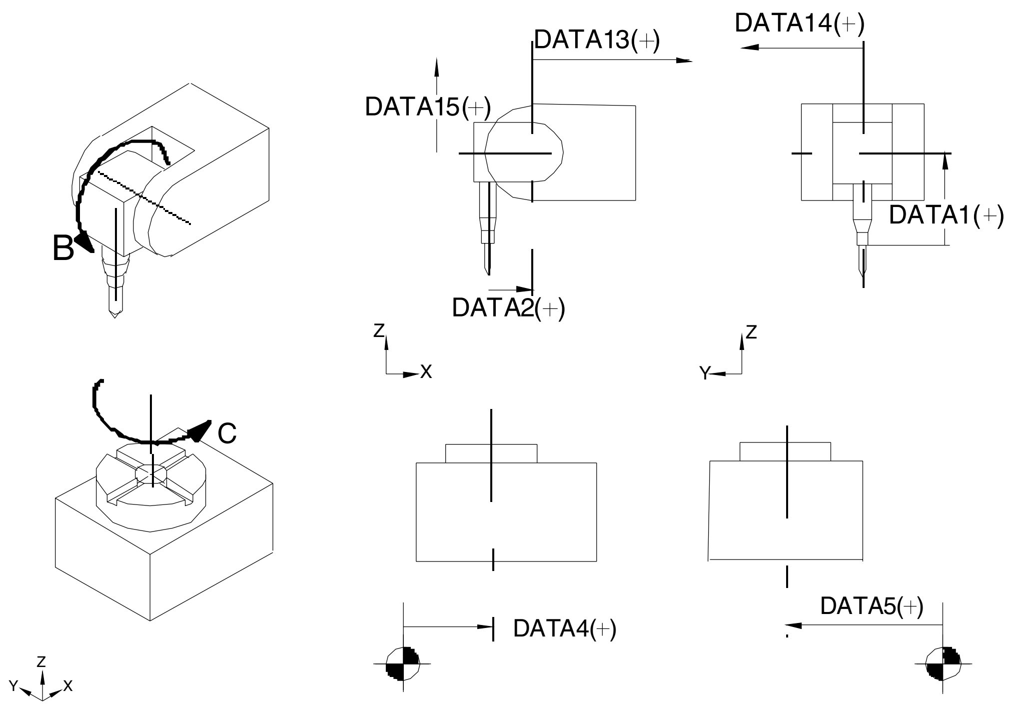
TYPE=14



TYPE=15



TYPE=16



DATA8 **Rest position of the main rotary axis**

DATA9 **Rest position of the secondary rotary axis.**

The rest position of the spindle is when the tool is perpendicular to the work plane (parallel to the longitudinal axis).

Possible values:	within ±99999.9999º.
By default:	0

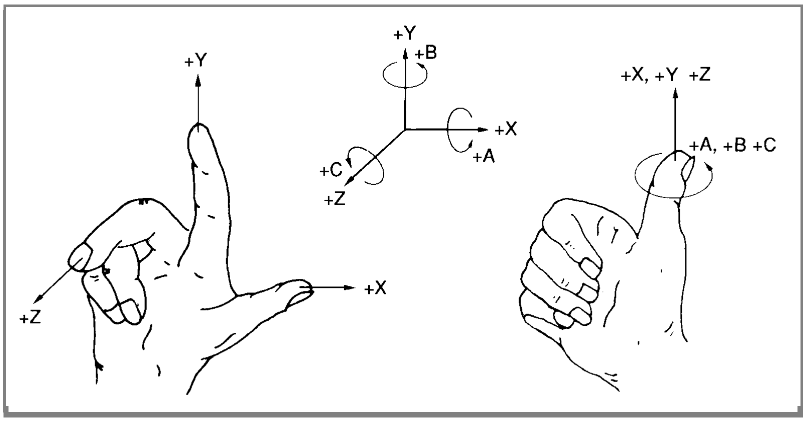
DATA10 **Turning direction of the main rotary axis**

DATA11 **Turning direction of the secondary rotary axis**

Possible values:	0	direction following the standard DIN 66217.
	1	direction contrary to the standard DIN 66217.
By default:	0	

The direction of the XYZ axes, according to the DIN 66217 standard is easy to remember using the rule of the right hand.

On rotary axes, the turning direction is established when bending your fingers (closing your hand) around the associated linear axis while your thumb is pointing in the positive direction of the linear axis.



DATA12 **Manual rotary axes or servo-controlled**

0 =Both axes are servo-controlled

1 = Main axis manual and secondary servo-controlled

2 = Main axis servo-controlled and secondary manual

3 = Both axes are manual.

DATA 13...DATA15 **Spindle placement**

DATA13 Distance defining the spindle placement, from the rotary axis, along the X axis.

DATA14 Distance defining the spindle placement, from the rotary axis, along the Y axis.

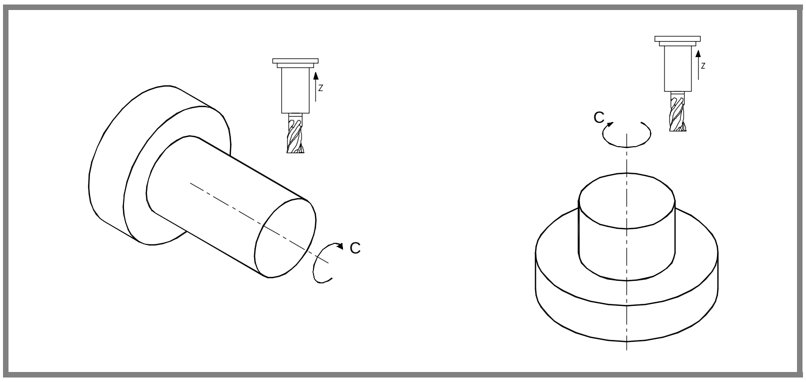
DATA15 Distance defining the spindle placement, from the rotary axis, along the Z axis.

DATA16...DATA42 **(Not being used at this time)**

Definition of the C axis kinematics (Types 41 through 43)

In this type of kinematics, one must define the physical location of the rotary axis with respect to the linear axes. These kinematics are assumed automatically when executing functions #CYL or #FACEvia part-program.

If these kinematics are defined, it assumes DATA2=0 or, which is the same, that the rotary axis coincides with the linear axis (e.g. the spindle of a lathe).

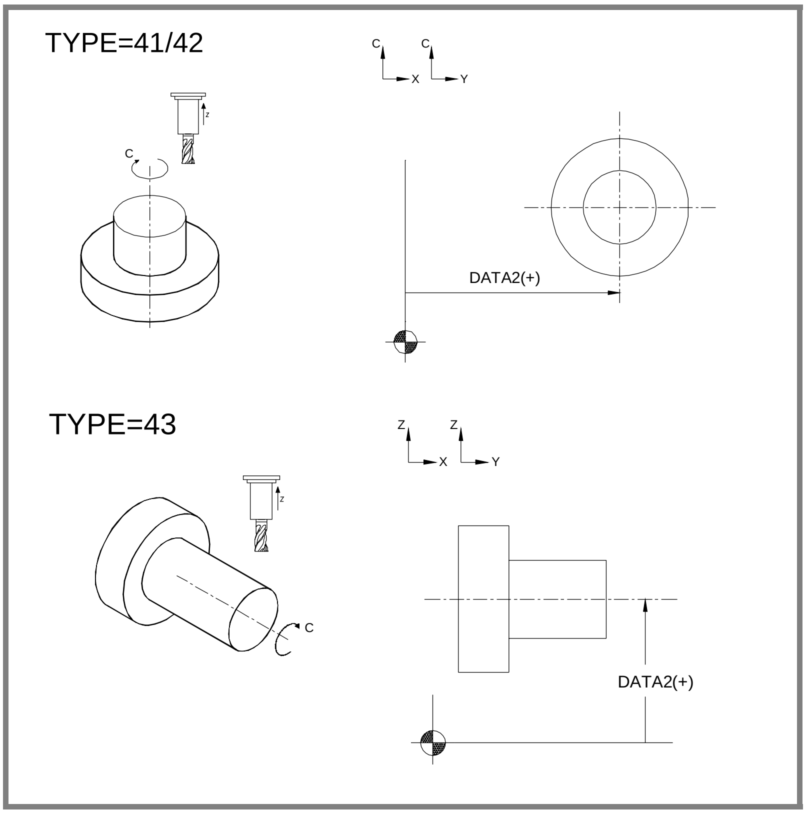


DATA2 **Rotary axis position**

For kinematics types 41 and 42, it indicates the distance from the rotary axis to the linear axis on which it develops.

For kinematics type 43, it indicates the position of the rotary axis along the tool axis.

They may be defined with a positive or negative value. The (+) sign in the illustrations indicates the direction it assumes as positive.



2.7 Machine parameters for the magazine

Tool magazine configuration

NTOOLMZ **Number of tool magazines**

Number of system's tool magazines.

Possible values:	from 1 to 4.
By default:	1

Although each channel has its own tool management, the tool magazines are not associated with any particular channel. They are not associated with any particular spindle either.

GROUND **Ground tools are permitted (manual load)**

They are the ones not located in the magazine. When programming them, the CNC requests them to be inserted in the spindle.

Possible values:	Yes / No.
By default:	No.

MAGAZINE **Tool magazine table**

It shows the table to enter the magazine data. There is a table per magazine.

Each table has the following machine parameters to configure it:

STORAGE MANAGEMENT

Storage data

STORAGE **Parameters related to storage**

Parameters `SIZE` and `RANDOM` must be set.

SIZE **Size of the magazine (number of pockets)**

Possible values:	from 0 to 1000.
By default:	20

RANDOM **Random magazine**

It indicates whether the tools must always occupy the same position (non random) or they may occupy any position (Random).

Possible values:	Yes / No.
By default:	No.

Tool magazine management

MANAGEMENT **Management related parameters**

It shows the parameters to configure the magazine management.

TYPE	CYCLIC	GROUND
OPTIMIZE	M6ALONE	

TYPE **Magazine type**

The CNC can manage different types of magazine. See *"2.7.1 Types of tool magazine"* on page 104.

Possible values:	Asynchronous. Synchronous. Turret. Synchronous + 2 arms Synchronous + 1 arm
By default:	Synchronous.

CYCLIC **Cyclic tool changer**

A "Cyclic tool changer" requires a tool change command (`M06`) after searching a tool and before searching the next one.

With a non-cyclic tool changer, it is possible to search for several tools in a row without necessarily having to make the actual tool change (M06 function).

Possible values:	Yes / No.
By default:	Yes.

OPTIMIZE **Tool management.**

When programming several `T`'s in a row without an `M6`, it indicates whether all the programmed tools are selected (`OPTIMIZE` = No) or just the ones involving a tool change (`OPTIMIZE` = Yes).

Optimizing only works when executing a program. In MDI mode, this parameter is ignored and all the blocks are executed.

`T2`	It is selected if Optimize = No
`T3 M6`	Is always selected. The M6 implies a tool change.
`T5`	Is always selected. M6 comes next.
`M6`	

M6ALONE **Action when executing an M06 without selecting a T**

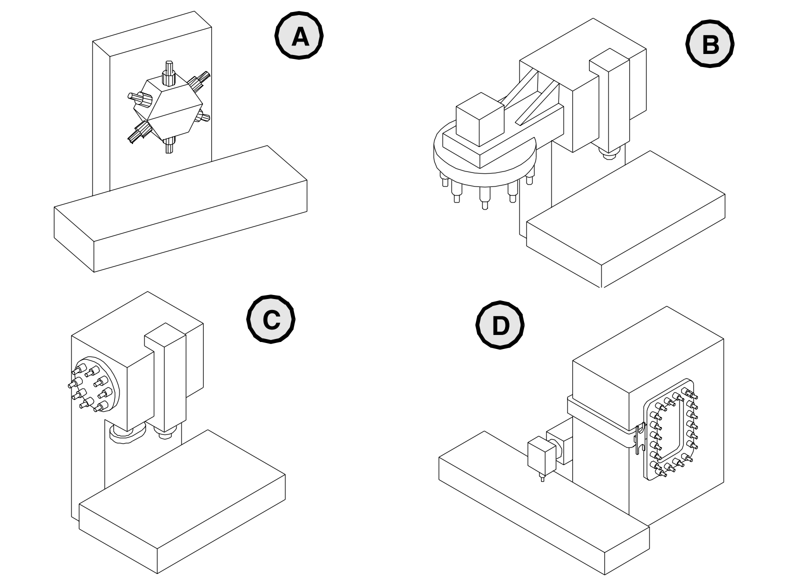
The `M06` function implies a tool change. This parameter indicates what happens when executing an `M06` without having selected a tool first.

Possible values:	Nothing. Show a warning. Show an error.
By default:	Show an error.

2.7.1 Types of tool magazine

The CNC can manage the following types of magazine:

- Turret.
- Asynchronous.
- Synchronous.
- Synchronous + 2 arms
- Synchronous + 1 arm



(A) Turret.
(B) Synchronous without changer arm.
(C) Synchronous with an arm with 1 or 2 holders.
(D) Asynchronous.

Turret type

It is a typical magazine for lathes. The tool cannot be changed while the part is being machined.

Synchronous magazine without changer arm

In a synchronous tool changer without arm, the magazine must move up to the spindle to change the tool. The tool cannot be changed while the part is being machined.

The tool change is carried out as follows:

1. It ends the movement of the axes.
2. The magazine approaches the spindle to grab the tool.
3. It selects the new tool and puts it in the spindle.
4. The magazine retracts.
5. The CNC resumes program execution.

Asynchronous magazines with changer arm (1 or 2 holders).

Synchronous magazines with tool changer arm (1 or 2 holders) have the magazine close to the spindle. The tool cannot be changed while machining the part because the arm would collide.

The tool change is carried out as follows: Example with 2 holders:

1. The new tool is selected in the magazine.
2. It ends the movement of the axes.
3. The arm takes a tool on each holder (from the magazine and from the spindle) and swaps them.
4. The arm retracts
5. The CNC resumes program execution.

Asynchronous magazines

An asynchronous magazine is located away from the spindle. Most of the movements may be carried out while machining the part, thus minimizing machining time.

The tool change is carried out as follows:

1. While machining, a new tool is selected at the magazine, the changer arm picks it up and brings it closer to the spindle.
2. It ends the movement of the axes.
3. The other holder picks up the tool that is in the spindle and makes the change.
4. The program resumes execution and the changer arm returns to the magazine to leave the tool.

2.8 Machine parameters for HMI (Interface)

These parameters are used to define the communication environment (interface) between the operator and the CNC.

To validate the changes made to these parameters, the CNC must be reset.

Customizing

WINDOW **Dimensions of the main window**

The following parameters must be set: `POSX`, `POSY`, `WIDTH` and `HEIGHT`.

POSX **X coordinate of the top left corner**
POSY **Y coordinate of the top left corner**
WIDTH **Width of the window**
HEIGHT **Height of the window**

They are defined in pixels. They are only used for the PC simulator version. They should not be changed at the CNC.

VMENU **Position of the vertical softkey-menu**

Depending on hardware, the vertical softkeys F8 through F12 appear on the left or on the right of the monitor.

Possible values:	Left / Right.
By default:	Right.

LANGUAGE **Work language**

Select a language among those available.

ENGLISH	SPANISH	ITALIAN
GERMAN	FRENCH	BASQUE
PORTUGUESE		

USERKEY **Customizing the user key**

To associate a function with the user key.

The `FUNCTION` parameter must be set. Depending on the option selected, either parameter `COMPONENT` or `APPLICATION` will have to be set.

FUNCTION

Function of the user key

Depending on the selected function

- Minimize the CNC and show the Windows screen.
- Access a component (work mode) without CNC hotkey.
- Execute an external application, for example the FGUIM.
- Disable the key.

Possible values:	Windows, Component, Application, Nothing.
By default:	Nothing.

COMPONENT

Access a component without a hotkey

Possible values:	Diagnosis mode. PLC. Machine parameters. DDSSETUP. TUNING Tool calibration.

APPLICATION

Execute a PC application

The whole application path must be indicated. For example:

C:\CNC8070 \FAGOR \ RELEASE \FGUIM.EXE

CHANGEKEY

Customizing the change key

To associate a function with the change key.

The `FUNCTION` parameter must be set.

FUNCTION

Function of the change key

It is possible to select between showing the next page of the active work mode, switching over to the next channel or showing the system menu.

If a menu is displayed, parameter `MENU` must be set with the options to be displayed on each menu of softkeys.

Possible values:	Next page / Next channel / Menu
By default:	Next page.

MENU

Set up the system menu

Parameter table to set up the softkey menu displayed when pressing the change key.

SYSMENUMODE

Behavior of the system menu

It determines when the system menu is disabled.

- If defined as "Volatile", it is disabled when selecting a menu option or when changing the active component.
- If it is defined as "Fixed", the softkey menu stays on until the change key is pressed again.

Possible values:	Volatile / Fixed.
By default:	Volatile.

SYSHMENU — Horizontal system-menu
SYSVMENU — Vertical system-menu

It sets the options that will appear on each softkey-menu.

- The menu will be disabled.
- The menu shows the various pages or screens of the active work mode.
- The menu shows the available channels.
- The menu shows the components or work modes of the CNC.

Possible values:	Disabled. Screens. Channels. Components.
By default:	Disabled.

ESCAPEKEY — Customizing the escape key

To associate a function with the escape key.

The `FUNCTION` parameter must be set. Depending on the option selected, parameter `NPREVIOUS` must be set.

FUNCTION — Function associated with the escape key

It is possible to choose between showing the previous softkey-menu, the previous work mode or both. If "both" is selected, every time this key is pressed, it will show the previous softkey menu until reaching the main menu. From then on, the work mode will change.

Possible values:	Pr. menu Pr component. Pr. Menu/Cmpnt
By default:	Pr. menu

NPREVIOUS — Maximum number of previous components stored.

Possible values:	1 to 5.
By default:	1.

SIMJOGPANEL — Simulated JOG panel

A simulated JOG panel is a window that lays over the CNC screen. It is used to simulate the JOG keys and the work modes accessing keys.

It may have to be used when working with Telediagnosis (remote control of the CNC).

This parameter indicates whether the simulated panel is available or not.

To select or deselect it, press [CTRL] + [J].

Possible values:	Yes / No.
By default:	No.

WINEXIT **Exit Windows when closing the CNC**

It indicates whether Windows is closed or not when exiting the CNC using [ALT] + [F4].

Possible values:	Yes / No.
By default:	No (Windows is not closed).

DIAGPSW **(Reserved)**

2.9 OEM machine parameters

Reading drive variables

DRIVEVAR **Drive variables table**

Configures the access to the drive variables from the CNC

It offers parameters SIZE and DATAto define it.

SIZE **Number of variables to consulted at the drive**

Possible values:	from 0 to 99.
By default:	0

DATA **List of drive variables**

The following parameters must be set for each variable.

MNEMONIC	AXIS	ID
TYPE	MODE	

MNEMONIC **Name of the variable at the drive**

Mnemonic to be used for the variable at the CNC. The access to the variable from the CNC will be as follows:

```
(V.)DRV.{mnemonic}.{axis}
(V.)DRV.{mnemonic}.{spindle}
```

AXIS **Axis or spindle that the variable belongs to**

The variable may be associated with a particular axis or spindle or may be good for all of them. When defined with the "*" character, it means all the axes and spindles.

Possible values:	Axes and spindles defined in AXISNAME. The "*" character means all the axes spindles.

ID **Variable identifier at the drive**

Sercos ID identifier that identifies the variable at the drive.

TYPE **Access type**

The access to the variable may be synchronous or asynchronous.

Variables of synchronous access are managed through the cyclic channel. Variables of asynchronous access are managed through the service channel, even if the default option at the drive is the opposite.

Not all the variables may be defined with synchronous access; only the ones allowed by the drive. Refer to the drive manual.

MODE **Access mode**

The access to the variable may be read-only or read-write.

Generic OEM-parameters

MTBPAR **OEM-parameter table**

They are generic parameters that the OEM can use as machine parameters.

It offers parameters `SIZE` and `DATA`to define it.

SIZE **Number of OEM parameters**

Possible values:	from 0 to 1000.
By default:	0

DATA **OEM parameters**

The access to these parameters using variables will be as follows.

```
(V.)MTB.P[n]
```

Cam editor

CAMTABLE **Table of electronic cams**

It offers parameters `SIZE` and `DATA`to define it.

SIZE **Number of electronic cams**

Possible values:	from 0 to 16.
By default:	0

DATA **Cam data**

It shows the available cams.

CAM1..16 **Electronic-cam editor**

Cam editor with friendly assistance to analyze the behavior of the cam projected through graphically assisted data entry for speed, acceleration and jerk.

This function has a specific manual.

Refer to the specific documentation to obtain further information regarding the requirements and operation of the electronic cam.

2.10 Alphabetical listing of machine parameters

CONCEPTS

3

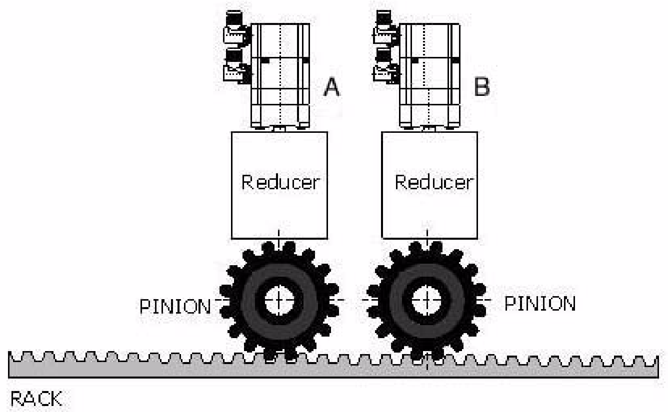
3.1 Tandem axis

A tandem axis consists of two electric motors mechanically coupled to each other forming a single transmission system (axis). It is normally used to move the axes on large machines.

The following aspects may be pointed out in this configuration:

- A tandem axis helps provide the necessary torque to move an axis when a single motor is not capable of supplying enough torque to do it.
- Applying a pre-load torque between the main motor and the slave motor reduces the backlash on the rack and pinion.
- The rigidity of the rack-and-pinion system is greater than that of long leadscrews.

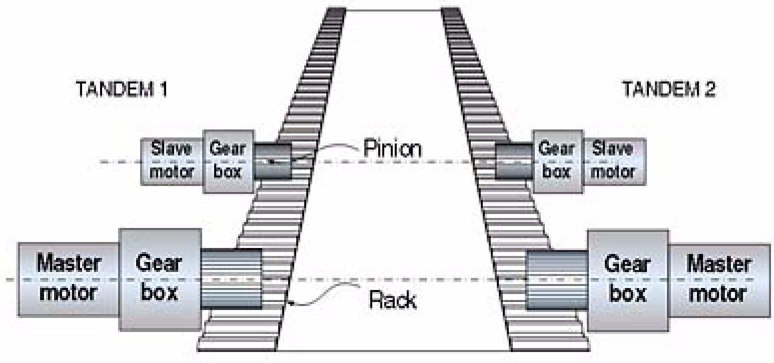
Example of a Tandem axis made up of a rack-and-pinion system driven by two motors.



(A) *Main or master motor. Besides generating torque, it is in charge of positioning.*

(B) *Slave motor. It only supplies torque.*

One of the many applications of controlling a tandem axis has to do with gantry machines.



Example diagram of a gantry machine with two tandem axes.

3.1.1 Tandem axis configuration. Machine parameters

TANDEM **Tandem axes**

There may be up to 8 pairs of Tandem axes. Every pair must meet the following requirements.

- Each master tandem axis admits one single slave tandem axis.
- The axes must be sercos in velocity.
- A preload may be applied between the two motors.
- Each motor may have a different rated torque.
- The turning direction of each motor may be different from the other's.
- The torque distribution between moth motors may be different from 1:1 ratio. For example, on motors whose rated torque is different.

Each pair has the following machine parameters:

TMASTERAXIS	TSLAVEAXIS	TORQDIST
PRELOAD	PRELFITI	TPROGAIN
TINTIME	TCOMPLIM	

TMASTERAXIS **Tandem. Master or main axis**

TSLAVEAXIS **Tandem. Slave axis**

The name of the axes 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.

Possible values:	*Any axis defined in the parameter "AXISNAME".*

TORQDIST **Torque distribution**

It sets the torque supplied by each motor to obtain the total necessary torque on the tandem axis.

This parameter refers to the master axis. It is defined as the percentage of the total torque required from the master axis. The difference between the value of this parameter and 100% is the percentage applied to the slave axis.

If the motors are identical and they're both supposed to output the same torque, this parameter should be set to 50%.

Possible values:	*From 0 to 100% (both included).*
By default:	*50%*

For example:

There are two motors, the master with 100 Nm and the slave with 20 Nm and they will be working at the same percentage of load.

TORQDIST = 20/100 = 20%

PRELOAD

Preload between both motors

It is the torque difference to be applied between the master axis and the slave axis. This sets a traction between them in order to eliminate the rack-and-pinion backlash when it is in rest position.

This parameter refers to the master axis. It is defined as the percentage of the rated torque to be applied as preload.

In order for the two axes to supply opposite torques, the preload value must be greater than the maximum torque needed at all times, including accelerations.

Possible values:	*from -100% to 100%.*
By default:	*0 (it disables the preload).*

This parameter may be modified from the oscilloscope and from the PLC.

Applying the preload necessarily implies mechanically joining the master and slave axes that make up the tandem axis. Otherwise, the motors will move even without the control velocity command.

PRELFITI

Filter time to apply the preload

It eliminates the torque steps at the input of the tandem compensator when setting a preload value. This avoids a step in the velocity commands of the master and slave axes of the tandem.

Setting it to zero disables the filter.

Possible values:	*from 0 to 65535 milliseconds.*
By default:	*1000ms*

This parameter may be modified from the oscilloscope.

TPROGAIN

Proportional gain (Kp) for the tandem axis

The proportional controller generates an output proportional to the torque error between the two motors.

$$k_p = \left(\frac{S_{max}}{T_{nom}}\right) \times TPROGAIN$$

$$T_{error} = (-T_{master} + T_{slave} + Preload)$$

$$Speed = k_p \bullet T_{error}$$

Possible values:	*from 0 to 100%.*
By default:	*0 (no proportional gain is applied).*

This parameter may be modified from the oscilloscope.

Example: A tandem axis has a maximum speed of 2000 rpm and a rated torque of 20 Nm. TPROGAIN has been set to 10%.

Kp = (2000 rpm / 20 Nm) · 0.1= 10 rpm/Nm.

TINTTIME

Integral gain (Kp) for the tandem axis

The integral controller generates an output proportional to the integral of the torque error between the two motors.

$$k_i = \frac{ControlTime}{IntegralTime} \times k_p$$

$$T_{error} = (-T_{master} + T_{slave} + Preload)$$

$$Speed = k_i \cdot \sum T_{error}$$

Possible values:	*from 0 to 65535 milliseconds.*
By default:	*0 (no integral gain is applied).*

This parameter may be modified from the oscilloscope.

TCOMPLIM

Compensation limit

This parameter limits the maximum compensation applied by the tandem axis. This limit is also applied to the integral.

This parameter refers to the master axis. It is defined as percentage of the maximum speed of the master motor. If programmed with a "0" value, the output of the tandem control will be zero, thus disabling the tandem.

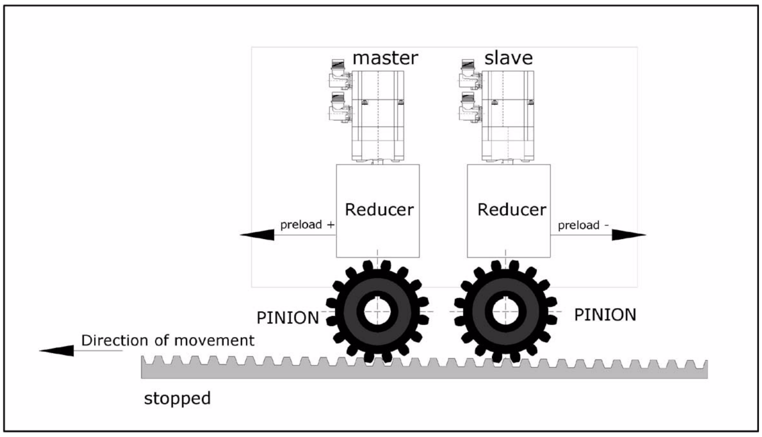
Possible values:	*from 0 to 100%.*
By default:	*0*

This parameter may be modified from the oscilloscope.

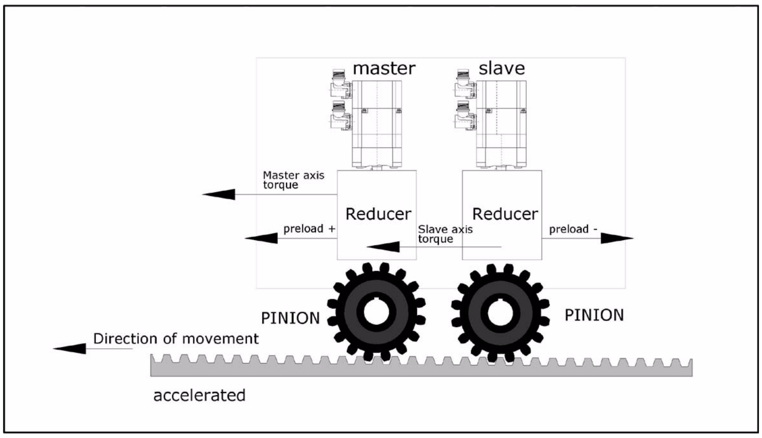
3.1.2 Effect of the preload

The following diagrams show the effect of preload in different situations.

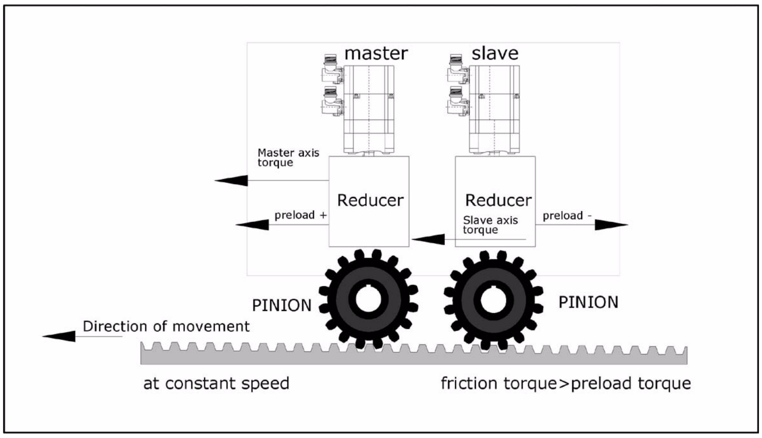
Preload at rest



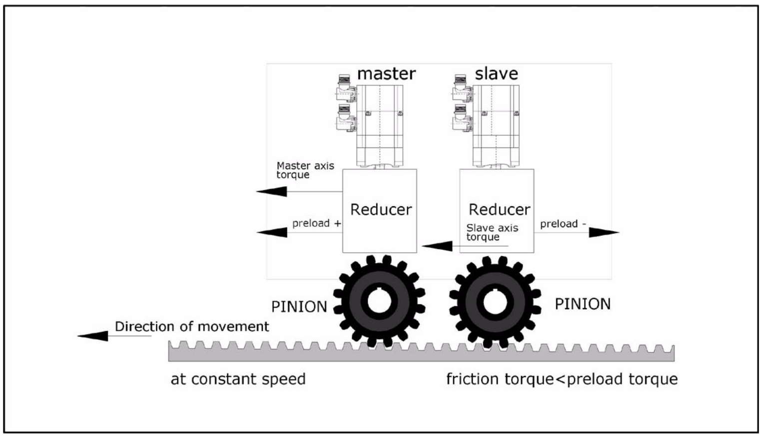
Preload with acceleration



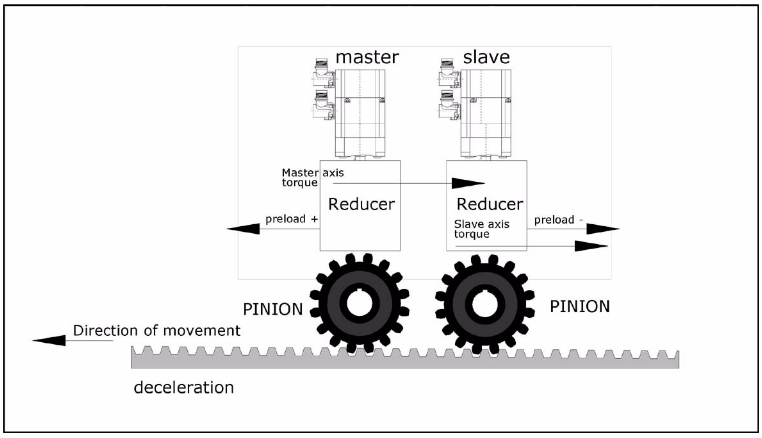
Preload at constant speed. Friction torque > Preload



Preload at constant speed. Friction torque < Preload

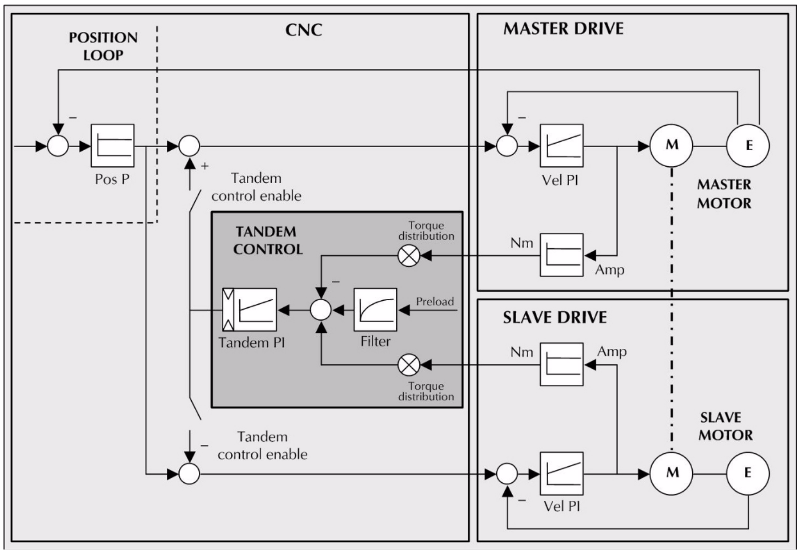


Preload with deceleration



3.1.3 Tandem axis configuration. Block diagram

The block diagram of the tandem control system shows the master tandem axis with its slave tandem axis. The block diagram for a gantry machine consists of two identical diagrams to the one shown in the figure.

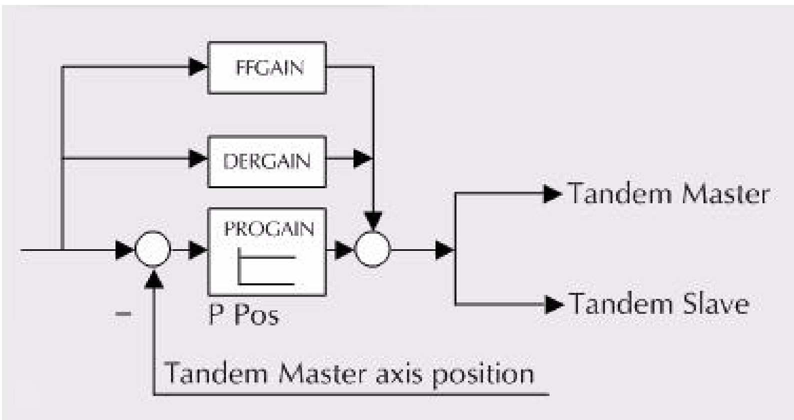


The block diagram has an area for the drive and another one for the CNC that comprises the position loop and the control of the tandem.

Position and velocity loop

The position loop is closed only with the position of the master axis of the tandem. The velocity command of the master tandem axis is also sent to the slave tandem axis closing the velocity loop.

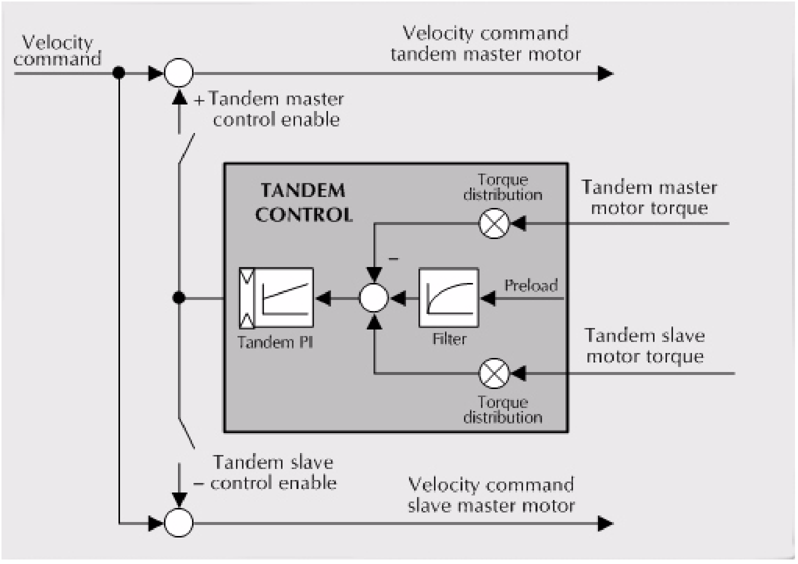
The tandem control changes the velocity command of the master axis and that of the slave axis according to the torque distribution and the selected preload.



The feed-forward and AC-forward values of the master axis are applied to the slave axis; consequently, they must have the same gear ratios.

Tandem axis control.

The block diagram showing the application of the tandem axis control is the following.



The meaning of the nomenclature being used is:

Torque of the master motor of the tandem

It is the percentage of the rated torque shown by the sercos variable TV2 of the drive that governs the master axis of the tandem. It is read through the sercos fast channel in each loop.

Torque of the slave motor of the tandem

It is the percentage of the rated torque shown by the sercos variable TV2 of the drive that governs the slave axis of the tandem. It is read through the sercos fast channel in each loop.

Torque distribution

Normalizing gain of the torque generated by the motors in order to distribute the torque in a ratio other than 1:1.

Preload

Previous torque applied to both tandem axes in opposite direction. This sets a traction between them in order to eliminate the rack-and-pinion backlash when it is in rest position. It is determined as the torque difference supplied by each axis.

Applying the preload necessarily implies mechanically joining the master and slave axes that make up the tandem axis. Otherwise, the motors will move even without the control velocity command.

Preload filter

First-order filter to prevent torque steps from coming in when configuring the preload.

PI of the tandem

PI for making each motor provide its corresponding torque. It increases its velocity command if the torque being supplied is too low and it decreases it if the torque being supplied is too high.

When defining a tandem axis, at each loop, the CNC reads via Sercos the torque supplied by the master and the slave axes. This may result in a smaller number of read and write variables available in the Sercos fast channel.

3.1.4 Tandem related variables

Related to general machine parameters

These variables are read-only synchronous and are evaluated during block preparation.

They have generic names. Replace the letter "i" with numbers, keeping the brackets.

		PRG	PLC	INT
(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

Tandem related

These variables are read/write (R/W) synchronous and are evaluated during execution. They correspond to linear and rotary axes and to the spindle.

The mnemonics of the variables have generic names.

- Replace "Xn" with the name or logic number of the axis.
- Replace the "n" character with the channel number, maintaining the brackets.

		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

(V.)A.TPIOUT.[Xn] (V.)A.TPIIN.[Xn]

Both variables return the values in rpm. The axis must be a valid tandem master, otherwise, it returns a value of zero.

(V.)A.TORQUE

The PLC reading of this variable comes in tenths (x10).

3.1.5 Tandem adjustment procedure

This procedure must bear in mind the type of machine. In general, the resonance frequency of a tandem machine is low; therefore, the CNC must generate position commands without frequency components higher than the resonance frequency.

The process should be initiated with low jerk values (lower than 10 m/sg3) and low Kv. They can always be increased in a later readjustment.

Steps for the adjustment

1. Move both axes independently.

The first step is to ensure the perfect operation of both the master and the slave axes separately. Also verify that both axes move in the same direction with similar dynamics.

2. Move one of the axes at a slow and constant speed.

2.1. Do not make jerky movements because the second motor is being dragged by the first one. In this situation, any acceleration or deceleration forces it to go from one side of the backlash to the other thus making it jerky.

2.2. Verify that the turning direction of both motors are coherent once the movement has been carried out.

Observe that reversing the turning direction of a motor reverses the direction of the torque and, consequently, it will be necessary to change the direction of its values monitored with drive parameters SP43 and TP85.

2.3. Verify that the gear ratio in both motors is the same (same feed for same turning speed).

2.4. Carry out a basic adjustment of the velocity loop so the machine can move. It will be readjusted later with both motors together.

2.5. Do not set the friction parameters (there is already enough torque to move the machine).

3. Repeat the procedure with the second axis.

When adjusting the loops, use the same parameters if the motors are identical and the torque distribution is 50%. If the motors are different, the axes must be adjusted so their dynamic response is the same or very similar. When using AC-forward ("ACFGAIN" = YES), remember that each motor has half the inertia for a 50% torque distribution.

4. Enable the tandem with both motors.

4.1. Disable the PI of the tandem.

4.2. Apply power and verify that the system is at rest.

4.3. Enter a low proportional value and eliminate the integral value of the PI of the tandem.

4.4. Without preload, verify that the machine moves and that each motor supplies its corresponding torque according to the parameter "TORQDIST" (e.g. half the torque for 50% distribution).

4.5. Monitor the torque of each motor (Sercos variable TV2). While stopped, increase the preload gradually until the motors supply torque in opposite directions.

4.6. Move slowly in both directions and verify that it works properly. Make sure that it is not jerky and that each motor supplies its corresponding torque according to parameters "TORQDIST" and "PRELOAD".

4.7. Readjust the velocity loop in both motors with the method used normally.

While changing the parameters of the velocity loop, the best thing would be to change them on both drives at the same time. But, since this is not possible, it is recommended to make small changes in the values or make them while the motor is stopped.

INTRODUCTION TO THE PLC

4

The PLC program (*.plc or *.c) may be either edited at the front panel or copied from a peripheral or PC.

It has a modular structure and may combine files in "C" language with files in mnemonic language.

Before executing it, its executable file must be generated.

On power-up, the CNC executes the executable PLC program stored in memory. If there is no such file, it will display the corresponding error message.

Data exchange takes place in automatic mode between the CNC and the PLC.

The following is possible from the PLC:

- Control physical inputs and outputs (remote modules).
- Consult and/or modify CNC-PLC exchange variables.
- Consult and/or modify the internal CNC variables.
- Display messages or errors at the CNC.

The following is possible from the CNC:

- Transfer auxiliary functions `M`, `H` and `S`.
- Access PLC resources from any part-program.

Abbreviations used in this chapter

(=0)	Low logic level
(=1)	High logic level
(g.m.p.)	General machine parameter

4.1 PLC program

The PLC program may combine a file in mnemonic language (extension: "plc") and another one in "C" language (extension: "c").

Both files must be located in the following directory:

C:\CNC8070 \MTB \PLC \PROJECT

It is recommended to use the mnemonic language file as the main PLC program, and the "C" language file for the auxiliary tasks (for example, temperature compensation).

Subroutines in the PLC program

The subroutines of the program in mnemonic language must be defined outside of the modules, for example at the end of the program after the END instruction.

The subroutines of the "C" language file must be defined as external at the beginning of the mnemonic language program (extension: "plc"). The name of the subroutine must be written in capital letters in both files.

"C" language programming offers a math library (trigonometric, logarithmic, etc.) that allows performing operations with tables, arrays, float type variables, etc.

PLC program with mnemonic language file.

Mnemonic.plc

```
PRG
()= MOV 1234 R201 = MOV 2345 R202
()= CAL SUMA
...
END

SUB SUMA
()= ADS R201 R202 R203
END
```

PLC program with a mnemonic language file and a "C" language file.

Mnemonic.plc

```
EXTERN SUMA
PRG
()= MOV 1234 R201 = MOV 2345 R202
()= CAL SUMA
...
END
```

Languagec.c

```
#include "plclib.h"
void SUMA (void)
(
R203=R201+R202
)
```

4.2 Modular structure of the PLC program

It may comprise the following modules:

- Main module (PRG).
- Periodic module (PE)
- First Cycle module (CY1).

Every module must begin with its defining instruction (PRG, PE, CY1) and end with the END instruction.

PRG **Main module**

It is executed cyclically and it is in charge of analyzing and modifying the CNC inputs, outputs and variables.

CY1 **First Cycle module**

It is optional. It will only be executed when starting the PLC up. It is used to initialized the various resources and variables before executing the main program.

PEt **Periodic execution module**

It is optional. It is executed every t milliseconds. The value of "t" is indicated by the instruction: PEt (between 1 and 2147483647 ms).

```
PE 1000
...
END
```

Defines the Periodic Module that will be executed every second (1000 ms).

It could be used to execute tasks that do not need to be executed at every PLC cycle. For example, a task to be performed every 30 seconds could be defined in a periodic module using the instruction PE 30000.

4.2.1 PLC module execution

CY1 **First Cycle module**

The First Cycle module (CY1) is only executed once when starting up the PLC program.

PRG **Main module**

The Main module (PRG) will be executed continuously as often as indicated by the (g.m.p.) PRGFREQ which sets after how many CNC cycles is the Main module executed.

For example, with sampling period of 4 ms and PRGFREQ=2, the PRG module will be executed every 4x2=8 ms.

The PRG execution takes about 100 µs.

PEt **Periodic execution module**

The Periodic Module is executed with the frequency indicated by the instruction PE t.

The periodic module PE 1000 will be executed every second (1000 ms)

4.3 PLC program execution

Main module (PRG).

The main module is processed as follows:

1. Assigns the current value of the physical inputs (remote modules) to the I resources of the PLC.
2. Assumes the current values of the internal CNC variables (CNCREADY, START, FHOUT, ...).
3. Executes the main program (PRG).
4. Updates the internal CNC variables (EMERGEN, STOP, FEEDHOL, ...) with the current values of the associated PLC resources.
5. It assigns the current value of the PLC's O resources to the physical outputs (remote modules).
6. The cycle ends and is ready for the next scan.

Periodic module (PE)

The periodic module is processed as follows:

1. It assumes the current values of the physical inputs (remote modules) at the beginning of the module.
2. Executes the periodic module.
3. It assigns the current value of the PLC's O resources to the physical outputs (remote modules).
4. The Periodic Module ends.

4.4 PLC resources

The PLC has the following resources.

