CNC 8070


# EXECUTION CHANNELS

(Soft V02.0x) Ref. 0501

FAGOR

REF. 0501

(SOFT V02.0X)

# 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



FAGOR

CNC 8070

(SOFT V02.0x)

# PARTE II. OPERATION AND PROGRAMMING

## 5. Operation and interface



## 6. Programming in a system with channels



## 7. Multiple spindles. Spindle control



FAGOR

CNC 8070

(SOFT V02.0X)

# INTRODUCTION 1

This manual is directed to the machine manufacturer as well as to the CNC user and it is meant to be a guide to better understand how to work with channels. It contains the necessary information to configure the CNC in multi-channel mode and a description of how to use it and program it.

**Configuration of a multi-channel system.**

The channels are configured through the machine parameter tables. These parameters set the number of channels, the axes and spindles of each channel, the possibility to exchange them between the different channels, etc.

There is only one PLC program for the whole system. The particularities of each channel will be dealt with inside the program itself. Each channel has its own marks and registers to communicate with the CNC. Refer to the installation manual for further detail.

**Configuration of a multi-spindle system.**

The CNC can control up to four spindles that may be distributed indistinctly through the different channels. The spindles are configured and distributed between the channels via machine parameters.

The PLC program manages the spindles. Each spindle has its own group of marks and registers.

**The part-program and the programming commands.**

This manual only shows the programming functions that are directly related to a multi-channel system. These functions deal with such matters as swapping axes or spindles, synchronizing channels, etc.

The rest of the functions, that are also valid both in a multi-channel CNC and in a single channel CNC, are described in the programming manual.

# 1.1 Basic notions about channels



## About a multi-channel system

A multi-channel CNC system may have up to four channels; each of which constitutes a different work environment that may act upon part of the CNC system or on the whole CNC system.

The difference between a multi-channel system and several independent CNC's is that the channels can not only act independently, but also together; in other words, they can communicate, synchronize with each other and carry out coordinated actions.

## What is a channel?

As mentioned earlier, each channel constitutes a different work environment inside the CNC. Each channel can execute a different program, be in an different work mode and have its own data.

If necessary, the channels can communicate and synchronize with each other and carry out actions that are coordinated with each other. They can also share information through variables and arithmetic parameters.

A channel can have a group of axes and spindles that act independently or in parallel with the rest of the channels. A channel can also be configured without assigning axes or spindles to it at first. Later on, it will be possible to add or remove axes and spindles through a program in execution or MDI.

A channel may be governed from the PLC, from the CNC or from both. Likewise, a channel may be configured as –hidden– so it cannot be selected from the interface and the screen does not show any information on its axes.

## The active channel

It is the channel selected with the channel selector switch. It is the channel being displayed and receiving the commands, e.g. START, STOP and RESET.

## Channel grouping

Two or more channels may be configured forming a group that will have the following characteristics.

- All the channels are in the same work mode, JOG or automatic.
- The reset command 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.

The groups of channels are defined via machine parameters.

## Configuring the axes and spindles of a channel

A channel may be configured at first with one, several, or no axis and no spindle, according to the machine parameter settings. The axes and spindles of each channel must be chosen among the ones available in the system. An axis or spindle cannot be in several channels at the same time; although it could happen that it is not assigned to any channel at first.

### Modifying the configuration of the axes and the spindles of a channel

Via a program in execution or MDI, a channel may release or request axes and spindles. This possibility is determined by machine parameter AXISEXCH, which establishes whether an axis or spindle can change channels or whether this change is permanent or not.

A permanent change is maintained after the end of the program, after a reset and on power-up. The original configuration may be restored either by validating the general parameters and restarting or by a part-program that undoes the changes.

*It also restore the machine parameter settings if a checksum error occurs when powering up the CNC. .*

### Main or master spindle of the channel.

It is the spindle that receives the commands when no specific spindle is mentioned. All the actions directed to a spindle through the operator panel will be directed to the master spindle.

In general, whenever a channel has a single spindle, it will be its master spindle. If a channel has several spindles, at first the master spindle will be the one configured by machine parameters. A new master spindle may be selected with the #MASTER instruction.

## Tool magazine and tool change

The CNC may have up to