- Inputs (I1-I1024) and outputs (O1-O1024)
- Marks (M1-M8192)
- Messages (MSG1-MSG256)
- Errors (ERR1-ERR256)
- Clocks (CLK)
- Registers (R1-R1024)
- Timers (T1-T256)
- Counters (C1-C256)
- Registers and marks for CNC-PLC communication

The MSG, ERR, CLK and T and T resources are initialized (=0) when starting up the PLC. M, C and R resources maintain their value between CNC start-ups.

Inputs (I1-I1024) and outputs (O1-O1024)

The inputs are elements that provide information to the PLC on the signals they receive from the outside world. They are represented by the letter I followed by an input number between I1 and I1024.

The outputs are elements that let the PLC activate or deactivate the various devices of the electrical cabinet. They are represented by the letter O followed by an output number between O1 and O1024.

Numbering of the physical inputs and outputs

There are two different ways to number the inputs and outputs. Depending on the order of the remote modules or via machine parameters. See *"4.4.1 Numbering of the physical inputs and outputs"* on page 139.

Marks (M1-M8192)

They are elements capable of memorizing in a bit (like an internal relay) the value set by the user. If the mark is (=0), it will be referred to as being set low. If the mark is (=1), it will be referred to as being set high.

They are represented by the letter M followed by a mark number between M1 and M8192.

Messages (MSG1-MSG256)

When activated (=1), they display a message on the CNC screen. The texts associated with the messages must be previously defined in the message and error table of the PLC.

They are represented by the letters MSG followed by a message number between MSG1 and MSG256.

All of them are initialized (=0) when starting up the PLC.

Errors (ERR1-ERR256)

When activating them (=1) they cause an error. An error interrupts the execution of the CNC and it shows an error message on the screen. The texts associated with the errors must be previously defined in the message and error table of the PLC.

They are represented by the letters ERR followed by an error number between ERR1 and ERR256.

All of them are initialized (=0) when starting up the PLC.

The errors do not activate the emergency of the CNC, (_ALARM) signal.

Clocks (CLK)

They are internal clocks with different time periods to be used in the PLC program.

They are represented by the word CLK followed by a clock number .

All of them are initialized (=0) when starting up the PLC.

These are the clock marks available. Their half-period (or after how long their state 0/1 changes) is shown next to them.

CLK1	1ms	CLK100	100ms	CLK1000	1s
CLK2	2ms	CLK200	200ms	CLK2000	2s
CLK4	4ms	CLK400	400ms	CLK4000	4s
CLK8	8ms	CLK800	800ms	CLK8000	8s
CLK16	16ms	CLK1600	1.6s	CLK16000	16s
CLK32	32ms	CLK3200	3.2s	CLK32000	32s
CLK64	64ms	CLK6400	6.4s	CLK64000	64s
CLK128	128ms	CLK12800	12.8s	CLK128000	128s

Registers (R1-R1024)

They are elements that can store a numeric variable in 32 bits. The value stored in each register is considered as a signed integer between ± 2.147.483.647. It can be processed as decimal or hexadecimal number (preceded by the "$" sign). For example:

156 (Decimal)

$9C (Hexadecimal)

They are represented by the letter R followed by a register number between R1 and R1024.

It is also possible to refer to a register bit with the letter B and a bit number (0/31). The PLC takes bit 0 as the least significant bit and as bit 31 as the most significant bit.

B7R155 refers to Bit 7 of Register 155.

Timers (T1-T256)

They are elements capable of maintaining their output at the same logic level (state) for a preset time period (time constant) after which their output changes states.

They are represented by the letter T followed by a time number between T1 and T256.

All timers are initialized (=0) when starting up the PLC.

See "4.5 Operation of a timer" on page 141.

Counters (C1-C256)

They are elements capable of counting up or down a preset amount of events.

They are represented by the letter C followed by a counter number between C1 and C256.

See "4.6 Operation of a counter" on page 152.

Registers and marks for CNC-PLC communication

The PLC has access to a some internal CNC data.

The PLC can consult and/or modify certain CNC signals (marks and registers).

- Consultation signals: CNCREADY, START, FHOUT, ...
- Modifiable signals: _EMERGEN, _STOP, _FEEDHOL, ...

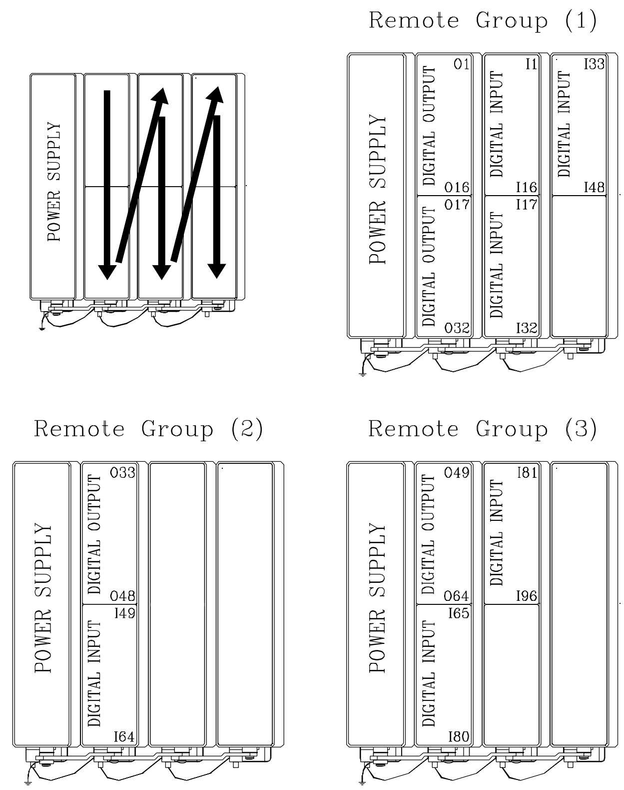
See chapter "7 Logic CNC inputs and outputs".

4.4.1 Numbering of the physical inputs and outputs

The numbering of the I/O modules may be set using the machine parameters. If these parameters are not defined, the CNC numbers the modules automatically according to the order of the remote modules.

Numbering according to the order of the remote groups

They are numbered following the order of the remote groups (rotary switch of the Power Supply element). Within each group, they are ordered from top to bottom and from left to right.

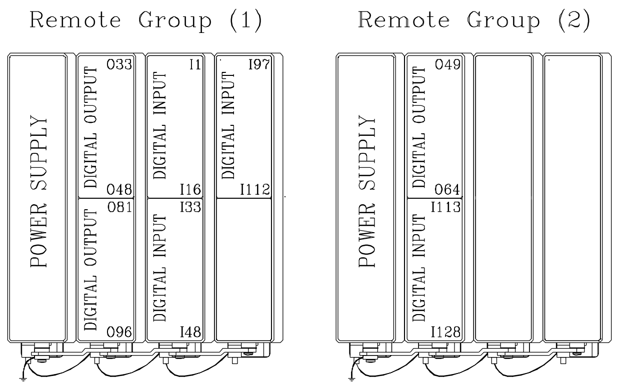


Group 1	Group 2	Group 3
O1...O32	O33...O48	O49...O64
I1...I48	I49...I64	I65...I96

Numbering by machine parameters

When the numbering is set by machine parameters, each module is assigned a base index and the inputs or outputs of that module are numbered after it. The values of the base index must be multiple of 16, plus 1 (i.e. 1, 17, 33, etc.). The base indexes may follow any order and they do not have to be sequential.

When inserting a new module, the first modules will be assigned the numbering of the table and the last one will be assigned the next valid base index after the highest one assigned until then.



	Digital inputs		Digital outputs	
	Index	Inputs	Index	Outputs
module 1	1	I1...I16	33	O33...O48
module 2	33	I33...I48	81	O81...O96
module 3	97	I97...I112	49	O49...O64
module 4	113	I113...I128		

4.5 Operation of a timer

All the timers have a status output T and the inputs: TEN, TRS, TG1, TG2, TG3 and TG4. It is also possible to check, at any time, the elapsed time t since the timer was triggered.

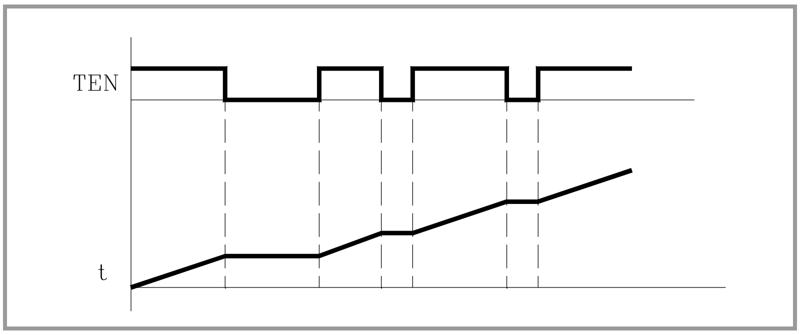
When starting up the PLC, all the timers are initialized by setting their status T and their time count to "0".

(TEN) Enable input

It can be used to interrupt and resume the timing.

It is referred to by the words TEN followed by the timer number. For example TEN 1, TEN 25, TEN 102, etc.

Once the timer is triggered, if input TEN is set low (=0), the PLC stops timing; input TEN must be set back high (=1) to resume timing.



By default, every time a timer is triggered, the PLC sets this input high (=1).

```
I2 = TEN 10
```

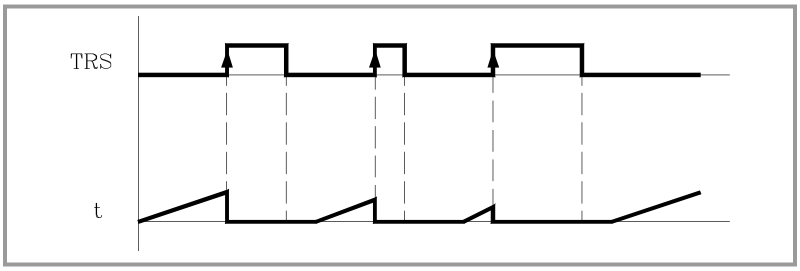
Input I2 controls the Enable input of timer T10.

(TRS) Reset input

It is used to initialize the timer by setting its status T and its timing to "0".

It is referred to by the words TRS followed by the timer number. For example `TRS 1`, `TRS 25`, `TRS 102`, etc.

Once the timer has been activated, at an up-flank (0 to 1 transition) of the TRS input, the PLC resets the timer. The timer is deactivated and its trigger input must be activated to turn the timer back on.



By default and every time a timer is triggered, the PLC sets this input high (=0).

```
I3 = TRS 10
```

Input I3 controls the Reset input of timer T10.

(TG1, TG2, TG3, TG4) Trigger inputs

They are used to activate the different work modes of the timer.

TG1	triggers the mono-stable mode.
TG2	triggers the delayed activation mode.
TG3	triggers the delayed deactivation mode.
TG4	triggers the signal limiting signal.

They are referred to by the words `TG1`, `TG2`, `TG3`, `TG4` followed by a timer number and the initial timing value (Time constant). For example `TG1 1 100`, `TG2 25 224`, `TG3 102 0`, etc.

Set the time constant

The time constant is defined by a numeric value or by the internal value of a register `R`. Its value must be between 0 and 4294967295 ms, equivalent to 1193 hours (almost 50 days).

```
TG1 20 100
```

Triggers timer T20 in Mono-stable mode (TG1) with a time constant of 100 ms.

```
TG2 22 R200
```

Triggers timer T22 in delayed activation mode (TG2) with the time constant stored in register R200 in ms.

Activating the timer

The timer is activated according to the selected input number at an up-flank (0 to 1 transition) or at a down-flank (1 to 0 transition).

Later on, this same section shows how to operate in each of these modes.

(T) Status output

It indicates the logic state of the timer.

It is referred to by the letter T followed by the timer number. For example: T1, T25, T102, etc.

As the timer's logic state depends on the selected work mode (TG1, TG2, TG3 and TG4) it will be explained later on.

(T) Elapsed time

It indicates the time elapsed at the timer since it was triggered.

It is referred to by the letter T followed by the timer number. It is represented by T123 which is the same as the status output, but is used in different types of instructions.

In binary instructions, it refers to the logic state of the timer.

```
T123 = M100
```

Assigns the state (1/0) of T123 to M100.

In arithmetic and comparison instructions, it refers to the elapsed time.

```
I2 = MOV T123 R200
```

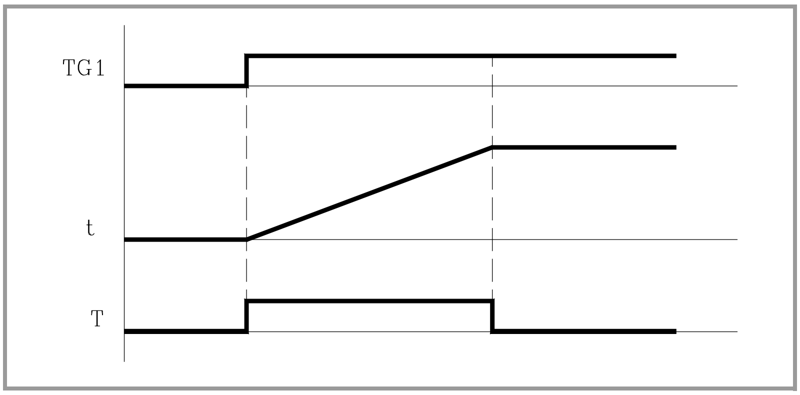
Transfers the time elapsed at T123 to register R200.

```
CPS T123 GT 1000 = M100
```

Compares the time elapsed at T123 is greater than 1000. If so, it activates mark M100.

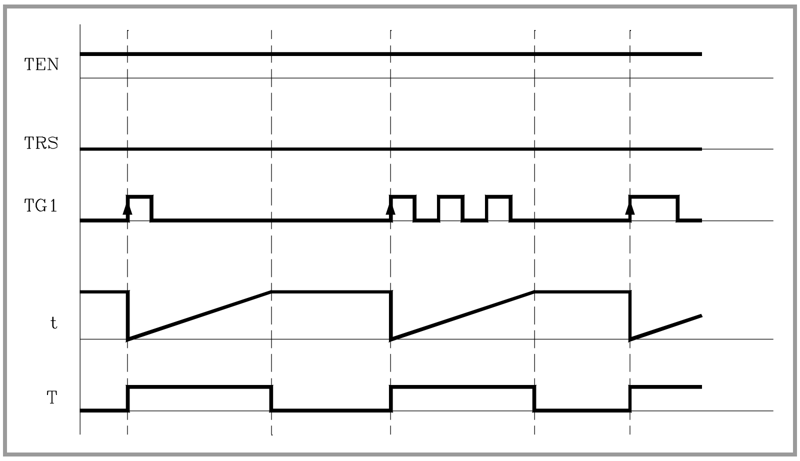
4.5.1 Monostable mode. TG1 input

In this operating mode, the status of the timer is maintained high (T=1) from when the TG1 input is activated until the indicated time period (constant) has elapsed.



With TEN=1 and TRS=0, the timer is activated with an up-flank at trigger input TG1. At that moment, the timer status output (T) changes states (T=1) and the timing t starts from "0".

Once the time period indicated by the time constant has elapsed, the timing is over. The status output (T) changes (T=0) and the elapsed time (t) is maintained.

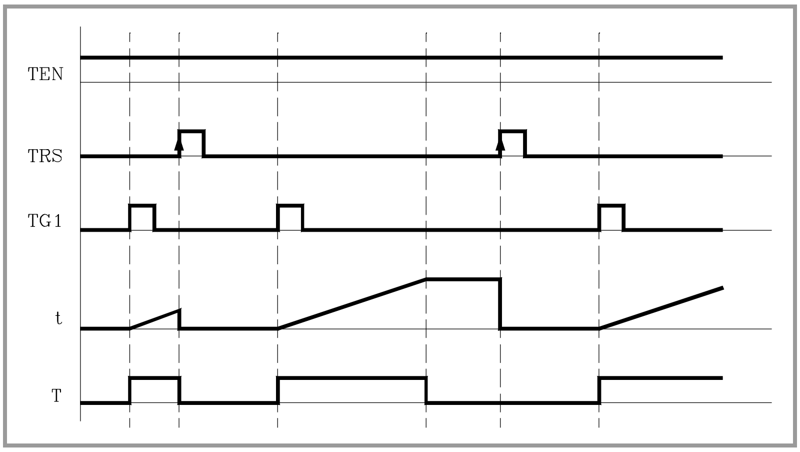


Any changes at the TG1 input (up or down-flank) while timing, has no effect.

Once the timing is over, an up-flank at trigger input TG1 is required to reactivate the timer.

Operation of the TRS input in this mode

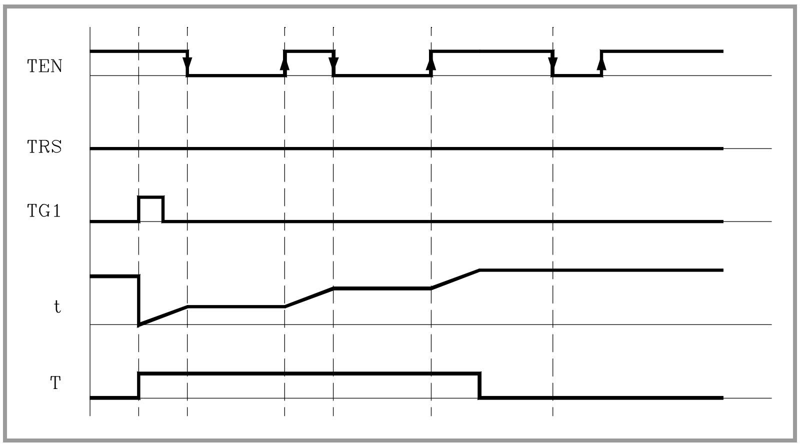
If an up-flank occurs at the TRS input while timing or after it, the PLC resets the timer setting its status output low (T=0) and resetting its timing (t=0).



Since the timer is reset, its trigger input must be activated again to turn it back on.

Operation of the TEN input in this mode

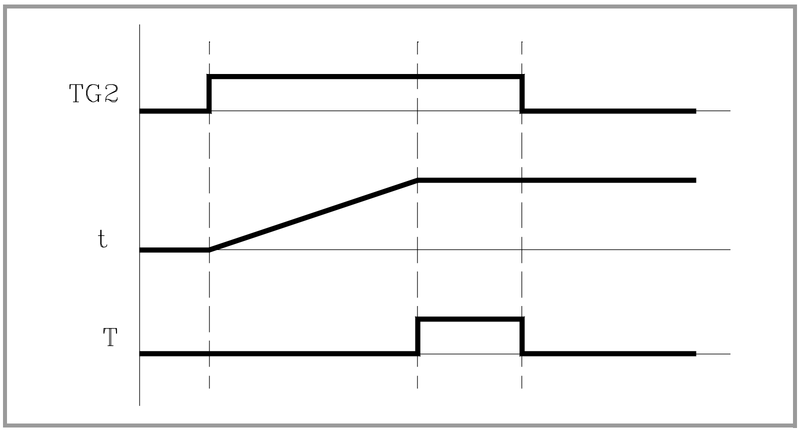
If once the timer has been activated, TEN = 0, the PLC interrupts the timing and TEN must be set to "1" to resume timing.



4.5.2 Delayed activation mode. TG2 input

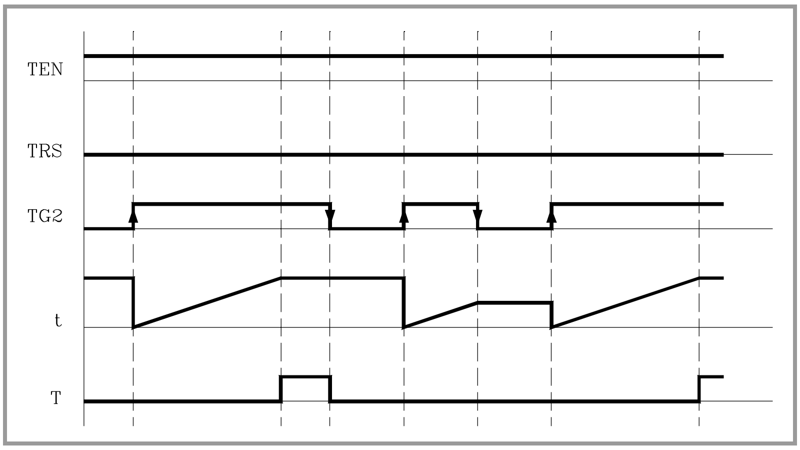
This mode applies a delay between the activation of the trigger input TG2 and that of the timer status output "T".

The time delay is set by the time constant.



With TEN=1 and TRS=0, the timer is activated with an up-flank at trigger input TG2. At that instant, the timing t begins from "0".

Once the time indicated with the time constant has elapsed, the timing is over, it activates the timer status output (T=1) which remains high that until a down-flank at trigger input TG2 occurs.

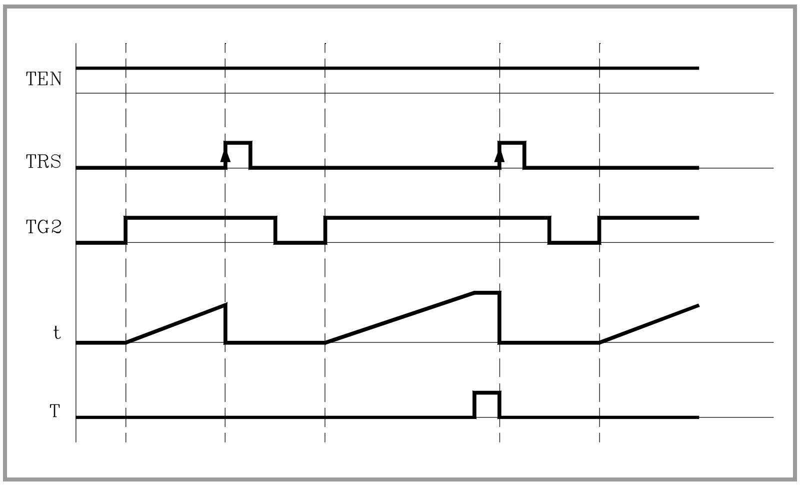


The elapsed time (t) is maintained until a new up-flank occurs at trigger input TG2.

If the down-flank at input TG2 occurs before the indicated time has elapsed, the PLC stops timing and it keeps the t value it has at the time.

Operation of the TRS input in this mode

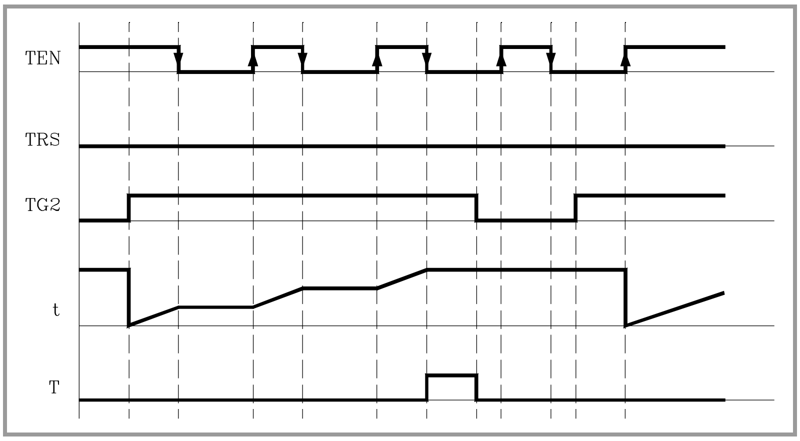
If an up-flank occurs at the TRS input while timing or after it, the PLC resets the timer setting its status output low (T=0) and resetting its timing (t=0).



Since the timer is reset, its trigger input must be activated again to turn it back on.

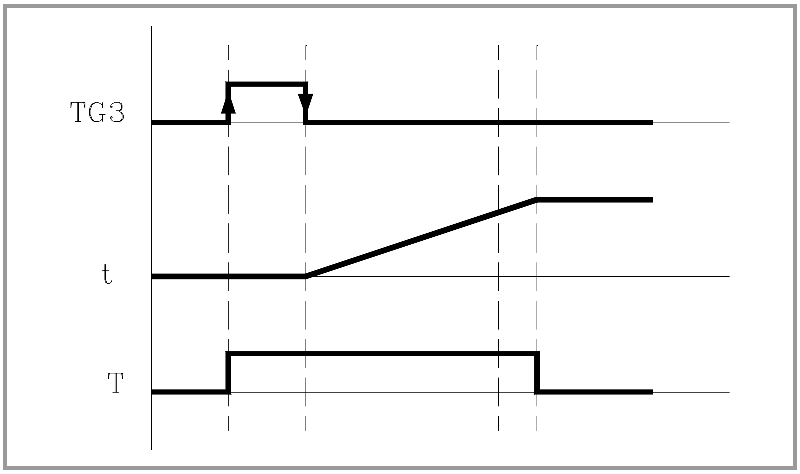
Operation of the TEN input in this mode

If once the timer has been activated, TEN = 0, the PLC interrupts the timing and TEN must be set to "1" to resume timing.



4.5.3 Delayed deactivation mode. TG3 input

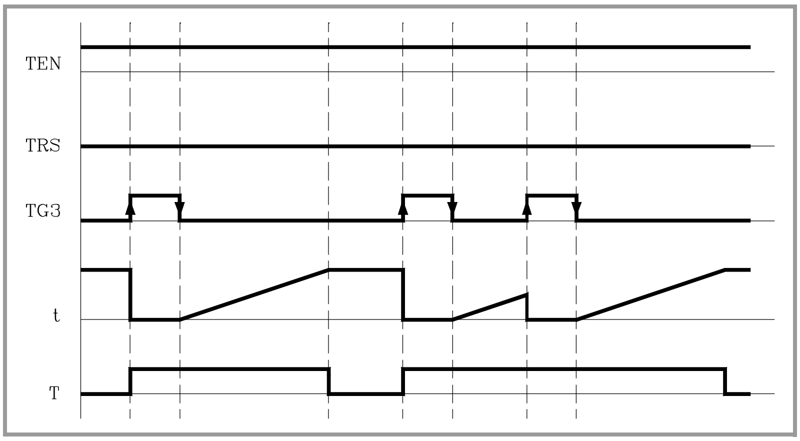
This operating mode is used to apply a delay between the deactivation of trigger input TG3 and that of the "T" output of the timer. The time delay is set by the time constant.



With TEN=1 and TRS=0, the timer is activated with an up-flank at trigger input TG3. At that instant, the status output of the timer goes high (T=1).

The timer waits for a down-flank at input TG3 to start the t timing from "0".

Once the time indicated by the time constant has elapsed, the timing stops and the status output of the timer goes low (T=0).

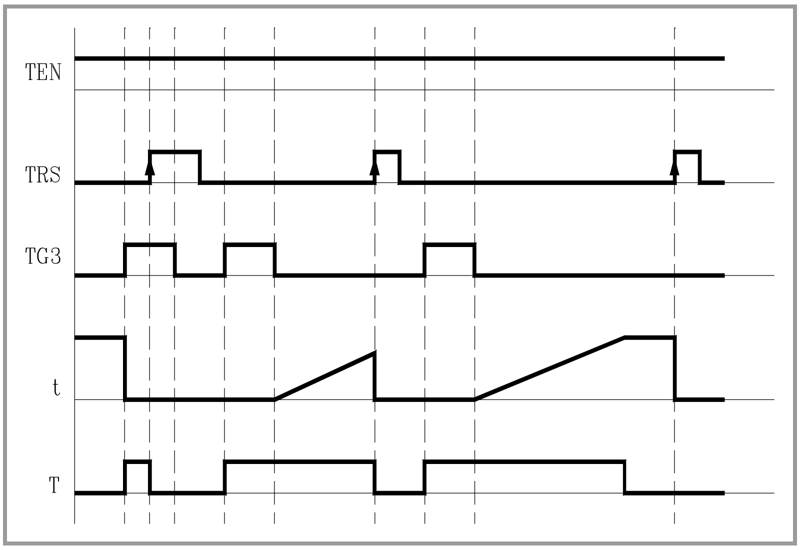


The elapsed time (t) is maintained until a new up-flank occurs at trigger input TG3.

If the up-flank at input TG3 takes place before the indicated time has elapsed, the PLC takes it as a new trigger and sets its status output high (T=1) and starts timing again from "0".

Operation of the TRS input in this mode

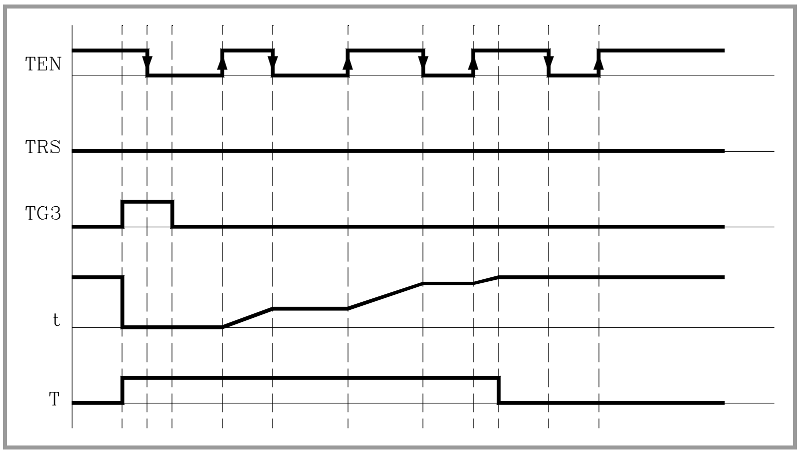
If an up-flank occurs at the TRS input while timing or after it, the PLC resets the timer setting its status output low (T=0) and resetting its timing (t=0).



Since the timer is reset, its trigger input must be activated again to turn it back on.

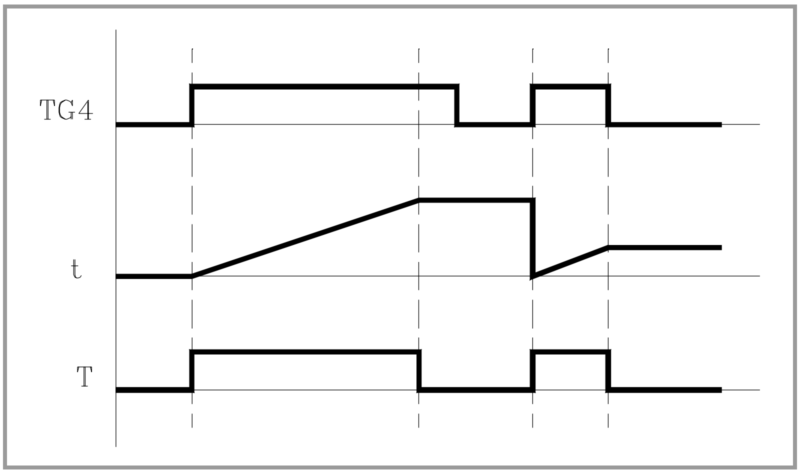
Operation of the TEN input in this mode

If once the timer has been activated, TEN = 0, the PLC interrupts the timing and TEN must be set to "1" to resume timing.



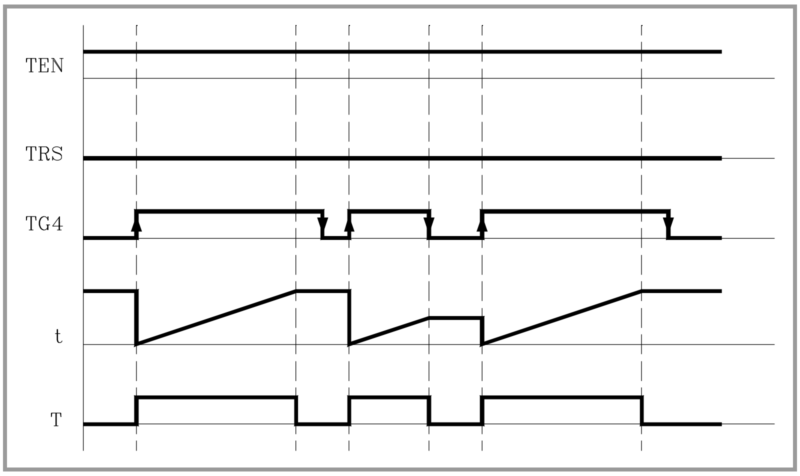
4.5.4 Signal limiting mode. TG4 Input

In this operating mode, the timer status is kept high (T=1) from the moment the TG4 input is activated until the time indicated by the time constant has elapsed or a down-flank occurs at input TG4.



With TEN=1 and TRS=0, the timer is activated with an up-flank at trigger input TG4. At that moment, the timer status output (T) changes states (T=1) and the timing t starts from "0".

Once the time indicated by the time constant has elapsed, the timing stops and the status output of the timer goes low (T=0).



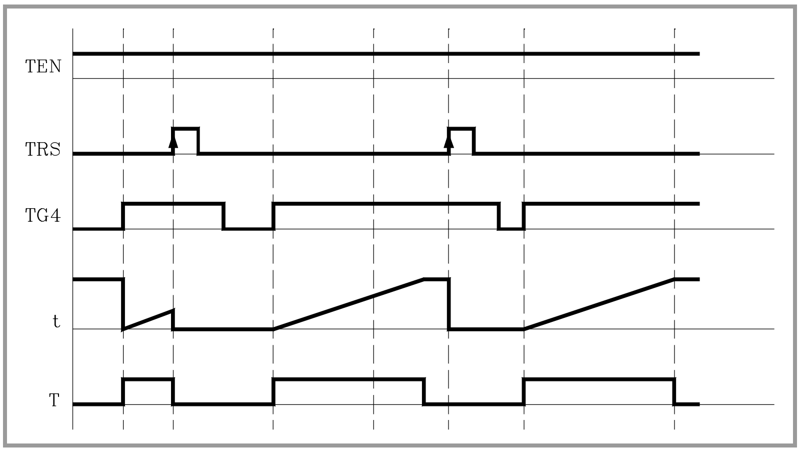
The elapsed time (t) is maintained until a new up-flank occurs at trigger input TG4.

If a down-flank occurs at trigger input TG4 before the time indicated by the time constant has elapsed, the PLC stops the timing, brings the status output low (T=0) and it keeps the current timing value (t).

To trigger the timer again, a new up-flank is required at input TG4.

Operation of the TRS input in this mode

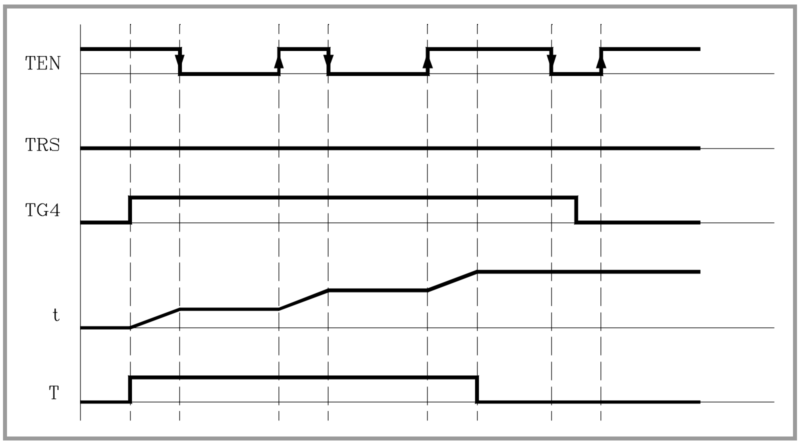
If an up-flank occurs at the TRS input while timing or after it, the PLC resets the timer setting its status output low (T=0) and resetting its timing (t=0).



Since the timer is reset, its trigger input must be activated again to turn it back on.

Operation of the TEN input in this mode

If once the timer has been activated, TEN = 0, the PLC interrupts the timing and TEN must be set to "1" to resume timing.



4.6 Operation of a counter

All the counters have a status output C and the inputs: CUP, CDW, CEN and CPR. Its internal count can also be checked at any time.

The counter's count is stored in a 32-bit variable. Consequently, its value will be in the ±2147483647 range.

(CUP) Count-up input

Every time an up-flank occurs at this input, the internal count of the counter increases one unit.

It is referred to by the letters CUP followed by the counter number, for example: `CUP 1`, `CUP 25`, `CUP 102`, etc.

```
I2 = CUP 10
```

Every time an up-flank occurs at I2, the count of the C10 counter increases one unit.

(CDW) Countdown input

Every time an up-flank occurs at this input, the internal count of the counter decreases one unit.

It is referred to by the letters CDW followed by the counter number, for example: `CDW1`, `CDW25`, `CDW102`, etc.

```
I3 = CDW 20
```

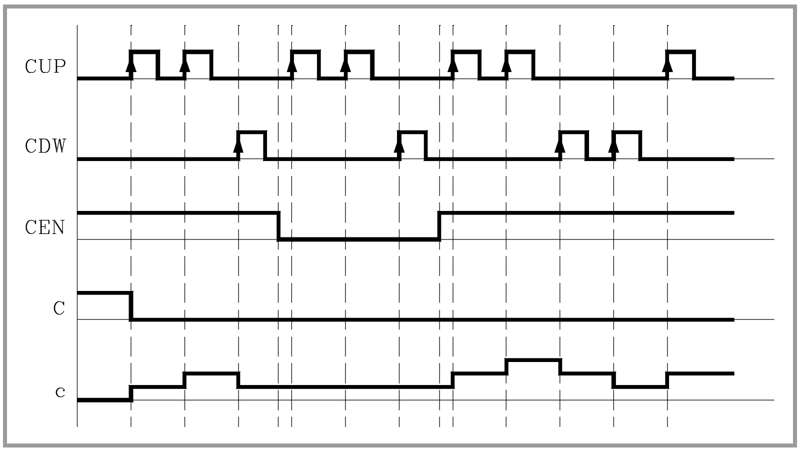
Every time an up-flank occurs at I3, the count of the C10 counter decreases one unit.

(CEN) Enable input

It enables the internal count of the counter.

It is referred to by the letters CEN followed by the counter number, for example: CEN 1, CEN 25, CEN 102, etc.

To change the internal count (CUP and CDW), the CEN input must be high (=1). Setting CEN = 0 stops the counter's count and ignores the CUP and CDW inputs.



(CPR) Preset input

To preset the counter with the desired value.

It is referred to with the letters CPR followed by the counter number and the preset count value. The counter is preset with the indicated value with an up-flank at the CPR input.

The count value may be indicated by a numeric value or by the internal value of a register R. Its value must be between 0 and ±2,147,483.647.

```
CPR 20 100
```

Presets counter C20 with a value of 100.

```
CPR 22 R200
```

Presets counter C22 with the value of register R200

(C) Status output

It indicates the logic state of the counter. It is referred to by the letter C followed by the counter number. For example: C1, C25, C102, etc.

The logic status of the counter will be C=1 when its count value is "0" and C=0 if otherwise.

(C) Count value

It indicates the value of the internal count of the counter.

It is referred to by the letter `C` followed by the counter number. For example: `C1`, `C25`, `C102`, etc. It is represented by `C123` which is the same as the status output, but is used in different types of instructions.

In binary instruction, it refers to the logic status of the counter.

```
C123 = M100
```

Assigns the status (0/1) of C123 to M100

In arithmetic and comparison instructions, it refers to the internal count of the counter.

```
I2 = MOV C123 R200
```

Transfers the count of C123 to register R200.

```
CPS C123 GT 1000 = M100
```

Compares whether the count of C123 is greater than 1000. If so, it activates mark M100.

The PLC has a 32-bit variable to store the count of each counter.

PLC PROGRAMMING

5

The PLC program is structured by modules and it may consist of:

- Main module (PRG).
- Periodic module (PE)
- First Cycle module (CY1).

All of them consist of a series of instructions that depending on their function may be either directing or executable instructions.

Directing instructions

The directing instructions provide the PLC with information on the type of module (PRG, CY1, ...) and on how it must be executed (REA, IMA, ...).

Executable instructions

With the executable instructions, it is possible to check and/or change the status of the PLC resources. They consist of:

- Logic or Boolean instructions (I28 AND I30).
- Action instructions (=O25).

Logic expressions consist of:

- Consulting instructions (I28, O25).
- Operators (AND).

To write a logic expression in 2 or more lines, put a "\" at the end of the line.

Comments

All comments must begin with ";". Lines beginning with a ";" are considered comments and are not executed.

Empty lines are also possible.

Programming example:

```
PRG            ; Directing instruction
; Example      Comment
I100 = M102    ; Executable instruction
I28 AND I30    ; Logic expression
= O25          ; Action instruction
I32 \          ; Consulting instruction (1st part of expression)
AND I36        ; Consulting instruction (2nd part of expression)
= M300         ; Action instruction
END            ; Directing instruction
```

5.1 Directing instructions

They provide the PLC with information on the type of program module and on how it must be executed.

The available directing instructions are:

PRG, PE t, CY1	Type of module
END	End of module
REA, IMA	Real or image values
L	Label
SUB	Subroutine definition
DEF:	Symbol definition.
NOMONIT	No monitoring
EXTERN	External subroutine definition

PRG, PE t, CY1

Type of module

The PLC program is structured by modules. Every module must begin with its defining instruction (`PRG`, `PE`, `CY1`) and end with the `END` instruction.

PRG	Main module
CY1	First Cycle module.
PE t	Periodic module. It is executed every "t" milliseconds.

See *"4.2 Modular structure of the PLC program"* on page 133.

END **End of module or subroutine**

It must be defined for each module or subroutine.

```
CY1                 ; Beginning of the CY1 module
...
END                 ; End of the CY1 module
PRG                 ; Beginning of the PRG module
...
...
END                 ; End of the PRG module
```

A carriage return is required after the last `END` (empty line).

REA, IMA Real or image values

They indicate whether the following consultations are carried out using the real values (REA) or image values (IMA) or the I, O, M resources. The rest of the resources have no image values, only real.

The real value is the one the resource has at that time and the image value is the one it had at the end of the previous cycle scan.

Image values (IMA) and real values (REA) may be combined in the same instruction.

```
IMA I3 AND REA M4 = O2
```

By default, all the modules (PRG, CY1, PEt) operate with real resource values. Action instructions (=O32) always update the real values of the PLC resources.

Understanding how real and image values work

The following example shows how the PLC acts when working with real or image values. For the given PLC program and with the resources initialized to zero, it shows the status of all the resources at the end of each scan or cycle.

		REA				IMA			
		M1	M2	M3	O5	M1	M2	M3	O5
()=M1		0	0	0	0	0	0	0	0
M1 = M2	Scan 1	1	1	1	1	1	0	0	0
M2 = M3	Scan 2	1	1	1	1	1	1	0	0
M3 = O5	Scan 3	1	1	1	1	1	1	1	0
	Scan 4	1	1	1	1	1	1	1	1

With real values (REA), output O5 goes high (=1) at the end of the first cycle scan, whereas it needs 4 cycle scans when using image values (IMA).

In the first cycle, ( )=M1 sets the real value of M1=1, but its image value is ·0·. Only at the end of this cycle scan will it be ·1·.

The system is faster when working with real values (REA) ; whereas with image values (IMA), it is possible to analyze the same resource throughout the whole program with the same value, regardless of its current value.

L Label

It is used to identify a program line. It may be defined in two ways:

- With L followed by up to 7 digits (L1 - L9999999).
- With L_ followed by 8 characters (L_GEAR).

If it is defined within a module (CY1, PRG or PE), it identifies a program line and allows making references or jumps.

If it is defined outside the modules, for example at the end of the program after END, it indicates the beginning of a subroutine. It is the same as the SUB directing instruction.

If a program has more than one label with the same name or number, the PLC program will issue the corresponding error message when generating the executable program.

SUB Subroutine definition

It indicates the beginning of a subroutine. A subroutine is a portion of the program that may be called upon by any executable instruction.

It is defined with SUB followed by a blank space and up to 24 characters. A subroutine must always end with an END instruction.

```
SUB A22
...
END
```

They must be defined outside the modules (PRG, CY1, PE), for example at the end of the program after the directing instruction END.

A subroutine can also start with the L instruction and end with the END instruction.

EXTERN External subroutine definition

The subroutines defined in the "C" language file used by the program must be defined as external at the beginning of the program before the DEF instructions and the modules: CY1, PRE and PEt.

With the EXTERN instruction, those subroutines may be defined one by one.

It is defined with EXTERN followed by a blank space and the name of the subroutine with up to 24 characters.

```
EXTERN SUMA
EXTERN TEMPERATURE
```

DEF, PDEF Symbol definition.

The PLC allows to define a number of symbols for easier programming and later understanding of the PLC program. These symbols are always programmed at the beginning of the program, before the modules CY1, PRE and PEt.

The DEF instruction may be used to define a limited number of symbols to be used only from the PLC whereas the PDEF instruction may be used to define up to 100 symbols that may be used in the PLC itself, in a part-program or in an external application. Symbols exceeding this limit will be ignored and the CNC will issue the corresponding warning.

They will consist of a sequence of up to 20 characters with capital letters (A.. Z) and digits (0 .. 9). They may also begin with the "/" character. In this case, the next character must be a letter. The name may have a "_" character, but it cannot be its first character. Words reserved for instructions CANNOT be used.

A symbol may be associated with any decimal or hexadecimal number or to PLC resources such as inputs (I), outputs (O), marks (M), registers (R), register bits, counters (C) and timers (T).

Duplicate symbols cannot be defined; but several symbols may be assigned to the same resource.

```
PDEF COOL I12
PDEF K1 $FFFF3
DEF K2 372893
DEF K3 -437289
DEF /FAN I23
```

Once a symbol has been associated with a resource or numeric value, it is possible to use the name of the resource, the number or its associated symbol.

Accessing the PDEF symbols from a part-program or from an application.

Accessing the symbols defined with the PDEF instruction from a part-program, MDI or external application is done using variables as follows. Consulting this variable from the part-program interrupts block preparation.

V.PLC.symbol	Access from a part-program or MDI.
PLC.symbol	Access from an external application.

The variables may be read or written depending on the resource assigned to the symbol defined with PDEF.

NOMONIT **No monitoring**

When programming this directing instruction, it does not generate information necessary to monitor the PLC program. In other words, the program is not monitored.

It must always be programmed at the beginning of the program, before the `DEF` instruction and the modules: `CY1`, `PRG` and `PEt`.

This instruction should only be used when the PLC program execution time is very critical. Define it after debugging the PLC program.

Programming example

```
;No monitoring
NOMONIT

;External subroutine
EXTERN TEMPERATURE

;Symbol definition
DEF COOL I12
DEF /FAN I23

;CY1 module
CY1
...
END

;PRG module

PRG
...
IMA I3 AND REA M4 = 02
...
L_GEAR
...
END

;PEt module
PE 100
...
END

;Subroutine
SUB A22
...
END
```

5.2 Consulting instructions

They may be used to check the status of PLC resources as well as the marks and registers for CNC-PLC communication. See chapter *"7 Logic CNC inputs and outputs"*.

There are the following consulting instructions.

- Simple consulting instructions.
- Flank detection instructions.
- Comparing instructions.

Simple consulting instructions

They test the status of the resources and they return their logic state.

- Inputs (`I1-I1024`)
- Outputs (`O1-O1024`)
- Marks (`M1-M8192`)
- Messages (`MSG1-MSG256`)
- Errors (`ERR1-ERR256`)
- Clocks (`CLK`)
- Registers (`R1-R1024`)
- Register bits (`B0-B31 R1-R1024`)
- Timer status (`T1-T256`)
- Counter status (`C1-C256`)
- CNC-PLC communication marks

`I12`

Returns a "1" if the I12 input is active and a "0" if otherwise.

`START`

Returns a "1" when the CYCLE START key of the front panel is pressed and a "0" if otherwise.

Flank detection instructions

They check whether the state of a resource has changed since the last time this consultation was made. This consultation may be made on real or image values.

The instructions available are:

DFU	Detect an up flank
DFD	Detect a down flank

DFU Detect an up flank

It detects an up-flank (0-to-1 change) at the indicated resource. It returns a "1" if it happened.

DFD Detect a down flank

It detects an down-flank (0-to-1 change) at the indicated resource. It returns a "1" if it happened.

The programming format for DFU and DFD is:

DFU	I1··1024
DFD	O1··1024
	M1··8192
	MSG1··256
	ERR1··256
	B0··31 R1··1024
	CLK
	CNC-PLC communication marks

```
DFU I23
DFU B3R120
DFU AUXEND
DFD O32
DFD M45
```

Comparing instructions

CPS Compare two operands

With the CPS instruction, it is possible to make comparisons between two operands, checking whether the first one is greater than (GT), greater than or equal to (GE), equal to (EQ), different from (NE), less than or equal to (LE) or less than (LT) the second one.

It is possible to use as operands, timers (internal count), counters (internal count), registers, registers for CNC-PLC communication and decimal (#) or hexadecimal numbers within ±2147483647 or between 0 and $FFFFFFFF.

If the required condition is met, the consulting instruction returns a logic value "1" and a "0" if otherwise.

The programming format is:

CPS	T1··256	GT	T1··256
	C1··256	GE	C1··256
	R1··1024	EQ	R1··1024
	R CNC-PLC	NE	R CNC-PLC
	#	LE	#
		LT	

```
CPS C12 GT R14 = M100
```

If the internal count of counter "C12" is GREATER than the value of register R14, the PLC will assign the value of "1" to mark M100 and a "0" if otherwise.

```
CPS T2 EQ 100 = TG1 5 2000
```

When the time elapsed at timer "T2" is equal to 100 milliseconds, it will trigger timer "T5" in monostable mode and with time constant of 2 seconds.

5.3 Operators and symbols

They are used to group and operate with different consulting instructions.

The available operators are NOT, AND, OR, XOR. The operators are associated from left to right and the priorities ordered from the highest to the lowest are NOT AND XOR OR.

The available symbols are \, (,).

NOT **Negates the result of the consultation.**

```
NOT I2 = O3
```

Output "O3" will be active when input I2 is not.

AND **Logic function "AND".**

```
I4 AND I5 = O6
```

Output "O6" will be active when both inputs (I4, I5) are active.

OR **Logic function "OR”.**

```
I7 OR I8 = O9
```

Output "O9" will be active when either one (or both) inputs are active.

XOR **Logic "Exclusive OR" function.**

```
I10 XOR I11 = O12
```

Output "O12" will be active when both inputs I10 and I11 have different logic states.

**** **Line feed**

It is used to write a logic expression in more than one line

The following may be programmed:

```
DFU MSTROBE AND CPS MFUN* EQ 3 = M1003
```

or also:

```
DFU MSTROBE \
AND CPS MFUN* EQ 3
= M1003
```

( ) Open and close parenthesis

They help clarify and select the order the logic expression is evaluated.

```
(I2 OR I3) AND (I4 OR (NOT I5 AND I6)) = O7
```

A consulting instruction consisting of only these two operators always has a value of "1".

```
( ) = O2
```

Output O2 will always be high (=1).

5.4 Action instructions.

Action instructions allow changing the status of PLC resources and CNC-PLC communication marks depending on the result of logic expression.

Logic expression = Action instruction

There may be several action instructions associated with a single logic expression. All the action instructions must be preceded by the "=" sign.

All the action instructions admit a prior `NOT` that inverts the result of the expression for that action.

```
I2 = O3 = NOT M100 = NOT TG1 2 100 = CPR 1 100
```

Output `O3` will show the status of input `I2`.

Mark `M100` will show the negated state of input `I2`.

A down-flank at input `I2` will activate the trigger input `TG1` of timer `T2`.

An up-flank at `I2` will preset counter `C1` with a value of 100.

The action instructions are divided into:

- Assignment binary action instructions.
- Conditional binary action instructions.
- Sequence breaking action instructions.
- Arithmetic action instructions.
- Logic action instructions.
- Specific action instructions.

Action instructions may change the status of all the PLC resources except for the physical inputs being used. When seeing the field "`I 1/1024`", one must understand that only the status of the unused inputs may be changed.

For example, if physical inputs `I1` through `I32` are used, only inputs `I33` through `I1024` may be changed.

5.4.1 Assignment binary instructions

They assign the value (0/1) resulting from a logic expression to the indicated resource.

= I 1/1024	= O 1/1024	= M 1/8192
= MSG 1/256	= ERR 1/256	= TEN 1/256
= TRS 1/256	= TGn 1/256 #/R	= CUP 1/256
= CDW 1/256	= CEN 1/256	= CPR 1/256 #/R
= B 0/31 R 1/499	= CNC-PLC mark	

```
I3 = TG1 4 100
```

Assigns the status of input I3 to the trigger input TG1 of timer T4. Thus, an up-flank at I3 will trigger the TG1 input of timer T4.

```
(I2 OR I3) AND (I4 OR (NOT I5 AND I6)) = M111
```

Assigns to mark M111 the result of evaluating the logic expression: (I2 OR I3) AND (I4 OR (NOT I5 AND I6)).

5.4.2 Conditional binary instructions

There are 3 instructions: SET, RES and CPL that are used to change the status of the indicated resource.

The programming format is:

= SET	I 1/1024
= RES	O 1/1024
= CPL	M 1/8192
	MSG 1/256
	ERR 1/256
	B 0/31 R 1/1024
	CNC-PLC mark

= SET **If expression = "1", it sets the resource to "1".**

If the result of the logic expression is a "1", it sets the indicated resource to "1". If the result is "0", it does not change the resource.

```
CPS T2 EQ 100 = SET B0R100
```

When the timing of timer T2 reaches 100 milliseconds, it sets bit 0 of register R100 to "1".

= RES **If expression = "1", it sets the resource to "0".**

If the result of the logic expression is a "1", it sets the indicated resource to "0". If the result is "0", it does not change the resource.

```
I12 OR NOT I22 = RES M55 = NOT RES M65
```

When the result of the logic expression is a "1", the PLC sets "M55 = 0"; but does not change M65.

If the result of the logic expression is a "0", the PLC does not change M55; but it sets "M65 = 0".

= CPL **If expression = "1", it complements the resource**

If the result of the logic expression is a "1", it complements the status of the indicated resource. If the result is "0", it does not change the resource.

```
DFU I8 OR DFD M22 = CPL B12R35
```

Every time an up-flank is detected at input I8 or down-flank of mark M22, the PLC complements the state of bit 12 of register R35.

5.4.3 Sequence breaking action instructions

These actions interrupt the sequence of a program, resuming it somewhere else in the program.

= JMP **Unconditional jump**

If the result of the logic expression is a "1", it jumps to the indicated label. If the result is a "0", it goes on to the next program line.

Its syntax depends on how the label to jump to was defined

= `JMP L123` If the label was defined as `L123`.

= `JMP L_ASA2` If the label was defined as `L_ASA2`.

I8 = JMP L12

If `I8=1` the program continues at `L12` and it does not execute the intermediate blocks.

```
NOT M14 AND NOT B7R120 = O8
CPS T2 EQ 2000 = O12
L12
(I12 AND I23) OR M54 = O6
```

= CAL **Call to a subroutine.**

If the result of the logic expression is a "1", this action executes the indicated subroutine. If the result of the logic expression is a "0", the PLC will ignore this action and the program will go on without executing that subroutine.

Once the subroutine execution is over, the PLC will continue at the action instruction or executable instruction programmed after `CAL`.

Its syntax depends on how its associated subroutine was defined.

= `CAL OILING` If it was defined as `SUB OILING`.

= `CAL L234` If it was defined as `L234`.

= `CAL L_GEAR` If it was defined as `L_GEAR`.

```
I2 = CAL L5 = O2
```

With `I2=1`, subroutine `L5` will be executed and once executed, the PLC will set `O2` to the value of input `I2` (=1).

If `I2=0`, the subroutine is not executed and the PLC sets output `O2` to the status of input `I2` (=0).

= RET **Return or end of a subroutine.**

If the result of the logic expression is a "1", the PLC will treat this action like an `END` instruction. If the result is a "0", the PLC will ignore it.

If while executing a subroutine, the PLC detects a validated `RET`, it will conclude the subroutine.

5.4.4 Arithmetic action instructions

= MOV Move

It is used to move data from one PLC resource to another.

The programming format is:

	Origin	Destination	Origin code	Destination code	Bits to transmit
= MOV	I1/1024	I1/1024	0(Bin)	0(Bin)	32
	O1/1024	O1/1024	1(BCD)	1(BCD)	28
	M1/8192	M1/8192			24
	MSG1/256	MSG1/256			20
	ERR1/256	ERR1/256			16
	T1/256	R1/1024			12
	C1/256	R CNC-PLC			8
	R1/1024				4
	R CNC-PLC				
	#				

The Origin and Destination codes indicate which format (binary or BCD) they have and in which format they will be deposited in the destination resource. 4, 8, 12, 16, 20, 24, 28 or 32 bits may be transmitted.

If the codes and number of bits to be moved are not indicated, 32 binary bits will be moved bit to bit (0032).

MOV	I12	M100	0032	binary to binary in 32 bits.
MOV	O21	R100	0012	binary to binary in 12 bits.
MOV	C22	O23	0108	binary to BCD in 8 bits.
MOV	T10	M112	1020	BCD to binary in 20 bits.

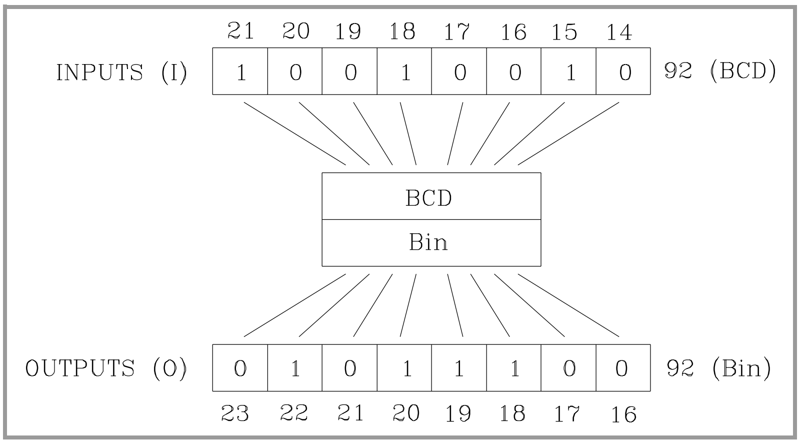
If the number to be converted from binary to BCD is larger than the maximum BCD, its value will be truncated ignoring the most significant bits.

The maximum BCD value that can be converted is: 9 (with 4 bits), 99 (with 8), 999 (with 12), 9999 (with 16), 99999 (with 20), 999999 (with 24), 9999999 (with 28) and 99999999 (con 32).

In these cases, it is recommended to make the move increasing the number of bits by using, if necessary, registers or marks in intermediate steps.

```
I11 = MOV I14 O16 108
```

If input `I11` is "=1", the PLC moves the logic states of the 8 inputs (`I14` plus the next 7) in BCD code to the 8 outputs (`O16` and the next 7) in binary code.



= NGU R 1/1024

It complements the register bits.

It changes the state of each one of the 32 bits of a register.

```
I15 = NGU R152
```

If input "`I15` is =1", the PLC changes the state of the 32 bits of register `R152`.

`R152` before: 0001 0001 0001 0001 0001 0001 0001 0001

`R152` after: 1110 1110 1110 1110 1110 1110 1110 1110

= NGS R 1/1024

Register sign change.

Changes the sign of the register.

```
I16 = NGS R89
```

If input "`I16` = 1", the PLC changes the sign of the contents of register `R89`.

`R89` before: 0001 0001 0001 0001 0001 0001 0001 0001

`R89` after: 1110 1110 1110 1110 1110 1110 1110 1111

= ADS, = SBS, = MLS, = DVS, = MDS
Arithmetic operations

for adding (ADS), subtracting (SBS), multiplying (MLS), dividing (DVS) and calculating the module or remainder of a division (MDS).

Its programming format is:

"Operation" "1st operand" "2nd operand" "Result".

The operands may be: Registers, CNC-PLC communication registers and numbers (#) in the ±2147483647 range or between 0 and $FFFFFFFF.

The result of the operation may be stored in a register or in a CNC-PLC communication register.

= ADS	R1/1024	R1/1024	R1/1024
= SBS	R CNC-PLC	R CNC-PLC	R CNC-PLC
= MLS	#	#	
= DVS			
= MDS			

Examples with R100=1234 and R101=100.

```
() = ADS   R100  R101  R102  ; R102 = 1234 +100      = 1334
() = SBS   R100  R101  R103  ; R103 = 1234 -100      = 1134
() = MLS   R100  R101  R104  ; R104 = 1234 x 100     = 123400
() = DVS   R100  R101  R105  ; R105 = 1234 : 100     = 12
() = MDS   R100  R101  R106  ; R106 = 1234 MOD 100   = 34

() = ADS   1563  R101  R112  ; R112 = 1563 +100      = 1663
() = SBS   R100  1010  R113  ; R113 = 1234 - 1010    = 224
() = MLS   1563  100   R114  ; R114 = 1563 x 100     = 156300

() = MLS   SANALOG     10000     R115
   = DVS   R115  32767 R115      ; Spindle speed command in mV.
```

5.4.5 Logic action instructions

= AND, = OR, = XOR
Logic operations

To perform logic operations: AND, OR and XOR between register contents or between a register content and a number. The result is always stored in a register.

Its programming format is:

AND	R1/1024	R1/1024	R1/1024
OR	R CNC-PLC	R CNC-PLC	R CNC-PLC
XOR	#	#	

Examples with R200 = B1001 0010

R201 = B0100 0101

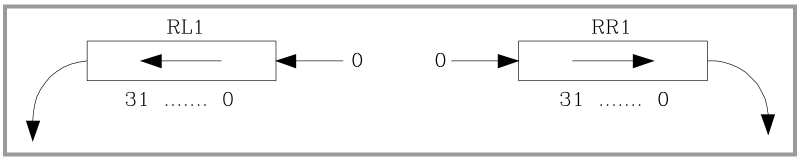
() = AND	R200	R201	R202	; R202 = B0
() = OR	R200	R201	R203	; R203 = B11010111
() = XOR	R200	R201	R204	; R204 = B11010111
() = AND	B1111	R201	R205	; R205 = B00000101
() = OR	R200	B1111	R206	; R206 = B10011111
() = XOR	B1010	B1110	R207	; R207 = B00000100

= RR, = RL Register rotation

Register contents may be rotated to the right (RR) or (RL) to the left and there are two types of rotations: type 1 (RR1 or RL1) and type 2 (RR2 or RL2).

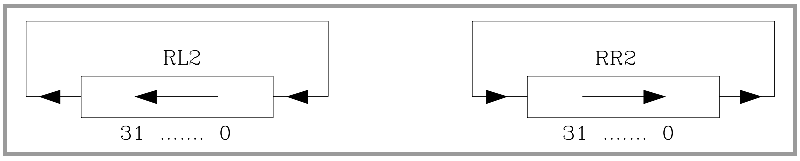
Rotation type 1 (RL1 or RR1):

It inserts a "0" at the least significant bit (RL1) or at the most significant bit (RR1), pushing the other bits of the register. The value of the last bit disappears.



Rotation type 2 (RL2 or RR2):

Circular rotation of the register in the indicated direction.



Its programming format is:

	Origin	Repetition Nr.	Destination
RR1	R1/1024	R1/1024	R1/1024
RR2	R CNC-PLC	R CNC-PLC	R CNC-PLC
RL1		0/31	
RL2			

The origin and destination registers must always be defined, even when they are both the same. The number of repetitions indicates the consecutive number of times the register will be rotated.

```
RR1 R100 1 R200
```

It does one type-1 right-hand rotation of `R100` leaving the result in `R200`.

```
RL2 R102 4 R101
```

It does one type-2 left-hand rotation of `R102` leaving the result in `R101`.

5.4.6 Specific action instructions

= ERA **Clears a group of resources.**

It is used to clear or initialize a group of same-type resources. Indicate the first and last resource to be erased.

Its programming format is:

= ERA	I 1/1024	1/1024
	O 1/1024	1/1024
	M 1/8192	1/8192
	MSG 1/256	1/256
	ERR 1/256	1/256
	T 1/256	1/256
	C 1/256	1/256
	R 1/1024	1/1024

When erasing a group of I, O, M, MSG, ERR or R, the PLC sets them all to "0".

Erasing a group of timers is like resetting them and erasing a group of counters is like presetting with a "0" value.

This action is especially handy when executed in the first cycle module (CY1) in order to set the desired resources to their initial work conditions (states).

```
I10 = ERA O5 12
```

If input "I10=1", the PLC sets outputs O5 through O12 (both included) to "0".

```
I23 = ERA C15 18
```

If input "I23 =1", the PLC presets counters C15 through C18 (both included) to "0".

= PAR **Parity of a register**

It analyzes the type of parity of a register. If the register has EVEN parity, this instruction sets the selected mark, message or error to "1" and to "0" if it has ODD parity.

Its programming format is:

= PAR	R1/1024	M1/8192
	R CNC-PLC	MSG1/256
		ERR1/256
		M CNC-PLC

```
I15 = PAR R123 M222
```

If input "I15=1", the PLC analyzes register R123 and sets mark M222 to "1" if EVEN parity or to "0" if ODD parity.

= CNCRD Reading of internal CNC variables.

= CNCWR Writing of internal CNC variables.

Used to read (CNCRD) and write (CNCWR) the internal CNC variables. Their programming format is:

CNCRD (Variable, Register, Mark)

CNCWR (Register, Variable, Mark)

The CNCRD action loads the contents of the variable into the register and the CNCWR action loads the contents of the register into the variable. The mark is set to "1" at the beginning of the operation and it keeps its value until the end of the operation.

When using CNCRD to read the variables of the arithmetic parameters and those of the OEM, it returns the value multiplied by 10000 (float mode reading).

When requesting information on a nonexistent variable (for example the position of an nonexistent axis), it will show the relevant error message. Likewise, when trying to read a value whose range is greater than that of the PLC register, the CNC will return a zero value and will issue the corresponding error message. Whenever an error occurs when reading a variable, the communication mark will remain at "1".

Synchronous and asynchronous variables

Synchronous variables are the ones resolved immediately whereas asynchronous variables are the ones requiring several cycle scans to be resolved.

Example of how to access asynchronous variables:

<condition> AND NOT M11 = CNCRD (TM.TOOL, R11, M11)

do not repeat this consultation until it ends.

DFD M11 AND CPS R11 EQ 3 = ...

wait for the consultation to end before comparing the data.

Examples of how to access synchronous variables:

<condition> = CNCRD (G.FREAL, R12, M12)

CPS R12 GT 2000 = ...

No need to wait before consulting the data because 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.

= CNCEX Execution of a CNC block.

It may be used to execute a CNC block in the indicated channel, including calls to subroutines or to complete programs. It works like when executing a block in MDI. The restrictions for the execution of commands are the same as for the MDI blocks.

Its programming format is:

CNCEX (block, mark, channel)

The mark is set to "1" at the beginning of the operation and it keeps its value until the end of the operation. If the channel is not indicated, the block is executed in the first or main channel.

```
... = CNCEX (G00 X0 Y0, M99, 2)
... = CNCEX (#CALL sub3.nc, M34)
```

Once the block has been executed, the CNC channel activates the FREE mark to let the PLC know that it is ready to accept a new block.

Executing independent movements from the PLC. Commands MOVEABS, MOVEADD and MOVEINF.

The movements of the independent axes may be programmed directly or with the CNCEX command; however, it is not recommended to use both methods in the same PLC program or subroutine.

The treatment for the execution of the commands is different and the order they are executed might not be the desired one. The CNCEX command is executed through a CNC channel whereas the MOVE* commands are executed directly at the interpolator (usually faster execution).

The independent movements may be executed as follows: The two should not be used in the same program or subroutine.

```
() = CNCEX(#MOVE ADD [X100,F100,NULL], M120,1)
() = MOVEADD(X,100000,100000,NULL)
```

5.4.7 Action instruction of the electronic cam

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 electronic cam.

= CAM ON **Activate the electronic cam**

= CAM OFF **Cancel the electronic cam**

The programming format for each of them is the following.

```
CAM ON (cam, master/"TIME", slave, master_off,
slave_off, range_master, range_slave, type)
```

```
CAM OFF (slave)
```

Executing the CAM OFF command involves eliminating the synchronization of the cam. Once this command has been programmed, the cam ends when reaching the end of its profile.

Parameter	Meaning
cam	Cam number.
master	Name of the master axis.
TIME	Time cam. When programming "TIME" instead of an axis name, the cam is interpreted as being a time cam.
slave	Name of the slave axis.
master_off	Offset for the master axis.
slave_off	Offset for the slave axis.
range_master	Master axis activation scale or range.
range_slave	Slave axis activation scale or range.
type	It defines the type of cam; periodic or not periodic. It is programmed using parameters "ONCE" (non-periodic cam) or "CONT" (periodic cam).

Cam mode.

Two types of cams may be activated; cams in time or cams according to the position of the master axis. The activation instruction is the same and it is selected by the call-parameters.

Cam number.

To activate a cam, it must have been previously defined at the cam editor, within the machine parameters.

Master axis activation range.

The cam is activated when the master axis is between the positions "master_off" and "master_off + range_master".

Range for the slave axis.

The cam applies it to the slave axis when the slave axis is between "slave_off" and "slave_off + range_slave".

Cam type.

Depending on the execution mode, the time cams and the position cams may be of two different types; i.e. periodic or non-periodic. It is selected with the `type` parameter.

Non-periodic — It is defined by assigning the "ONCE" value to the `type` parameter.

This mode maintains the synchronization for the range defined for the master axis. If the master axis moves backwards or if it is a module, the slave axis will keep on executing the cam profile until the cancellation is programmed.

Periodic — It is defined by assigning the "CONT" value to the `type` parameter.

In this mode, when reaching the end of the range of the master axis, it calculates the offset to execute the cam again shifted in the amount of that range. In other words, identical cams are executed along the path of the master axis.

If the master axis is a rotary module and the cam definition range is that module, the two execution modes are equivalent.

Either mode maintains synchronization until the `#CAM OFF` command is executed. When reaching that command, the execution of the cam will end the next time the end of the cam profile is reached.

5.4.8 Action instructions for 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.

= MOVE ABS **Absolute positioning move**

= MOVE ADD **Incremental positioning move**

= MOVE INF **Infinite (endless) positioning move**

The programming format for each of them is the following.

```
MOVE ABS (axis, pos, feed, blend)
MOVE ADD (axis, pos, feed, blend)
MOVE INF (axis, direction, feed, blend)
```

Parameter	Meaning
axis	Axis to position.
pos	Position to reach.
direction	Moving direction. It is programmed with parameters "DIRPOS" (positive direction) or "DIRNEG" (negative direction).
feed	Positioning feedrate
blend	Dynamic blend with the next block It is programmed using parameters "PRESENT", "NULL", "NEXT" or "WAITINPOS.

Position to reach.

With MOVE ABS it will be defined in absolute coordinates whereas with MOVE ADD it will be defined in incremental coordinates. For positioning, the zero offset active in the channel is ignored.

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 it is unidirectional, it positions in the preset direction; otherwise, it positions via the shortest path.

Moving direction.

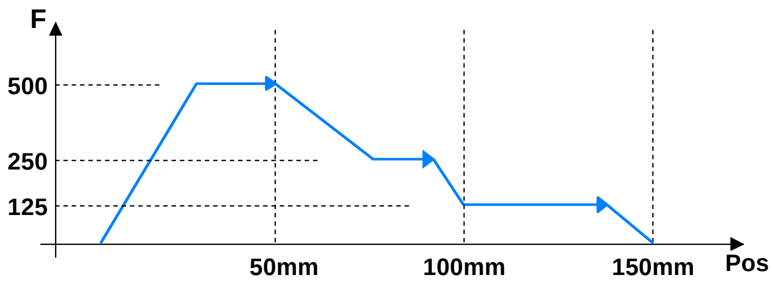
Moving direction. It is used with MOVE INF to execute an endless (infinite) movement until the axis limit is reached or until the movement is interrupted.

Dynamic blend with the next block

It sets the feedrate used to reach the position (dynamic blend with the next block). It is programmed with one of the following parameters.

PRESENT	The axis reaches the indicated position at the feedrate indicated by the block itself.
NEXT	The axis reaches the indicated position at the feedrate indicated in the next block.
NULL	The axis reaches the indicated position at zero feedrate.
WAITINPOS	The axis reaches the indicated position at zero feedrate and it waits to be in position before executing the next block.

```
.. = MOVE ABS (X, 50, 500, PRESENT)
.. = MOVE ABS (X, 100, 250, NEXT)
.. = MOVE ABS (X, 150, 125, NULL)
```



= FOLLOW ON — Activates the synchronization movement

= FOLLOW OFF — Cancels the synchronization movement.

The programming format for each of them is the following.

```
FOLLOW ON (master, slave, nratio, dratio, synctype)
FOLLOW OFF (slave)
```

Parameter	Meaning
master	Name of the master axis.
slave	Name of the slave axis.
nratio	Numerator of the gear ratio. Turns of the slave axis.
dratio	Denominator of the gear ratio. Rotations of the master axis.
synctype	Type of synchronism. It is programmed with parameters "POS" (position synchronism) or "VEL" (velocity synchronism).

Example from the PLC program.

```
FOLLOW ON (A1, Z, 3, 1, VEL)
FOLLOW OFF (Z)
```

5.5 Summary programming commands

RESOURCES AVAILABLE AT THE PLC

Inputs	I1··1024
Outputs	O1··1024
Marks	M1··8192
Message mark	MSG1··256
Error Mark	ERR1··256

Clocks

CLK1	1ms	CLK100	100ms	CLK1000	1s
CLK2	2ms	CLK200	200ms	CLK2000	2s
CLK4	4ms	CLK400	400ms	CLK4000	4s
CLK8	8ms	CLK800	800ms	CLK8000	8s
CLK16	16ms	CLK1600	1.6s	CLK16000	16s
CLK32	32ms	CLK3200	3.2s	CLK32000	32s
CLK64	64ms	CLK6400	6.4s	CLK64000	64s
CLK128	128ms	CLK12800	12.8s	CLK128000	128s

CNC-PLC communication marks.

Timers:	T1··256
Counters:	C1··256
Registers:	R1··1024

CNC-PLC communication registers.

The register value may be treated as a decimal or hexadecimal ("S") number. It is also possible to refer to a register bit using the letter s B (0/31) R (1/1024).

DIRECTING INSTRUCTIONS

PRG	Main module.
CY1	First cycle module.
PE t	Periodic module. It is executed every "t" milliseconds.
END	End of module.

L	Label	L with up to 7 digits	(L1 ... L9999999)
		L_ with up to 8 characters	(L_GEAR)

SUB	Subroutine definition.
DEF	Symbol definition.
PDEF	External symbol definition.
REA	The consultations will use real values.
IMA	The consultations will use image values.

SIMPLE CONSULTING INSTRUCTIONS

Inputs	I1··1024
Outputs	O1··1024
Marks	M1··8192
Message mark	MSG1··256
Error Mark	ERR1··256
Timers (status)	T1··256
Counters (status)	C1··256
Register bit	B0··31 R1··1024
CNC-PLC communication marks	

FLANK DETECTION CONSULTING INSTRUCTIONS

DFU	Up flank detection.
DFD	Down flank detection.

DFU	I1··1024
DFD	O1··1024
	M1··8192
	MSG1··256
	ERR1··256
	B0··31 R1··1024
	CNC-PLC communication marks
	Clock marks (CLK)

COMPARISON CONSULTING INSTRUCTIONS

CPS to do the following comparisons.

CPS	T1··256	GT	T1··256
	C1··256	GE	C1··256
	R1··1024	EQ	R1··1024
	R CNC-PLC	NE	R CNC-PLC
	#	LE	#
		LT	

OPERATORS

NOT	Negates the result of the consultation.
AND	Logic function "AND".
OR	Logic function "OR”.
XOR	Logic "Exclusive OR" function.
"\\"	New line.
"( )"	Consulting instruction whose value is always "1".

ASSIGNMENT BINARY ACTION INSTRUCTIONS

= I 1/1024	= O 1/1024	= M 1/8192
= MSG 1/256	= ERR 1/256	= TEN 1/256
= TRS 1/256	= TGn 1/256 #/R	= CUP 1/256
= CDW 1/256	= CEN 1/256	= CPR 1/256 #/R
= B 0/31 R 1/1024	= CNC-PLC mark	

CONDITIONAL BINARY ACTION INSTRUCTIONS

= SET	If expression = "1", it sets the resource to "1"
= RES	If expression = "1", it sets the resource to "0"
= CPL	If expression = "1", it complements the resource.

		<Resource>
= SET	<Resource>	I 1/1024
= RES	<Resource>	O 1/1024
= CPL	<Resource>	M 1/8192
		MSG 1/256
		ERR 1/256
		B 0/31 R 1/1024
		CNC-PLC mark

SEQUENCE BREAKING ACTION INSTRUCTIONS

= JMP L	Unconditional jump.
= CAL	Call to a subroutine.
= RET	Return or End of a subroutine.

ARITHMETIC ACTION INSTRUCTIONS

= MOV Move

	Origin	Destination	Orig.code	Des.cod	Bits
MOV	I1/1024	I1/1024	0(Bin)	0(Bin)	32
	O1/1024	O1/1024	1(BCD)	1(BCD)	28
	M1/8192	M1/8192			24
	MSG1/256	MSG1/256			20
	ERR1/256	ERR1/256			16
	T1/256	R1/1024			12
	C1/256	R CNC-PLC			8
	R1/1024				4
	R CNC-PLC				
	#				

= NGU R 1/1024	It complements the register bits.
= NGS R 1/1024	Register sign change.
= ADS	Addition.
= SBS	Subtraction.
= MLS	Multiplication.
= DVS	Division.
= MDS	Module or remainder of the division.

ADS	R1/1024	R1/1024	R1/1024
SBS	R CNC-PLC	R CNC-PLC	R CNC-PLC
MLS	#	#	
DVS			
MDS			

LOGIC ACTION INSTRUCTIONS

= AND Logic operation "AND".

= OR Logic operation "OR".

= XOR Logic operation "XOR".

AND	R1/1024	R1/1024	R1/1024
OR	R CNC-PLC	R CNC-PLC	R CNC-PLC
XOR	#	#	

= RR 1/2 Clockwise register rotation.

= RL 1/2 Counterclockwise register rotation.

	Origin	Repetition Nr.	Destination
RR1	R1/1024	R1/1024	R1/1024
RR2	R CNC-PLC	R CNC-PLC	R CNC-PLC
RL1		0/31	
RL2			

SPECIFIC ACTION INSTRUCTIONS

= ERA Erases or resets a group of resources.

ERA	I 1/1024	1/1024
	O 1/1024	1/1024
	M 1/8192	1/8192
	MSG 1/256	1/256
	ERR 1/256	1/256
	T 1/256	1/256
	C 1/256	1/256
	R 1/1024	1/1024

= CNCRD Reading of internal CNC variables.

CNCRD (Variable, R1/1024, M1/8192)

= CNCWR Writing of internal CNC variables.

CNCWR (R1/1024, Variable, M1/8192)

= PAR Parity of a register.

PAR	R1/1024	M1/8192
	R CNC-PLC	MSG1/256
		ERR1/256
		M CNC-PLC

ACTION INSTRUCTIONS OF THE ELECTRONIC CAM

= CAM ON Activate de electronic cam

= CAM OFF Cancel de electronic cam

```
CAM ON (cam, master/"TIME", slave, master_off,
slave_off, range_master, range_slave, type)
CAM OFF (slave)
```

ACTION INSTRUCTIONS FOR INDEPENDENT AXES

- Positioning move.

= MOVE ABS Absolute positioning move.

= MOVE ADD Incremental positioning move.

= MOVE INF Infinite (endless) positioning move.

```
MOVE ABS (axis, pos, feed, blend)
MOVE ADD (axis, pos, feed, blend)
MOVE INF (axis, direction, feed, blend)
```

- Synchronization movement.

= FOLLOW ON Activates the synchronization movement.

= FOLLOW OFF Cancels the synchronization movement.

```
FOLLOW ON (master, slave, nratio, dratio, synctype)
FOLLOW OFF (slave)
```

CNC-PLC COMMUNICATION 6

With the data exchange between the CNC and the PLC, it is possible to:

- Control the logic CNC inputs and outputs in peripheral mode using specific PLC marks and registers.
- Transfer auxiliary M, H and S functions from the CNC to the PLC.
- Generate messages and errors at the CNC using PLC marks.
- Read and modify internal CNC variables from the PLC.
- Access all PLC resources from any part-program.
- Monitor PLC resources on the CNC screen.

Abbreviations used in this chapter

(=0)	Low logic level.
(=1)	High logic level.
(g.m.p.)	General machine parameter.
(a.m.p.)	Machine parameter for Axes and Spindles.

M and H functions with channels

The `M` and `H` functions are exchanged by channel. When using several channels, the marks and registers of these functions must indicate the channel number they refer to. If no channel number is indicated, the marks and registers refer to the first channel.

S functions with multiple spindles

The exchange of `S` functions is independent from the channel. When using several spindles, the marks and registers of these functions refer to the spindle number. The spindle number is determined by its logic number.

6.1 Auxiliary -M- functions

Up to 7 M functions in the same block. The CNC indicates to the PLC which auxiliary M functions are programmed in the execution block using 32-bit registers MFUN1 through MFUN7. Each one of them indicates the number of one of the M functions programmed in the block. If all the registers are not used, the CNC assigns $FFFFFFFF to the unused ones (those with the highest numbers).

This way, if a block contains functions M100, M120 and M135, the CNC will transfer the following information to the PLC:

MFUN1	MFUN2	MFUN3	MFUN4 - MFUN7
100	120	135	$FFFFFFFF

Command MFUN*. Checking if a function has been programmed in the block.

To know whether a particular M function is programmed in the execution block, use one of the following methods:

- Check all the MFUN registers one by one until that particular M function is found or until one of them has the value of $FFFFFFFF.
- Use the "MFUN*" command to check all the registers at the same time.

Example to detect M30: If it has been programmed, it will return a "1", and a "0" if otherwise.

```
CPS MFUN* EQ 30 = ...
```

Sending the function and synchronizing the execution

Within the CNC machine parameters, the auxiliary M function table indicates when the function is sent and when the PLC execution is synchronized. In either case, it may be before or after the movement. See *"2.5 Machine parameters for the M function table"* on page 84.

The sending and synchronizing types may be the following:

M not synchronized.

M sent out and synchronized before the movement.

M sent out before the move and synchronized after the movement.

M sent out and synchronized after the movement.

M functions with different types of synchronization may be programmed in the same block. Each one of them will be sent out to the PLC at the right moment. The transfer of auxiliary M functions is described later on in this chapter. See *"6.4 Transferring auxiliary functions -M-, -H-, -S-"* on page 198.

The functions may be set as follows:

M11 not synchronized.

M12 is sent and synchronized before the movement.

M13 is sent before and synchronized after the movement.

M14 is sent and synchronized after the movement.

When executing a block like this:

```
X100 F1000 M11 M12 M13 M14
```

The functions are transferred as follows:

1. sends the M11, M12 and M13 out to the PLC.
2. waits for the PLC to execute the M12.
3. moves the axis to X100.
4. sends function M14 to the PLC.
5. waits until the PLC executes the M13 and M14.

6.1.1 Special considerations with the multi-spindle option and channels

The CNC may have up to 4 channels and each channel can execute a part-program in parallel with the rest. This means that each channel can execute seven auxiliary functions simultaneously. The auxiliary functions executed from each channel are treated independently; to do that, each channel has its own marks and registers.

Since each channel may have four spindles, it is possible to program in the same block 6 non-spindle-related `M` functions, the startup of all four spindles `M3` / `M4` and a speed for each of them involving an automatic gear change. This means that, because some functions are generated automatically, it may exceed the maximum of seven auxiliary functions per block. In this case, the CNC will send the `M` functions out to the PLC in two stages.

Marks and registers in the channels option

Each channel has 32-bit registers `MFUN1` a `MFUN7` to indicate to the PLC which auxiliary `M` functions are programmed in the execution block.

`MFUN1C1` - `MFUN7C1`	for the first channel.
`MFUN1C2` - `MFUN7C2`	for the second channel.
`MFUN1C3` - `MFUN7C3`	for the third channel.
`MFUN1C4` - `MFUN7C4`	for the fourth channel.

Each one of them indicates the number of one of the `M` functions programmed in the block. If all the registers are not used, the CNC assigns $FFFFFFFF to the unused ones (those with the highest numbers).

This way, if functions `M100` and `M135` are programmed in the first channel and functions `M88` and `M75` in the second channel, the CNC will transfer the following data.

MFUN1C1	MFUN2C1	MFUN3C1 - MFUN7C1
100	135	$FFFFFFFF

MFUN1C2	MFUN2C2	MFUN3C2 - MFUN7C2
88	75	$FFFFFFFF

Commands `MFUNC1*` - `MFUNC4*`. Checking if a function has been programmed in the channel.

To know whether a particular `M` function is programmed in the execution block, use one of the following methods:

- Check all the `MFUN` registers one by one until that particular `M` function is found or until one of them has the value of $FFFFFFFF.
- Use one of the following commands to check all the `MFUN` registers of the channel at the same time.

`MFUNC1*`	For channel 1
`MFUNC2*`	For channel 2.
`MFUNC3*`	For channel 3.
`MFUNC4*`	For channel 4.

Example for detecting `M04` in channel 1. If programmed, it will return a "1" and a "0" if otherwise.

```
CPS MFUNC1* EQ 4 = ...
```

6.2 Auxiliary H functions

Up to 7 M and 7 H functions may be programmed in a block. The treatment of the auxiliary H functions is similar to the M functions without synchronization.

The CNC indicates to the PLC which auxiliary H functions are programmed in the execution block using 32-bit registers HFUN1 through HFUN7. Each one of them indicates the number of one of the H functions programmed in the block. If all the registers are not used, the CNC assigns $FFFFFFFF to the unused ones (those with the highest numbers).

This way, if a block contains functions H12, H20 and H35, the CNC will transfer the following information to the PLC:

HFUN1	HFUN2	HFUN3	HFUN4 - HFUN7
12	20	35	$FFFFFFFF

Command HFUN*. Checking if a function has been programmed in the block.

To know whether a particular H function is programmed in the execution block, use one of the following methods:

- Check all the HFUN registers one by one until that particular H function is found or until one of them has the value of $FFFFFFFF.
- Use the “HFUN*” format to check all the HFUN registers at the same time.

Example to detect H77: If it has been programmed, it will return a “1”, and a "0" if otherwise.

```
CPS HFUN* EQ 77 = ...
```

Sending and synchronizing the function

The H functions are not synchronized and are sent out to the PLC at the beginning of block execution.

The transfer of auxiliary H functions is described later on in this chapter. See *"6.4 Transferring auxiliary functions -M-, -H-, -S-"* on page 198.

When executing a block like this:

```
X100 F1000 H11 H12
```

The functions are transferred as follows:

1. functions H11 and H12 are sent out to the PLC
2. It does not wait for confirmation and the CNC moves the axis to X100.

6.2.1 Special considerations with the multi-spindle option and channels

The CNC may have up to 4 channels and each channel can execute a part-program in parallel with the rest. This means that each channel can execute seven auxiliary functions simultaneously. The auxiliary functions executed from each channel are treated independently; to do that, each channel has its own marks and registers.

Marks and registers in the channels option

Each channel has 32-bit registers HFUN1 to HFUN7 to indicate to the PLC which auxiliary H functions are programmed in the execution block.

HFUN1C1 - HFUN7C1	for the first channel.
HFUN1C2 - HFUN7C2	for the second channel.
HFUN1C3 - HFUN7C3	for the third channel.
HFUN1C4 - HFUN7C4	for the fourth channel.

Each one of them indicates the number of one of the H functions programmed in the block. If all the registers are not used, the CNC assigns $FFFFFFFF to the unused ones (those with the highest numbers).

This way, if functions H10 and H13 are programmed in the first channel and functions H8 and H10in the second channel, the CNC will transfer the following data.

HFUN1C1	HFUN2C1	HFUN3C1 - HFUN7C1
10	13	$FFFFFFFF

HFUN1C2	HFUN2C2	HFUN3C2 - HFUN7C2
8	10	$FFFFFFFF

Commands HFUNC1* - HFUNC4*. Checking if a function has been programmed in the channel.

To know whether a particular H function is programmed in the execution block, use one of the following methods:

- Check all the HFUN registers one by one until that particular H function is found or until one of them has the value of $FFFFFFFF.
- Use one of the following commands to check all the HFUN registers of the channel at the same time.

HFUNC1*	For channel 1
HFUNC2*	For channel 2.
HFUNC3*	For channel 3.
HFUNC4*	For channel 4.

6.3 Auxiliary S function

The auxiliary S function indicates the spindle turning speed with M03 and M04 or the angular position with M19.

The S function with M3 and M4 is always executed at the beginning of the block and the CNC waits for confirmation before going on executing the program. When working with M19, the CNC treats the spindle like a regular linear axis. It only sends the M19 out to the PLC.

The CNC uses 32-bit register SFUN1 to indicate to the PLC the value of the S function programmed in the block. If not programmed, the value of $FFFFFFFF is sent out. The SFUN command only assumes the value of the programmed S if spindle parameter SPDLTIME is other than zero.

The transfer of the S function is described later on in this chapter. See *"6.4 Transferring auxiliary functions -M-, -H-, -S-"* on page 198.

6.3.1 Special considerations with the multi-spindle option and channels

The CNC may have up to four spindles. All of them may be controlled independently in the same block; in other words, each spindle may be given a different command.

When using channels, the spindles may be distributed indistinctly between them. In this case, a channel can control a spindle of another channel. The marks and registers refer to the spindle regardless of the channel they belong to.

The spindle number is determined by its logic number that is set by the order they were defined in the machine parameter SPDLNAME.

Marks and registers in the multi-spindle version

The CNC indicates to the PLC which S functions are programmed in the execution block using 32-bit registers SFUN1 through SFUN4. These registers refer to the spindle number; they are independent from the channel where the spindle is.

Each one of them indicates the value of one of the S functions programmed. If all the registers are not used, the CNC assigns $FFFFFFFF to the unused ones (those with the highest numbers).

This way, if a block contains functions S1000 and S1=550, the CNC will transfer the following information to the PLC:

SFUN1	SFUN2	SFUN3	SFUN4
1000	550	$FFFFFFFF	$FFFFFFFF

Commands SP1FUN* - SP4FUN*. Check if an auxiliary function has been programmed for a spindle.

Considering the possible channels/spindles combinations, these functions are available to make it easier to manage the auxiliary M functions associated with each spindle. Each one indicates if any M3, M4, etc. type M function has been programmed in any channel.

SP1FUN* For the spindle 1.

SP2FUN* For the spindle 2.

SP3FUN* For the spindle 3.

SP4FUN* For the spindle 4.

Checks if the M5 function has been sent to spindle 1 from a channel.

```
CPS SP1FUN* EQ 5 = ...
```

6.4 Transferring auxiliary functions -M-, -H-, -S-

The M and H functions are transferred per channel. Transferring S functions does not depend on the channel.

When executing a block that contains M, H, S functions, the following information is transferred to the PLC.

Transferring -M- functions

The CNC assigns the numbers of the M functions programmed in the block to registers MFUN1 through MFUN7 Some M functions have an associated function that is activated when sending the M to the PLC.

M00	M01	M02	M03	M04
M05	M06	M08	M09	M19
M30	M41	M42	M43	M44

The CNC activates the general logic output MSTROBE to "tell" the PLC that it must execute them. This mark is kept high (=1) for a time period indicated by (g.m.p.) MINAENDW.

Depending on the type of synchronization, the CNC will either wait or not for the general input AUXEND to be activated indicating the end of the PLC execution. The type of synchronization is defined in the machine parameters See ***"2.5 Machine parameters for the M function table"*** on page 84.

The CNC cancels the general logic output "MSTROBE" to conclude the execution.

Transferring -H- functions

The CNC assigns the numbers of the H functions programmed in the block to registers HFUN7 through MFUN7

The CNC activates the general logic output HSTROBE to "tell" the PLC that it must execute them. This mark is kept high (=1) for a time period indicated by (g.m.p.) MINAENDW.

After this time period, the CNC considers its execution completed because there is no synchronization.

When sending several blocks in a row just having H functions, the CNC waits twice the time indicated in g.m.p. MINAENDW.

```
N10 H60
N20 H30 H18
N30 H40
```

Transferring -S- functions

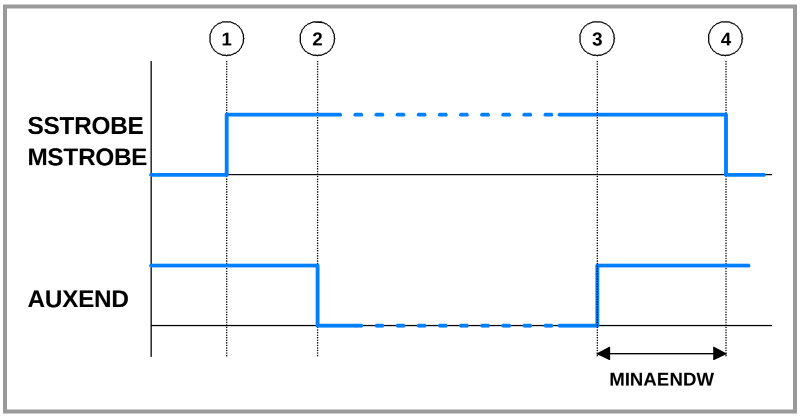
The CNC assigns the values of the S programmed in each spindle to registers SFUN1 through SFUN4.

The CNC activates the general logic output SSTROBE to "tell" the PLC that it must execute it. The CNC waits for the general input AUXEND to be activated indicating the end of the PLC execution.

The CNC cancels the general logic output “SSTROBE” to conclude the execution.

6.4.1 Synchronized transfer

This type of transfer takes place with the S function and with the M functions set with synchronization. See *"2.5 Machine parameters for the M function table"* on page 84.



When the PLC is requested to execute several M or S functions at the same time, the corresponding SSTROBE or MSTROBE signals are activated; but the CNC waits for a single "AUXEND" signal to end all of them.

Transferring -M- functions

1. The CNC indicates in registers MFUN1 a MFUN7 of the channel the M functions programmed in the block and it activates the MSTROBE mark so the PLC executes them.
2. The PLC must deactivate the AUXEND mark to let the CNC know that the execution has begun.
3. Once the required auxiliary functions have been executed, the PLC must activate the AUXEND mark to let the CNC know that the execution has ended.

   The AUXEND mark must be kept high (=1) longer than the time period established by (g.m.p.) MINAENDW.
4. After this time, the CNC deactivates the MSTROBE mark thus ending the execution of the function.

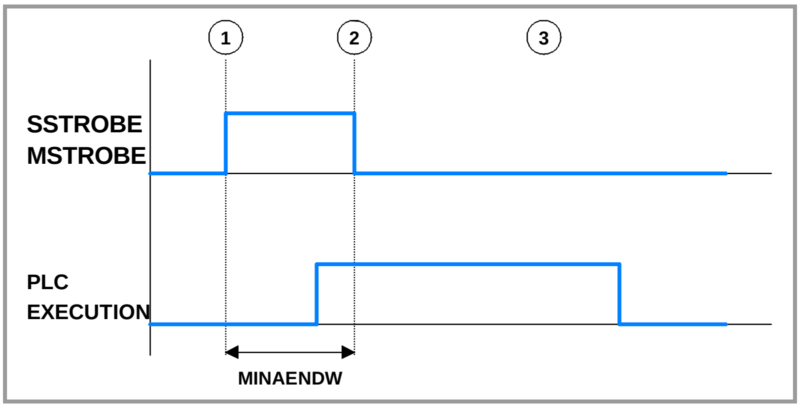
Transferring -S- functions

1. The CNC indicates in registers SFUN1 through SFUN4 the S value programmed in the block and activates the SSTROBE mark so the PLC executes them.
2. The PLC must deactivate the AUXEND mark to let the CNC know that the execution has begun.
3. After selecting the requested S, the PLC must activate the AUXEND mark to let the CNC know that the execution has ended.

   The AUXEND mark must be kept high (=1) longer than the time period established by (g.m.p.) MINAENDW.
4. After this time, the CNC deactivates the SSTROBE mark thus ending the execution of the function.

6.4.2 Non-synchronized transfer

This type of transfer takes place with the H function and with the M functions set without synchronization. See *"2.5 Machine parameters for the M function table"* on page 84.



Transferring -M- functions

1. The CNC indicates in registers MFUN1 a MFUN7 of the channel the M functions programmed in the block and it activates the MSTROBE mark so the PLC executes them.
2. The CNC keeps the MSTROBE mark high (=1) for a time period indicated by (g.m.p.) MINAENDW.
3. After this time, the CNC goes on executing the program regardless of the time required by the PLC to execute that function.

Transferring -H- functions

1. The CNC indicates in registers HFUN1 a HFUN7 of the channel the H functions programmed in the block and it activates the HSTROBE mark so the PLC executes them.
2. The CNC keeps the HSTROBE mark high (=1) for a time period indicated by (g.m.p.) MINAENDW.
3. After this time, the CNC goes on executing the program regardless of the time required by the PLC to execute that function.

Considerations for transferring these functions

The value of (g.m.p.) MINAENDW should be the same or longer than the PLC program execution period (g.m.p.) PRGFREQ in order to ensure that the PLC detects that signal.

When sending non-synchronized H or M functions corresponding to consecutive blocks of the same program, the CNC waits between blocks for a time period indicated by MINANEDW so the PLC can read all the functions.

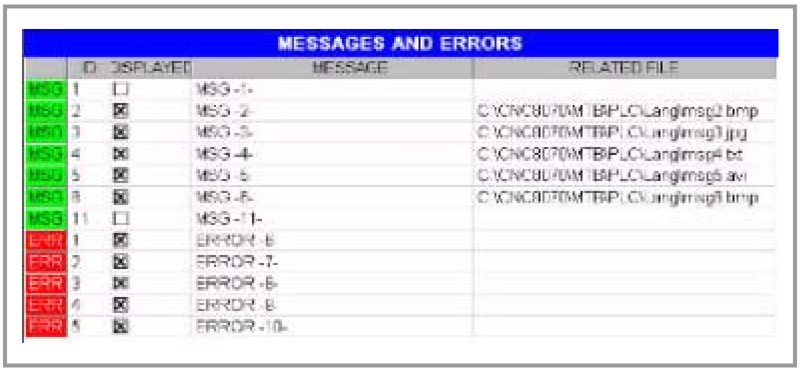
6.5 Displaying PLC errors and messages

The PLC has 256 marks for displaying messages and another 256 marks for displaying errors at the CNC. When the mark is high (=1) the message or the error is active.

`MSG1 - MSG256`	for displaying messages.
`ERR1 - ERR256`	for displaying errors.

There is a message and error table where each message or error may have associated the following:

- A text ("Message" field).
- In the case of messages, if it is only shown at the window for PLC messages or it is also shown to full screen ("Display" field selected).
- In the case of messages, a file with additional information ("help" field) It may be a "bmp, txt, jpg, gif, htm, html or avi" type file.



For more information on how to edit this table, refer to the operation manual.

PLC messages

When activating one of the marks `MSG1` through `MSG256`, the CNC window for PLC messages shows the message number and its associated text.

If the "Display" field is selected, the screen shows the additional information file, or a blue window if otherwise, with the text of the message. To close this window, press [ESC].

When there are more than one message activated, it always shows the one with the highest priority (the one with the lowest number). The PLC-messages window shows the "+" sign meaning that there are more messages activated by the PLC. To display the whole list, press [CTRL] + [M].

Displaying errors

When activating one of the marks ERR1 through ERR256, the CNC interrupts the execution of the part-program and it displays the error number and its associated text in the middle of the screen.

External inputs should be used to activate and deactivate error marks, thus preventing the CNC from receiving those errors at every new PLC cycle scan.

LOGIC CNC INPUTS AND OUTPUTS

7

CNC's physical inputs and outputs are the set of system inputs and outputs that are governed by the PLC and communicate with the outside world through CNC connectors.

The CNC also has a number of logic inputs and outputs to exchange the internal data with the marks and registers of the PLC. This way, the PLC has access to some internal CNC data.

Each of these logic inputs and outputs may be referred to using its associated mnemonic. The mnemonics that begin with a "_" sign indicate that the signal is active low.

CNCREADY	_ALARM
AUXEND	_EMERGEN
MANUAL	_STOP

The CNC's logic outputs or PLC consulting signals are grouped into:

- General consulting signals.
- Axis consulting signals.
- Spindle consulting signals.
- Consultation signals of the independent interpolator.
- Tool manager consulting signals.
- Keystroke consulting signals.

The CNC's logic inputs or signals that may be modified via PLC are grouped into:

- General modifiable signals.
- Axis modifiable signals.
- Spindle modifiable signals.
- Modifiable signals of the independent interpolator.
- Tool manager modifiable signals.
- Keystroke modifiable signals.

Abbreviations used in this chapter

(=0)	Low logic level.
(=1)	High logic level.
(g.m.p.)	General machine parameter.
(a.m.p.)	Machine parameter for Axes and Spindles

7.1 General consulting signals

CNCREADY

This mark indicates the CNC status. It is set to "0" when the CNC is in an error state (red status window) and to "1" if otherwise.

Include this mark in the PLC maneuver to enable the drives.

```
CNCREADY  AND  (rest of conditions)  = O1
```

If there are no errors, output O1 will be high (=1).

START

There is one mark for each channel. The mnemonics for each channel are the following.

STARTC1 (can also be programmed as START)

STARTC2 STARTC3 STARTC4

The CNC channel sets this mark high (=1) to indicate to the PLC that the START key has been pressed at the operator panel.

If the rest of the conditions are met (hydraulic, safety, etc.), the PLC must set the CYSTART mark high (=1) in order for the program to start running. See *"CYSTART"* on page 229.

```
START  AND  (rest of conditions) = CYSTART
```

FHOUT

There is one mark for each channel. The mnemonics for each channel are the following.

FHOUTC1 (can also be programmed as FHOUT)

FHOUTC2 FHOUTC3 FHOUTC4

The CNC channel sets this mark high (=1) when the execution of the part-program is interrupted. To interrupt and later resume the execution of a program, proceed as follows:

- If the "_STOP" mark has been set low (=0).

  To resume execution, set "_STOP" high (=1) and press the [START] key.

- If the "_FEEDHOL" mark is low (=0).

  To resume execution, set "_FEEDHOL" high (=1).

RESETOUT

There is one mark for each channel. The mnemonics for each channel are the following.

RESETOUTC1 (can also be programmed as RESETOUT)

RESETOUTC2 RESETOUTC3 RESETOUTC4

When pressing the [RESET] key or when the PLC activates the "RESETIN" mark (=1), the CNC assumes the initial conditions and sets the "RESETOUT" mark high (=1). This mark stays high (=1) for a time period set by (g.m.p.) MINAENDW. See *"RESETIN"* on page 230.

_ALARM

There is one mark for each channel. The mnemonics for each channel are the following.

_ALARMC1 (can also be programmed as _ALARM)

_ALARMC2 _ALARMC3 _ALARMC4

The CNC channel sets this mark low (=0) when there is an alarm or an emergency generated by the CNC channel. It is not activated when the emergency is caused by the PLC by activating the _EMERGEN mark (=0).

It goes back high (=1) when the CNC channel message is removed and the cause of the alarm has been eliminated.

There is no output associated with this mark. The following example shows how to associate output O1.

```
_ALARM  AND  (rest of conditions) = O1
```

If there are no errors, output O1 will be high (=1)

MANUAL

There is one mark for each channel. The mnemonics for each channel are the following.

MANUALC1 (can also be programmed as MANUAL)

MANUALC2 MANUALC3 MANUALC4

The CNC sets this mark high (=1) when the Manual Operation (JOG) mode is selected.

AUTOMAT

There is one mark for each channel. The mnemonics for each channel are the following.

AUTOMATC1 (can also be programmed as AUTOMAT)

AUTOMATC2 AUTOMATC3 AUTOMATC4

The CNC channel sets this mark high (=1) when the automatic operation mode is selected.

MDI

There is one mark for each channel. The mnemonics for each channel are the following.

MDIC1 (can also be programmed as MDI)

MDIC2 MDIC3 MDIC4

The CNC sets this mark high (=1) when the MDI (Manual Data Input) mode is selected.

- If the MDI mode is selected while in automatic mode, the AUTOMAT and MDI marks will be activated.
- If the MDI mode is selected while in manual (jog) mode, the MANUAL and MDI marks will be activated.

SBOUT

There is one mark for each channel. The mnemonics for each channel are the following.

SBOUTC1 (can also be programmed as SBOUT)

SBOUTC2 SBOUTC3 SBOUTC4

The CNC sets this mark high (=1) when the Single Block mode is selected.

- If the Single Block mode is selected while in Automatic mode, the AUTOMAT and SBOUT marks are activated.
- If the Single Block mode is selected while in manual (jog) mode, the MANUAL and SBOUT marks are activated.

INCYCLE

There is one mark for each channel. The mnemonics for each channel are the following.

INCYCLEC1 (can also be programmed as INCYCLE)

INCYCLEC2 INCYCLEC3 INCYCLEC4

The CNC sets this mark high (=1) when executing a block or moving an axis.

- In execution. It is set high (=1) at the beginning of the execution and it stays high until the end, until the [STOP] key is pressed or until the "_STOP" mark is set low (=0).
- In MDI or Single Block execution, it is set low (=0) at the end of the block.
- In JOG mode, it stays high (=1) while any of the JOG keys is pressed.

RAPID

There is one mark for each channel. The mnemonics for each channel are the following.

RAPIDC1 (can also be programmed as RAPID)

RAPIDC2 RAPIDC3 RAPIDC4

The CNC channel sets this mark high (=1) when executing a rapid traverse movement (G0).

ZERO

There is one mark for each channel. The mnemonics for each channel are the following.

ZEROC1 (can also be programmed as ZERO)

ZEROC2 ZEROC3 ZEROC4

The CNC sets this mark high (=1) when searching home (G74).

PROBE

There is one mark for each channel. The mnemonics for each channel are the following.

PROBEC1 (can also be programmed as PROBE)

PROBEC2 PROBEC3 PROBEC4

The CNC channel sets this mark high (=1) when executing a probing movement (G100).

THREAD

There is one mark for each channel. The mnemonics for each channel are the following.

THREADC1 (can also be programmed as THREAD)

THREADC2 THREADC3 THREADC4

The CNC sets this mark high (=1) when executing an electronic threading block (G33).

TAPPING

There is one mark for each channel. The mnemonics for each channel are the following.

TAPPINGC1 (can also be programmed as TAPPING)

TAPPINGC2 TAPPINGC3 TAPPINGC4

The CNC sets this mark high (=1) when executing a tapping canned cycle (G84).

RIGID

There is one mark for each channel. The mnemonics for each channel are the following.

RIGIDC1 (can also be programmed as RIGID)

RIGIDC2 RIGIDC3 RIGIDC4

The CNC sets this mark high (=1) when executing a rigid tapping block (G63).

CSS

There is one mark for each channel. The mnemonics for each channel are the following.

CSSC1 (can also be programmed as CSS)

CSSC2 CSSC3 CSSC4

The CNC sets this mark high (=1) when Constant Surface Speed is selected (G96).

INTEREND

INPOS

There is one mark for each channel. The mnemonics for each channel are the following.

INTERENDC1 (can also be programmed as INTEREND)

INTERENDC2 INTERENDC3 INTERENDC4

INPOSC1 (can also be programmed as INPOS)

INPOSC2 INPOSC3 INPOSC4

The CNC sets the "INTEREND" mark high (=1) when the theoretical movement of the axes (when it no longer outputs the velocity command) and sets the "INPOS" mark high (=1) when all of them reach their position. The INPOS mark also stays at (=1) while the independent axes are moving.

An axis is in position when it stays within the in-position zone (window) (a.m.p.) "INPOSW" for a time period indicated by (a.m.p.) "INPOSTIME".

The "INTEREND" mark may be used to activate external devices before the axes reach their position. The mark ADVINPOS can also be used. See *"ADVINPOS"* on page 213.

MFUN1···MFUN7

HFUN1···HFUN7

There is one register for each channel. The mnemonics for each channel are the following. Here is an example of the mnemonics for MFUN1 and HFUN1; it is the same for the rest of the registers.

MFUN1C1 (can also be programmed as MFUN1)

MFUN1C2 MFUN1C3 MFUN1C4

HFUN1C1 (can also be programmed as HFUN1)

HFUN1C2 HFUN1C3 HFUN1C4

The channel uses these registers to indicate to the PLC the M or H auxiliary functions selected for execution.

Each channel can have up to 7 M and 7 H functions in a block. If all the registers are not used, the hexadecimal value $FFFFFFFF is assigned to the unused ones (those with the highest numbers).

This way, if functions M100 and M135 are programmed in the first channel and functions M88 and M75 in the second channel, the CNC will transfer the following data.

MFUN1C1	MFUN2C1	MFUN3C1 - MFUN7C1
100	135	$FFFFFFFF

MFUN1C2	MFUN2C2	MFUN3C2 - MFUN7C2
88	75	$FFFFFFFF

If, then, the M88 function is executed in the first channel, then:

MFUN1C1	MFUN2C1	MFUN3C1 - MFUN7C1
88	$FFFFFFFF	$FFFFFFFF

Commands MFUNC1* - MFUNC4* and HFUNC1* - HFUNC4*. Checking if a function has been programmed in the channel.

In order to know whether a particular function is programmed in the block currently being executed, all the registers may be checked one by one or the following commands may be used to check them all at the same time.

MFUNC1*/HFUNC1*	For channel 1. They can also be programmed as MFUN*/HFUN*.
MFUNC2*/HFUNC2*	For channel 2.
MFUNC3*/HFUNC3*	For channel 3.
MFUNC4*/HFUNC4*	For channel 4.

```
CPS MFUNC1* EQ 4 = ...
```

Example for detecting M04 in channel 1. If programmed, it will return a "1" and a "0" if otherwise.

See chapter *"6 CNC-PLC communication"*.

SPN1···SPN7

There is one register for each channel. The mnemonics for each channel are the following. Here is an example of the mnemonics for SPN1; it is the same for the rest of the registers.

SPN1C1 SPN1C2 SPN1C3 SPN1C4

The channel uses these registers to indicate to the PLC which spindle of the channel each auxiliary M function selected for execution is addressed to.

Each channel can have up to 7 M functions in a block. If all the registers are not used, the hexadecimal value $FFFFFFFF is assigned to the unused ones (those with the highest numbers).

This way, if the next block is programmed in the first channel, the CNC will pass on to the PLC the following information.

```
M3.S1 S1=1000 M4.S2 S2=500
```

Clockwise rotation of spindle S1 at 1000 rpm and counterclockwise rotation of spindle S2 at 500 rpm.

MFUN1C1	MFUN2C1	MFUN3C1 - MFUN7C1
3	4	$FFFFFFFF

SPN1C1	SPN2C1	SPN2C1 - SPN2C1
1	2	$FFFFFFFF

If a function is programmed in the block without mentioning the spindle, it will assume the master spindle of the channel.

Commands SP1FUN* - SP4FUN*. Checking if a spindle receives a function from any channel

In order to know whether a particular spindle has received a particular function or not, it is possible to check all the registers one by one or use the following commands to check all of them at the same time.

SP1FUN*	For the spindle 1.
SP2FUN*	For the spindle 2.
SP3FUN*	For the spindle 3.
SP4FUN*	For the spindle 4.

```
CPS SP1FUN* EQ 5 = ...
```

Example to check if the first spindle has received an M5 function from any channel. If it has been programmed, it will return a “1”, and a "0" if otherwise.

MSTROBE

There is one mark for each channel. The mnemonics for each channel are the following.

MSTROBEC1 (can also be programmed as MSTROBE)

MSTROBEC2 MSTROBEC3 MSTROBEC4

The CNC channel sets this mark high (=1) to indicate to the PLC that it must execute the auxiliary M functions indicated in registers “MFUN1” through “MFUN7”.

HSTROBE

There is one mark for each channel. The mnemonics for each channel are the following.

HSTROBEC1 (can also be programmed as HSTROBE)

HSTROBEC2 HSTROBEC3 HSTROBEC4

The CNC channel sets this mark high (=1) to indicate to the PLC that it must execute the auxiliary H functions indicated in registers “HFUN1” through “HFUN7”.

SFUN1···SFUN4

There is one register for each spindle. The mnemonics for each spindle are the following:

SFUN1 SFUN2 SFUN3 HFUN4

These registers indicate the programmed speed for each spindle. These registers refer to the spindle number; they are independent from the channel where the spindle is.

Each one of them indicates the value of one of the S functions programmed. If all the registers are not used, the CNC assigns $FFFFFFFF to the unused ones (those with the highest numbers).

This way, if a block contains functions S1000 and S1=550, the CNC will transfer the following information to the PLC:

SFUN1	SFUN2	SFUN3	SFUN4
1000	550	$FFFFFFFF	$FFFFFFFF

See chapter *"6 CNC-PLC communication"*.

SSTROBE

There is one mark for each spindle. The mnemonics for each spindle are the following:

SSTROBE1 (can also be programmed as SSTROBE)

SSTROBE2 SSTROBE3 SSTROBE4

The CNC channel sets this mark high (=1) to indicate to the PLC that a new spindle speed has been selected .

BLKSEARCH

There is one mark for each channel. The mnemonics for each channel are the following.

BLKSEARCHC1 (can also be programmed as BLKSEARCH)

BLKSEARCHC2 BLKSEARCHC3 BLKSEARCHC4

The CNC channel sets this mark high (=1) to indicate that the "Block Search" option is active in the Automatic Operation mode.

DMxx

This mark is associated with some M auxiliary functions.

The marks associated with functions M00, M01, M02, M06, M08, M09, M30 have a mark for each channel. Here is an example of the mnemonics for DM00; it is the same for the rest of the marks (DM01, DM02, DM06, DM08, DM09, DM30).

DM00C1 (can also be programmed as DM00)

DM00C2	DM00C3	DM00C4

The marks associated with functions M03, M04, M05, M19, M41, M42, M43, M44 have a mark for each channel. Here is an example of the mnemonics for DM03; it is the same for the rest of the marks (DM04, DM05, DM19, DM41, DM42, DM43, DM44).

DM03SP1 (can also be programmed as DM03)

DM03SP2	DM03SP3	DM03SP4

The CNC indicates in these marks the status of the spindle auxiliary M functions. The mark is set to (=1) if the function is active and to (=0) if otherwise.

ADVINPOS

There is one mark for each channel. The mnemonics for each channel are the following.

ADVINPOSC1 (can also be programmed as ADVINPOS)

ADVINPOSC2	ADVINPOSC3	ADVINPOSC4

The CNC channel sets this signal high for some time before the axes reach position. This time is set by (g.m.p.) ANTIME.

If the total duration of the movement is lower than the value set by (g.m.p.) ANTIME, the mark goes high (=1) immediately.

If (g.m.p.) ANTIME has been set to 0, the mark is always active.

It is used on punch presses that have an eccentric cam as a punching system. This signal may be used to start the movement of the punch before the axes reach the position. This reduces idle time, thus increasing the number of punches per minute.

FREE

There is one mark for each channel. The mnemonics for each channel are the following.

FREEC1	FREEC2
FREEC3	FREEC4

The CNC channel sets this signal high (=1) to indicate to the PLC that it is ready to accept a new block, sent using the CNCEX command.

WAITOUT

There is one mark for each channel. The mnemonics for each channel are the following.

WAITOUTC1	WAITOUTC2
WAITOUTC3	WAITOUTC4

It is applied to channel synchronization. The CNC channel sets this signal high (=1) to indicate to the PLC that it is waiting for a synchronization signal. Synchronization signals may be executed from the part-program using the #WAIT or #MEET instructions.

SYNC

There is one register for each channel. The mnemonics for each channel are the following.

SYNC1	SYNC2
SYNC3	SYNC4

This register is used when using, from one channel, a particular spindle for synchronization even if the spindle is in another channel. For example, in the case of dual-turret lathe with a single spindle.

- With the G33 function , when threading with a particular spindle.
- With the G95 function, when programming the feedrate as a function of the turning speed of a particular spindle.

To do that, the PLC indicates in channel register SYNC the spindle to be used, only for synchronization. The SYNC register will take values 1 through 4; when assigning a 0 value, it will use the master spindle of the channel.

The CNC will check the contents of this register at the beginning of the block. If the PLC modifies this register during the execution of the block, the change will not effective until the beginning of the next block.

MMCWDG

This mark indicates the status of the operating system. Its value is (=0) when the CNC operating system is working properly and (=1) when the operating system is locked up.

Include this mark in the PLC maneuver to enable the emergencies when the operating system locks up.

7.2 Consulting signals for axes and spindles

When the spindle is working in closed loop (M19 or G63), it behaves like an axis.

The signal names are generic. Replace the word (axis) with the spindle name or with the name or logic number of the axis.

For example, the name of the `ENABLE(axis)` mark for a machine with the X, Y, Z, Z2, B axes and spindle S:

ENABLEX, ENABLEZ2, ENABLEB, ENABLES

ENABLE3 for the Z axis.

ENABLE5 for the B axis.

ENABLE(axis)

The CNC sets this mark high (=1) to "tell" the PLC to enable the movement of the relevant axis or spindle.

For the independent axes, this mark will stay at (=1) until the synchronization is canceled.

DIR(axis)

The CNC sets this mark to (=1) to indicate that the axis is moving in the negative direction and to (=0) when moving in the positive direction. When the axis is stopped, it keeps the last value.

If the PLC stops the movement of the axes (mark `_FEEDHOL=0`) the `ENABLE(axis)` signal stays at (=1).

REFPOIN(axis)

This mark is set low (=0) on CNC power-up and is set high (=1) after searching home.

DRSTAF(axis)

DRSTAS(axis)

The CNC uses these marks when communicating with the drive via Sercos® and they indicate the drive's status.

In case of error, both marks stay at (=1) if the axis is moving.

Normal

1. After actuating the main switch at the electrical cabinet, the drive is supplied with 24 Vdc.
2. The drive runs an internal test. If OK, it activates the SYSTEM OK output. From that instant on, apply power to the Power Supply.

   Marks `DRSTAF(*)=0  DRSTAS(*)=1`
3. When there is power at the bus, the drive is ready to provide torque. To do that, activate the Drive enable and Speed enable inputs.

   Marks `DRSTAF(*)=1  DRSTAS(*)=0`
4. Once the Drive enable and Speed enable inputs are activated, the drive is running properly.

   Marks `DRSTAF(*)=1  DRSTAS(*)=1`

Mark consultation

When consulting the DRSTAF(*) and DRSTAS(*) marks, the following values may result:

- DRSTAF(*)=0 DRSTAS(*)=0

  The drive is in error or it does not exist.
- DRSTAF(*)=0 DRSTAS(*)=1

  The DC bus has no power. The drive cannot be enabled; but it is possible to provide power to the drives' power supply.
- DRSTAF(*)=1 DRSTAS(*)=0

  The drive's DC bus now has power. The drive may be enabled.
- DRSTAF(*)=1 DRSTAS(*)=1

  The drive is enabled.

INPOS(axis)

The CNC sets this mark high (=1) when the relevant axis or spindle is in position.

An axis is in position when it stays within the in-position zone (window) (a.m.p.) "INPOSW" for a time period indicated by (a.m.p.) "INPOSTIME".

There is an INPOS(axis) mark for each axis and for the spindle and a general INPOS that indicates whether all axes and the spindle have reached their position or not. See *"INPOS"* on page 209.

LUBR(axis)

The CNC sets this mark high (=1) when the relevant axis or spindle needs to be lubricated. See *"LUBRENA(axis)"* on page 236.

(a.m.p.) DISTLUBRI indicates the distance to be moved before being lubricated.

HIRTHON(axis)

MATCH(axis)

All these marks are related to Hirth axes. A Hirth axis is a rotary axis that can only be positioned at specific positions, every so many degrees.

The HIRTHON mark indicates whether it is working as a regular rotary axis (=0) or as a Hirth axis (=1). The MATCH mark indicates whether the Hirth axis is properly positioned (=1) or not (=0).

Both linear and rotary axes can be Hirth axes. (a.m.p.) HIRTH indicates whether the axis can work as Hirth axis or not. Functions G170 and G171 indicate whether it is a Hirth axis (G171) or a normal linear or rotary axis (G170).

PARK(axis)
UNPARK(axis)

The CNC sets mark PARK(axis) to (=1) to indicate that an axis or a spindle is being parked and mark UNPARK(axis) to (=1) to indicate that it is being unparked. The mark PARKED(axis) is also used which may be modified and is described later on. See *"PARKED(axis)"* on page 235.

When parking an axis or spindle, the CNC will not control the axis (it ignores the drive signals, feedback systems, etc.) because, it interprets that the axis is not present in the new machine configuration. When unparking an axis, the CNC will control the axis again because it interprets that the axis is present again in the new machine configuration.

The axes may be parked and unparked from the CNC or from the PLC.

Application example

There are machines that, depending on the type of machining, may have 2 different axis configurations. For example, a machine that interchanges a regular spindle with an orthogonal one may have the following configurations:

- With a normal spindle, X Y Z axes configuration.
- With an orthogonal spindle, X Y Z A B axes configuration.

When working with a regular spindle, the A and B axes are not present and the CNC issues an error because it takes them into account (drives, feedback systems, etc.). The A and B axes must be parked to avoid this.

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.

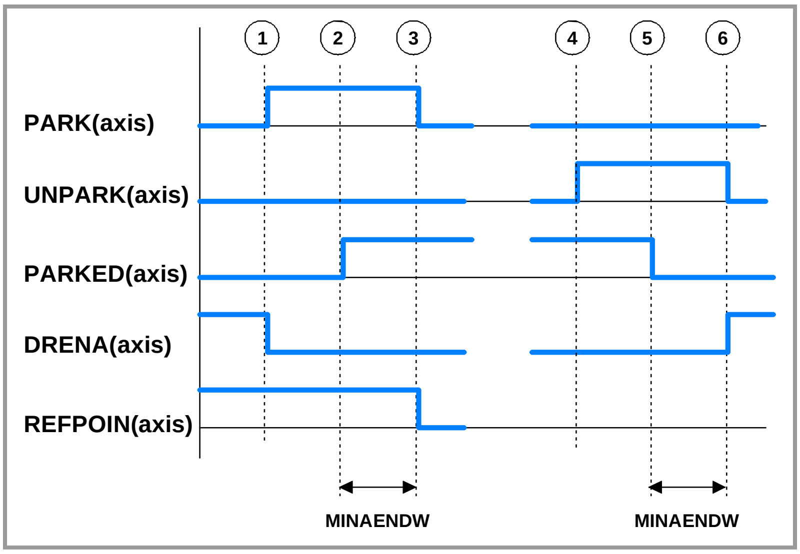
If after parking the spindles, there is only one spindle left in the channel, it will become the new master. If a spindle is unparked and it is the only spindle of the channel, it is also assumed as the new master spindle.

Park/Unpark from the CNC

This type of maneuver is good for applications that need the axes or spindles to be parked automatically from a part program (for example from a part-program). The parking/unparking maneuver via part-program or MDI is controlled using the programming instructions #PARK and #UNPARK.

The CNC uses the PARK and UNPARK signals to inform the PLC that the parking or unparking process has been engaged.

To park an axis or a spindle, its enable signals must be (=0). Likewise, after unparking the axis, the axis enable signals must be set to (=1).



Maneuver to park an axis or spindle from the CNC.

1. When executing the programming instruction #PARK the CNC checks whether the requested axis can be parked or not. If so, the CNC sets the PARK mark high (=1) to let the PLC know that it must park the corresponding axis.

   For digital axes, the PLC must first remove the enable of the drive of the axis to be parked (DRENA). Also, the CNC sends to the drive the command to park the axis.

2. The PLC, after receiving the PARK signal, parks the requested axis. After verifying that the axis has been parked (presence sensors) the PLC sets mark PARKED to (=1).

3. The CNC will recognize that the axis has been parked when detecting that the PARKED signal has been activated. The PARK and REFPOIN signals are reset completing this process.

Maneuver to unpark an axis or spindle from the CNC.

4. When executing the programming instruction #UNPARK, the CNC sets the UNPARK mark high (=1) to let the PLC know that it must unpark the corresponding axis.

   For digital axes, the CNC sends to the drive the command to unpark the axis.

5. The PLC, after receiving the UNPARK signal, unparks the requested axis. After verifying that the axis has been unparked (presence sensors) the PLC sets mark PARKED to (=0).

6. The CNC will recognize that the process has been completed when detecting that the PARKED signal has been deactivated. The UNPARK and REFPOIN signals are reset.

   For digital axes, the PLC must activate the enable of the drive of the axis (DRENA).

Example for parking and unparking an axis:

Input I15 corresponds to the presence sensor of the "B" axis

```
PARKB AND NOT I15 = SET PARKEDB
```

If there is a request to park the "B" axis (PARKB) and the axis is not present (NOT I15), the axis is parked (SET PARKEDB).

```
UNPARKB AND I15 = RES PARKEDB
```

If there is a request to unpark the "B" axis (UNPARKB) and the axis is not present (I15), the axis is unparked (RES PARKEDB).

```
NOT (PARKB OR UNPARKB OR PARKEDB) AND··· = DRENAB = SPENAB
= SERVOBON
```

If the axis is neither parked nor being parked and the enabling conditions are met, the axis gets enabled.

Park/Unpark from the PLC

This type of maneuver is good for applications that need manual parking of the axes either while the machine is off or on (with or without power).

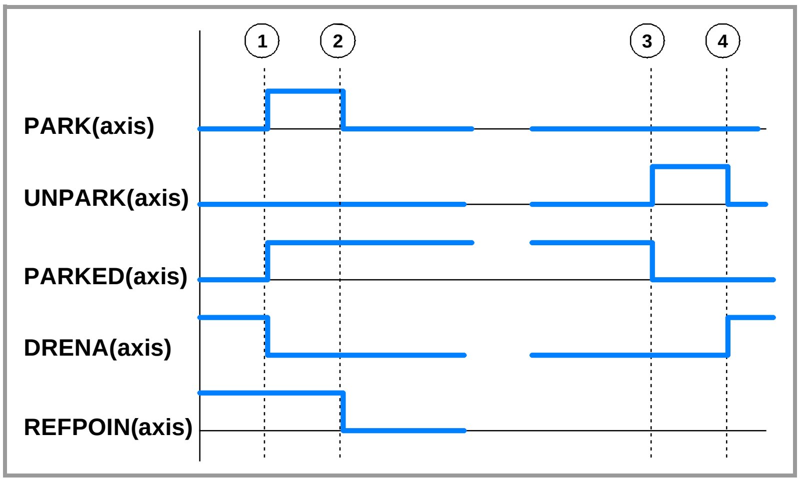
The axis parking/unparking maneuver, handled from the PLC program, is controlled using the PARKED signal. This mark is usually affected by the input for the axis presence sensor. The state of this signal is maintained even if the CNC is turned off.

The CNC uses logic signals PARK and UNPARK to inform the PLC that the parking or unparking processes are in execution.

To park an axis, the axis enable signals must be (=0). Likewise, after unparking the axis, the axis enable signals must be set to (=1).

For safety, after parking and unparking an axis, the REFPOIN signal of the axis is set to (=0).

In versions older than V2.00, at the end of the parking or unparking maneuver, the PLC must generate a reset (RESETIN).



Maneuver to park an axis or spindle from the PLC.

1. From the PLC program, the PARKED mark is set to (=1) to park the relevant axis. The CNC sets the PARK mark to (=1) and begins parking the axis.

   For digital axes, the PLC must first remove the enable of the drive of the axis to be parked (DRENA). Also, the CNC sends to the drive the command to park the axis.

2. The CNC considers the operation completed. It resets the PARK signal.

Maneuver to unpark an axis or spindle from the PLC.

3. From the PLC program, the PARKED mark is set to (=0) to unpark the relevant axis. The CNC sets the UNPARK mark to (=1) and begins unparking the axis.

   For digital axes, the CNC sends to the drive the command to unpark the axis.

4. The CNC considers the operation completed. It resets the UNPARK signal.

   For digital axes, the PLC must activate the enable of the drive of the axis (DRENA).

Example for parking and unparking an axis:

```
I10 = PARKEDV
```

Axis present. "V" axis presence sensor

```
NOT (PARKV OR UNPARKV OR PARKEDV) AND··· = DRENAV = SPENAV
= SERVOVON
```

If the axis is neither parked nor being parked and the enabling conditions are met, the axis gets enabled.

7.3 Consulting signals for the spindle

CAXIS

There is one mark for each channel. The mnemonics for each channel are the following.

CAXISC1 (can also be programmed as CAXIS)

CAXISC2 CAXISC3 CAXISC4

The CNC channel sets this mark to (=1) when the spindle is working as C axis. This mark is kept active while any of the functions #CAX, #FACE or #CYL are kept active.

REVOK

There is one mark for each spindle. The mnemonics for each spindle are the following:

REVOK1 (can also be programmed as REVOK)

REVOK2 REVOK3 REVOK4

It indicates whether the actual (real) spindle rpm match the ones programmed (=1) or not (=0). In other words, whether they are within the percentages set by (a.m.p.) UPSPDLIM and LOSPDLIM.

- When the spindle is stopped, M5, REVOK (=1).
- With M3 and M4, the CNC sets this mark high (=1) when the actual spindle rpm match the ones programmed.
- When working in closed loop (M19 or G63), the CNC sets this mark low (=0) while the spindle is moving and high (=1) when the spindle is in position.

The REVOK signal may be used to handle the Feedhold signal and avoid machining at lower or higher rpm than desired.

7.4 Consultation signals of the independent interpolator

The signal names are generic. Replace the text (axis) with the name or logic axis number.

IBUSY(axis)

For movements of the independent axis, the independent interpolator of the axis sets this mark to (=1) when it has an instruction pending to execute.

IFREE(axis)

For movements of the independent axis, the independent interpolator of the axis sets this mark to (=1) to indicate to the PLC that it is ready to accept a new motion block.

IFHOUT(axis)

For movements of the independent axis, the independent interpolator of the axis sets this mark to (=1) when the execution has been interrupted.

IEND(axis)

For movements of the independent axis, the independent interpolator of the axis sets this mark to (=1) when the movement has finished and the final position has been reached.

INSYNC(axis)

For synchronization movements of an independent axis and of an electronic cam, the interpolator of the axis sets this mark to (=1) when synchronism has been reached. This mark stays at (=1) while maintaining synchronism.

7.5 Tool manager consulting signals

TMOPERATION

There is one mark for each channel. The mnemonics for each channel are the following.

TMOPERATIONC1 (same as TMOPERATION)

TMOPERATIONC2 TMOPERATIONC3 TMOPERATIONC4

This register indicates the type of operation to be carried out by the tool manager.

0	Do nothing.
1	Take a tool from the magazine and insert it in the spindle.
2	Leave the spindle tool in the magazine.
3	Insert a ground tool in the spindle.
4	Leave the spindle tool on the ground.
5	Leave the spindle tool in the magazine and take another one from the magazine.
6	Leave the spindle tool in the magazine and take another one from ground.
7	Leave the spindle tool on the ground and take another one from the magazine.
8	Leave the spindle tool on the ground and take another one from ground.
9	Take a ground tool to the magazine going through the spindle.
10	Take a tool from the magazine and leave on the ground going through the spindle.
11	Orient the magazine.
12	Leave the spindle tool in the magazine and take another one from the same magazine. Special for a synchronous magazine in the following cases: • Non-random having a tool changer arm with two claws. • Random when having special tools.
13	Orienting two magazines
14	Leave the spindle tool in a magazine and take another one from another magazine.

TMOPSTROBE

There is one mark for each channel. The mnemonics for each channel are the following.

TMOPSTROBEC1 (same as TMOPSTROBE)

TMOPSTROBEC2 TMOPSTROBEC3 TMOPSTROBEC4

The CNC sets this mark high (=1) to let the PLC know that it must execute the operation indicated by TMOPERATION.

LEAVEPOS

There is one register for each magazine. The mnemonics for each magazine are the following.

LEAVEPOSMZ1 (can also be programmed as LEAVEPOS)

LEAVEPOSMZ2 LEAVEPOSMZ3 LEAVEPOSMZ4

This register indicates the magazine position to leave the tool.

TAKEPOS

There is one register for each magazine. The mnemonics for each magazine are the following.

TAKEPOSMZ1 (can also be programmed as TAKEPOS)

TAKEPOSMZ2 TAKEPOSMZ3 TAKEPOSMZ4

This register indicates the magazine position of the tool to be taken.

NEXTPOS

There is one register for each magazine. The mnemonics for each magazine are the following.

NEXTPOSMZ1 (can also be programmed as NEXTPOS)

NEXTPOSMZ2 NEXTPOSMZ3 NEXTPOSMZ4

This register indicates the magazine position occupied by the next tool.

TWORNOUT

There is one mark for each channel. The mnemonics for each channel are the following.

TWORNOUTC1 (can also be programmed as TWORNOUT)

TWORNOUTC2 TWORNOUTC3 TWORNOUTC4

The CNC channel sets this mark high (=1) to "tell" the PLC that the tool has been rejected because it is worn out (real life > maximum life span).

TMINEM

There is one register for each magazine. The mnemonics for each magazine are the following.

TMINEMZ1 (can also be programmed as TMINEM)

TMINEMZ2 TMINEMZ3 TMINEMZ4

The CNC sets this mark high (=1) to inform the PLC that an emergency has occurred at the tool manager.

MZID

There is one register for each channel. The mnemonics for each channel are the following.

MZIDC1 MZIDC2 MZIDC3 MZIDC4

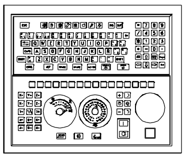
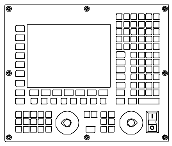
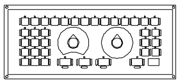
This register indicates which magazine contains the tool requested by the channel.

7.6 Keystroke consulting signals

KEYBD1, KEYBD2

These registers indicate which key has been pressed, relevant bit high (=1).

- On the standard keyboard, the user keys "User1" through "User16" are numbered from left to right.
- On the compact keyboard, the keys "User1" through "User6" are located next to the jog keys. They are numbered from left to right and from top down.
- On the flat keyboard, the keys "Jogkey1" through "Jogkey3" are located next to the jog keys. The are numbered from left to right.

	Bit	OP-Panel-H/E	LCD-10K	Jog Panel
				
KEYBD1	0	User1	User1	User1
KEYBD1	1	User2	User2	User2
KEYBD1	2	User3	User3	User3
KEYBD1	3	User4	User4	User4
KEYBD1	4	User5	User5	User5
KEYBD1	5	User6	User6	User6
KEYBD1	6	User7		User7
KEYBD1	7	User8		User8
KEYBD1	8	User9		User9
KEYBD1	9	User10		User10
KEYBD1	10	User11		User11
KEYBD1	11	User12		User12
KEYBD1	12	User13		User13
KEYBD1	13	User14		User14
KEYBD1	14	User15		
KEYBD1	15	User16		
KEYBD1	16	X+	1	X
KEYBD1	17	Y+	2	Y
KEYBD1	18	Z+	3	Z
KEYBD1	19	X-	4	4
KEYBD1	20	Y-	5	5
KEYBD1	21	Z-	6	6
KEYBD1	22	7-	-	+
KEYBD1	23	RAPID	RAPID	RAPID
KEYBD1	24	7+	+	-
KEYBD1	25	4+		jogkey1
KEYBD1	26	5+		jogkey2

	Bit	OP-Panel-H/E	LCD-10K	Jog Panel
KEYBD1	27	6+		jogkey3
KEYBD1	28	4-		
KEYBD1	29	5-		
KEYBD1	30	6-		
KEYBD1	31			

	Bit	Key	Bit	Key
KEYBD2	0	Spindle Ovr+	16	Feed override
KEYBD2	1	Spindle clockwise	17	Feed override
KEYBD2	2	Spindle positioning	18	Feed override
KEYBD2	3	Spindle stop	19	Feed override
KEYBD2	4	Spindle Ovr-	20	Feed override
KEYBD2	5	Spindle c.clockwise	21	
KEYBD2	6	START	22	
KEYBD2	7	STOP	23	
KEYBD2	8		24	Mode selector
KEYBD2	9		25	Mode selector
KEYBD2	10		26	Mode selector
KEYBD2	11	ZERO	27	Mode selector
KEYBD2	12		28	
KEYBD2	13	Single block	29	
KEYBD2	14		30	
KEYBD2	15	RESET	31	

Mode Selector

KEYBD2				
27	26	25	24	
0	0	0	0	Handwheel 100
0	0	0	1	Handwheel 10
0	0	1	0	Handwheel 1
0	0	1	1	JOG 1
0	1	0	0	JOG 10
0	1	0	1	JOG 100
0	1	1	0	JOG 1000
0	1	1	1	JOG 10000
1	0	0	0	JOG Continuous

Feed Override

KEYBD2					
20	19	18	17	16	
0	0	0	0	0	0 %
0	0	0	0	1	2 %
0	0	0	1	0	4 %
0	0	0	1	1	10 %
0	0	1	0	0	20 %
0	0	1	0	1	30 %
0	0	1	1	0	40 %
0	0	1	1	1	50 %
0	1	0	0	0	60 %
0	1	0	0	1	70 %
0	1	0	1	0	80 %
0	1	0	1	1	90 %
0	1	1	0	0	100 %
0	1	1	0	1	110 %
0	1	1	1	0	120 %
0	1	1	1	1	130 %
1	0	0	0	0	140 %
1	0	0	0	1	150 %
1	0	0	1	0	160 %
1	0	0	1	1	170 %
1	0	1	0	0	180 %
1	0	1	0	1	190 %
1	0	1	1	0	200 %

7.7 General modifiable signals

The _EMERGEN, _STOP, _FEEDHOL, _XFERINH and _CYSTART signals must be defined in the PLC program.

_EMERGEN

There is one mark for each channel. The mnemonics for each channel are the following.

_EMERGENC1 (can also be programmed as _EMERGEN)

_EMERGENC2 _EMERGENC3 _EMERGENC4

If the PLC sets this mark low (=0), the CNC stops the axes and the spindle and displays the corresponding error message.

While the "_EMERGEN" mark is low (=0), the CNC does not allow executing programs and aborts any axis or spindle movement.

```
I-EMERG  AND  (rest of conditions) = _EMERGEN
```

If the emergency-stop button is pressed (I-EMERG=0) or any other emergency situation occurs (=0), the _EMERGEN mark is set low (=0) causing an emergency at the CNC.

_STOP

There is one mark for each channel. The mnemonics for each channel are the following.

_STOPC1 (can also be programmed as _STOP)

_STOPC2 _STOPC3 _STOPC4

If the PLC sets this mark low (=0), the CNC channel interrupts the execution of the part-program; but it keeps the spindle turning. It is similar to pressing the [STOP] button at the keyboard.

The movement of the independent axes is not affected by this mark. Nor is affected by the [STOP] key of the operator panel.

To resume the execution of the program, besides setting this mark high (=1), the CYSTART mark must also be activated. See ***"CYSTART"*** on page 229.

_XFERINH

There is one mark for each channel. The mnemonics for each channel are the following.

_XFERINHC1 (can also be programmed as _XFERINH)

_XFERINHC2 _XFERINHC3 _XFERINHC4

If the PLC sets this mark low (=0), the CNC channel inhibits the execution of the next block, but it ends the current block. When this mark is set back high (=1), the CNC resumes the execution of the program.

_FEEDHOL

There is one mark for each channel. The mnemonics for each channel are the following.

_FEEDHOLC1 (can also be programmed as _FEEDHOL)

_FEEDHOLC2 _FEEDHOLC3 _FEEDHOLC4

If the PLC sets this mark low (=0), the CNC channel interrupts momentarily the movement of the axes; but keeping the spindle turning. When this mark goes back high (=1), the axes resume their motion.

The movement of the independent axes is not affected by this mark.

If the _FEEDHOL mark is set low (=0) in a motionless block, the CNC continues executing the program until it detects a motion block.

Hirth axes are stopped at positions multiple of the Hirth pitch.

All the stops and starts of the axes are carried out with the corresponding acceleration and deceleration.

CYSTART

There is one mark for each channel. The mnemonics for each channel are the following.

CYSTARTC1 (can also be programmed as CYSTART)

CYSTARTC2 CYSTARTC3 CYSTARTC4

When the operator presses the [START] key, the CNC lets the PLC know by activating the START mark.

If the rest of the conditions are met (hydraulic, safety, etc.), the PLC must set the CYSTART mark high (=1) in order for the program to start running.

```
START  AND  (rest of conditions) = CYSTART
```

SBLOCK

There is one mark for each channel. The mnemonics for each channel are the following.

SBLOCKC1 (can also be programmed as SBLOCK)

SBLOCKC2 SBLOCKTC3 SBLOCKC4

If the PLC sets this mark high (=1), the CNC switches to SINGLE BLOCK operation mode.

MANRAPID

This mark is treated in a similar way to the Rapid traverse key.

If the PLC sets this mark high (=1), the CNC selects rapid traverse for all the movements in JOG mode. When the mark goes back low (=0), all the JOG movements are carried out at the feedrate that was selected.

Feedrate changes in JOG mode are always carried out with linear acceleration.

OVRCAN

There is one mark for each channel. The mnemonics for each channel are the following.

OVRCANC1 (can also be programmed as OVRCAN)

OVRCANC2 OVRCANTC3 OVRCANC4

If the PLC sets this mark high (=1), the CNC selects 100 % of feedrate regardless of the feedrate override currently selected.

While the OVRCAN mark is high (=1), the CNC channel will apply 100% of the feedrate corresponding to each work mode.

LATCHM

With this mark, it is possible to select how the JOG keys will work in JOG mode.

- If this mark is low (=0), the axes will move while the corresponding JOG key kept pressed.
- If it is high (=1), the axes will move from the instant the JOG key is pressed until the software limits are reached or the [STOP] key is pressed or another JOG key is pressed (in this case the new axis will start moving).

RESETIN

There is one mark for each channel. The mnemonics for each channel are the following.

RESETOUTC1 (can also be programmed as RESETOUT)

RESETOUTC2 RESETOUTC3 RESETOUTC4

With an up-flank, (0-to-1 transition), the CNC assumes the machining conditions selected by machine parameter and it activates the RESETOUT mark. See *"RESETOUT"* on page 206.

The treatment of this mark is similar to that of the [RESET] key.

AUXEND

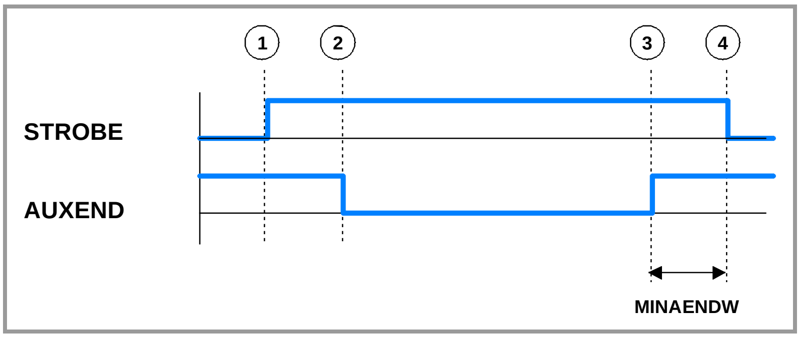
There is one mark for each channel. The mnemonics for each channel are the following.

AUXENDC1 (can also be programmed as AUXEND)

AUXENDC2 AUXENDC3 AUXENDC4

It is used when executing the auxiliary functions M and S. See chapter *"6 CNC-PLC communication"*.

It works as follows:



1. The CNC channel indicates to the PLC, in registers MFUN and SFUN of the channel, the functions that it must execute and it activates the MSTROBE or SSTROBE mark to start the execution.
2. The PLC, when detecting that one of these marks has been activated, it must cancel the "AUXEND" mark to let the CNC know that the execution begins.
3. Once the required auxiliary functions have been executed, the PLC must activate the AUXEND mark to let the CNC know that the execution has ended.

   The AUXEND mark must be kept high (=1) longer than the time period established by (g.m.p.) MINAENDW.
4. After this time, the CNC deactivates the corresponding SSTROBE or MSTROBE mark thus ending the execution of the function.

BLKSKIP1

There is one mark for each channel. The mnemonics for each channel are the following.

BLKSKIP1C1 (can also be programmed as BLKSKIP1)

BLKSKIP1C2 BLKSKIP1C3 BLKSKIP1C4

The PLC sets this mark high (=1) to let the CNC channel know that the block skip condition "/" is met. Therefore, blocks with that condition will not be executed.

M01STOP

There is one mark for each channel. The mnemonics for each channel are the following.

M01STOPC1 (can also be programmed as M01STOP)

M01STOPC2 M01STOPC3 M01STOPC4

The PLC sets this mark high (=1) to "tell" the CNC channel not to ignore the conditional stops (M01).

TIMERON

The CNC has a timer to be used at will.

It times in seconds and it is enabled and disabled with this mark. It will be enabled (counting) with TIMERON set high (=1).

To initialize and/or know its count, use the (V.)PLC.TIMER variable This variable may be accessed from the PLC, program, MDI or Interface (any supported application).

Example of how to know the machining time.

```
CY1
()= MOV 0 R100
()= CNCWR (R100, PLC.TIMER, M11)
```

Resets the timer to "0" on power-up.

```
END

PRG
AUTOMAT AND INCYCLE = TIMERON
```

Timer active while machining.

```
()= CNCRD (PLC.TIMER, R300, M12)
```

Register R300 shows the value of the timer.

```
END
```

PLCREADY

This mark indicates whether the PLC is running (=1) or stopped (=0).

- It must be high (=1) for the CNC to allow moving the axes and turning the spindle.
- If set low (=0), the PLC program stops and an error message is displayed.

NOWAIT

There is one mark for each channel. The mnemonics for each channel are the following.

`NOWAITC1`	`NOWAITC2`
`NOWAITC3`	`NOWAITC4`

It is applied to channel synchronization. The PLC sets this mark high (=1) to cancel all the synchronizations with the CNC channel.

For example, with the `NOWAITC1` signal set to (=1), the waits programmed in any channel with the `#WAIT,` instruction and that refer to a mark of channel 1, they finish immediately and the program execution resumes.

DISCROSS1...9

It is applied to the cross compensation tables defined in the machine parameters. There is one mark for each table.

The PLC sets the mark to (=1) to disable the tool. The `DISCROSS1` mark is for table 1, `DISCROSS2` for table 2 and so on.

7.8 Modifiable signals for axes and spindles

When the spindle is working in closed loop (M19 or G63), it behaves like an axis.

The signal names are generic. Replace the word (axis) with the spindle name or with the name or logic number of the axis.

For example, the name of the `LIMITPOS(axis)` mark for a machine with the X, Y, Z, Z2, B axes and spindle S:

LIMITPOSX, LIMITPOSZ2, LIMITPOSB, LIMITPOSS

LIMITPOS3 for the Z axis.

LIMITPOS5 for the B axis.

LIMITPOS(axis)

LIMITNEG(axis)

If the PLC sets this mark high (=1), the CNC interprets that the corresponding axis or spindle has exceeded the positive (POS) or negative (NEG) travel limits.

It stops the axes and the spindle and displays the relevant error message.

To take the axis to the work zone, access the JOG mode and move the axis or spindle that overran the travel limit. They can only be moved in the proper direction.

DECEL(axis)

This mark is used during home search. The PLC sets this mark high (=1) to indicate that the home switch is pressed.

The CNC decelerates the axis, it switches from the fast homing feedrate indicated by (a.m.p.) "`REFEED1`", to the slow homing feedrate indicated by the (a.m.p.) "`REFEED2`".

INHIBIT(axis)

If the PLC sets this mark high (=1) the CNC inhibits any movement of the corresponding axis or spindle. This movement will resume when the PLC sets this mark back low (=0).

If the axis or spindle is moving with other axes, all the axes will stop.

For independent axes and electronic cam, if the PLC sets this signal to (=1), it interrupts the synchronization movement switching to zero speed. The system waits for the signal to deactivate before resuming the execution and the movement from the interruption point.

For independent axes, this signal also stops the monitoring of the synchronization.

AXISPOS(axis)

AXISNEG(axis)

The CNC uses these marks when operating in JOG mode.

If the PLC sets one of these marks high (=1), the CNC will move the relevant axis in the indicated direction: positive (POS) or negative (NEG). The CNC will move the axis at the corresponding feedrate and selected override (%).

The treatment of these marks is similar to the JOG keys of the operator panel.

SERVO(axis)ON

This mark must be high (=1) in order to be able to move the corresponding axis. If the SERVOnON mark is set low (=0) while moving the axis or spindle, the CNC stops the axes and the spindle. It also displays the relevant error message.

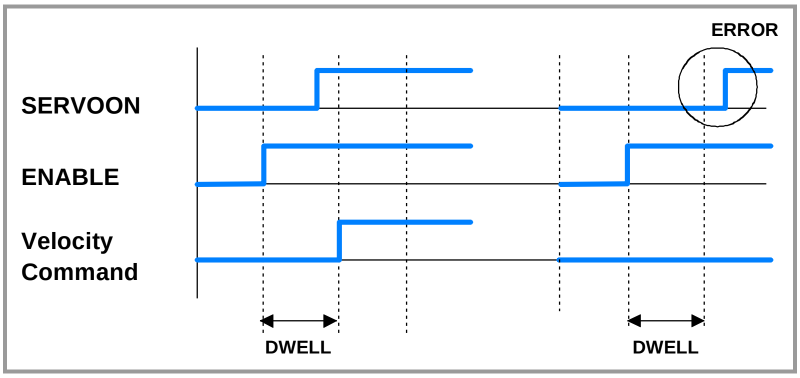
- In order to be able to continuously control the axis, the SERVOON mark must always be high (=1).

  `(there are no errors) AND (axis drive OK) = SERVOnON`

- To control the axis only when they move, use the ENABLE mark. The CNC sets it high (=1) whenever it has to move the axis. See *"ENABLE(axis)"* on page 215.

  `(no errors) AND (drive OK) AND ENABLE = SERVOON`

If the axis moves while locked (meaning SERVOON=0), the CNC stores that displacement as axis lag (following error). When controlling it back (SERVOON=1) it restores its position.



The CNC, after activating the ENABLE mark, waits for a time period indicated by (a.m.p.) "DWELL" before checking that SERVOnON is high (=1). If it is not high, it stops the movement of the axes and the spindle and it displays the error message: "axis locked".

To control the axis only while moving, the (a.m.p.) "DWELL" must be assigned a value greater than 2 PLC cycle scans in order to avoid the error message "axis locked".

DRO(axis)

This mark, together with the relevant "SERVOnON" mark allows the axis or spindle as a dro axis. To do that, the DRO mark must be set high (=1) and SERVOnON low (=0).

When working as a dro, the position loop is open and the following error generated by the displacements ignored.

The programmed movements are executed without making any moves or causing error messages.

When programming an interpolation with other axes, the other ones will move at the corresponding feedrate; but the dro axis will not move.

When the DRO marks goes back low (=0), the axis is no longer a dro axis and it assumes the current position having a following error of "0".

SPENA(axis)

DRENA(axis)

The CNC uses these marks when it communicates with the drive through Sercos®. Whenever the PLC sets one of these marks low (=0) or high (=1), the CNC informs the relevant drive about it via Sercos®.

These marks correspond to the "Speed enable" and "Drive enable" signals of the drive.

The operation of these two signals is described in the drive manual. Nevertheless, remember that:

- Both signals must be initialized low (=0) when powering-up the PLC.
- For the normal operation of the drive, the “Drive Enable” and “Speed Enable” signals must be high (=1). The motor will respond to any changes of velocity command.
- If the Drive Enable input is set low (=0), the power circuit turns off and the motor loses its torque. In this situation, the motor is no longer governed and will turn freely stopping by inertia.
- If the Speed Enable input is set low (=0), the "internal velocity command" of the drive is switched to "0 rpm". The motor breaks while keeping its torque. Once the motor has stopped, the drive's power circuit is turned off and the motor has no torque.

LIM(axis)OFF

If the PLC sets this mark high (=1), the CNC ignores the software travel limits set for the corresponding axis. With `LIM(axis)OFF` = 0, those software limits are no longer ignored.

The software travel limits are set with machine parameters or using functions `G198` and `G199`.

Bear in mind that after a Reset, either from the keyboard or with the `RESETIN` mark, it maintains those travel limits set by functions `G198` and `G199`. These limits are canceled after turning the CNC off or validating the machine parameters.

PARKED(axis)

The PLC sets this mark high (=1) to let the CNC know that the corresponding axis or spindle is parked.

For further information, see the consulting signals for axes and spindles: `PARK` and `UNPARK` described earlier in this chapter. See ***"PARK(axis) UNPARK(axis)"*** on page 217.

LUBRENA(axis)

LUBROK(axis)

These marks together with the axis consulting signal LUBR(axis) must be used to lubricate the axes. See "LUBR(axis)" on page 216.

The LUBRENA mark indicates whether to use this feature (=1) or not (=0). With LUBRENA high (=1), the CNC acts as follows:

1. When the axis has traveled the distance set by (a.m.p.) DISTLUBRI, it sets the LUBR mark high (=1) to let the PLC know that the axis must be lubricated.
2. After lubricating the axis, the PLC sets the LUBROK(axis) mark high (=1) to let the CNC know that the axis has been lubricated.
3. The CNC sets the LUBR mark low (=0) and resets its count to "0".

Be careful not to set the LUBROK mark low (=0) so the feature works properly. Otherwise, its count will always stay at "0".

DIFFCOMP(axis)

This mark is used on Gantry axes to correct the position difference between the master and the slave axes. The slave axis will move until reaching the position of the master axis at the feedrate set by parameter REFEED2.

The correction must be enabled in machine parameter DIFFCOMP and it is applied in the following cases.

- With the up-flank of SERVO*ON if DIFFCOMP is set high.
- With the up-flank of DIFFCOMP if SERVO*ON is set high.

This process can only be interrupted with RESET.

7.9 Spindle modifiable signals

GEAR1, GEAR2, GEAR3, GEAR4

There is one mark for each spindle. The mnemonics for each spindle are the following: Here is an example of the mnemonics for GEAR1; it is the same for the rest of the marks.

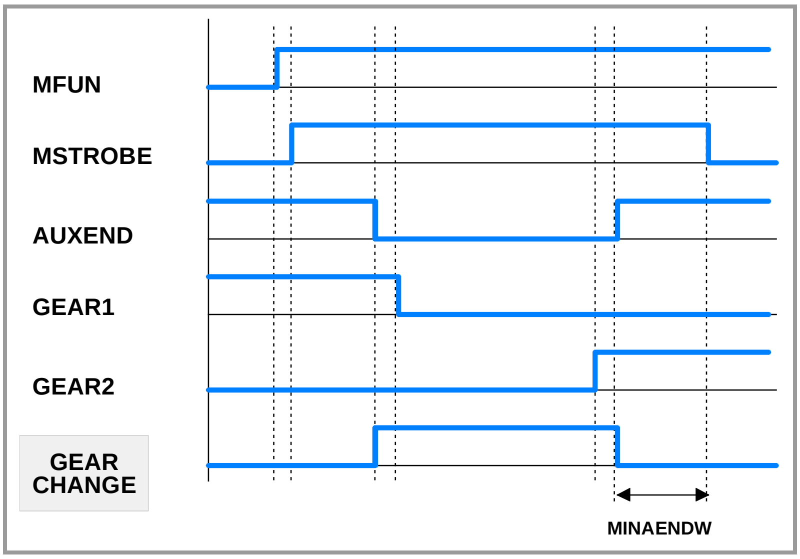
GEAR1SP1 (can also be programmed as GEAR1)

GEAR1SP2 GEAR1SP3 GEAR1SP4

The PLC uses these marks to let the CNC know which spindle range (gear) is currently selected (=1). When requesting a gear change, the CNC informs the PLC about it using auxiliary functions: M41, M42, M43 or M44.

Example of a GEAR1 GEAR2 change

If Gear 2 (M42) is requested while gear 1 is active.



1. The CNC indicates to the PLC the gear requested with MFUN1=42 and sets the MSTROBE mark high (=1).
2. When detecting the request, the PLC sets an internal indicator.

```
DFU MSTROBE AND CPS MFUN* EQ 42 = SET M1002
```

3. The change begins and lets the CNC know by setting AUXEND (=0)

```
NOT M1002 AND <rest of conditions> \
= AUXEND \
= (starts the gear change)
```

During the change, the CNC is "told" that gear 1 is unselected and gear 2 is selected. The active gear indicator GEAR1 through GEAR4 must be set before activating the AUXEND signal.

```
I21 = GEAR1
I22 = GEAR2
```

4. Once the gear change is over, it cancels the indicator (M1002) and it lets the CNC know by setting AUXEND high (=1).

```
(GEAR change completed) = RES M1002
```

Keep the AUXEND mark high (=1) longer than the time period set by (g.m.p.) “MINAENDW” so the CNC cancels the “MSTROBE” mark and concludes the gear change.

SPDLEREV

There is one mark for each spindle. The mnemonics for each spindle are the following:

SPDLREV1 (can also be programmed as SPDLREV)

SPDLREV2 SPDLREV3 SPDLREV4

When the PLC sets this mark high (=1), the CNC reverses the spindle turning direction. To do this, it decelerates and accelerates applying the ramps set by machine parameters.

If an M3 or M4 function is executed while the SPDLEREV mark is high (=1), the spindle will turn in the opposite direction to the one assigned to the function.

PLCCNTL

SANALOG

There is one signal for each spindle. The mnemonics for each spindle are the following:

PLCCNTL1 (can also be programmed as PLCCNTL)

PLCCNTL2 PLCCNTL3 PLCCNTL4

SANALOG1 (can also be programmed as SANALOG)

SANALOG2 SANALOG3 SANALOG4

These signals are used with analog spindles, Sercos in position and Sercos in velocity.

When the PLC sets the PLCCNTL mark high (=1), it indicates that the spindle is directly controlled by the PLC and that the SANALOG register sets the spindle velocity command to be applied. It is used, for example, for oscillating the spindle during a gear change.

- Analog spindle.

  10 V of velocity command correspond to SANALOG = 32767. In other words:

  For 4V, program SANALOG = (4x32767)/10 = 13107

  For -4V, program SANALOG = (-4x32767)/10 = -13107

- Sercos spindle in velocity.

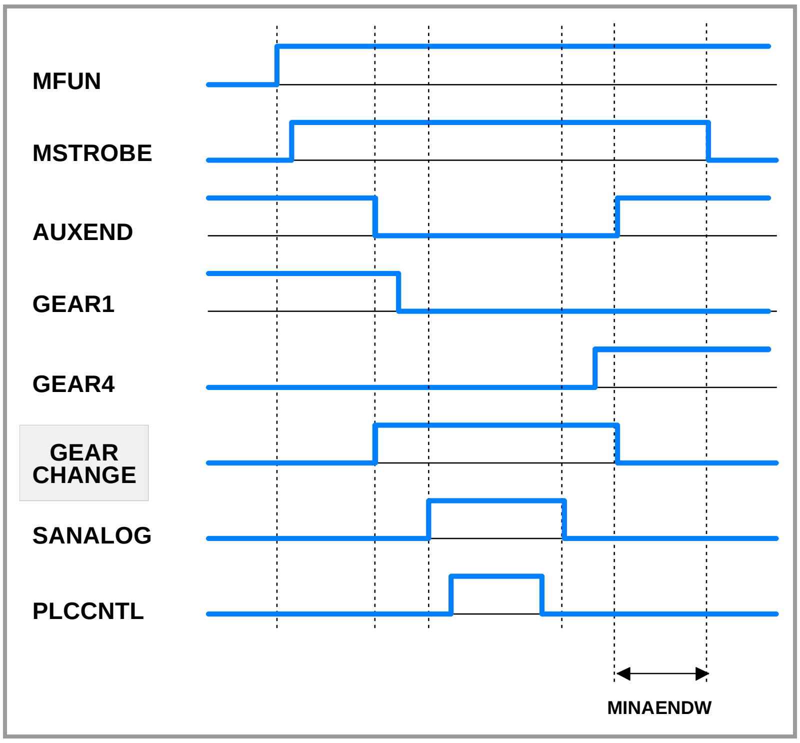
  The command in SANALOG will be given in 0.0001 rpm.

- Sercos spindle in position.

  The command in SANALOG will be given in 0.0001 degrees.

Example similar to the one used for GEAR1 through GEAR4

The spindle oscillation during a gear change is controlled by the PLC. Gear 4 is requested while gear 1 is active.



The example for GEAR1 through GEAR4 signals describes how to detect and carry out the gear change. This example shows how to control the spindle oscillation during a gear change.

The PLC sets SANALOG to the value corresponding to the residual analog voltage and activates the PLCCNTL mark to indicate that the spindle is controlled by the PLC.

When done, the PLCCNTL mark must be set low (=0) and the SANALOG signal must be set to "0".

7.10 Modifiable signals of the independent interpolator

The signal names are generic. Replace the text (axis) with the name or logic axis number.

IRESET(axis)

For movements of an independent axis, if the PLC sets this mark to (=1), it interrupts the instruction in execution and eliminates the instructions pending execution.

For electronic-cam movements, it interrupts the cam synchronization movement switching to zero speed.

It sets the initial conditions at the independent interpolator of the axis.

IABORT(axis)

For movements of an independent axis, if the PLC sets this mark to (=1), the positioning block being executed (if any) stops, eliminating also the rest of the positioning blocks pending execution.

It only affects positioning blocks; neither the pending instructions nor the synchronization movement are eliminated.

7.11 Tool manager modifiable signals

SETTMEM

There is one mark for each magazine. The mnemonics for each magazine are the following.

SETTMEMZ1 (can also be programmed as SETTMEM)

SETTMEMZ2 SETTMEMZ3 SETTMEMZ4

The PLC sets this mark high (=1) to activate the tool manager emergency.

RESTMEM

There is one mark for each magazine. The mnemonics for each magazine are the following.

RESTMEMZ1 (can also be programmed as RESTMEM)

RESTMEMZ2 RESTMEMZ3 RESTMEMZ4

The PLC sets this mark high (=1) to cancel the tool manager emergency.

CUTTINGON

There is one mark for each channel. The mnemonics for each channel are the following.

CUTTINGONC1 (same as CUTTINGON)

CUTTINGONC2 CUTTINGONC3 CUTTINGONC4

When associating a maximum life span to a tool (monitoring), the CNC checks this mark to find out whether the tool is machining (=1) or not (=0).

```
PRG
()= CNCRD (G.GS0, R300, M12)
```

Register R300 shows the status of the G functions

```
AUTOMAT AND INCYCLE AND NOT B0R300 = CUTTINGON
```

If it is in automatic mode (AUTOMAT), a (INCYCLE) block is being executed and the G00 function is not active, then, the tool is considered to be machining.

```
END
```

TREJECT

There is one mark for each channel. The mnemonics for each channel are the following.

TREJECTC1 (can also be programmed as TREJECT)

TREJECTC2 TREJECTC3 TREJECTC4

If the PLC sets this mark high (=1), the CNC interprets that the tool must be rejected.

MZTOCH1

There is one mark for each magazine. The mnemonics for each magazine are the following.

MZTOCH1MZ1 (can also be programmed as MZTOCH1)

MZTOCH1MZ2 MZTOCH1MZ3 MZTOCH1MZ4

Use it with an asynchronous magazine or synchronous with arm.

The PLC must set this mark high (=1) after taking the tool from the magazine to the changer arm 1.

CH1TOSPDL

There is one mark for each magazine. The mnemonics for each magazine are the following.

CH1TOSPDLMZ1 (same as CH1TOSPDL)

CH1TOSPDLMZ2 CH1TOSPDLMZ3 CH1TOSPDLMZ4

Use it with an asynchronous magazine or synchronous with arm.

The PLC must set this mark high (=1) after taking the tool from the changer arm 1 to the spindle.

SPDLTOCH1

There is one mark for each magazine. The mnemonics for each magazine are the following.

SPDLTOCH1MZ1 (same as SPDLTOCH1)

SPDLTOCH1MZ2 SPDLTOCH1MZ3 SPDLTOCH1MZ4

Use it with an asynchronous magazine with one changer arm.

The PLC must set this mark high (=1) after taking the tool from the spindle to the changer arm 1.

SPDLTOCH2

There is one mark for each magazine. The mnemonics for each magazine are the following.

SPDLTOCH2MZ1 (same as SPDLTOCH2)

SPDLTOCH2MZ2 SPDLTOCH2MZ3 SPDLTOCH2MZ4

Use it with an asynchronous magazine or synchronous with arm.

The PLC must set this mark high (=1) after taking the tool from the spindle to the changer arm 2.

CH1TOMZ

There is one mark for each magazine. The mnemonics for each magazine are the following.

CH1TOMZ1 (can also be programmed as CH1TOMZ)

CH1TOMZ2 CH1TOMZ3 CH1TOMZ4

Use it with an asynchronous magazine or synchronous with arm.

The PLC must set this mark high (=1) after taking the tool from the tool changer arm 1 to the magazine.

CH2TOMZ

There is one mark for each magazine. The mnemonics for each magazine are the following.

`CH2TOMZ1` (can also be programmed as `CH2TOMZ`)

`CH2TOMZ2` `CH2TOMZ3` `CH2TOMZ4`

Use it with an asynchronous magazine or synchronous with arm.

The PLC must set this mark high (=1) after taking the tool from the tool changer arm 2 to the magazine.

SPDLTOGR

There is one mark for each channel. The mnemonics for each channel are the following.

`SPDLTOGRC1` (can also be programmed as `SPDLTOGR`)

`SPDLTOGRC2` `SPDLTOGRC3` `SPDLTOGRC4`

Use it with a magazine that admits ground tools.

The PLC must set this mark high (=1) after taking the tool from the spindle to ground.

GRTOSPDL

There is one mark for each channel. The mnemonics for each channel are the following.

`GRTOSPDLC1` (can also be programmed as `GRTOSPDL`)

`GRTOSPDLC2` `GRTOSPDLC3` `GRTOSPDLC4`

Use it with a magazine that admits ground tools.

The PLC must set this mark high (=1) after taking the tool from the ground to the spindle.

MZTOSPDL

There is one mark for each magazine. The mnemonics for each magazine are the following.

`MZTOSPDLMZ1` (can also be programmed as `MZTOSPDL`)

`MZTOSPDLMZ2` `MZTOSPDLMZ3` `MZTOSPDLMZ4`

Use it with a synchronous magazine (without arm).

The PLC must set this mark high (=1) after taking the tool from the magazine to the spindle.

SPDLTOMZ

There is one mark for each magazine. The mnemonics for each magazine are the following.

`SPDLTOMZ1` (can also be programmed as `SPDLTOMZ`)

`SPDLTOMZ2` `SPDLTOMZ3` `SPDLTOMZ4`

Use it with a synchronous magazine (without arm).

The PLC must set this mark high (=1) after taking the tool from the spindle to the magazine.

MZROT

There is one mark for each magazine. The mnemonics for each magazine are the following.

`MZROTMZ1` (can also be programmed as `MZROT`)

`MZROTMZ2` `MZROTMZ3` `MZROTMZ4`

Use it with a turret type magazine

The PLC must set this mark high (=1) when the turret has rotated.

TCHANGEOK

There is one mark for each magazine. The mnemonics for each magazine are the following.

`TCHANGEOKMZ1` (same as `TCHANGEOK`)

`TCHANGEOKMZ2` `TCHANGEOKMZ3` `TCHANGEOKMZ4`

The PLC must set this mark high (=1) when the tool change has ended (`M06`).

MZPOS

There is one register for each magazine. The mnemonics for each magazine are the following.

`MZPOSMZ1` (can also be programmed as `MZPOS`)

`MZPOSMZ2` `MZPOSMZ3` `MZPOSMZ4`

The PLC must indicate the current magazine position in this register.

7.12 Keystroke modifiable signals

KEYLED1, KEYLED2

These registers control the LED's of the operator panel keys.

The user keys "User1" through "User16" are numbered from left to right.

	Bit	Led Key
KEYLED1	0	User1
KEYLED1	1	User2
KEYLED1	2	User3
KEYLED1	3	User4
KEYLED1	4	User5
KEYLED1	5	User6
KEYLED1	6	User7
KEYLED1	7	User8
KEYLED1	8	User9
KEYLED1	9	User10
KEYLED1	10	User11
KEYLED1	11	User12
KEYLED1	12	User13
KEYLED1	13	User14
KEYLED1	14	User15
KEYLED1	15	User16
KEYLED1	16	X+
KEYLED1	17	Y+
KEYLED1	18	Z+
KEYLED1	19	X-
KEYLED1	20	Y-
KEYLED1	21	Z-
KEYLED1	22	7-
KEYLED1	23	Rapid
KEYLED1	24	7+
KEYLED1	25	4+
KEYLED1	26	5+
KEYLED1	27	6+
KEYLED1	28	4-
KEYLED1	29	5-
KEYLED1	30	6-
KEYLED1	31	

	Bit	Led Key
KEYLED2	0	Spdl Ovr +
KEYLED2	1	Spindle clockwise
KEYLED2	2	Spindle posit.
KEYLED2	3	Spindle stop
KEYLED2	4	Spdl Ovr -
KEYLED2	5	Spindle c.clockwise
KEYLED2	6	
KEYLED2	7	
KEYLED2	8	
KEYLED2	9	
KEYLED2	10	
KEYLED2	11	ZERO
KEYLED2	12	
KEYLED2	13	Single Block
KEYLED2	14	
KEYLED2	15	RESET
KEYLED2	16	
KEYLED2	17	
KEYLED2	18	
KEYLED2	19	
KEYLED2	20	
KEYLED2	21	
KEYLED2	22	
KEYLED2	23	
KEYLED2	24	
KEYLED2	25	
KEYLED2	26	
KEYLED2	27	
KEYLED2	28	
KEYLED2	29	
KEYLED2	30	
KEYLED2	31	

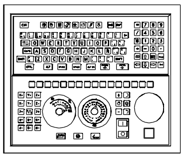
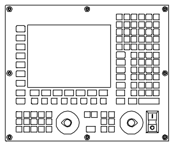
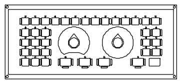
The following instruction changes the status of the led every time the key is pressed.

```
DFU B0KEYBD1 = CPL B0KEYLED1
```

KEYDIS1, KEYDIS2, KEYDIS3

These registers inhibit the operation of the keys individually.

- On the standard keyboard, the user keys "User1" through "User16" are numbered from left to right.
- On the compact keyboard, the keys "User1" through "User6" are located next to the jog keys. They are numbered from left to right and from top down.
- On the flat keyboard, the keys "Jogkey1" through "Jogkey3" are located next to the jog keys. The are numbered from left to right.

	Bit	OP-Panel-H/E	LCD-10K	Jog Panel
				
KEYDIS1	0	User1	User1	User1
KEYDIS1	1	User2	User2	User2
KEYDIS1	2	User3	User3	User3
KEYDIS1	3	User4	User4	User4
KEYDIS1	4	User5	User5	User5
KEYDIS1	5	User6	User6	User6
KEYDIS1	6	User7		User7
KEYDIS1	7	User8		User8
KEYDIS1	8	User9		User9
KEYDIS1	9	User10		User10
KEYDIS1	10	User11		User11
KEYDIS1	11	User12		User12
KEYDIS1	12	User13		User13
KEYDIS1	13	User14		User14
KEYDIS1	14	User15		
KEYDIS1	15	User16		
KEYDIS1	16	X+	1	X
KEYDIS1	17	Y+	2	Y
KEYDIS1	18	Z+	3	Z
KEYDIS1	19	X-	4	4
KEYDIS1	20	Y-	5	5
KEYDIS1	21	Z-	6	6
KEYDIS1	22	7-	-	+
KEYDIS1	23	RAPID	RAPID	RAPID
KEYDIS1	24	7+	+	-
KEYDIS1	25	4+		jogkey1
KEYDIS1	26	5+		jogkey2

	Bit	OP-Panel-H/E	LCD-10K	Jog Panel
KEYDIS1	27	6+		jogkey3
KEYDIS1	28	4-		
KEYDIS1	29	5-		
KEYDIS1	30	6-		
KEYDIS1	31			

	Bit	Key	Bit	Key
KEYDIS2	0	Spindle Ovr+	9	
KEYDIS2	1	Spindle clockwise	10	
KEYDIS2	2	Spindle positioning	11	ZERO
KEYDIS2	3	Spindle stop	12	
KEYDIS2	4	Spindle Ovr-	13	Single block
KEYDIS2	5	Spindle c.clockwise	14	
KEYDIS2	6	START	15	RESET
KEYDIS2	7	STOP	16-31	
KEYDIS2	8			

	Bit	Key	Bit	Key
KEYDIS3	0	0%	16	140%
KEYDIS3	1	2%	17	150%
KEYDIS3	2	4%	18	160%
KEYDIS3	3	10%	19	170%
KEYDIS3	4	20%	20	180%
KEYDIS3	5	30%	21	190%
KEYDIS3	6	40%	22	200%
KEYDIS3	7	50%	23	Handwheel 100
KEYDIS3	8	60%	24	Handwheel 10
KEYDIS3	9	70%	25	Handwheel 1
KEYDIS3	10	80%	26	JOG 1
KEYDIS3	11	90%	27	JOG 10
KEYDIS3	12	100%	28	JOG 100
KEYDIS3	13	110%	29	JOG 1000
KEYDIS3	14	120%	30	JOG 10000
KEYDIS3	15	130%	31	JOG continuous

The following line inhibits the [X+] key.

```
( ) = B16 KEYDIS1
```

When selecting one of the inhibited positions of the feedrate override, the CNC will take the value of the lowest position allowed. If all of them are inhibited, it will take the value of 0%.

For example, being allowed only the 110% and 120% positions, if the 50% position is selected, the CNC will take the value of 0%.

7.13 Alphabetical listing of marks (M) and registers (R)

TOOL AND MAGAZINE MANAGEMENT

8

To properly configure the tool magazines and the tool change, you must:

- Set the machine parameters.
- Set the tool table and the tool magazine table.
- Write the PLC program.
- Program the subroutine associated with the tool and with the `M06` function

The machine parameters define the number of tool magazines available and their characteristics. Up to four magazines may be used and each may be of a different type. See ***"8.1 Types of tool magazine"*** on page 252.

When creating the PLC program and the subroutine associated with the tool and with the `M06` function, you must bear in mind the number of magazines and channels available. For the communication between the tool manager and the PLC, each channel and each magazine has its own group of marks and registers.

The PLC maneuver will be different depending on the type of tool magazine. Later sections of this chapter show an example for each type of magazine.

About the magazines.

The CNC can have up to four magazines and each may be of a different type. Each magazine has its own configuration parameters. See ***"2.7 Machine parameters for the magazine"*** on page 101.

The number of magazines is independent from the number of spindles and channels available. A magazine is not associated with any particular channel or spindle; i.e. a magazine may be shared by several channels and a channel can request tools from different magazines.

The access to the magazine depends on the mechanical configuration of the machine; in other words, to the physical possibility of the machine to access the magazines.

All the magazines can carry out tool changes simultaneously. However, one magazine can only be involved in a tool change process. If from one channel, one wishes to pick up or leave a tool in a magazine already involved in a tool change, the tool manager will wait for the tool change to be done before attending to the new request.

Two magazines may be involved in a tool change. The magazine receiving the tool and the magazine from where the new tool is picked up may be different.

About the tools

Each tool is identified by its number, that is unique for the whole system; it cannot be repeated in different magazines nor in ground tools.

The list of available tools is saved in the tool table, the only one for the whole system. This table indicates the position and the magazine where each tool is located, whether it is a ground tool or the tool is active in a channel.

The tools are always stored in the same magazine. When doing a tool change, it is always stored in the magazine from which it was picked up.

Ground tools

A ground tool is a tool that is not stored in any magazine and is loaded manually when requested. Ground tools are also defined in the tool table, but they are not associated with any magazine position.

Ground tool loading and unloading is global to the system; it is not associated with any particular magazine or channel.

Tool manager

The CNC has a tool magazine management that knows at all times the location of each tool. When requesting a tool change or tool search, the tool manager "tells" the PLC the operations to be carried out.

- Take a tool from the magazine and insert it in the spindle.
- Leave the tool of the spindle in the magazine and take another one.
- Leave the spindle tool on the ground.
- Etc.

Depending on the type of magazine and the operation requested, several actions may be needed sometimes. For example, in some magazines, to take a tool from the spindle to the main magazine, the tool must be carried from the spindle to the changer arm and then from the changer arm to the magazine.

It is up to the PLC to control those movements. It must inform the manager about the actions carried out so it updates its information.

For the communication between the tool manager and the PLC, each channel and magazine has its own group of marks and registers. See ***"7.5 Tool manager consulting signals"*** on page 223. See ***"7.11 Tool manager modifiable signals"*** on page 241.

Subroutines associated with the tool change

There are two subroutines associated with the tool change.

Subroutines associated with the tool.

The subroutine associated with the tool is executed automatically every time a T function (tool selection) is executed.

There is one subroutine in each channel.

Subroutines associated with the M06 function.

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

It is recommended to set this function in the "M" function table so it executes the subroutine for the tool changer installed on the machine.

This subroutine is common to the whole system.

Within both subroutines, the "modal" condition of the canned cycles no longer has any influence. This characteristic is restored when the execution of the subroutine ends.

8.1 Types of tool magazine

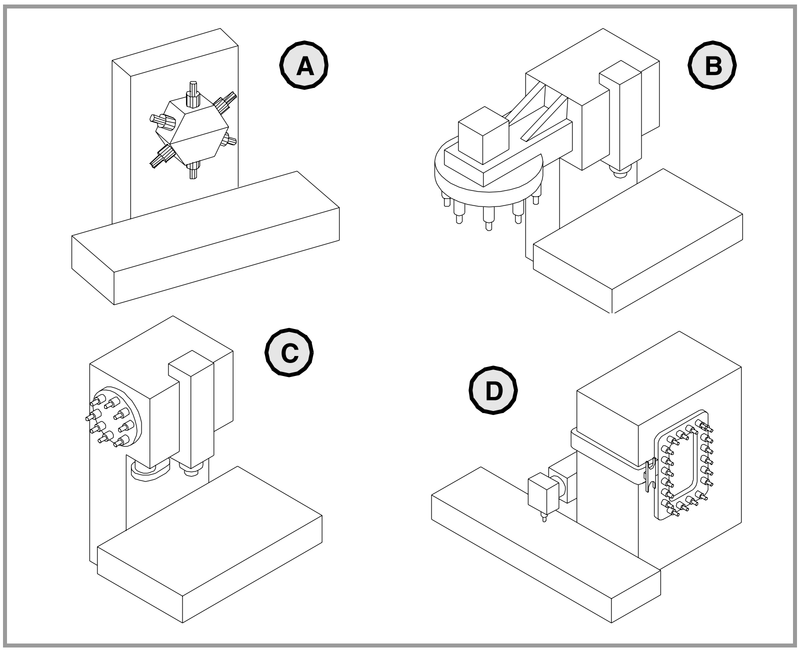
Tool magazines may be divided into 4 large groups:

A. Turret type.

B. Synchronous magazine without changer arm.

C. Synchronous magazine with changer arm (1 or 2 holders). The arm cannot move while machining the part.

D. Asynchronous magazine. It has a tool changer arm of independent movements that may be moved while machining.



"Random" or "Non-Random" magazine

Depending on how the tools are stored in the magazine during a tool change, the magazine may be either random or non-random. In a random magazine, the tool may occupy any position whereas in a non-random magazine, the tools must always occupy the same position.

In any case and even when all the magazines are random, the tools are always stored in the same magazine from which they were picked up.

"Cyclic" or "Non-Cyclic" magazine

A cyclic magazine requires a tool change command `M06` after searching a tool and before searching the next one. In a non-cyclic magazine, it is possible to perform several consecutive tool searches without necessarily having to change the tool.

8.2 Tool table, active tool table and tool magazine table

After setting the machine parameters for the tool magazines, define the tool table and then the magazine table.

Tool table

The tool table must contain, among other things, the geometry, type of monitoring and tool size. All the tools must be defined, including the ground ones.

The tool manager inserts the special tools always in the same tool pocket regardless of the number of pockets they occupy.

Tool magazine table

There is one table for each magazine. Each table shows which tool is in each pocket and on each holder of the changer arm (if any).

Although the magazine table may be initialized manually, it is up to the tool manager to dynamically update all its data.

Example:

Having 10 tools and a 10-position magazine. The tools are small, except T2 that is too large and T4 that has a special size (0 to the left and 1 to the right).

Load the tools one by one in the magazine using the corresponding softkey.

- T1 in position 1.
- T2 in position 3. Verify that it cannot go in position 2 because the tool is too large
- T3 in position 5. T2 occupies positions 2-3-4.
- T4 in position 6. T4 occupies positions 6-7.
- T5 in position 8.
- T6 in position 9.
- T7 in position 10.

T8, T9 and T10 do not fit in the magazine; therefore, they will be ground tools.

Active-tools table

The active-tools table shows which tools are active in the spindles.

8.3 Communication between manager and PLC

For the communication between the tool manager and the PLC, each channel and magazine has its own group of marks and registers. See *"7.5 Tool manager consulting signals"* on page 223. See *"7.11 Tool manager modifiable signals"* on page 241.

The CNC maintains compatibility with the marks and registers of previous versions whose mnemonics do not refer to any particular channel or magazine.

From software version V2.00 on, these signals refer to the first channel or magazine accordingly.

The communication between the manager and the PLC takes place in two stages; when executing the T function and when executing the M06 function.

- When executing the T function, the CNC lets the tool manager know about it.

  The tool manager sends a command to the PLC to select the next tool in the magazine (if possible).

  The CNC continues with the execution of the program without waiting for the tool manager to complete the operation.

- When executing the M06 function, its associated subroutine is called upon. The M06 function must also be programmed inside that subroutine so the CNC "tells" the tool manager to start making the tool change.

  The tool manager sends a command to the PLC to make the change.

  The CNC waits for the tool manager to complete the operation before continuing with the execution of the program.

Considerations and recommendations

The management of the tool change should be included in the subroutine associated with the M06 and leave the control of the external devices up to the PLC.

Use the auxiliary functions to govern the various devices (magazine rotation, magazine movement, tool changer arm, etc.) from the M06 subroutine.

On asynchronous magazines (changer arm with independent movements) when the change implies leaving a tool in the magazine, the TCHANGEOK mark may be activated so the CNC goes on executing the program while the tool is being taken to the magazine.

8.3.1 Manager --> PLC communication

The manager uses the following registers and marks to inform the PLC about the operations it must carry out. Some signals are per channel whereas others are per magazine. See *"7.5 Tool manager consulting signals"* on page 223.

The following table shows the mnemonics for each mark (M) or register (R) in each channel or magazine.

M/R	Channel ·1·	Channel ·2·	Channel ·3·	Channel ·4·
M	TMOPSTROBE TMOPSTROBEC1	TMOPSTROBEC2	TMOPSTROBEC3	TMOPSTROBEC4
R	TMOPERATION TMOPERATIONC1	TMOPERATIONC2	TMOPERATIONC3	TMOPERATIONC4
R	MZIDC1	MZIDC2	MZIDC3	MZIDC4

M/R	Magazine ·1·	Magazine ·2·	Magazine ·3·	Magazine ·4·
R	LEAVEPOS LEAVEPOSMZ1	LEAVEPOSMZ2	LEAVEPOSMZ3	LEAVEPOSMZ4
R	TAKEPOS TAKEPOSMZ1	TAKEPOSMZ2	TAKEPOSMZ3	TAKEPOSMZ4
R	NEXTPOS NEXTPOSMZ1	NEXTPOSMZ2	NEXTPOSMZ3	NEXTPOSMZ4

TMOPSTROBE

The channel tool manager sets this mark high (=1) to let the PLC know that it must execute the operation indicated by the `TMOPERATION` mark of the channel.

LEAVEPOS

This register indicates the magazine position to leave the tool.

TAKEPOS

This register indicates the magazine position of the tool to be taken.

NEXTPOS

This register indicates the magazine position occupied by the next tool.

MZID

This register indicates which magazine contains the tool requested by the channel.

When two magazines are involved in a tool change, the lower portion of this register indicates the destination magazine for the tool and the higher portion the source magazine for the tool.

TMOPERATION

This register indicates the type of operation to be carried out by the tool manager.

0	Do nothing.
1	Take a tool from the magazine and insert it in the spindle.
2	Leave the spindle tool in the magazine.
3	Insert a ground tool in the spindle.
4	Leave the spindle tool on the ground.
5	Leave the spindle tool in the magazine and take another one from the same magazine.

6	Leave the spindle tool in the magazine and take another one from ground.
7	Leave the spindle tool on the ground and take another one from the magazine.
8	Leave the spindle tool on the ground and take another one from ground.
9	Take a ground tool to the magazine going through the spindle.
10	Take a tool from the magazine and leave on the ground going through the spindle.
11	Orient the magazine.
12	Leave the spindle tool in the magazine and take another one from the same magazine. Special for a synchronous magazine in the following cases: • Non-random having a tool changer arm with two claws. • Random when having special tools.
13	Orienting two magazines.
14	Leave the spindle tool in a magazine and take another one from another magazine.

8.3.2 PLC --> Manager communication

The PLC uses the following marks to inform the manager about the operations that it has carried out. There is one group of marks for each magazine. See *"7.11 Tool manager modifiable signals"* on page 241.

The PLC, depending on the type of magazine, must take some actions to execute the operations requested by the manager. After ending each one of them, it must activate certain marks to inform the manager that the action has been completed. The manager sets them back to "0" once they are read.

The following table shows the mnemonics for each mark (M) or register (R) in each channel or magazine.

M/R	Magazine ·1·	Magazine ·2·	Magazine ·3·	Magazine ·4·
M	MZTOCH1 MZTOCH1MZ1	MZTOCH1MZ2	MZTOCH1MZ3	MZTOCH1MZ4
M	CH1TOSPDL CH1TOSPDLMZ1	CH1TOSPDLZ2	CH1TOSPDLMZ3	CH1TOSPDLMZ4
M	SPDLTOCH1 SPDLTOCH1MZ1	SPDLTOCH1MZ2	SPDLTOCH1MZ3	SPDLTOCH1MZ4
M	SPDLTOCH2 SPDLTOCH2MZ1	SPDLTOCH2MZ2	SPDLTOCH2MZ3	SPDLTOCH2MZ4
M	CH1TOMZ CH1TOMZ1	CH1TOMZ2	CH1TOMZ3	CH1TOMZ4
M	CH2TOMZ CH2TOMZ1	CH2TOMZ2	CH2TOMZ3	CH2TOMZ4
M	SPDLTOGR SPDLTOGRMZ1	SPDLTOGRMZ2	SPDLTOGRMZ3	SPDLTOGRMZ4
M	GRTOSPDL GRTOSPDLMZ1	GRTOSPDLMZ2	GRTOSPDLMZ3	GRTOSPDLMZ4
M	MZTOSPDL MZTOSPDLMZ1	MZTOSPDLMZ2	MZTOSPDLMZ3	MZTOSPDLMZ4
M	MZTOSPDL MZTOSPDLMZ1	MZTOSPDLMZ2	MZTOSPDLMZ3	MZTOSPDLMZ4
M	MZROT MZROTMZ1	MZROTMZ2	MZROTMZ3	MZROTMZ4
M	TCHANGEOK TCHANGEOKMZ1	TCHANGEOKMZ2	TCHANGEOKMZ3	TCHANGEOKMZ4
R	MZPOS MZPOSMZ1	MZPOSZ2	MZPOSMZ3	MZPOSMZ4

MZTOCH1

Use it with an asynchronous magazine or synchronous with arm.

The PLC must set this mark high (=1) after taking the tool from the magazine to the changer arm 1.

CH1TOSPDL

Use it with an asynchronous magazine or synchronous with arm.

The PLC must set this mark high (=1) after taking the tool from the changer arm 1 to the spindle.

SPDLTOCH1	Use it with an asynchronous magazine with one changer arm. The PLC must set this mark high (=1) after taking the tool from the spindle to the changer arm 1.
SPDLTOCH2	Use it with an asynchronous magazine or synchronous with arm. The PLC must set this mark high (=1) after taking the tool from the spindle to the changer arm 2.
CH1TOMZ	Use it with an asynchronous magazine or synchronous with arm. The PLC must set this mark high (=1) after taking the tool from the tool changer arm 1 to the magazine.
CH2TOMZ	Use it with an asynchronous magazine or synchronous with arm. The PLC must set this mark high (=1) after taking the tool from the tool changer arm 2 to the magazine.
SPDLTOGR	Use it with a magazine that admits ground tools. The PLC must set this mark high (=1) after taking the tool from the spindle to ground.
GRTOSPDL	Use it with a magazine that admits ground tools. The PLC must set this mark high (=1) after taking the tool from the ground to the spindle.
MZTOSPDL	Use it with a synchronous magazine (without arm). The PLC must set this mark high (=1) after taking the tool from the magazine to the spindle.
SPDLTOMZ	Use it with a synchronous magazine (without arm). The PLC must set this mark high (=1) after taking the tool from the spindle to the magazine.
MZROT	Use it with a turret-type magazine and with a synchronous magazine. The PLC must set this mark high (=1) when the turret has rotated. In the synchronous magazine, it is used to optimize the change by orienting the magazine while machining. The PLC must set this mark to (=1) to indicate that the operation has been completed, whether it has been oriented or not.
TCHANGEOK	On asynchronous magazines (with changer arm of independent movements) the following mark may be activated so the CNC goes on executing the program while the tool is being taken to the magazine. The PLC must set this mark high (=1) to "tell" the manager to go on executing the program while the tool is being taken to the magazine.
MZPOS	On random magazines, magazine orientations may be optimized if the manager knows the position selected at all times. The PLC must indicate the current magazine position in this register. When not using this register, the PLC must set it to "0".

8.3.3 Manager Emergency

The manager may be set in an Emergency state when a malfunction occurs (the PLC has executed the wrong action, incomplete tool change, etc.) or if the PLC activates the emergency.

The PLC marks related to manager emergency are the following: There is one group of marks for each magazine.

The following table shows the mnemonics for each mark (M) in each channel or magazine.

M/R	Magazine ·1·	Magazine ·2·	Magazine ·3·	Magazine ·4·
M	SETTMEM SETTMEMZ1	SETTMEMZ2	SETTMEMZ3	SETTMEMZ4
M	RESTMEM RESTMEMZ1	RESTMEMZ2	RESTMEMZ3	RESTMEMZ4
M	TMINEM TMINEMZ1	TMINEMZ2	TMINEMZ3	TMINEMZ4

SETTMEM The PLC sets this mark high (=1) to activate the tool manager emergency.

RESTMEM The PLC sets this mark high (=1) to cancel the tool manager emergency.

TMINEM The CNC sets this mark high (=1) to inform the PLC that an emergency has occurred at the tool manager.

To generate an emergency at the manager from the PLC:

1. Activate the emergency.

```
DFU (error condition) = SET SETTMEM
```

2. Make sure that the emergency has occurred before canceling it.

```
TMINEM AND DFU (removal condition) = SET RESTMEM
```

The manager sets the `SETMEM` and `RESTMEM` signals low (=0).

8.3.4 Tool monitoring

The PLC marks related to tool monitoring are the following. There is one group of marks for each channel.

The following table shows the mnemonics for each mark (M) in each channel or magazine.

M/R	Channel ·1·	Channel ·2·	Channel ·3·	Channel ·4·
M	CUTTINGON CUTTINGONC1	CUTTINGONC2	CUTTINGONC3	CUTTINGONC4
M	TREJECT TREJECTC1	TREJECTC2	TREJECTC3	TREJECTC4
M	TWORNOUT TWORNOUTC1	TWORNOUTC2	TWORNOUTC3	TWORNOUTC4

The PLC marks related to tool monitoring are:

CUTTINGON

When a tool is assigned a maximum life span (monitoring), the CNC checks this mark in order to know whether the tool is machining (=1) or not (=0).

It is usually considered to be machining when the following conditions are met:

- The spindle is turning (M3 or M4) or when the tool is threadcutting and the 0% of feedrate is not selected.
- The automatic operating mode is selected, there is a block in execution and the G00 function is not active.
- The execution is not interrupted.

TREJECT

If the PLC sets this mark high (=1), the CNC interprets that the tool must be rejected.

TWORNOUT

The CNC sets this mark high (=1) to "tell" the PLC that the tool has been rejected because it is worn out (real life > maximum life span).

8.4 Variables related to tool magazine management

The variables associated with the magazine that are involved in the tool magazine management are the following. There is one group of variables for each channel. Replace the [n] character with the channel number, maintaining the brackets.

V.[n].TM.MZMODE

Operating mode of the tool magazine manager.

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

V.[n].TM.MZSTATUS

Tool manager status.

0	Normal operation.
1	An error has occurred.
2	An error has occurred; but it will wait for the current maneuver to be completed.
4	Emergency.

V.[n].TM.MZRUN

Tool manager running.

0	There is no sequence in execution.
1	There is a sequence in execution.

V.[n].TM.MZWAIT

Tool manager executing a maneuver. It indicates whether to wait or not for the maneuver to end.

0	No need to wait.
1	It has to wait.

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.

8.5 Tool loading and unloading from the magazines

Tool loading and unloading from the magazine

Each magazine table has softkeys for initializing, loading and unloading tools in the magazine either manually or automatically. Refer to the operation manual.

The tools may also be loaded and unloaded in the magazine by program or in MDI mode.

Tool loading and unloading from the spindle

The tools must always be loaded and unloaded in the spindle and on the changer arm in manual mode from the magazine table. Refer to the operation manual.

Loading the magazine by program or in MDI mode

The tools are loaded in the magazine by taking them one by one from ground to the magazine going through the spindle.

Set the variable: `V.TM.MZMODE=1` to "tell" the tool manager that the magazine loading mode has be chosen.

If a `T1 M6` is executed next, the tool manager interprets that `T1` must be taken from ground to the magazine going through the spindle and it will let the PLC know by setting `TMOPERATION=9`.

The subroutine associated with the `M06` function and the PLC program must contain the maneuver needed to make the tool change.

After loading the tool, the tool manager updates the magazine table.

Unloading the magazine by program or MDI

The tools are unloaded from the magazine taking them one by one from the magazine to the ground going through the spindle.

Set variable: `V.TM.MZMODE=2` to "tell" the tool manager that the magazine unloading mode has been chosen.

If a `T1 M6` is executed next, the tool manager interprets that `T1` must be taken from the magazine to ground going through the spindle and it will let the PLC know by setting `TMOPERATION=10`.

The subroutine associated with the `M06` function and the PLC program must contain the maneuver needed to make the tool change.

After unloading the tool, the tool manager updates the magazine table.

Placing a tool in the spindle by program or in MDI mode

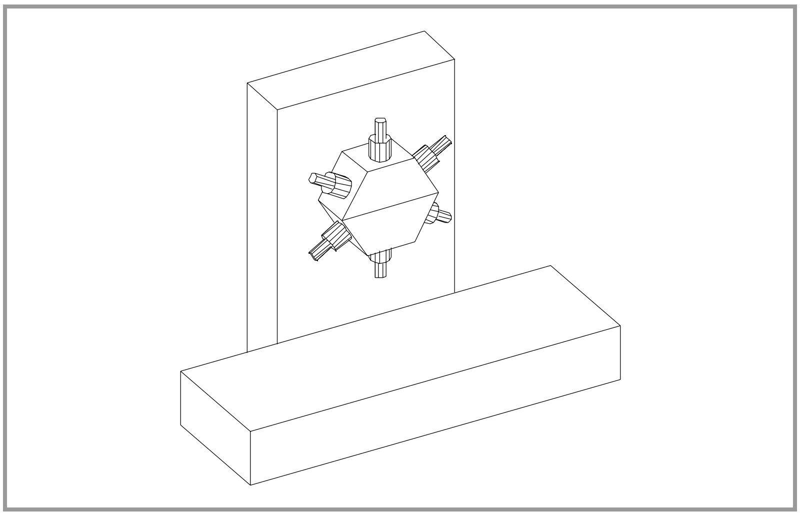
Set variable: `V.TM.MZMODE=0` to "tell" the tool manager that the normal mode has been chosen.

If `T1 M6` is executed next, the tool manager checks whether there is already a tool in the spindle (to remove it first) and whether the requested tool is in the magazine or it has to be picked up from the ground. In any case, it lets the PLC know which operation it must carry out by setting `TMOPERATION` to the proper value.

The subroutine associated with the `M06` function and the PLC program must contain the maneuver needed to make the tool change.

After placing the tool, the tool manager updates the magazine table.

8.6 Turret type



When selecting a tool, the subroutine associated with the T executes the M06 function (tool change), machining stops and the tool manager sends to the PLC the code for the operation to be carried out.

The management of the tool change should be included in the subroutine associated with the M06 and leave the control of the external devices up to the PLC.

This type of magazine does not admit ground tools when it is in normal mode; but it does when the magazine is in load/download mode.

T0 cannot be programmed for this type of magazine.

8.6.1 Values of the TMOPERATION and marks to be activated by the PLC

The possible TMOPERATION values in this type of magazine and the associated marks and registers are the following:

TMOPERATION=1

Assume the tool as active tool.

TAKEPOS Indicates the position of the tool to be picked up.

When the operation is completed, activate the mark MZTOSPDL.

TMOPERATION=11

Orient the turret.

When the operation is completed, activate the mark MZROT.

Summary

The values sent by the tool manager to the PLC in each operation and the marks to be activated by the PLC are the following:

TM ==> PLC			PLC ==> TM
TMOPERATION	TAKEPOS	LEAVEPOS	
1	#	0	MZTOSPDL
11	0	0	MZROT

The TAKEPOS and LEAVEPOS values indicate the following:

0 Do nothing.

Magazine position number.

-4 Ground position.

Application example

It shows the functions executed by the CNC, the values sent by the tool manager to the PLC in each operation and the marks to be activated by the PLC in each case.

CNC	TM ==> PLC			PLC ==> TM
	TMOPERATION	TAKEPOS	LEAVEPOS	
T1 M6	11 1	0 1	0 0	MZROT MZTOSPDL
T2 M6	11 1	0 1	0 0	MZROT MZTOSPDL

8.6.2 Communication between the PLC and the M06 subroutine

The communication between the PLC and the M06 subroutine takes place using a series of generic marks and registers. The program of the M06 subroutine offered as an example uses the following marks and registers.

Communication between the PLC and the M06 subroutine

Registers used by the PLC to transfer information to the M06 subroutine:

R101 Type of operation requested by the tool manager (value of TMOPERATION).

Communication between the PLC and the M06 subroutine

Marks activated by the M06 subroutine so the PLC activates the relevant mark of the tool manager.

M1111 MZROT

The magazine has rotated already.

M functions at the PLC

M functions that imply movements controlled by the PLC:

M111 Select at the magazine the pocket indicated by TAKEPOS.

Set the M function with "before-before" synchronization so the program continues after the M function is completed.

8.6.3 Program of the M06 subroutine

```
%L SUB_MZ_ROT
    M111
        Auxiliary function to execute an action.
    V.PLC.M[1111]=1
        MZROT mark to the tool manager.
#RET

%SUB_M6.nc
M6
        Order the tool manager to start the tool change.
$IF V.PLC.R[101] == 1
    LL SUB_MZ_ROT
$ENDIF
        If type of operation=1, execute the subroutine.

$WHILE V.TM.MZWAIT == 1
$ENDWHILE
        Wait for the tool manager.
#RET
        End of M06 subroutine.
```

8.6.4 Basic PLC programming

When executing the M06 function

When executing the M06 function, the tool manager sends to the PLC, in the TMOPERATION register, the code for the operation to be carried out.

```
DFU TMOPSTROBE = MOV TMOPERATION R101
```

This instruction transfers the TMOPERATION value to register R101 so it is managed by the M06 subroutine.

Every time the M06 subroutine ends the action, it lets the PLC know so it activates the relevant mark of the tool manager.

```
DFU M1111 = SET MZROT
DFD MZROT = RES M1111
```

The magazine has rotated already.

Define the auxiliary function M111 to select the magazine position indicated by TAKEPOS. The auxiliary function will conclude after executing the requested movement.

Manager emergency signal

Treatment of the tool manager emergency signal.

```
DFU B11KEYBD1 = SET SETTMEM
```

Pressing the USER12 key activates the emergency.

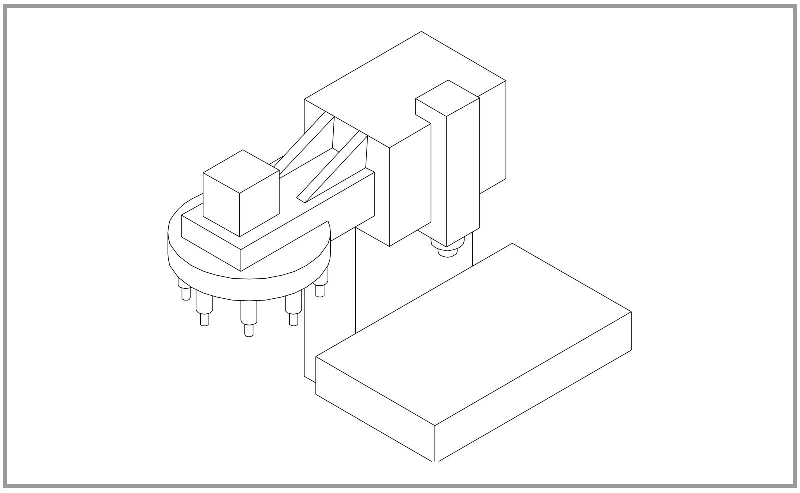
```
TMINEM = B11KEYLED1
```

The lamp of the USER12 key turns on when there is an emergency.

```
TMINEM AND DFU B12KEYBD1 = SET RESTMEM
```

Pressing the USER13 key removes the emergency.

8.7 Synchronous magazine without changer arm



The communication between the tool manager and the PLC takes place in two stages.

- When executing the T function, the CNC lets the tool manager know about it.

  The tool manager sends a command to the PLC to select the next tool in the magazine (if possible).

  The CNC continues with the execution of the program without waiting for the tool manager to complete the operation.

- When executing the M06 function, its associated subroutine is called upon.

  The M06 function must also be programmed inside that subroutine so the CNC "tells" the tool manager to start making the tool change.

  The tool manager sends a command to the PLC to make the change.

  The CNC waits for the tool manager to complete the operation before continuing with the execution of the program.

The management of the tool change should be included in the subroutine associated with the M06 and leave the control of the external devices up to the PLC.

Use the auxiliary functions to govern the various devices (magazine rotation, magazine movement, tool changer arm, etc.) from the M06 subroutine.

8.7.1 Values of the TMOPERATION and marks to be activated by the PLC

The possible TMOPERATION values in this type of magazine and the marks and registers associated with each case are:

TMOPERATION=1

Take a tool from the magazine and insert it in the spindle.

TAKEPOS	Position occupied by the tool.

When the operation is completed, activate the mark MZTOSPDL.

TMOPERATION=2

Leave the spindle tool in the magazine.

LEAVEPOS	Position to leave the tool.

When the operation is completed, activate the mark SPDLTOMZ.

TMOPERATION=3

Insert a ground tool in the spindle.

TAKEPOS=-4	Pick up the ground tool.

When the operation is completed, activate the mark GRTOSPDL.

TMOPERATION=4

Leave the spindle tool on the ground.

LEAVEPOS=-4	Leave the tool on the ground.

When the operation is completed, activate the mark SPDLTOGR.

TMOPERATION=5

Leave the spindle tool in the magazine and take another one from the magazine.

TAKEPOS	Position occupied by the tool.
LEAVEPOS	Position to leave the tool.

1. Leave the tool of the spindle in the magazine and activate the mark SPDLTOMZ.
2. Leave the tool of the magazine in the spindle and activate the mark MZTOSPDL.

TMOPERATION=6

Leave the spindle tool in the magazine and take another one from ground.

TAKEPOS=-4	Pick up the ground tool.
LEAVEPOS	Position to leave the tool.

1. Leave the tool of the spindle in the magazine and activate the mark SPDLTOMZ.
2. Leave the ground tool in the spindle and activate the mark GRTOSPDL.

TMOPERATION=7

Leave the spindle tool on the ground and take another one from the magazine.

TAKEPOS	Position occupied by the tool.
LEAVEPOS=-4	Leave the tool on the ground.

1. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.
2. Leave the tool of the magazine in the spindle and activate the mark MZTOSPDL.

TMOPERATION=8 Leave the spindle tool on the ground and take another one from ground.

TAKEPOS=-4	Pick up the ground tool.
LEAVEPOS=-4	Leave the tool on the ground.

1. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.
2. Leave the ground tool in the spindle and activate the mark GRTOSPDL.

TMOPERATION=9 Take a ground tool to the magazine going through the spindle.

TAKEPOS=-4	Pick up the ground tool.
LEAVEPOS	Position to leave the tool.

1. Leave the ground tool in the spindle and activate the mark GRTOSPDL.
2. Leave the tool of the spindle in the magazine and activate the mark SPDLTOMZ.

TMOPERATION=10 Take a tool from the magazine and leave on the ground going through the spindle.

TAKEPOS	Position occupied by the tool.
LEAVEPOS=-4	Leave the tool on the ground.

1. Leave the tool of the magazine in the spindle and activate the mark MZTOSPDL.
2. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.

TMOPERATION=11 In general, it is an optimization of the change that permits orienting the magazine while machining.

Activate the MZROT mark to indicate that the operation has been completed, whether it has been oriented or not.

TMOPERATION=12 Same as TMOPERATION=5. Only for random magazines and when having special tools.

TMOPERATION=13 In general, it is an optimization of the change that permits orienting two magazines while machining.

Activate the MZROT mark in both magazines to indicate that the operation has been completed, whether it has been oriented or not.

NEXTPOS	Position to leave the tool.
TAKEPOS	Position of the tool to be picked up.
MZID	Destination magazine to leave the tool and source magazine from which to pick up the new one.

TMOPERATION=14 Leave the spindle tool in a magazine and take a tool from another magazine.

TAKEPOS	Position of the tool to be picked up.
LEAVEPOS	Position to leave the tool.
MZID	Destination magazine to leave the tool and source magazine from which to pick up the new one.

1. Leave the tool of the spindle in the magazine and activate the mark SPDLTOMZ.
2. Leave the tool of the magazine in the spindle and activate the mark MZTOSPDL.

Summary

The values sent by the tool manager to the PLC in each operation and the marks to be activated by the PLC are the following:

TM ==> PLC		
TMOPERATION	TAKEPOS	LEAVEPOS
1	#	0
2	0	#
3	-4	0
4	0	-4
5	#	#
6	-4	#
7	#	-4
8	-4	-4
9	-4	#
10	#	-4
11	0	0
12	#	#
13	0	0
14	0	#

PLC ==> TM
MZTOSPDL
SPDLTOMZ
GRTOSPDL
SPDLTOGR
SPDLTOMZ + MZTOSPDL
SPDLTOMZ + GRTOSPDL
SPDLTOGR + MZTOSPDL
SPDLTOGR + GRTOSPDL
GRTOSPDL + SPDLTOMZ
MZTOSPDL + SPDLTOGR
MZROT
SPDLTOMZ + MZTOSPDL
MZROT + MZROT
SPDLTOMZ + MZTOSPDL

The TAKEPOS and LEAVEPOS values indicate the following:

0	Do nothing.
#	Magazine position number.
-4	Ground position.

Application example

Assuming that there is no tool in the spindle, it shows the functions executed by the CNC, the values sent by the tool manager to the PLC in each operation and the marks to be activated by the PLC in each case.

It is a non-random magazine, each tool occupies the position of its own number and TG7, TG8 and TG9 are ground tools.

CNC	TM ==> PLC		
	TMOPERATION	TAKEPOS	LEAVEPOS
T1 M6	11 1	0 1	0 0
T2 M6	11 5	0 2	0 1
TG7 M6	11 6	0 -4	0 2
TG8 M6	11 8	0 -4	0 -4
T3 T4 M6	11 11 7	0 0 4	0 0 -4
T0 M6	11 2	0 0	0 4
TG9 M6	11 3	0 -4	0 0
T0 M6	11 4	0 0	0 -4

PLC ==> TM
MZROT MZTOSPDL
MZROT SPDLTOMZ + MZTOSPDL
MZROT SPDLTOMZ + GRTOSPDL
MZROT SPDLTOGR + GRTOSPDL
MZROT MZROT SPDLTOGR + MZTOSPDL
MZROT SPDLTOMZ
MZROT GRTOSPDL
MZROT SPDLTOGR

8.7.2 Communication between the PLC and the M06 subroutine

The communication between the PLC and the M06 subroutine takes place using a series of generic marks and registers. The program of the M06 subroutine offered as an example uses the following marks and registers.

Communication between the PLC and the M06 subroutine

Registers used by the PLC to transfer information to the M06 subroutine:

R101 Type of operation requested by the tool manager (value of TMOPERATION).

Communication between the PLC and the M06 subroutine

Marks activated by the M06 subroutine so the PLC activates the relevant mark of the tool manager.

M1107 SPDLTOGR

The tool has been taken from the spindle to ground.

M1108 GRTOSPDL

The tool has been taken from ground to the spindle.

M1109 MZTOSPDL

The tool has been taken from the magazine to the spindle.

M1110 SPDLTOMZ

The tool has been taken from the spindle to the magazine.

M functions at the PLC

M functions that imply movements controlled by the PLC:

M109 Select in the magazine the position indicated by TAKEPOS and insert the tool in the spindle.

M110 Select in the magazine the position indicated by LEAVEPOS and leave the spindle tool.

Set all the M functions with "before-before" synchronization to the program continues after the M function is completed.

8.7.3 Program of the M06 subroutine

%L SUB_SPD_TO_GR

Remove the tool from the spindle (take it to ground).

It shows the message for the operator to extract the tool and waits for the operation to be completed before removing the message.

#MSG ["Extract Tool. T%D and press cycle-start", V.TM.TOOL]
M0
#MSG [""]
V.PLC.M[1107]=1

SPDLTOGR mark to the tool manager.

#RET

%L SUB_GR_TO_SPD

Insert the ground tool in the spindle.

It shows the message for the operator to insert the tool and waits for the operation to be completed before removing the message.

#MSG ["Insert Tool. T%D and press cycle-start", V.TM.NXTOOL]
M0
#MSG [""]
V.PLC.M[1108]=1

GRTOSPDL mark to the tool manager.

#RET

%L SUB_MZ_TO_SPD

Take the tool from the magazine and insert it in the spindle.

M109

Auxiliary function to execute an action.

V.PLC.M[1109]=1

MZTOSPDL mark to the tool manager.

#RET

%L SUB_SPD_TO_MZ

Leave the spindle tool in the magazine.

M110

Auxiliary function to execute an action.

V.PLC.M[1110]=1

SPDLTOMZ mark to the tool manager.

#RET

%L SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

G1 Z_ F_

Move the spindle.

#RET

%L SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

G1 Z_ F_

Move the spindle.

#RET

%SUB_M6.nc

M6

Order the tool manager to start the tool change.

$SWITCH V.PLC.R[101]

It analyzes the type of operation.

$CASE 1

Take a tool from the magazine and insert it in the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_MZ_TO_SPD

Take the tool from the magazine to the spindle.

$BREAK

$CASE 2

Leave the spindle tool in the magazine.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_SPD_TO_MZ

Leave the spindle tool in the magazine.

$BREAK

$CASE 3

Insert the ground tool in the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 4

Leave the spindle tool on the ground.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

$BREAK

$CASE 5

Leave the spindle tool in the magazine and take another one from the magazine.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_SPD_TO_MZ

Leave the spindle tool in the magazine.

LL SUB_MZ_TO_SPD

Take the tool from the magazine to the spindle.

$BREAK

$CASE 6

Leave the spindle tool in the magazine and take another one from ground.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_SPD_TO_MZ

Leave the spindle tool in the magazine.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 7

Leave the spindle tool on the ground and take another one from the magazine.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_MZ_TO_SPD

Take the tool from the magazine to the spindle.

$BREAK

$CASE 8

Leave the spindle tool on the ground and take another one from ground.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 9

Take a ground tool to the magazine going through the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_SPD_TO_MZ

Leave the spindle tool in the magazine.

$BREAK

$CASE 10

Take a tool from the magazine and leave on the ground going through the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_MZ_TO_SPD

Take the tool from the magazine to the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

$BREAK

$ENDSWITCH

End of analysis of type of operation.

$WHILE V.TM.MZWAIT == 1
$ENDWHILE

Wait for the tool manager.

#RET

End of M06 subroutine.

8.7.4 Basic PLC programming

When executing the -T- function

When executing a T function, the tool manager sends the code TMOPERATION=11 to the PLC. In general, it is an optimization of the change that permits orienting the magazine while machining.

In this case, the magazine is not oriented and the MZROT mark is activated to indicate that the operation has been completed.

```
DFU TMOPSTROBE AND CPS TMOPERATION EQ 11 = SET MZROT
```

Activate the MZROT mark to "tell" the tool manager that the operation has finished.

When executing the M06 function

When executing the M06 function, the tool manager sends to the PLC, in the TMOPERATION register, the code for the operation to be carried out.

```
DFU TMOPSTROBE = MOV TMOPERATION R101
```

This instruction transfers the TMOPERATION value to register R101 so it is managed by the M06 subroutine.

Every time the M06 subroutine ends an action, it lets the PLC know so it activates the relevant mark of the tool manager.

```
DFU M1107 = SET SPDLTOGR
DFD SPDLTOGR = RES M1107
```

The tool has been taken from the spindle to ground.

```
DFU M1108 = SET GRTOSPDL
DFD GRTOSPDL = RES M1108
```

The tool has been taken from ground to the spindle.

```
DFU M1109 = SET MZTOSPDL
DFD MZTOSPDL = RES M1109
```

The tool has been taken from the magazine to the spindle.

```
DFU M1110 = SET SPDLTOMZ
DFD SPDLTOMZ = RES M1110
```

The tool has been taken from the spindle to the magazine.

```
DFU M1111 = SET MZROT
DFD MZROT = RES M1111
```

The magazine has rotated already.

The M06 subroutine uses the following M functions to "tell" the PLC which movements it must carry out.

M109	Select in the magazine the position indicated by TAKEPOS and insert the tool in the spindle.
M110	Select in the magazine the position indicated by LEAVEPOS and leave the spindle tool.

Programming it depends on the type of machine. The auxiliary function will conclude after executing the requested movement.

Certain operations require using the information transferred by the tool manager in the following registers:

LEAVEPOS This register indicates the magazine position to leave the tool.

TAKEPOS This register indicates the magazine position of the tool to be taken (picked up).

Manager emergency signal

Treatment of the tool manager emergency signal.

```
DFU B11KEYBD1 = SET SETTMEM
```

Pressing the USER12 key activates the emergency.

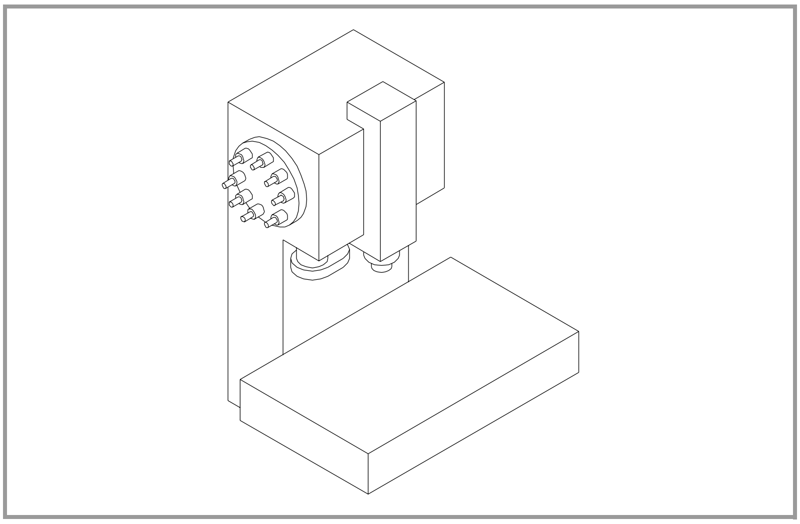
```
TMINEM = B11KEYLED1
```

The lamp of the USER12 key turns on when there is an emergency.

```
TMINEM AND DFU B12KEYBD1 = SET RESTMEM
```

Pressing the USER13 key removes the emergency.

8.8 Synchronous magazine. Changer arm with independent movements.



Usually the movements of the changer arm depend on machining because it cannot move while the part is being machined.

The communication between the tool manager and the PLC takes place in two stages.

- When executing the T function, the CNC lets the tool manager know about it.

  The tool manager sends a command to the PLC to select the next tool in the magazine (if possible).

  The CNC continues with the execution of the program without waiting for the tool manager to complete the operation.

- When executing the M06 function, its associated subroutine is called upon.

  The M06 function must also be programmed inside that subroutine so the CNC "tells" the tool manager to start making the tool change.

  The tool manager sends a command to the PLC to make the change.

  The CNC waits for the tool manager to complete the operation before continuing with the execution of the program.

The management of the tool change should be included in the subroutine associated with the M06 and leave the control of the external devices up to the PLC.

Use the auxiliary functions to govern the various devices (magazine rotation, magazine movement, tool changer arm, etc.) from the M06 subroutine.

8.8.1 Values of the TMOPERATION and marks to be activated by the PLC

The possible TMOPERATION values in this type of magazine and the marks and registers associated with each case are:

TMOPERATION=1

Take a tool from the magazine and insert it in the spindle.

TAKEPOS	Position occupied by the tool.

1. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.
2. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

TMOPERATION=2

Leave the spindle tool in the magazine.

LEAVEPOS	Position to leave the tool.

1. Take the tool from the spindle with holder 1 and activate the mark SPDLTOCH1.
2. Leave the tool of holder 1 in the magazine and activate the markCH1TOMZ.

TMOPERATION=3

Insert a ground tool in the spindle.

TAKEPOS=-4	Pick up the ground tool.

When the operation is completed, activate the mark GRTOSPDL.

TMOPERATION=4

Leave the spindle tool on the ground.

LEAVEPOS=-4	Leave the tool on the ground.

When the operation is completed, activate the mark SPDLTOGR.

TMOPERATION=5

Leave the spindle tool in the magazine and take another one from the magazine.

TAKEPOS	Position occupied by the tool.
LEAVEPOS	Position to leave the tool.

1. Take the tool from the spindle with holder 1 and activate the mark SPDLTOCH1.
2. Leave the tool of holder 1 in the magazine and activate the markCH1TOMZ.
3. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.
4. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

TMOPERATION=6

Leave the spindle tool in the magazine and take another one from ground.

TAKEPOS=-4	Pick up the ground tool.
LEAVEPOS	Position to leave the tool.

1. Take the tool from the spindle with holder 1 and activate the mark SPDLTOCH1.
2. Leave the tool of holder 1 in the magazine and activate the markCH1TOMZ.
3. Leave the ground tool in the spindle and activate the mark GRTOSPDL.

The tool manager admits the sequence 1-3-2.

TMOPERATION=7

Leave the spindle tool on the ground and take another one from the magazine.

TAKEPOS	Position occupied by the tool.
LEAVEPOS=-4	Leave the tool on the ground.

1. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.
2. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.
3. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

The tool manager admits the sequence 2-1-3.

TMOPERATION=8

Leave the spindle tool on the ground and take another one from ground.

TAKEPOS=-4	Pick up the ground tool.
LEAVEPOS=-4	Leave the tool on the ground.

1. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.
2. Leave the ground tool in the spindle and activate the mark GRTOSPDL.

TMOPERATION=9

Take a ground tool to the magazine going through the spindle.

TAKEPOS=-4	Pick up the ground tool.
LEAVEPOS	Position to leave the tool.

1. Leave the ground tool in the spindle and activate the mark GRTOSPDL.
2. Take the tool from the spindle with holder 1 and activate the mark SPDLTOCH1.
3. Leave the tool of holder 1 in the magazine and activate the markCH1TOMZ.

TMOPERATION=10

Take a tool from the magazine and leave on the ground going through the spindle.

TAKEPOS	Position occupied by the tool.
LEAVEPOS=-4	Leave the tool on the ground.

1. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.
2. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.
3. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.

TMOPERATION=11

In general, it is an optimization of the change that permits orienting the magazine while machining.

Activate the MZROT mark to indicate that the operation has been completed, whether it has been oriented or not.

TMOPERATION=12

Same as TMOPERATION=5. Only for random magazines and when having special tools.

TMOPERATION=13 In general, it is an optimization of the change that permits orienting two magazines while machining.

Activate the MZROT mark in both magazines to indicate that the operation has been completed, whether it has been oriented or not.

NEXTPOS	Position to leave the tool.
TAKEPOS	Position of the tool to be picked up.
MZID	Destination magazine to leave the tool and source magazine from which to pick up the new one.

TMOPERATION=14 Leave the spindle tool in a magazine and take a tool from another magazine.

TAKEPOS	Position of the tool to be picked up.
LEAVEPOS	Position to leave the tool.
MZID	Destination magazine to leave the tool and source magazine from which to pick up the new one.

1. Take the tool from the spindle with holder 1 and activate the mark SPDLTOCH1.
2. Leave the tool of holder 1 in the magazine and activate the markCH1TOMZ.
3. Take the tool from the other magazine with holder (claw) 1 and activate the mark MZTOCH1.
4. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

Summary

The values sent by the tool manager to the PLC in each operation and the marks to be activated by the PLC are the following:

TM ==> PLC			PLC ==> TM
TMOPERATION	TAKEPOS	LEAVEPOS	
1	#	0	MZTOCH1 + CH1TOSPDL
2	0	#	SPDLTOCH1 + CH1TOMZ
3	-4	0	GRTOSPDL
4	0	-4	SPDLTOGR
5	#	#	SPDLTOCH1 + CH1TOMZ + MZTOCH1 + CH1TOSPDL
6	-4	#	SPDLTOCH1 + + (CH1TOMZ / GRTOSPDL)
7	#	-4	(SPDLTOGR / MZTOCH1) + + CH1TOSPDL
8	-4	-4	SPDLTOGR + GRTOSPDL
9	-4	#	GRTOSPDL + SPDLTOCH1 + + CH1TOMZ
10	#	-4	MZTOCH1 + CH1TOSPDL + + SPDLTOGR
11	0	0	MZROT
12	#	#	SPDLTOCH1 + CH1TOMZ + MZTOCH1 + CH1TOSPDL
13	0	0	MZROT + MZROT
14	0	#	SPDLTOCH1 + CH1TOMZ + MZTOCH1 + CH1TOSPDL

The marks in parenthesis may be executed in any order; but both must be executed. Possibilities with TMOPERATION = 6:

SPDLTOCH1 + CH1TOMZ + GRTOSPDL

SPDLTOCH1 + GRTOSPDL + CH1TOMZ

The TAKEPOS and LEAVEPOS values indicate the following:

0	Do nothing.
#	Magazine position number.
-4	Ground position.

Application example

Assuming that there is no tool in the spindle, it shows the functions executed by the CNC, the values sent by the tool manager to the PLC in each operation and the marks to be activated by the PLC in each case.

It is a non-random magazine, each tool occupies the position of its own number and TG7, TG8 and TG9 are ground tools.

CNC	TM ==> PLC			PLC ==> TM
	TMOPERATION	TAKEPOS	LEAVEPOS	
T1 M6	11 1	0 1	0 0	MZROT MZTOCH1 + CH1TOSPDL
T2 M6	11 5	0 2	0 1	MZROT SPDLTOCH1 + CH1TOMZ + + MZTOCH1 + CH1TOSPDL
TG7 M6	11 6	0 -4	0 2	MZROT SPDLTOCH1 + CH1TOMZ + + GRTOSPDL
TG8 M6	11 8	0 -4	0 -4	MZROT SPDLTOGR + GRTOSPDL
T3 T4 M6	11 11 7	0 0 4	0 0 -4	MZROT MZROT SPDLTOGR + MZTOCH1 + + CH1TOSPDL
T0 M6	11 2	0 0	0 4	MZROT SPDLTOCH1 + CH1TOMZ
TG9 M6	11 3	0 -4	0 0	MZROT GRTOSPDL
T0 M6	11 4	0 0	0 -4	MZROT SPDLTOGR

8.8.2 Communication between the PLC and the M06 subroutine

The communication between the PLC and the M06 subroutine takes place using a series of generic marks and registers. The program of the M06 subroutine offered as an example uses the following marks and registers.

Communication between the PLC and the M06 subroutine

Registers used by the PLC to transfer information to the M06 subroutine:

R101 Type of operation requested by the tool manager (value of TMOPERATION)

Communication between the PLC and the M06 subroutine

Marks activated by the M06 subroutine so the PLC activates the relevant mark of the tool manager.

M1101 MZTOCH1

The tool has been taken from the magazine to holder 1.

M1102 CH1TOSPDL

The tool has been taken from holder 1 to the spindle.

M1103 SPDLTOCH1

The tool has been taken from the spindle to holder 1.

M1105 CH1TOMZ

The tool has been taken from holder 1 to the magazine.

M1107 SPDLTOGR

The tool has been taken from the spindle to ground.

M1108 GRTOSPDL

The tool has been taken from ground to the spindle.

M functions at the PLC

M functions that imply movements controlled by the PLC:

M101 Select in the magazine the position indicated by TAKEPOS and take the tool with holder 1.

M102 Take the tool of holder 1 to the spindle.

M103 Take the spindle tool with holder 1.

M105 Select in the magazine the position indicated by LEAVEPOS and leave the tool of holder 1.

Set all the M functions with "before-before" synchronization to the program continues after the M function is completed.

8.8.3 Program of the M06 subroutine

%L SUB_MZ_TO_CH1

Take the tool from the magazine with holder 1.

M101

Auxiliary function to execute an action.

V.PLC.M[1101]=1

MZTOCH1 mark to the tool manager.

#RET

%L SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

M102

Auxiliary function to execute an action.

V.PLC.M[1102]=1

CH1TOSPDL mark to the tool manager.

#RET

%L SUB_SPD_TO_CH1

Take the spindle tool with holder 1.

M103

Auxiliary function to execute an action.

V.PLC.M[1103]=1

SPDLTOCH1 mark to the tool manager.

#RET

%L SUB_CH1_TO_MZ

Take the tool of holder 1 to the magazine.

M105

Auxiliary function to execute an action.

V.PLC.M[1105]=1

CH1TOMZ mark to the tool manager.

#RET

%L SUB_SPD_TO_GR

Remove the tool from the spindle (take it to ground).

It shows the message for the operator to extract the tool and waits for the operation to be completed before removing the message.

#MSG ["Extract Tool. T%D and press cycle-start", V.TM.TOOL]
M0
#MSG [""]
V.PLC.M[1107]=1

SPDLTOGR mark to the tool manager.

#RET

%L SUB_GR_TO_SPD

Insert the ground tool in the spindle.

It shows the message for the operator to insert the tool and waits for the operation to be completed before removing the message.

```
#MSG ["Insert Tool. T%D and press cycle-start", V.TM.NXTOOL]
M0
#MSG [""]
V.PLC.M[1108]=1
```

GRTOSPDL mark to the tool manager.

```
#RET
```

%L SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

```
G1 Z_ F_
```

Move the spindle.

```
#RET
```

%L SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

```
G1 Z_ F_
```

Move the spindle.

```
#RET
```

%SUB_M6.nc

```
M6
```

Order the tool manager to start the tool change.

```
$SWITCH V.PLC.R[101]
```

It analyzes the type of operation.

```
$CASE 1
```

Take a tool from the magazine and insert it in the spindle.

```
LL SUB_SPD_AUTCHG
```

Move the spindle to the automatic tool change point

```
LL SUB_MZ_TO_CH1
```

Take the tool from the magazine with holder 1.

```
LL SUB_CH1_TO_SPD
```

Take the tool of holder 1 to the spindle.

```
$BREAK
```

```
$CASE 2
```

Leave the spindle tool in the magazine.

```
LL SUB_SPD_AUTCHG
```

Move the spindle to the automatic tool change point

```
LL SUB_SPD_TO_CH1
```

Take the spindle tool with holder 1.

```
LL SUB_CH1_TO_MZ
```

Leave the tool of holder 1 in the magazine.

```
$BREAK
```

$CASE 3

Insert the ground tool in the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 4

Leave the spindle tool on the ground.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

$BREAK

$CASE 5

Leave the spindle tool in the magazine and take another one from the magazine.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_SPD_TO_CH1

Take the spindle tool with holder 1.

LL SUB_CH1_TO_MZ

Leave the tool of holder 1 in the magazine.

LL SUB_MZ_TO_CH1

Take the tool from the magazine with holder 1.

LL SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

$BREAK

$CASE 6

Leave the spindle tool in the magazine and take another one from ground.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_SPD_TO_CH1

Take the spindle tool with holder 1.

LL SUB_CH1_TO_MZ

Leave the tool of holder 1 in the magazine.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 7

Leave the spindle tool on the ground and take another one from the magazine.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_MZ_TO_CH1

Take the tool from the magazine with holder 1.

LL SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

$BREAK

$CASE 8

Leave the spindle tool on the ground and take another one from ground.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 9

Take a ground tool to the magazine going through the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_SPD_TO_CH1

Take the spindle tool with holder 1.

LL SUB_CH1_TO_MZ

Leave the tool of holder 1 in the magazine.

$BREAK

$CASE 10

Take a tool from the magazine and leave on the ground going through the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_MZ_TO_CH1

Take the tool from the magazine with holder 1.

LL SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

$BREAK

$ENDSWITCH

End of analysis of type of operation.

$WHILE V.TM.MZWAIT == 1
$ENDWHILE

Wait for the tool manager.

#RET

End of M06 subroutine.

8.8.4 Basic PLC programming

When executing the -T- function

When executing a T function, the tool manager sends the code `TMOPERATION=11` to the PLC. In general, it is an optimization of the change that permits orienting the magazine while machining.

In this case, the magazine is not oriented and the `MZROT` mark is activated to indicate that the operation has been completed.

```
DFU TMOPSTROBE AND CPS TMOPERATION EQ 11 = SET MZROT
```

Activate the MZROT mark to "tell" the tool manager that the operation has finished.

When executing the M06 function

When executing the M06 function, the tool manager sends to the PLC, in the `TMOPERATION` register, the code for the operation to be carried out.

```
DFU TMOPSTROBE = MOV TMOPERATION R101
```

This instruction transfers the TMOPERATION value to register R101 so it is managed by the M06 subroutine.

Every time the M06 subroutine ends an action, it lets the PLC know so it activates the relevant mark of the tool manager.

```
DFU M1101 = SET MZTOCH1
DFD MZTOCH1 = RES M1101
```

The tool has been taken from the magazine to holder 1.

```
DFU M1102 = SET CH1TOSPDL
DFD CH1TOSPDL = RES M1102
```

The tool has been taken from holder 1 to the spindle.

```
DFU M1103 = SET SPDLTOCH1
DFD SPDLTOCH1 = RES M1103
```

The tool has been taken from the spindle to holder 1.

```
DFU M1105 = SET CH1TOMZ
DFD CH1TOMZ = RES M1105
```

The tool has been taken from holder 1 to the magazine.

```
DFU M1107 = SET SPDLTOGR
DFD SPDLTOGR = RES M1107
```

The tool has been taken from the spindle to ground.

```
DFU M1108 = SET GRTOSPDL
DFD GRTOSPDL = RES M1108
```

The tool has been taken from ground to the spindle.

The M06 subroutine uses the following M functions to "tell" the PLC which movements it must carry out.

M101	Select in the magazine the position indicated by TAKEPOS and take the tool with holder 1.
M102	Take the tool of holder 1 to the spindle.
M103	Take the spindle tool with holder 1.
M105	Select in the magazine the position indicated by LEAVEPOS and leave the tool of holder 1.

Programming it depends on the type of machine. The auxiliary function will conclude after executing the requested movement.

Certain operations require using the information transferred by the tool manager in the following registers:

LEAVEPOS This register indicates the magazine position to leave the tool.

TAKEPOS This register indicates the magazine position of the tool to be taken (picked up).

Manager emergency signal

Treatment of the tool manager emergency signal.

```
DFU B11KEYBD1 = SET SETTMEM
```

Pressing the USER12 key activates the emergency.

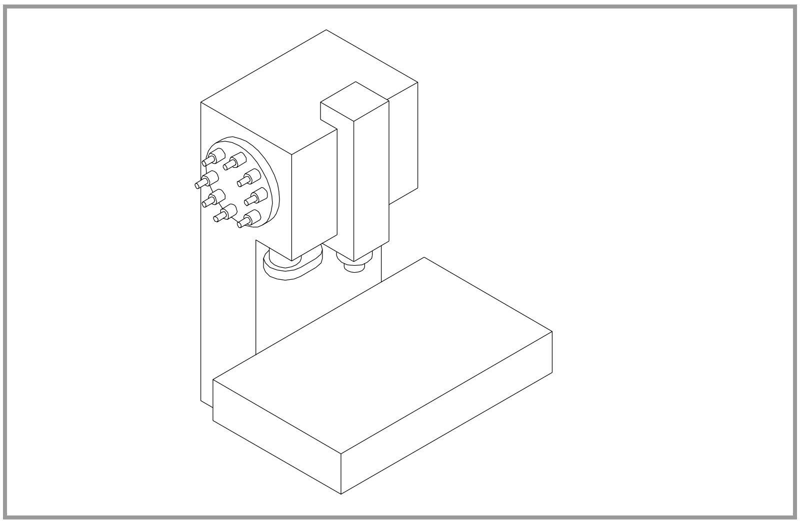
```
TMINEM = B11KEYLED1
```

The lamp of the USER12 key turns on when there is an emergency.

```
TMINEM AND DFU B12KEYBD1 = SET RESTMEM
```

Pressing the USER13 key removes the emergency.

8.9 Synchronous magazine. Tool changer arm with 2 holders



Usually the movements of the changer arm depend on machining because it cannot move while the part is being machined.

The communication between the tool manager and the PLC takes place in two stages.

- When executing the T function, the CNC lets the tool manager know about it.

  The tool manager sends a command to the PLC to select the next tool in the magazine (if possible).

  The CNC continues with the execution of the program without waiting for the tool manager to complete the operation.

- When executing the M06 function, its associated subroutine is called upon.

  The M06 function must also be programmed inside that subroutine so the CNC "tells" the tool manager to start making the tool change.

  The tool manager sends a command to the PLC to make the change.

  The CNC waits for the tool manager to complete the operation before continuing with the execution of the program.

The management of the tool change should be included in the subroutine associated with the M06 and leave the control of the external devices up to the PLC.

Use the auxiliary functions to govern the various devices (magazine rotation, magazine movement, tool changer arm, etc.) from the M06 subroutine.

8.9.1 Values of the TMOPERATION and marks to be activated by the PLC

The possible TMOPERATION values in this type of magazine and the marks and registers associated with each case are:

TMOPERATION=1

Take a tool from the magazine and insert it in the spindle.

TAKEPOS Position occupied by the tool.

1. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.
2. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

TMOPERATION=2

Leave the spindle tool in the magazine.

LEAVEPOS Position to leave the tool.

1. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
2. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.

TMOPERATION=3

Insert a ground tool in the spindle.

TAKEPOS=-4 Pick up the ground tool.

When the operation is completed, activate the mark GRTOSPDL.

TMOPERATION=4

Leave the spindle tool on the ground.

LEAVEPOS=-4 Leave the tool on the ground.

When the operation is completed, activate the mark SPDLTOGR.

TMOPERATION=5

Leave the spindle tool in the magazine and take another one from the magazine.

TAKEPOS Position occupied by the tool.

LEAVEPOS Position to leave the tool.

1. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
2. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.
3. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.
4. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

The tool manager also admits the sequences 1-2-4-3, 2-1-3-4, 2-1-4-3

TMOPERATION=6

Leave the spindle tool in the magazine and take another one from ground.

TAKEPOS=-4 Pick up the ground tool.

LEAVEPOS Position to leave the tool.

1. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
2. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.
3. Leave the ground tool in the spindle and activate the mark GRTOSPDL.

The tool manager admits the sequence 1-3-2.

TMOPERATION=7

Leave the spindle tool on the ground and take another one from the magazine.

TAKEPOS	Position occupied by the tool.
LEAVEPOS=-4	Leave the tool on the ground.

1. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.
2. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.
3. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

The tool manager admits the sequence 2-1-3.

TMOPERATION=8

Leave the spindle tool on the ground and take another one from ground.

TAKEPOS=-4	Pick up the ground tool.
LEAVEPOS=-4	Leave the tool on the ground.

1. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.
2. Leave the ground tool in the spindle and activate the mark GRTOSPDL.

TMOPERATION=9

Take a ground tool to the magazine going through the spindle.

TAKEPOS=-4	Pick up the ground tool.
LEAVEPOS	Position to leave the tool.

1. Leave the ground tool in the spindle and activate the mark GRTOSPDL.
2. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
3. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.

TMOPERATION=10

Take a tool from the magazine and leave on the ground going through the spindle.

TAKEPOS	Position occupied by the tool.
LEAVEPOS=-4	Leave the tool on the ground.

1. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.
2. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.
3. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.

TMOPERATION=11

In general, it is an optimization of the change that permits orienting the magazine while machining.

Activate the MZROT mark to indicate that the operation has been completed, whether it has been oriented or not.

TMOPERATION=12

Same as TMOPERATION=5. It is only used when the magazine is non-random. Also for random magazines when having special tools.

TMOPERATION=13

In general, it is an optimization of the change that permits orienting two magazines while machining.

Activate the MZROT mark in both magazines to indicate that the operation has been completed, whether it has been oriented or not.

NEXTPOS	Position to leave the tool.
TAKEPOS	Position of the tool to be picked up.
MZID	Destination magazine to leave the tool and source magazine from which to pick up the new one.

TMOPERATION=14

Leave the spindle tool in a magazine and take a tool from another magazine.

TAKEPOS	Position of the tool to be picked up.
MZID	Destination magazine to leave the tool and source magazine from which to pick up the new one.

1. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
2. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.
3. Take the tool from the other magazine with holder (claw) 1 and activate the mark MZTOCH1.
4. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

Summary

The values sent by the tool manager to the PLC in each operation and the marks to be activated by the PLC are the following:

TM ==> PLC			PLC ==> TM
TMOPERATION	TAKEPOS	LEAVEPOS	
1	#	0	MZTOCH1 + CH1TOSPDL
2	0	#	SPDLTOCH2 + CH2TOMZ
3	-4	0	GRTOSPDL
4	0	-4	SPDLTOGR
5	#	#	(SPDLTOCH2 / MZTOCH1) + + (CH1TOSPDL / CH2TOMZ)
6	-4	#	SPDLTOCH2 + + (CH2TOMZ / GRTOSPDL)
7	#	-4	(SPDLTOGR / MZTOCH1) + + CH1TOSPDL
8	-4	-4	SPDLTOGR + GRTOSPDL
9	-4	#	GRTOSPDL + SPDLTOCH2 + + CH2TOMZ
10	#	-4	MZTOCH1 + CH1TOSPDL + + SPDLTOGR
11	0	0	MZROT
12	#	#	(SPDLTOCH2 / MZTOCH1) + + (CH1TOSPDL / CH2TOMZ)
13	0	0	MZROT + MZROT
14	#	0	SPDLTOCH2 + CH2TOMZ + + MZTOCH1 + CH1TOSPDL

The marks in parenthesis may be executed in any order; but both must be executed. Possibilities with TMOPERATION = 6:

SPDLTOCH2 + CH2TOMZ + GRTOSPDL

SPDLTOCH2 + GRTOSPDL + CH2TOMZ

The TAKEPOS and LEAVEPOS values indicate the following:

0	Do nothing.
#	Magazine position number.
-4	Ground position.

Application example

Assuming that there is no tool in the spindle, it shows the functions executed by the CNC, the values sent by the tool manager to the PLC in each operation and the marks to be activated by the PLC in each case.

It is a non-random magazine, each tool occupies the position of its own number and TG7, TG8 and TG9 are ground tools.

CNC	TM ==> PLC			PLC ==> TM
	TMOPERATION	TAKEPOS	LEAVEPOS	
T1 M6	11 1	0 A	0 0	MZROT MZTOCH1 + CH1TOSPDL
T2 M6	11 5	0 B	0 B	MZROT MZTOCH1 + SPDLTOCH2 + + CH1TOSPDL + CH2TOMZ
TG7 M6	11 6	0 -4	0 A	MZROT SPDLTOCH2 + CH2TOMZ + + GRTOSPDL
TG8 M6	11 8	0 -4	0 -4	MZROT SPDLTOGR + GRTOSPDL
T3 T4 M6	11 11 7	0 0 D	0 0 -4	MZROT MZROT SPDLTOGR + MZTOCH1 + + CH1TOSPDL
T0 M6	11 2	0 0	0 D	MZROT SPDLTOCH2 + CH2TOMZ
TG9 M6	11 3	0 -4	0 0	MZROT GRTOSPDL
T0 M6	11 4	0 0	0 -4	MZROT SPDLTOGR

8.9.2 Communication between the PLC and the M06 subroutine

The communication between the PLC and the M06 subroutine takes place using a series of generic marks and registers. The program of the M06 subroutine offered as an example uses the following marks and registers.

Communication between the PLC and the M06 subroutine

Registers used by the PLC to transfer information to the M06 subroutine:

R101 Type of operation requested by the tool manager (value of TMOPERATION)

Communication between the PLC and the M06 subroutine

Marks activated by the M06 subroutine so the PLC activates the relevant mark of the tool manager.

M1101 MZTOCH1

The tool has been taken from the magazine to holder 1.

M1102 CH1TOSPDL

The tool has been taken from holder 1 to the spindle.

M1104 SPDLTOCH2

The tool has been taken from the spindle to holder 2.

M1106 CH2TOMZ

The tool has been taken from holder 2 to the magazine.

M1107 SPDLTOGR

The tool has been taken from the spindle to ground.

M1108 GRTOSPDL

The tool has been taken from ground to the spindle.

M functions at the PLC

M functions that imply movements controlled by the PLC:

M101 Select in the magazine the position indicated by TAKEPOS and take the tool with holder 1.

M102 Take the tool of holder 1 to the spindle.

M104 Take the spindle tool with holder 2.

M106 Select in the magazine the position indicated by LEAVEPOS and leave the tool of holder 2.

Set all the M functions with "before-before" synchronization to the program continues after the M function is completed.

8.9.3 Program of the M06 subroutine

%L SUB_MZ_TO_CH1

Take the tool from the magazine with holder 1.

M101

Auxiliary function to execute an action.

V.PLC.M[1101]=1

MZTOCH1 mark to the tool manager.

#RET

%L SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

M102

Auxiliary function to execute an action.

V.PLC.M[1102]=1

CH1TOSPDL mark to the tool manager.

#RET

%L SUB_SPD_TO_CH2

Take the spindle tool with holder 2.

M104

Auxiliary function to execute an action.

V.PLC.M[1104]=1

SPDLTOCH2 mark to the tool manager.

#RET

%L SUB_CH2_TO_MZ

Take the tool of holder 2 to the magazine.

M106

Auxiliary function to execute an action.

V.PLC.M[1106]=1

CH2TOMZ mark to the tool manager.

#RET

%L SUB_SPD_TO_GR

Remove the tool from the spindle (take it to ground).

It shows the message for the operator to extract the tool and waits for the operation to be completed before removing the message.

#MSG ["Extract Tool. T%D and press cycle-start", V.TM.TOOL]
M0
#MSG [""]
V.PLC.M[1107]=1

SPDLTOGR mark to the tool manager.

#RET

%L SUB_GR_TO_SPD

Insert the ground tool in the spindle.

It shows the message for the operator to insert the tool and waits for the operation to be completed before removing the message.

```
#MSG ["Insert Tool. T%D and press cycle-start", V.TM.NXTOOL]
M0
#MSG [""]
V.PLC.M[1108]=1
```

GRTOSPDL mark to the tool manager.

```
#RET
```

%L SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

```
G1 Z_ F_
```

Move the spindle.

```
#RET
```

%L SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

```
G1 Z_ F_
```

Move the spindle.

```
#RET
```

%SUB_M6.nc

```
M6
```

Order the tool manager to start the tool change.

```
$SWITCH V.PLC.R[101]
```

It analyzes the type of operation.

```
$CASE 1
```

Take a tool from the magazine and insert it in the spindle.

```
LL SUB_SPD_AUTCHG
```

Move the spindle to the automatic tool change point

```
LL SUB_MZ_TO_CH1
```

Take the tool from the magazine with holder 1.

```
LL SUB_CH1_TO_SPD
```

Take the tool of holder 1 to the spindle.

```
$BREAK
```

```
$CASE 2
```

Leave the spindle tool in the magazine.

```
LL SUB_SPD_AUTCHG
```

Move the spindle to the automatic tool change point

```
LL SUB_SPD_TO_CH2
```

Take the spindle tool with holder 2.

```
LL SUB_CH2_TO_MZ
```

Leave the tool of holder 2 in the magazine.

```
$BREAK
```

$CASE 3

Insert the ground tool in the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 4

Leave the spindle tool on the ground.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

$BREAK

$CASE 5

Leave the spindle tool in the magazine and take another one from the magazine.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_SPD_TO_CH2

Take the spindle tool with holder 2.

LL SUB_MZ_TO_CH1

Take the tool from the magazine with holder 1.

LL SUB_CH2_TO_MZ

Leave the tool of holder 2 in the magazine.

LL SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

$BREAK

$CASE 6

Leave the spindle tool in the magazine and take another one from ground.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_SPD_TO_CH2

Take the spindle tool with holder 2.

LL SUB_CH2_TO_MZ

Leave the tool of holder 2 in the magazine.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 7

Leave the spindle tool on the ground and take another one from the magazine.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_MZ_TO_CH1

Take the tool from the magazine with holder 1.

LL SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

$BREAK

$CASE 8

Leave the spindle tool on the ground and take another one from ground.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 9

Take a ground tool to the magazine going through the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_SPD_TO_CH2

Take the spindle tool with holder 2.

LL SUB_CH2_TO_MZ

Leave the tool of holder 2 in the magazine.

$BREAK

$CASE 10

Take a tool from the magazine and leave on the ground going through the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_MZ_TO_CH1

Take the tool from the magazine with holder 1.

LL SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

$BREAK

$ENDSWITCH

End of analysis of type of operation.

$WHILE V.TM.MZWAIT == 1
$ENDWHILE

Wait for the tool manager.

#RET

End of M06 subroutine.

8.9.4 Basic PLC programming

When executing the -T- function

When executing a T function, the tool manager sends the code TMOPERATION=11 to the PLC. In general, it is an optimization of the change that permits orienting the magazine while machining.

In this case, the magazine is not oriented and the MZROT mark is activated to indicate that the operation has been completed.

```
DFU TMOPSTROBE AND CPS TMOPERATION EQ 11 = SET MZROT
```

Activate the MZROT mark to "tell" the tool manager that the operation has finished.

When executing the M06 function

When executing the M06 function, the tool manager sends to the PLC, in the TMOPERATION register, the code for the operation to be carried out.

```
DFU TMOPSTROBE = MOV TMOPERATION R101
```

This instruction transfers the TMOPERATION value to register R101 so it is managed by the M06 subroutine.

Every time the M06 subroutine ends an action, it lets the PLC know so it activates the relevant mark of the tool manager.

```
DFU M1101 = SET MZTOCH1
DFD MZTOCH1 = RES M1101
```

The tool has been taken from the magazine to holder 1.

```
DFU M1102 = SET CH1TOSPDL
DFD CH1TOSPDL = RES M1102
```

The tool has been taken from holder 1 to the spindle.

```
DFU M1104 = SET SPDLTOCH2
DFD SPDLTOCH2 = RES M1104
```

The tool has been taken from the spindle to holder 2.

```
DFU M1106 = SET CH2TOMZ
DFD CH2TOMZ = RES M1106
```

The tool has been taken from holder 2 to the magazine.

```
DFU M1107 = SET SPDLTOGR
DFD SPDLTOGR = RES M1107
```

The tool has been taken from the spindle to ground.

```
DFU M1108 = SET GRTOSPDL
DFD GRTOSPDL = RES M1108
```

The tool has been taken from ground to the spindle.

The M06 subroutine uses the following M functions to "tell" the PLC which movements it must carry out.

M101 Select in the magazine the position indicated by TAKEPOS and take the tool with holder 1.

M102 Take the tool of holder 1 to the spindle.

M104 Take the spindle tool with holder 2.

M106 Select in the magazine the position indicated by LEAVEPOS and leave the tool of holder 2.

Programming it depends on the type of machine. The auxiliary function will conclude after executing the requested movement.

Certain operations require using the information transferred by the tool manager in the following registers:

LEAVEPOS	This register indicates the magazine position to leave the tool.
TAKEPOS	This register indicates the magazine position of the tool to be taken (picked up).

Manager emergency signal

Treatment of the tool manager emergency signal.

```
DFU B11KEYBD1 = SET SETTMEM
```

Pressing the USER12 key activates the emergency.

```
TMINEM = B11KEYLED1
```

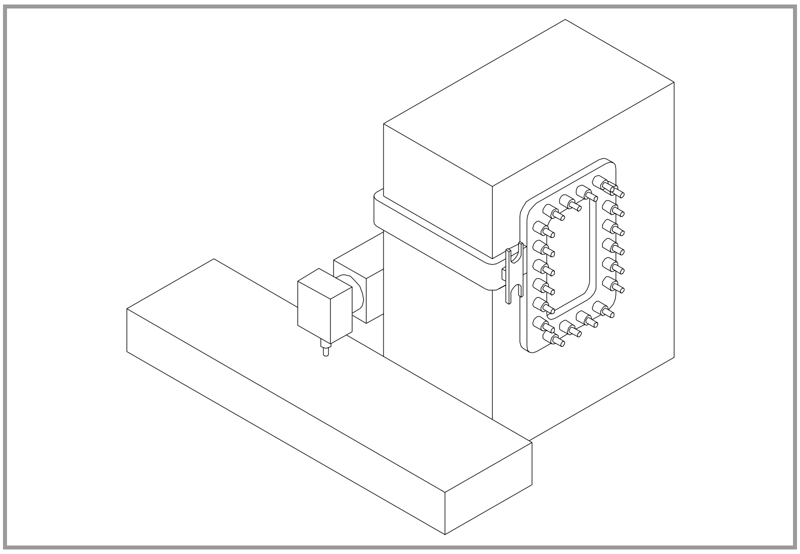
The lamp of the USER12 key turns on when there is an emergency.

```
TMINEM AND DFU B12KEYBD1 = SET RESTMEM
```

Pressing the USER13 key removes the emergency.

8.10 Asynchronous magazine

Usually, the tool magazine is located away from the spindle and the movements of the changer arm are independent. It is recommended to take and leave the tool in the magazine while machining the part.



The communication between the tool manager and the PLC takes place in two stages.

- When executing the T function, the CNC lets the tool manager know about it.

  The tool manager sends a command to the PLC to select the next tool in the magazine (if possible).

  The CNC continues with the execution of the program without waiting for the tool manager to complete the operation.

- When executing the M06 function, its associated subroutine is called upon.

  The M06 function must also be programmed inside that subroutine so the CNC "tells" the tool manager to start making the tool change.

  The tool manager sends a command to the PLC to make the change.

  The CNC waits for the tool manager to complete the operation before continuing with the execution of the program.

The management of the tool change should be included in the subroutine associated with the M06 and leave the control of the external devices up to the PLC.

Use the auxiliary functions to govern the various devices (magazine rotation, magazine movement, tool changer arm, etc.) from the M06 subroutine.

If the tool change implies leaving a tool in the magazine, once executed the change and with the tool in the arm, it is possible to activate the TCHANGEOK mark for the CNC to continue executing the program while the tool is being left in the magazine.

8.10.1 Values of the TMOPERATION and marks to be activated by the PLC

The possible TMOPERATION values in this type of magazine and the marks and registers associated with each case are:

TMOPERATION=1

Take a tool from the magazine and insert it in the spindle.

Previously, while machining (when executing the T), it sends the TMOPERATION=11 code to pick up the tool from the magazine with holder 1. Now it is done with TMOPERATION=1.

TAKEPOS=-1	Pick up the tool from holder 1.

1. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

TMOPERATION=2

Leave the spindle tool in the magazine.

LEAVEPOS	Position to leave the tool.

1. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
2. Start sending the changer arm to the magazine to leave the tool in holder 2.
3. When the arm leaves the collision zone, activate TCHANGEOK, if so desired, to continue executing the program.
4. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.

TMOPERATION=3

Insert a ground tool in the spindle.

TAKEPOS=-4	Pick up the ground tool.

When the operation is completed, activate the mark GRTOSPDL.

TMOPERATION=4

Leave the spindle tool on the ground.

LEAVEPOS=-4	Leave the tool on the ground.

When the operation is completed, activate the mark SPDLTOGR.

TMOPERATION=5

Leave the spindle tool in the magazine and take another one from the magazine.

Previously, while machining (when executing the T), it sends the TMOPERATION=11 code to pick up the tool from the magazine with holder 1. Now it is done with TMOPERATION=5.

TAKEPOS=-1	Pick up the tool from holder 1.
LEAVEPOS	Position to leave the tool.

The tool manager admits 2 sequences. First sequence.

1. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
2. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.
3. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

Second sequence.

1. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
2. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.
3. Start sending the changer arm to the magazine to leave the tool in holder 2.

4. When the arm leaves the collision zone, activate TCHANGEOK, if so desired, to continue executing the program.
5. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.

TMOPERATION=6

Leave the spindle tool in the magazine and take another one from ground.

TAKEPOS=-4 Pick up the ground tool.

LEAVEPOS Position to leave the tool.

The tool manager admits 2 sequences. First sequence.

1. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
2. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.
3. Leave the ground tool in the spindle and activate the mark GRTOSPDL.

Second sequence.

1. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
2. Leave the ground tool in the spindle and activate the mark GRTOSPDL.
3. Start sending the changer arm to the magazine to leave the tool in holder 2.
4. When the arm leaves the collision zone, activate TCHANGEOK, if so desired, to continue executing the program.
5. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.

TMOPERATION=7

Leave the spindle tool on the ground and take another one from the magazine.

Previously, while machining (when executing the T), it sends the TMOPERATION=11 code to pick up the tool from the magazine with holder 1. Now it is done with TMOPERATION=7.

TAKEPOS=-1 Pick up the tool from holder 1.

LEAVEPOS=-4 Leave the tool on the ground.

1. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.
2. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

TMOPERATION=8

Leave the spindle tool on the ground and take another one from ground.

TAKEPOS=-4 Pick up the ground tool.

LEAVEPOS=-4 Leave the tool on the ground.

1. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.
2. Leave the ground tool in the spindle and activate the mark GRTOSPDL.

TMOPERATION=9 Take a ground tool to the magazine going through the spindle.

TAKEPOS=-4	Pick up the ground tool.
LEAVEPOS	Position to leave the tool.

1. Leave the ground tool in the spindle and activate the mark GRTOSPDL.
2. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
3. Start sending the changer arm to the magazine to leave the tool in holder 2.
4. When the arm leaves the collision zone, activate TCHANGEOK, if so desired, to continue executing the program.
5. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.

TMOPERATION=10 Take a tool from the magazine and leave on the ground going through the spindle.

TAKEPOS	Position occupied by the tool.
LEAVEPOS=-4	Leave the tool on the ground.

1. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.
2. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.
3. Leave the tool of the spindle on the ground and activate the mark SPDLTOGR.

TMOPERATION=11 It is an optimization of the tool change that is used in the following cases.

When executing a T function while machining.

TAKEPOS	Position occupied by the tool.

1. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.

When requesting a new tool and there is another one in holder 1 of the arm.

TAKEPOS	Position occupied by the tool.
LEAVEPOS	Position to leave the tool.

1. Leave the tool of holder 1 in the magazine and activate the markCH1TOMZ.
2. Take the tool from the magazine with holder 1 and activate the mark MZTOCH1.

TMOPERATION=14 Leave the spindle tool in a magazine and take a tool from another magazine.

Previously, while machining (when executing the T), it sends the TMOPERATION=11 code to pick up the tool from the magazine with holder (claw) 1. Now it is done with TMOPERATION=14.

TAKEPOS	Position of the tool to be picked up.
LEAVEPOS	Position to leave the tool.
MZID	Destination magazine to leave the tool and source magazine from which to pick up the new one.

1. Take the tool from the spindle with holder 2 and activate the mark SPDLTOCH1.
2. Leave the tool of holder 2 in the magazine and activate the markCH2TOMZ.
3. Insert the tool of holder 1 in the spindle and activate CH1TOSPDL.

Summary

The values sent by the tool manager to the PLC in each operation and the marks to be activated by the PLC are the following:

TM ==> PLC			PLC ==> TM
TMOPERATION	TAKEPOS	LEAVEPOS	
1	-1	0	CH1TOSPDL
2	0	#	SPDLTOCH2 + [TCHANGEOK] + + CH2TOMZ
3	-4	0	GRTOSPDL
4	0	-4	SPDLTOGR
5	-1	#	a) SPDLTOCH2 + CH2TOMZ + + CH1TOSPDL b) SPDLTOCH2 + CH1TOSPDL + [TCHANGEOK] + CH2TOMZ
6	-4	#	a) SPDLTOCH2 + CH2TOMZ + + GRTOSPDL b) SPDLTOCH2 + GRTOSPDL + [TCHANGEOK] + CH2TOMZ
7	-1	-4	SPDLTOGR + CH1TOSPDL
8	-4	-4	SPDLTOGR + GRTOSPDL
9	-4	#	GRTOSPDL + SPDLTOCH2 + + [TCHANGEOK] + CH2TOMZ
10	#	-4	MZTOCH1 + CH1TOSPDL + + SPDLTOGR
11	?	?	1) IF LEAVEPOS=# CH1TOMZ 2) IF TAKEPOS=# MZTOCH1
14	-1	#	SPDLTOCH2 + CH2TOMZ + CH1TOSPDL

There are 2 possible sequences in operations 5 and 6. There may be two cases in operation 11.

The TCHANGEOK mark is optional. Once the tool change is done and the tool is in the arm, it must be used when it is possible to execute the program while leaving the tool in the magazine.

The TAKEPOS and LEAVEPOS values indicate the following:

0	Do nothing.
#	Magazine position number.
-1	Holder 1 of the tool changer arm.
-4	Ground position.

Application example

Assuming that there is no tool in the spindle, it shows the functions executed by the CNC, the values sent by the tool manager to the PLC in each operation and the marks to be activated by the PLC in each case.

It is a non-random magazine, each tool occupies the position of its own number and TG7, TG8 and TG9 are ground tools.

CNC	TM ==> PLC			PLC ==> TM
	TMOPERATION	TAKEPOS	LEAVEPOS	
T1 M6	11 1	1 -1	0 0	MZTOCH1 CH1TOSPDL
T2 M6	11 5	2 -1	0 1	MZTOCH1 SPDLTOCH2 + CH1TOSPDL + + CH2TOMZ
TG7 M6	11 6	0 -4	0 0	MZROT SPDLTOCH2 + CH2TOMZ + + GRTOSPDL
TG8 M6	11 8	0 -4	0 -4	MZROT SPDLTOGR + GRTOSPDL
T3 T4 M6	11 11 7	3 4 -1	0 3 -4	MZTOCH1 CH1TOMZ + MZTOCH1 SPDLTOGR + + CH1TOSPDL
T0 M6	11 2	0 0	0 4	MZROT SPDLTOCH2 + CH2TOMZ
TG9 M6	11 3	0 -4	0 0	MZROT GRTOSPDL
T0 M6	11 4	0 0	0 -4	MZROT SPDLTOGR

8.10.2 Communication between the PLC and the M06 subroutine

The communication between the PLC and the M06 subroutine takes place using a series of generic marks and registers. The program of the M06 subroutine offered as an example uses the following marks and registers.

Communication between the PLC and the M06 subroutine

Registers used by the PLC to transfer information to the M06 subroutine:

R101 Type of operation requested by the tool manager (value of TMOPERATION)

Communication between the PLC and the M06 subroutine

Marks activated by the M06 subroutine so the PLC activates the relevant mark of the tool manager.

M1100 TCHANGEOK

Continue executing the program.

M1101 MZTOCH1

The tool has been taken from the magazine to holder 1.

M1102 CH1TOSPDL

The tool has been taken from holder 1 to the spindle.

M1104 SPDLTOCH2

The tool has been taken from the spindle to holder 2.

M1107 SPDLTOGR

The tool has been taken from the spindle to ground.

M1108 GRTOSPDL

The tool has been taken from ground to the spindle.

The PLC sets the CH2TOMZ mark when the tool has been left.

M functions at the PLC

M functions that imply movements controlled by the PLC:

M101 Select in the magazine the position indicated by TAKEPOS and take the tool with holder 1.

M102 Take the tool of holder 1 to the spindle.

M104 Take the spindle tool with holder 2.

M106 Start sending the changer arm to the magazine to leave the tool in holder 2.

M121 Take the changer arm to the change point.

M122 Take the changer arm to the magazine.

M123 Retract the changer arm.

Set all the M functions with "before-before" synchronization to the program continues after the M function is completed.

The PLC must consider the M106 completed when the arm exits the collision zone and machining is possible.

8.10.3 Program of the M06 subroutine

%L SUB_MZ_TO_CH1

Take the tool from the magazine with holder 1.

M101

Auxiliary function to execute an action.

V.PLC.M[1101]=1

MZTOCH1 mark to the tool manager.

#RET

%L SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

M102

Auxiliary function to execute an action.

V.PLC.M[1102]=1

CH1TOSPDL mark to the tool manager.

#RET

%L SUB_SPD_TO_CH2

Take the spindle tool with holder 2.

M104

Auxiliary function to execute an action.

V.PLC.M[1104]=1

SPDLTOCH2 mark to the tool manager.

#RET

%L SUB_CH2_TO_MZ

Start sending the changer arm to the magazine to leave the tool in holder 2.

M106

Auxiliary function to execute an action.

The PLC must consider the M106 completed when the arm exits the collision zone and machining is possible.

The PLC sets the CH2TOMZ mark when the tool has been left.

#RET

%L SUB_SPD_TO_GR

Remove the tool from the spindle (take it to ground).

It shows the message for the operator to extract the tool and waits for the operation to be completed before removing the message.

#MSG ["Extract Tool. T%D and press cycle-start", V.TM.TOOL]
M0
#MSG [""]
V.PLC.M[1107]=1

SPDLTOGR mark to the tool manager.

#RET

%L SUB_GR_TO_SPD

Insert the ground tool in the spindle.

It shows the message for the operator to insert the tool and waits for the operation to be completed before removing the message.

```
#MSG ["Insert Tool. T%D and press cycle-start", V.TM.NXTOOL]
M0
#MSG [""]
V.PLC.M[1108]=1
```

GRTOSPDL mark to the tool manager.

```
#RET
```

%L SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

```
G1 Z_ F_
```

Move the spindle.

```
#RET
```

%L SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

```
G1 Z_ F_
```

Move the spindle.

```
#RET
```

%L SUB_ARM_TO_CHG

Take the changer arm to the change point.

```
M121
```

Auxiliary function to execute an action.

```
#RET
```

%L SUB_ARM_TO_MZ

Take the changer arm to the magazine.

```
M122
```

Auxiliary function to execute an action.

```
#RET
```

%L SUB_ARM_BACK

Retract the changer arm.

```
M123
```

Auxiliary function to execute an action.

```
#RET
```

%SUB_M6.nc

```
M6
```

Order the tool manager to start the tool change.

```
$SWITCH V.PLC.R[101]
```

It analyzes the type of operation.

$CASE 1

Take a tool from the magazine and insert it in the spindle.

Previously, While machining (when executing the T), the tool is taken from the magazine to holder 1.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_ARM_TO_CHG

Take the changer arm to the change point.

LL SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

LL SUB_ARM_BACK

Retract the changer arm.

$BREAK

$CASE 2

Leave the spindle tool in the magazine.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_ARM_TO_CHG

Take the changer arm to the change point.

LL SUB_SPD_TO_CH2

Take the spindle tool with holder 2.

LL SUB_ARM_BACK

Retract the changer arm.

LL SUB_CH2_TO_MZ

Start sending the changer arm to the magazine to leave the tool in holder 2.

V.PLC.M[1100]=1

Order the PLC to activate the TCHANGEOK mark "telling" the tool manager that it can continue executing.

$BREAK

$CASE 3

Insert the ground tool in the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 4

Leave the spindle tool on the ground.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

$BREAK

$CASE 5

Leave the spindle tool in the magazine and take another one from the magazine.

Previously, While machining (when executing the T), the tool is taken from the magazine to holder 1.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_ARM_TO_CHG

Take the changer arm to the change point.

LL SUB_SPD_TO_CH2

Take the spindle tool with holder 2.

LL SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

LL SUB_ARM_BACK

Retract the changer arm.

LL SUB_CH2_TO_MZ

Start sending the changer arm to the magazine to leave the tool.

V.PLC.M[1100]=1

Order the PLC to activate the TCHANGEOK mark "telling" the tool manager that it can continue executing.

$BREAK

$CASE 6

Leave the spindle tool in the magazine and take another one from ground.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_ARM_TO_CHG

Take the changer arm to the change point.

LL SUB_SPD_TO_CH2

Take the spindle tool with holder 2.

LL SUB_ARM_BACK

Retract the changer arm.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

LL SUB_CH2_TO_MZ

Start sending the changer arm to the magazine to leave the tool.

V.PLC.M[1100]=1

Order the PLC to activate the TCHANGEOK mark "telling" the tool manager that it can continue executing.

$BREAK

$CASE 7

Leave the spindle tool on the ground and take another one from the magazine.

Previously, While machining (when executing the T), the tool is taken from the magazine to holder 1.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_ARM_TO_CHG

Take the changer arm to the change point.

LL SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

LL SUB_ARM_BACK

Retract the changer arm.

$BREAK

$CASE 8

Leave the spindle tool on the ground and take another one from ground.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

$BREAK

$CASE 9

Take a ground tool to the magazine going through the spindle.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_GR_TO_SPD

Insert the ground tool in the spindle.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_ARM_TO_CHG

Take the changer arm to the change point.

LL SUB_SPD_TO_CH2

Take the spindle tool with holder 2.

LL SUB_ARM_BACK

Retract the changer arm.

LL SUB_CH2_TO_MZ

Start sending the changer arm to the magazine to leave the tool.

V.PLC.M[1100]=1

Order the PLC to activate the TCHANGEOK mark "telling" the tool manager that it can continue executing.

$BREAK

$CASE 10

Take a tool from the magazine and leave on the ground going through the spindle.

LL SUB_MZ_TO_CH1

Take the tool from the magazine with holder 1.

LL SUB_SPD_AUTCHG

Move the spindle to the automatic tool change point

LL SUB_ARM_TO_CHG

Take the changer arm to the change point.

LL SUB_CH1_TO_SPD

Take the tool of holder 1 to the spindle.

LL SUB_ARM_BACK

Retract the changer arm.

LL SUB_SPD_GMCHG

Move the spindle to the manual tool change point.

LL SUB_SPD_TO_GR

Remove the tool from the spindle.

$BREAK

$ENDSWITCH

End of analysis of type of operation.

$WHILE V.TM.MZWAIT == 1
$ENDWHILE

Wait for the tool manager.

#RET

End of M06 subroutine.

8.10.4 Basic PLC programming

When executing the -T- function

When executing the T function, the tool manager sends to the PLC the code: TMOPERATION=11 to take the next tool in the arm and approach it to the spindle while machining.

```
DFU TMOPSTROBE AND CPS TMOPERATION EQ 11 = ···
```

The following operations must be carried out:

Send the changer arm to the magazine.

If LEAVEPOS indicates a magazine position, leave the tool of holder 1 in that position and activate the CH1TOMZ mark.

Take the tool of the TAKEPOS position of the magazine with holder 1. Activate the MZTOCH1 mark to "tell" the tool manager that the tool has been taken.

When executing the M06 function

When executing the M06 function, the tool manager sends to the PLC, in the TMOPERATION register, the code for the operation to be carried out.

```
DFU TMOPSTROBE = MOV TMOPERATION R101
```

This instruction transfers the TMOPERATION value to register R101 so it is managed by the M06 subroutine.

Every time the M06 subroutine ends an action, it lets the PLC know so it activates the relevant mark of the tool manager.

```
DFU M1100 = SET TCHANGEOK
DFD TCHANGEOK = RES M1100
```

Continue executing the program.

```
DFU M1101 = SET MZTOCH1
DFD MZTOCH1 = RES M1101
```

The tool has been taken from the magazine to holder 1.

```
DFU M1102 = SET CH1TOSPDL
DFD CH1TOSPDL = RES M1102
```

The tool has been taken from holder 1 to the spindle.

```
DFU M1104 = SET SPDLTOCH2
DFD SPDLTOCH2 = RES M1104
```

The tool has been taken from the spindle to holder 2.

```
DFU M1107 = SET SPDLTOGR
DFD SPDLTOGR = RES M1107
```

The tool has been taken from the spindle to ground.

```
DFU M1108 = SET GRTOSPDL
DFD GRTOSPDL = RES M1108
```

The tool has been taken from ground to the spindle.

The M06 subroutine uses the following M functions to "tell" the PLC which movements it must carry out.

M101	Select in the magazine the position indicated by TAKEPOS and take the tool with holder 1.
M102	Take the tool of holder 1 to the spindle.
M104	Take the spindle tool with holder 2.
M106	Start sending the changer arm to the magazine to leave the tool in holder 2.
M121	Take the changer arm to the change point.
M122	Take the changer arm to the magazine.
M123	Retract the changer arm.

Programming it depends on the type of machine. The auxiliary functions will be completed after executing the requested movement.

Treatment of the M106 function:

- Completed when the changer arm exits the collision zone and machining is possible.
- Activate the CH2TOMZ mark when the tool has been left to let the tool manager know that the tool has been taken from holder 2 to the magazine".

Certain operations require using the information transferred by the tool manager in the following registers:

LEAVEPOS	This register indicates the magazine position to leave the tool.
TAKEPOS	This register indicates the magazine position of the tool to be taken (picked up).

Manager emergency signal

Treatment of the tool manager emergency signal.

```
DFU B11KEYBD1 = SET SETTMEM
```

Pressing the USER12 key activates the emergency.

```
TMINEM = B11KEYLED1
```

The lamp of the USER12 key turns on when there is an emergency.

```
TMINEM AND DFU B12KEYBD1 = SET RESTMEM
```

Pressing the USER13 key removes the emergency.

CNC VARIABLES

9

9.1 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.

9.1.1 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.

9.1.2 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

9.1.3 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.

9.2 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 326.

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

9.2.1 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 326.

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

9.3 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 326.

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

9.3.1 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 326.

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	—

9.4 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)

9.5 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

9.6 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

9.7 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

9.8 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.

9.9 User tables related

These variables are read/write (R/W) synchronous and are evaluated during execution.

They have generic names.

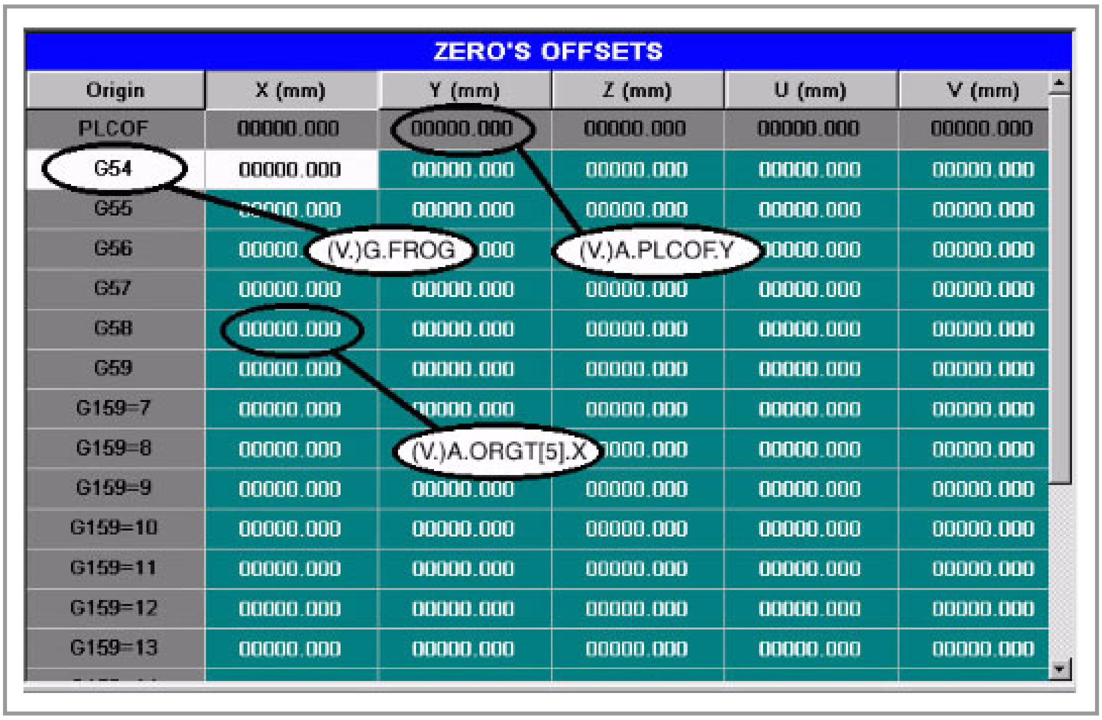
- Replace the "n" character with the channel number, maintaining the brackets. The first channel is identified with the number 1, "0" is not a valid number.
- Replace the letters "m" and "i" with a number, keeping the brackets.
- Replace the "Xn" character by the name, logic number or index in the channel 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]	

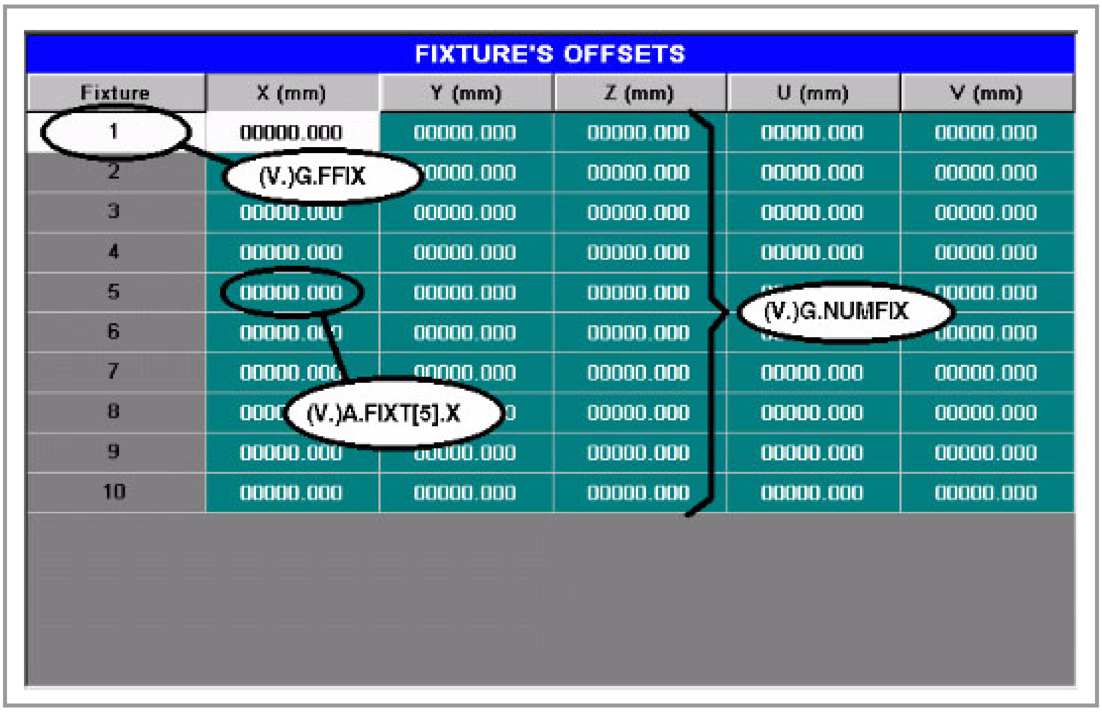
ZERO OFFSET TABLE		Lin Rot	Spd	PRG	PLC	INT	Exec
(V.)G.FORG	First zero offset in the table	—	—	R	R	R	Yes
(V.)G.NUMORG	Number of zero offsets in the table	—	—	R	R	R	Yes
(V.)[n].A.ORG.Xn	Offset of current origin for the Xn axis	Yes	No	R	R	R	No
(V.)[n].A.ORGT[i].Xn	Offset of [i] origin for the Xn axis	Yes	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



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
```

9.10 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.

9.10.1 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

9.11 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.

9.12 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.

9.13 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

9.14 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.

9.15 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

9.16 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 326.

9.17 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 326.

9.18 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 326.

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 326.

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

9.19 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.

9.20 Alphabetical listing of variables

9.

CNC VARIABLES

Alphabetical listing of variables

FAGOR

CNC 8070

(SOFT V02.0x)

APPENDIX

CNC GENERAL CHARACTERISTICS

- PC-bassed open system.
- Windows® XP operating system.
- Full customizing.
  - INI configuration files.
  - FGUIM visual configuration tool.
  - Visual Basic®, Visual C++®, etc.
  - Internal databases in Microsoft® Access.
  - OPC compatible interface
- Integrated PLC.
  - Up to 1024 digital inputs.
  - Up to 1024 digital outputs.
  - Up to 8192 marks.
  - Up to 1024 registers.
  - Up to 256 timers.
  - Up to 256 counters.
  - Unlimited number of symbols.
- Block processing time < 1 ms.
- PLC execution time < 1 ms/K.
- SERCOS® and CAN field bus.
- Up to 28 axes and 3 handwheels.
- Up to 4 spindles.
- Up to four execution channels. The axes and spindles may be distributed at will between the channels.
- Up to four tool magazines.
- Digital (Sercos) and analog drives may be used.

CNC MAINTENANCE

Cleaning

The accumulated dirt inside the unit may act as a screen preventing the proper dissipation of the heat generated by the internal circuitry which could result in a harmful overheating of the CNC and, consequently, possible malfunctions.

On the other hand, accumulated dirt can sometimes act as an electrical conductor and shortcircuit the internal circuitry, especially under high humidity conditions.

To clean the operator panel and the monitor, a smooth cloth should be used which has been dipped into de-ionized water and /or non abrasive dish-washer soap (liquid, never powder) or 75º alcohol.

Do not use highly compressed air to clean the unit because it could generate electrostatic discharges.

The plastics used on the front panel are resistant to :

- Grease and mineral oils.
- Bases and bleach.
- Dissolved detergents.
- Alcohol.
- Avoid the action of solvents such as Chlorine hydrocarbons, Benzole, Esters and Ether which can damage the plastics used to make the unit's front panel.

Preventive inspection

If the CNC does not turn on when actuating the start-up switch, verify the connections.

- 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 handling these connectors (I/O, feedback, etc.), make sure that the unit is not connected to main AC power.

Fagor Automation shall not be held responsible for any material or physical damage derived from the violation of these basic safety requirements.

SUMMARY OF CNC MACHINE PARAMETERS

General machine parameters

General machine parameters. Channels

Feedrate override .... ***Page 35***

MAXOVR	Maximum axis override (%)
RAPIDOVR	Override acts in G00 (from 0 to 100%)
FEEDND	Apply the programmed feedrate to all the axes of the channel
IMOVEMACH	Movement of the independent axis referred to machine coordinates

Related to subroutines .... ***Page 37***

SUBTABLE	OEM-subroutines table
TOOLSUB	Name of the subroutine associated with T
REFPSUB (G74)	Subroutine associated with function G74
OEMSUB (G18x)	Subroutines associated with functions G180 through G189
SUBPATH	Path of program subroutines

Tabletop probe position .... ***Page 38***

PROBEDATA	Channel related probe parameters
PRB1MAX	Maximum probe coordinate (abscissa axis)
PRB1MIN	Minimum probe coordinate (abscissa axis)
PRB2MAX	Maximum probe coordinate (ordinate axis)
PRB2MIN	Minimum probe coordinate (ordinate axis)
PRB3MAX	Maximum probe coordinate (axis perpendicular to the plane)
PRB3MIN	Minimum probe coordinate (axis perpendicular to the plane)

Machine parameters for the axes

Belonging to the channel .... ***Page 40***

AXISEXCH	Channel changing permission	(L R S) (A S X)

Type of axis and drive .... ***Page 40***

AXISTYPE	Type of axis	(L R S) (A S X)
DRIVETYPE	Drive type	(L R S) (A S X)
SERCOSDATA	SERCOS drive data	(L R S) (S)
DRIVEID	Sercos drive address	(L R S) (S)
OPMODEP	Sercos drive operation mode	(L R S) (S)
FBACKSRC	Type of feedback	(L R S) (S)

Hirth axis .... ***Page 41***

HIRTH	Hirth axis	(L R) (A S X)
HPITCH	Hirth axis pitch	(L R) (A S X)

Axis configuration for lathe type machines .... ***Page 42***

FACEAXIS	Face axis (lathe)	(L) (A S X)
LONGAXIS	Longitudinal axis (lathe)	(L) (A S X)

Rotary axes .... ***Page 42***

AXISMODE	Operating mode of the rotary axis	(R) (A S X)
UNIDIR	Unidirectional rotation	(R) (A S X)
SHORTESTWAY	Via shortest way	(R) (A S X)

Rotary axes and spindle .... ***Page 44***

MODCOMP	Module compensation	(R S) (A Ss X)
CAXIS	Works as a "C" axis	(R S) (A S X)
CAXSET	Work set for "C" axis	(R S) (A S X)

Spindle .... ***Page 44***

AUTOGEAR	Automatic gear change	(S) (A S X)
LOSPDLIM	lower percentage for rpm OK	(S) (A S X)
UPSPDLIM	Upper percentage for rpm OK	(S) (A S X)
SPDLTIME	Estimated time for an S function	(S) (A S X)
SPDLSTOP	M2, M30 and Reset stop the spindle	(S) (A S X)
SREVM05	G84. Reversal stops the spindle	(S) (A S X)
STEPOVR	Spindle Override step	(S) (A S X)
MINOVR	Minimum spindle override	(S) (A S X)
MAXOVR	Maximum spindle override	(S) (A S X)

Software axis limits .... ***Page 46***

LIMIT+	Positive software limit	(L R) (A S X)
LIMIT-	Negative software limit	(L R) (A S X)
SWLIMITTOL	Software limits tolerance	(L R) (A S X)

Machine parameters for the axes. Work sets

Machine parameters for JOG mode

Machine parameters for the M function table

Machine parameters for kinetics

Machine parameters for the magazine

Machine parameters for HMI (Interface)

OEM machine parameters

SUMMARY OF PLC PROGRAMMING COMMANDS

RESOURCES AVAILABLE AT THE PLC

Inputs (I1··1024)

Outputs (O1··1024)

Marks (M1··8192)

Message marks (MSG1··256)

Error marks (ERR1··256)

Clocks (CLK)

CNC-PLC marks

Timers (T1··256)

Counters (C1··256)

Registers (R1··1024)

CNC-PLC registers

The register value may be treated as a decimal or hexadecimal ("S") number.

It is also possible to refer to a register bit using the letter B (0··31) R (1··1024).

CLK1	1ms	CLK100	100ms	CLK1000	1s
CLK2	2ms	CLK200	200ms	CLK2000	2s
CLK4	4ms	CLK400	400ms	CLK4000	4s
CLK8	8ms	CLK800	800ms	CLK8000	8s
CLK16	16ms	CLK1600	1.6s	CLK16000	16s
CLK32	32ms	CLK3200	3.2s	CLK32000	32s
CLK64	64ms	CLK6400	6.4s	CLK64000	64s
CLK128	128ms	CLK12800	12.8s	CLK128000	128s

DIRECTING INSTRUCTIONS

PRG	Main module
PE t	Periodic module. It is executed every "t" milliseconds.
CY1	First Cycle module.
END	End of module.
L	Label
SUB	Subroutine definition.
DEF	Symbol definition.
PDEF	External symbol definition.
REA	The consultations will use real values.
IMA	The consultations will use image values.
NOMONIT	No PLC program monitoring.
EXTERN	External subroutine definition.

CONSULTING INSTRUCTIONS

- Simple consulting instructions.

I1··1024	Inputs
O1··1024	Outputs
M1··8192	Marks
MSG1··256	Messages
ERR1··256	Errors
T1··256	Timers (status)
C1··256	Counters (status)
B0··31 R1··1024	Register bit
CLK	Clocks
M <CNC-PLC>	Marks for CNC-PLC communication

- Flank detection instructions.

DFU	Up flank detection.
DFD	Down flank detection.

DFU	I1··1024
DFD	O1··1024
	M1··8192
	MSG1··256
	ERR1··256
	B0··31 R1··1024
	CLK
	M <CNC-PLC>

- Comparing instructions.

CPS	For comparisons.

CPS	T1··256	GT	T1··256
	C1··256	GE	C1··256
	R1··1024	EQ	R1··1024
	R CNC-PLC	NE	R CNC-PLC
	#	LE	#
		LT	

OPERATORS

NOT	Negates the result of the consultation.
AND	Logic function "AND".
OR	Logic function "OR”.
XOR	Logic "Exclusive OR" function.
\	Line feed.
( )	Consulting instruction whose value is always "1".

ACTION INSTRUCTION.

- Assignment binary action instructions.

= I 1/1024	= O 1/1024	= M 1/8192
= MSG 1/256	= ERR 1/256	= TEN 1/256
= TRS 1/256	= TGn 1/256 #/R	= CUP 1/256
= CDW 1/256	= CEN 1/256	= CPR 1/256 #/R
= B 0/31 R 1/499	= CNC-PLC mark	

- Conditional binary action instructions.

= SET	If expression = "1", it sets the resource to "1".
= RES	If expression = "1", it sets the resource to "0".
= CPL	If expression = "1", it complements the resource.

= SET	I1··1024
= RES	O1··1024
= CPL	M1··8192
	MSG1··256
	ERR1··256
	B0··31 R1··1024
	M <CNC-PLC>

- Sequence breaking action instructions.

= JMP L	Unconditional jump.
= CAL	Call to a subroutine.
= RET	Return or end of a subroutine.

- Arithmetic action instructions.

= MOV	Move.

	Origin	Destination	Origin code	Destination code	Bits to transmit
= MOV	I1/1024	I1/1024	0(Bin)	0(Bin)	32
	O1/1024	O1/1024	1(BCD)	1(BCD)	28
	M1/8192	M1/8192			24
	MSG1/256	MSG1/256			20
	ERR1/256	ERR1/256			16
	T1/256	R1/1024			12
	C1/256	R <CNC-PLC>			8
	R1/1024				4
	R <CNC-PLC>				
	#				

= NGU R1··1024 Complements the bits of a register.

= NGS R1··1024 Register sign change.

= ADS Add.

= SBS Substract.

= MLS Multiplication.

= DVS Division.

= MDS Module or remainder of a division.

	Operand	Operand	Result
= ADS	R1··1024	R1··1024	R1··1024
= SBS	R <CNC-PLC>	R <CNC-PLC>	R <CNC-PLC>
= MLS	#	#	
= DVS			
= MDS			

- Logic action instructions.

= AND Logic operation "AND".

= OR Logic operation "OR".

= XOR Logic operation "XOR".

= AND	R1··1024	R1··1024	R1··1024
= OR	R <CNC-PLC>	R <CNC-PLC>	R <CNC-PLC>
= XOR	#	#	

= RR 1/2 Right-hand register rotation.

= RL 1/2 Left-hand register rotation.

	Origin	Repetion Nr.	Destination
= ADS	R1··1024	R1··1024	R1··1024
= SBS	R <CNC-PLC>	R <CNC-PLC>	R <CNC-PLC>
= MLS		0··31	
= DVS			
= MDS			

- Specific action instructions.

= ERA Erases or resets a group of resources.

= ERA	I1··1024	1··1024
	O1··1024	1··1024
	M1··8192	1··8192
	MSG1··256	1··256
	ERR1··256	1··256
	T1··256	1··256
	C1··256	1··256
	R1··1024	1··1024

=CNCRD Reading of internal variables.

CNCRD (Variable, R1··1024, M1··8192)

=CNCWR Writing of internal variables.

CNCRD (Variable, R1··1024, M1··8192)

=PAR Parity of a register.

= PAR	R1··1024 R CNC-PLC	M1··8192 MSG1··256 ERR1··256 M CNC-PLC

- Action instructions of the electronic cam

= CAM ON Activate de electronic cam

```
CAM ON (cam, master/"TIME", slave, master_off, slave_off, range_master,
range_slave, type)
```

= CAM OFF Cancel de electronic cam

```
CAM OFF (slave)
```

- Actions instructions for independent axes

= MOVE ABS Absolute positioning move.

```
MOVE ABS (axis, pos, feed, blend)
```

= MOVE ADD Incremental positioning move.

```
MOVE ADD (axis, pos, feed, blend)
```

= MOVE INF Infinite (endless) positioning move.

```
MOVE INF (axis, direction, feed, blend)
```

= FOLLOW ON Activates the synchronization movement.

```
FOLLOW ON (master, slave, nratio, dratio, synctype)
```

= FOLLOW OFF Cancels the synchronization movement.

```
FOLLOW OFF (slave)
```

LOGIC CNC INPUTS AND OUTPUTS

CONSULTATION - GENERAL

CNCREADY	(M)	The CNC is ready (No problems).
START	(M)	The CYCLE START key has been pressed.
FHOUT	(M)	The execution of the program is stopped.
RESETOUT	(M)	A Reset has occurred (by key or by PLC).
_ALARM	(M)	Alarm or emergency generated by the CNC.
MANUAL	(M)	Manual operating mode selected.
AUTOMAT	(M)	Automatic operating mode selected.
MDI	(M)	MDI mode selected.
SBOUT	(M)	Single-block mode selected.
INCYCLE	(M)	Block in execution or axis in motion.
RAPID	(M)	Rapid positioning (traverse, G0).
ZERO	(M)	Home search (G74).
PROBE	(M)	Probing (G100).
THREAD	(M)	Threading (G33).
TAPPING	(M)	Tapping canned cycle (G84).
RIGID	(M)	Rigid tapping (G63).
CSS	(M)	Constant Surface Speed (G96).
MFUN1..7	(R)	Auxiliary M functions.
MSTROBE	(M)	Execute auxiliary M function.
HFUN1..7	(R)	Auxiliary H functions.
HSTROBE	(M)	Execute auxiliary H function.
SFUN1	(R)	Selected spindle speed.
SSTROBE	(M)	Select new spindle speed.
INTEREND	(M)	Theoretical movement ended.
INPOS	(M)	All axes in position.
DMxx	(M)	Mxx in the program history.
BLKSEARCH	(M)	"Block search" option.
ADVINPOS	(M)	Anticipation signal for axes in position.
FREE	(M)	CNC ready to accept a block using CNCEX.
WAITOUT	(M)	The CNC is waiting for a synchronization signal from a channel.
SYNC	(R)	Spindle to be used for synchronization.
MMCWDG	(M)	Status of the operating system.

CONSULTATION - AXES AND SPINDLES

ENABLE(axis)	(M)	Enable axis or spindle movement.
DIR(axis)	(M)	Axis moving in the negative direction.
REFPOIN(axis)	(M)	Home search done.
DRSTAF(axis)	(M)	With Sercos. Drive status
DRSTAS(axis)	(M)	With Sercos. Drive status
INPOS(axis)	(M)	Axis or spindle in position.
LUBR(axis)	(M)	The axis or spindle must be lubricated.
HIRTHON(axis)	(M)	Axis working as Hirth axis.
MATCH(axis)	(M)	Hirth axis positioned properly.
PARK(axis)	(M)	Park the axis.
UNPARK(axis)	(M)	Unpark the axis.

CONSULTATION - SPINDLE

CAXIS	(M)	Spindle working as "C" axis.
REVOK	(M)	Real revolutions = programmed.

CONSULTATION - INDEPENDENT INTERPOLATOR

IBUSY(axis)	(M)	There is an instruction pending execution.
IFREE(axis)	(M)	The PLC is ready to accept a motion block.
IFHOUT(axis)	(M)	The execution is interrupted.
IEND(axis)	(M)	The final position has been reached.
INSYNC(axis)	(M)	Synchronism has been reached.

CONSULTATION - TOOL MANAGER

TMOPERATION	(R)	Type of operation.
TMOPSTROBE	(M)	Execute operation indicated in TMOPERATION.
LEAVEPOS	(R)	Magazine position to leave the tool.
TAKEPOS	(R)	Magazine position to pick up the tool.
NEXTPOS	(R)	Magazine position for next tool.
TWORNOUT	(M)	Rejected tool (worn out, real life > max.).
TMINEM	(M)	Emergency of tool manager.
MZID	(R)	Magazine containing the requested tool.

CONSULTATION - KEYS

KEYBD1, 2	(R)	Indicate which key has been pressed.

MODIFIABLE - GENERAL

_EMERGEN	(M)	Stops (=0) axis feed and spindle speed. It displays an error message on the screen.
_STOP	(M)	Stops (=0) part program execution and keeps the spindle turning.
_FEEDHOL	(M)	Stops (=0) axis feed temporarily and keeps the spindle turning.
_XFERINH	(M)	Inhibits (=0) the execution of the next block, but it ends the one that is being executed.
CYSTART	(M)	Start (=1) program execution.
SBLOCK	(M)	Operate (=1) in Single block mode.
MANRAPID	(M)	Rapid jog (=1).
OVRCAN	(M)	Set (=1) feedrate at 100%.
LATCHM	(M)	JOG keys. The axis moves while the key is pressed (=0) or until another one is pressed (=1).
RESETIN	(M)	Assume (up flank) the machining conditions set by machine parameters.
AUXEND	(M)	M and S function management.
TIMERON	(M)	Enable timer (=1).
PLCREADY	(M)	PLC ready (=1).
BLKSKIP1	(M)	Do not execute (=1) blocks with block skip indicator "/".
M01STOP	(M)	Do not ignore (=1) conditional stop (M01).
NOWAIT	(M)	It cancels the synchronizations with the channel.
DISCROSS1..9	(M)	It disables the cross compensation table.

MODIFIABLE - AXES AND SPINDLES

LIMITPOS(axis)	(M)	Positive limit overrun (=1).
LIMITNEG(axis)	(M)	Negative limit overrun (=1).
DECEL(axis)	(M)	Change (=1) home searching feedrate from fast to slow.
INHIBIT(axis)	(M)	Inhibit (=1) axis or spindle movement.
AXISPOS(axis)	(M)	Movement (=1) in the positive direction in JOG mode.
AXISNEG(axis)	(M)	Movement (=1) in the negative direction in JOG mode.
SERVO(axis)ON	(M)	Enable (=1) axis or spindle movement. When (=0), it stops the axis and the spindle displaying an error message.
DRO(axis)	(M)	When (=1) and SERVOnON (=0) it works as a DRO (open loop and ignoring the following error).
SPENA(axis)	(M)	"Speed enable" signal of the drive (Sercos).
DRENA(axis)	(M)	"Drive enable" signal of the drive (Sercos).
LIM(axis)OFF	(M)	Ignore (=1) software limits set with G198 and G199.
PARKED(axis)	(M)	The axis is parked (=1).
LUBRENA(axis)	(M)	Use (=1) axis lubricating feature.
LUBROK(axis)	(M)	Axis lubrication done (=1).
DIFFCOMP(axis)	(M)	Correct the coordinate difference between the two axes of a Gantry axis.

MODIFIABLE - SPINDLE

SPDLEREV	(M)	Reverse (=1) spindle turning direction.
GEAR1,2,3,4	(M)	Selected gear (=1).
PLCCNTL	(M)	Spindle controlled by the PLC (=1).
SANALOG	(R)	Spindle analog voltage to be applied.

MODIFIABLE - INDEPENDENT INTERPOLATOR

IRESET(axis)	(M)	It interrupts the running independent-axis instruction and eliminates the pending instructions. It interrupts cam synchronization. It sets the initial conditions at the interpolator.
IABORT(axis)	(M)	It interrupts the running positioning movement and eliminates the pending positioning movements.

MODIFIABLE - TOOL MANAGER

SETTMEM	(M)	Activate (=1) Tool manager emergency.
RESTMEM	(M)	Cancel (=0) tool manager emergency.
CUTTINGON	(M)	Tool in execution (=1).
TREJECT	(M)	The tool must be rejected (=1).
MZTOCH1	(M)	(=1) Tool taken from magazine to changer arm 1.
CH1TOSPDL	(M)	(=1) Tool taken from changer arm 1 to the spindle.
SPDLTOCH1	(M)	(=1) Tool taken from spindle to tool changer arm 1.
SPDLTOCH2	(M)	(=1) Tool taken from spindle to tool changer arm 2.
CH1TOMZ	(M)	(=1) Tool taken from changer arm 1 to the magazine.
CH2TOMZ	(M)	(=1) Tool taken from changer arm 2 to the magazine.
SPDLTOGR	(M)	(=1) Tool taken from the spindle to ground.
GRTOSPDL	(M)	(=1) Tool taken from ground to the spindle.
MZTOSPDL	(M)	(=1) Tool taken from magazine to spindle.
SPDLTOMZ	(M)	(=1) Tool taken from spindle to magazine.
MZROT	(M)	The turret has been rotated (=1).
TCHANGEOK	(M)	Tool change done (=1).
MZPOS	(R)	Current tool magazine position.

MODIFIABLE - KEYS

KEYLED1, 2	(R)	Turn the key lights on (=1).
KEYDIS1, 2, 3	(R)	Inhibit keys (=1).

SUMMARY OF CNC VARIABLES

Related to general machine parameters

(V.)MPG.NCHANNEL	Number of CNC channels.

(V.)MPG.NAXIS	Number of axes governed by the CNC
(V.)MPG.AXISNAMEx	Name of the "n" logic axis
(V.)MPG.TMASTERAXIS[i]	Tandem [i]. Logic number of the master axis
(V.)MPG.TSLAVEAXIS[i]	Tandem [i]. Logic number of the slave axis
(V.)MPG.TORQDIST[i]	Tandem [i]. Torque distribution
(V.)MPG.PRELOAD[i]	Tandem [i]. Preload
(V.)MPG.PRELFITI[i]	Tandem [i]. Time to apply the preload
(V.)MPG.TPROGAIN[i]	Tandem [i]. Proportional gain
(V.)MPG.TINTIME[i]	Tandem [i]. Integral gain
(V.)MPG.TCOMPLIM[i]	Tandem [i]. Compensation Limit
(V.)MPG.MASTERAXIS[i]	Gantry [i]. Logic number of the master axis
(V.)MPG.SLAVEAXIS[i]	Gantry [i]. Logic number of the slave axis
(V.)MPG.WARNCOUPE[i]	Gantry [i]. Maximum difference to issue a warning
(V.)MPG.MAXCOUPE[i]	Gantry [i]. Maximum difference allowed
(V.)MPG.DIFFCOMP[i]	Gantry [i]. Error difference compensation.

(V.)MPG.NSPDL	Number of spindles governed by the CNC
(V.)MPG.SPDLNAMEx	Name of the "x" spindle

(V.)MPG.LOOPTIME	Loop time
(V.)MPG.PRGFREQ	Frequency of the PRG module (in cycles)

(V.)MPG.SERBRATE	Sercos transmission speed
(V.)MPG.SERPOWSE	Sercos optical power
(V.)MPG.CANLENGTH	Can bus cable length (in meters)

(V.)MPG.INCHES	Default work units

(V.)MPG.MAXLOCP	Maximum local arithmetic parameter
(V.)MPG.MINLOCP	Minimum local arithmetic parameter
(V.)MPG.MAXGLBP	Maximum global arithmetic parameter
(V.)MPG.MINGLBP	Minimum global arithmetic parameter
(V.)MPG.ROPARMAX	Maximum global read-only arithmetic parameter
(V.)MPG.ROPARMIN	Minimum global read-only arithmetic parameter
(V.)MPG.MAXCOMP	Maximum common arithmetic parameter
(V.)MPG.MINCOMP	Maximum common arithmetic parameter

(V.)MPG.MOVAXIS[m]	Table [m]. Master axis
(V.)MPG.COMPAXIS[m]	Table [m]. Axis to be compensated
(V.)MPG.NPCROSS[m]	Table [m]. Number of points
(V.)MPG.TYPCROSS[m]	Table [m]. Type of compensation
(V.)MPG.BIDIR[m]	Table [m]. Bi-directional compensation
(V.)MPG.REFNEED[m]	Table [m]. Mandatory home search
(V.)MPG.POSITION[m][i]	Table [m]. Master axis position for point [i]
(V.)MPG.POSERROR[m][i]	Table [m]. Error of point [i] in the positive direction
(V.)MPG.NEGERROR[m][i]	Table [m]. Error of point [i] in the negative direction

Channel related

Related to axis machine parameters

Related to gear parameters

Related to jog mode parameters

Related to "M" function parameters

Related to kinematic parameters

Related to magazine parameters

(V.)TM.NTOOLMZ	Number of tool magazines
(V.)TM.MZGROUND[z]	Ground tools allowed
(V.)TM.MZSIZE[z]	Magazine size
(V.)TM.MZRANDOM[z]	Random magazine
(V.)TM.MZTYPE[z]	Type of magazine
(V.)TM.MZCYCLIC[z]	Cyclic tool changer
(V.)TM.MZOPTIMIZED[z]	Tool management
(V.)TM.MZM6ALONE[z]	Action when executing an M6 without a tool

Related to OEM parameters

(V.)MTB.PLCDATASIZE	Size of the PLC's shared data area

(V.)MTB.SIZE	Number of OEM parameters
(V.)MTB.P[i]	Value of the OEM parameter [i]
(V.)MTB.PF[i]	Value of the OEM parameter [i] Value per 10000

(V.)DRV.SIZE	Number of variables to be consulted at the drive
(V.)DRV.name	Value of the variable

User tables related

(V.)G.FORG	First zero offset in the table
(V.)G.NUMORG	Number of zero offsets in the table
(V.)[n].A.ORG.Xn	Offset of current origin for the Xn axis
(V.)[n].A.ORGT[i].Xn	Offset of [i] origin for the Xn axis
(V.)[n].A.PLCOF.Xn	Offset of PLC origin for the Xn axis

(V.)G.FFIX	First fixture of the table
(V.)G.NUMFIX	Number of fixtures in the table
(V.)[n].G.FIX	Number of current fixture
(V.)[n].A.FIX.Xn	Offset of current fixture for Xn axis
(V.)[n].A.FIXT[i].Xn	Offset of [i] fixture for the Xn axis

(V.)G.CUP[i]	Value of the common arithmetic parameter [i]
(V.)G.CUPF[i]	Value of the common arithmetic parameter [i]. Value per 10000
(V.)[n].G.GUP[i]	Value of the global arithmetic parameter [i]
(V.)[n].G.GUPF[i]	Value of the global arithmetic parameter [i]. Value per 10000
(V.)[n].G.LUPACT[i]	Value of local arithmetic parameter [i] active level
(V.)[n].G.LUPm[i]	Value of local arithmetic parameter [i] of m level
(V.)[n].G.LUPmF[i]	Value of local arithmetic parameter [i] of m level. Value per 1000

Tool related

(V.)TM.T[z][j]	Tool in the [j] position of the [z] magazine
(V.)TM.P[z][m]	Position of the [m] tool in the [z] magazine
(V.)[n].TM.TOOL	Number of the active tool
(V.)[n].TM.TOD	Number of the active tool offset
(V.)[n].TM.NXTOOL	Number of the next tool
(V.)[n].TM.NXTOD	Number of the next tool offset

Monitoring .... Page 349

(V.)[n].TM.TOMON[i]	Monitoring type of the [i] offset of the active tool
(V.)TM.TOMONT[m][i]	Monitoring type of the [i] offset of the [m] tool
(V.)[n].TM.TLFN[i]	Maximum life of the [i] offset of the active tool
(V.)TM.TLFNT[m][i]	Maximum life of the [i] offset of the [m] tool
(V.)[n].TM.TLFR[i]	Real life of the [i] offset of the active tool
(V.)TM.TLFRT[m][i]	Real life of the [i] offset of the [m] tool
(V.)[n].TM.REMLIFE	Remaining life of the active tool

Magazine... Page 349

(V.)[n].TM.TSTATUS	Status of the active tool
(V.)TM.TSTATUST[m]	Status of the [m] tool
(V.)[n].TM.TLFF	Family of the active tool
(V.)TM.TLFFT[m]	Family of the [m] tool
(V.)[n].TM.ACTUALMZ	Tool Magazine being used by each channel
(V.)TM.MZRESPECTSIZE[z]	In a random magazine [z], the tool always in the same position.
(V.)TM.MZACTUALCH[z]	Channel being used by the tool magazine [z]

Geometry... Page 350

(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
(V.)[n].TM.TOFL1	Offset of the tool in the first axis of the channel
(V.)[n].TM.TOFL2	Offset of the tool in the second axis of the channel
(V.)[n].TM.TOFL3	Offset of the tool in the third axis of the channel
(V.)TM.TOFLT[m][i].Xn	Xn axis deviation of the [i] offset of the [m] tool
(V.)[n].TM.TOFLW[i].Xn	Xn axis deviation of the [i] offset of the active tool
(V.)[n].TM.TOFLW1	Wear offset of the tool in the first axis of the channel
(V.)[n].TM.TOFLW2	Wear offset of the tool in the second axis of the channel
(V.)[n].TM.TOFLW3	Wear offset of the tool in the third axis of the channel
(V.)TM.TOFLWT[m][i].Xn	Xn axis deviation wear of the [i] offset of the [m] tool

"Custom" data .... Page 350

(V.)[n].TM.TOTP1	Additional parameter 1 of the active tool
(V.)[n].TM.TOTP2	Additional parameter 2 of the active tool
(V.)[n].TM.TOTP3	Additional parameter 3 of the active tool
(V.)[n].TM.TOTP4	Additional parameter 4 of the active tool
(V.)TM.TOTP1T[i]	Additional parameter 1 of the [i] tool
(V.)TM.TOTP2T[i]	Additional parameter 2 of the [i] tool
(V.)TM.TOTP3T[i]	Additional parameter 3 of the [i] tool
(V.)TM.TOTP4T[i]	Additional parameter 4 of the [i] tool

Tool manager .... Page 350

(V.)[n].TM.MZSTATUS	Status of the tool manager
(V.)[n].TM.MZRUN	Tool manager running
(V.)[n].TM.MZMODE	Operating mode of the tool manager
(V.)[n].TM.MZWAIT	Tool manager executing a maneuver

Variables only used during block preparation

(V.)[n].G.TOOL	Number of the tool being prepared
(V.)[n].G.TOD	Number of tool offset being prepared
(V.)[n].G.NXTOOL	Number of next tool being prepared
(V.)[n].G.NXTOD	Number of next tool offset being prepared
(V.)[n].G.TOR	Radius of the tool offset being prepared
(V.)[n].G.TOI	Radius wear of the tool offset being prepared
(V.)[n].G.TOL	Length of the tool offset being prepared
(V.)[n].G.TOK	Length wear of the tool offset being prepared
(V.)[n].G.TOTIPR	Tip radius of the offset being prepared
(V.)[n].G.TOWTIPR	Tip radius wear of the offset being prepared
(V.)[n].G.TOCUTL	Cutting length of the tool offset being prepared
(V.)[n].G.TOAN	Penetration angle of the tool offset being prepared
(V.)[n].A.TOFL.Xn	Deviation of the active offset on the Xn axis
(V.)[n].A.TOFLW.Xn	Deviation of the active wear offset on the Xn axis
(V.)[n].G.TOFL1	Offset of the tool in the first axis of the channel
(V.)[n].G.TOFL2	Offset of the tool in the second axis of the channel
(V.)[n].G.TOFL3	Offset of the tool in the third axis of the channel
(V.)[n].G.TOFLW1	Wear offset of the tool in the first axis of the channel
(V.)[n].G.TOFLW2	Wear offset of the tool in the second axis of the channel
(V.)[n].G.TOFLW3	Wear offset of the tool in the third axis of the channel
(V.)[n].G.TOMON	Monitoring type of the tool offset being prepared
(V.)[n].G.TLFN	Nominal life of the tool offset being prepared
(V.)[n].G.TLFR	Real life of the tool offset being prepared
(V.)[n].G.REMLIFE	Remaining life of the tool offset being prepared
(V.)[n].G.TSTATUS	Status of the tool being prepared
(V.)[n].G.TLFF	Family of the tool offset being prepared
(V.)[n].G.TOTP1	Additional parameter 1 of the active tool
(V.)[n].G.TOTP2	Additional parameter 2 of the active tool
(V.)[n].G.TOTP3	Additional parameter 3 of the active tool
(V.)[n].G.TOTP4	Additional parameter 4 of the active tool

PLC related

(V.)PLC.STATUS	PLC status

(V.)PLC.I[i]	Status of PLC input [i]
(V.)PLC.O[i]	Status of PLC output [i]
(V.)PLC.M[i]	Status of PLC mark [i]
(V.)PLC.R[i]	Status of PLC register [i]
(V.)PLC.T[i]	Status of PLC timer [i]
(V.)PLC.C[i]	Status of PLC counter [i]
(V.)PLC.signal	Status of exchange signals with CNC

(V.)PLC.symbol	Status of the external symbols defined at the PLC

(V.)PLC.MSG[i]	Status of PLC message [n]
(V.)PLC.PRIORMSG	Active message with the highest priority (the one with the lowest number among the active ones)
(V.)PLC.EMERGMSG	Active emerging message (the one shown at full screen)

(V.)PLC.ERR[i]	Status of PLC error [n]
(V.)PLC.PRIORERR	Active error with the highest priority (the one with the lowest number among the active ones)

(V.)PLC.TIMER	Value of the timer enabled by PLC

Jog mode related

(V.)G.MANMODE	Active for all the axes
(V.)G.CNCMANMODE	At the switch for all of the axes
(V.)PLC.MANMODE	By PLC for all the axes
(V.)[n].A.MANMODE.Xn	Active for the Xn axis
(V.)[n].A.CNCMMODE.Xn	At the switch for the Xn axis
(V.)[n].A.PLCMMODE.Xn	By PLC for the Xn axis

(V.)G.MPGIDX	Active position for all the handwheels
(V.)G.CNCMPGIDX	Position selected at the switch
(V.)PLC.MPGIDX	Position selected by PLC

(V.)G.INCJOGIDX	Active position for all the axes
(V.)G.CNCINCJOGIDX	Position selected by the switch
(V.)PLC.INCJOGIDX	Position selected by PLC

(V.)[n].G.FMAN	JOG feedrate in G94
(V.)[n].G.MANFPR	JOG feedrate in G95

Coordinate related

(V.)[n].A.PPOS.Xn	Programmed coordinates (of the tool tip)
(V.)[n].G.PLPPOS1	Programmed coordinate (of the tool tip) First axis of the channel
(V.)[n].G.PLPPOS2	Programmed coordinate (of the tool tip) Second axis of the channel
(V.)[n].G.PLPPOS3	Programmed coordinate (of the tool tip) Third axis of the channel
(V.)[n].A.FLWE.Xn	Following error of the axis
(V.)[n].A.APOS.Xn	Part coordinates. Real of the tool base
(V.)[n].A.ATPOS.Xn	Part coordinates. Theoretical of the tool base
(V.)[n].A.ATIPPOS.Xn	Part coordinates. Real of the tool tip
(V.)[n].A.ATIPTPOS.Xn	Part coordinates. Theoretical of the tool tip
(V.)[n].A.POS.Xn	Machine coordinates. Real of the tool base
(V.)[n].A.TPOS.Xn	Machine coordinates. Theoretical of the tool base
(V.)[n].A.TIPPOS.Xn	Machine coordinates. Real of the tool tip
(V.)[n].A.TIPTPOS.Xn	Machine coordinates. Theoretical of the tool tip

(V.)[n].A.POS.Sn	Real spindle position
(V.)[n].A.TPOS.Sn	Theoretical spindle position
(V.)[n].A.PPOS.Sn	Programmed spindle position
(V.)[n].A.FLWE.Sn	Spindle following error

Feedrate related

(V.)[n].G.FREAL	Real CNC feedrate
(V.)[n].G.FEED	Active feedrate in G94
(V.)[n].PLC.F	Feedrate by PLC in G94
(V.)[n].G.PRGF	Feedrate by program in G94
(V.)[n].G.FPREV	Active feedrate in G95
(V.)[n].PLC.FPR	Feedrate by PLC in G95
(V.)[n].G.PRGFPR	Feedrate by program in G95

(V.)G.FTIME	Machining time in G93

Related to the spindle speed

Related to the programmed functions

	the second axis
(V.)[n].G.CSMAT12	Offset of the current coordinate system referred to machine zero on the third axis

(V.)[n].G.MEETST[i]	Status of the MEET type [i] mark in the [n] channel
(V.)[n].G.WAITST[i]	Status of the WAIT type [i] mark in the [n] channel
(V.)[n].G.MEETCH[i]	MEET type mark expected by the [n] channel of the [i] channel
(V.)[n].G.WAITCH[i]	WAIT type mark expected by the [n] channel from the [i] channel

(V.)[n].A.FFGAIN.Xn	Active percentage of feed-forward
(V.)[n].A.ACFGAIN.Xn	Active percentage of AC-forward

Related to the independent axes

(V.)[n].G.IBUSY	An independent axis is in execution

(V.)[n].A.IORG.Xn	Offset for the independent axis
(V.)[n].A.IPRGF.Xn	Feedrate programmed in the independent axis
(V.)[n].A.IPPOS.Xn	Coordinate programmed for the independent axis
(V.)[n].A.ITPOS.Xn	Theoretical coordinate of the independent axis

(V.)[n].A.SYNCTOUT.Xn	Maximum time to establish synchronism
(V.)[n].A.SYNCVEL.Xn	Synchronization speed
(V.)[n].A.SYNCPOSW.Xn	Maximum position difference to start correcting it
(V.)[n].A.SYNCVELW.Xn	Maximum velocity difference to start correcting it
(V.)[n].A.SYNCPOSOFF.Xn	Position offset for synchronization
(V.)[n].A.SYNCVELOFF.Xn	Velocity offset for synchronization
(V.)[n].A.GEARADJ.Xn	Fine adjustment of the gear ratio for the synchronization movement

Related to the machine configuration

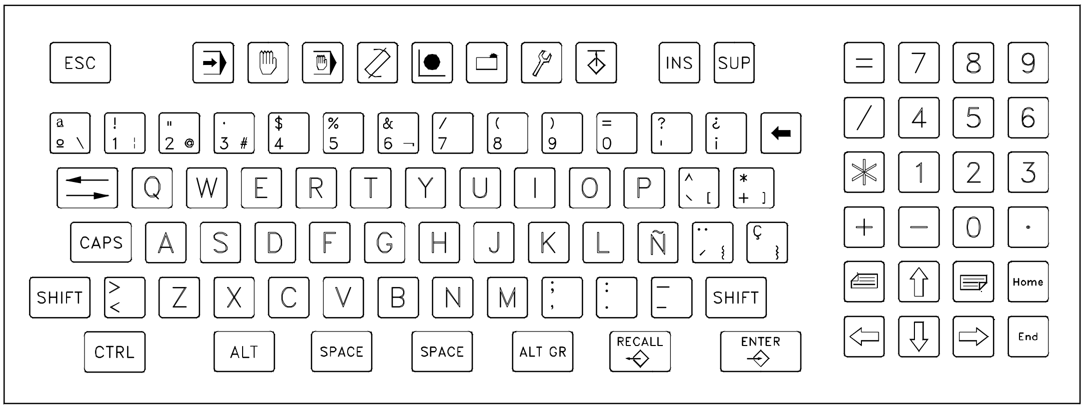
(V.)G.NUMCH	Number of channels
(V.)[n].G.AXISCH	Name the axes of the channel
(V.)[n].A.ACTCH.Xn	Current channel of the axis or of the spindle
(V.)[n].A.ACTIVSET.Xn	Active axis or spindle set
(V.)[n].G.AXIS	Number of axes of the channel
(V.)[n].G.NAXIS	Number of axes of the channel including the empty positions of the yielded axes
(V.)[n].G.AXISNAMEx	Name of the "x" axis of the channel
(V.)G.GAXISNAMEx	Name of the "x" axis of the system
(V.)[n].G.NSPDL	Number of spindles of the channel
(V.)[n].G.SPDLNAMEx	Name of the "x" spindle of the channel
(V.)G.GSPDLNAMEx	Name of the "x" spindle of the system
(V.)[n].G.MASTERSP	Master spindle of the channel

(V.)[n].A.POSLIMIT.Xn	Positive software limit
(V.)[n].A.NEGLIMIT.Xn	Negative software limit
(V.)[n].A.RTPOSLIMIT.Xn	Second positive software travel limit
(V.)[n].A.RTNEGLIMIT.Xn	Second negative software travel limit
(V.)[n].G.SOFTLIMIT	Software limits reached

(V.)[n].A.HEADOF.Xn	Dimension of the kinematics

Other variables

KEY CODES (QWERTY KEYBOARD)



	Press key	Release key		Press key	Release key		Press key	Release key
Alphanumeric keyboard			Y	$15	$95	+	$1B	$9B
0	$0B	$8B	Z	$2C	$AC	´	$28	$A8
1	$02	$82				Ç	$2B	$AB
2	$03	$83	**Numeric keypad**			,	$33	$B3
3	$04	$84	0	$52	$D2	.	$34	$B4
4	$05	$85	1	$4F	$CF	_	$35	$B5
5	$06	$86	2	$50	$D0	<	$56	$D6
6	$07	$87	3	$51	$D1			
7	$08	$88	4	$4B	$CB	**Motion keys**		
8	$09	$89	5	$4C	$CC	[PAGUP]	$E0 $49	$E0 $C9
9	$0A	$8A	6	$4D	$CD	[PAGDN]	$E0 $51	$E0 $D1
A	$1E	$9E	7	$47	$C7	[UP]	$E0 $48	$E0 $C8
B	$30	$B0	8	$48	$C8	[DN]	$E0 $50	$E0 $D0
C	$2E	$AE	9	$49	$C9	[LEFT]	$E0 $4B	$E0 $CB
D	$20	$A0	+	$4E	$CE	[RIGHT]	$E0 $4D	$E0 $CD
E	$12	$92	-	$4A	$CA	[HOME]	$E0 $47	$E0 $C7
F	$21	$A1	*	$37	$B7	[END]	$0E $4F	$E0 $CF
G	$22	$A2	/	$E0 $35	$E0 $B5	[INS]	$E0 $52	$E0 $D2
H	$23	$A3	.	$53	$E8	[SUP]	$E0 $53	$E0 $D3
I	$17	$97						
J	$24	$A4	**Other keys**			**Function keys**		
K	$25	$A5	[ESC]	$01	$81	F1	$3B	$BB
L	$26	$A6	º	$29	$A9	F2	$3C	$BC
M	$32	$B2	'	$0C	$8C	F3	$3D	$BD
N	$31	$B1	¡	$0D	$8D	F4	$3E	$BE
Ñ	$27	$A7	[BACK]	$0E	$8E	F5	$3F	$BF
O	$18	$98	[TAB]	$0F	$8F	F6	$40	$C0
P	$19	$99	[CAPSLOCK]	$3A	$BA	F7	$41	$C1
Q	$10	$90	[LSHIFT]	$2A	$AA	F8	$42	$C2
R	$13	$93	[RSHIFT]	$36	$B6	F9	$43	$C3
S	$1F	$9F	[CTRL]	$1D	$9D	F10	$44	$C4
T	$14	$94	[ALT]	$38	$B8	F11	$57	$D7
U	$16	$96	[ALT GR]	$E0 $38	$E0 $B8	F12	$58	$D8
V	$2F	$AF	[ENTER]	$1C	$9C			
W	$11	$91	[SPACE]	$39	$B9			
X	$2D	$AD	`	$1A	$9A			

The specific function keys of the CNC are assigned the codes of the following keys (Hotkeys).

WORK MODES

Task window	[CTRL] + A
Automatic mode	[CTRL] + F6
Manual (jog) mode	[CTRL] + F7
MDI mode	[CTRL] + F8
Editing / simulation mode	[CTRL] + F9
User tables	[CTRL] + F10
Tool and magazine table	[CTRL] + F11
Utilities mode	[CTRL] + F12

USING THE SCREENS

Horizontal softkeys	F1 through F7
Vertical softkeys	F8 through F12
Previous horizontal menu	[CTRL] + F1
Previous vertical menu	[CTRL] + [SHIFT] + F1
Window change	[CTRL] + F2
Screen change	[CTRL] + F3

OPERATIONS AT THE CNC

HELP	[CTRL] + F4
RECALL	[CTRL] + F5
"Single block" mode	[CTRL] + B
Home search	
Show/Hide virtual operator panel	[CTRL] + J
Show/Hide PLC messages	[CTRL] + M
Minimize/Restore the CNC	[CTRL] + W
Turn the CNC off	[ALT] + F4

The [START], [STOP] and [RESET] keys may be actuated from the PLC using the CYSTART, _STOP and RESETIN keys respectively.

EXAMPLE FOR SIMULATING THE KEYBOARD FROM THE PLC

The CNC keyboard may be simulated from the PLC using the following variable:

(V.)G.KEY Code of the last key accepted by the CNC.

And using the following functions.

CNCWR Allows writing a variable.

CNCRD Allows reading a variable.

The CNCWR(Rxxx, G.KEY, Mxxx) function indicates to the CNC that the key whose code is kept in register Rxxx has been pressed.

The CNCRD(G.KEY, Rxxx, Mxxx) function reads the code of the last key sent from the PLC and it stores it in register Rxxx.

In either case, the Mxxx mark is set to "1" at the beginning of the operation and it keeps its value until the end of the operation.

Programming example

Pressing the first customizable key (B0KEYBD1) carries out the following operations:

1. The manual mode is accessed at the CNC.
2. Then, the MDI mode is accessed.
3. It homes the X axis.
4. After the home search, it exits the MDI mode.

For each key to be sent from the PLC, the codes for "press key" and for the "release key" must be written. When sending both codes, the example uses a 200 ms delay between them (for safety).

```
START OR DFU M313 = CYSTART

()= MOV $1D R200            ;CTRL
()= MOV $9D R201
() = MOV $41 R202           ;F7
()= MOV $C1 R203
() = MOV $42 R204           ;F8
()= MOV $C2 R205
() = MOV $22 R206           ;G
()= MOV $A2 R207
() = MOV $08 R208           ;7
()= MOV $88 R209
() = MOV $05 R210           ;4
()= MOV $85 R211
() = MOV $2D R212           ;X
()= MOV $AD R213
() = MOV $02 R214           ;1
()= MOV $82 R215
() = MOV $01 R216           ;ESC
()= MOV $81 R217
```

;CTRL F7 (JOG MODE)

```
DFU B0KEYBD1 = CNCWR(R200,G.KEY,M200) = CNCWR(R202,G.KEY,M201)=TG1 200 200
T200 = M300
DFD M300 = CNCWR(R201,G.KEY,M202)= CNCWR(R203,G.KEY,M203)=TG1 201 200
T201 = M301
```

;CTRL F8 (MDI MODE)

```
DFD M301 = CNCWR(R200,G.KEY,M200) = CNCWR(R204,G.KEY,M204) = TG1 202 200
```

```
T202 = M302
DFD M302 = CNCWR(R201,G.KEY,M202)= CNCWR(R205,G.KEY,M205) = TG1 203 200
T203 =M303
```

;Home search

```
DFD M303 = CNCWR(R206,G.KEY,M206) = TG1 204 200    ;G
T204 = M304
DFD M304 = CNCWR(R207,G.KEY,M207)=TG1 205 200
T205 = M305
DFD M305 = CNCWR(R208,G.KEY,M208) = TG1 206 200    ;7
T206 = M306
DFD M306 = CNCWR(R209,G.KEY,M209)=TG1 207 200
T207 = M307
DFD M307 = CNCWR(R210,G.KEY,M210) = TG1 208 200    ;4
T208 = M308
DFD M308 = CNCWR(R211,G.KEY,M211)=TG1 209 200
T209 = M309
DFD M309 = CNCWR(R212,G.KEY,M212) = TG1 210 200    ;X
T210 = M310
DFD M310 = CNCWR(R213,G.KEY,M213)=TG1 211 200
T211 = M311
DFD M311 = CNCWR(R214,G.KEY,M214) = TG1 212 200    ;1
T212 = M312
DFD M312 = CNCWR(R215,G.KEY,M215)=TG1 213 200
T213 = M313                                        ;Execute Cycle Start (CYSTART=1)
DFD M313 = SET M500
DFD ZERO = SET M501                                ;Home search finished.
```

;Quit MDI mode

```
()= CNCRD(G.STATUS,R220,M220)                      ;CNC status ("1"=READY)
M500 AND M501 AND (CPS R220 EQ 1) = CNCWR(R216,G.KEY,M216) = TG1 214 200
                                                   ;ESC
T214 = M314
DFD M314 = CNCWR(R217,G.KEY,M217)= RES M500=RES M501
```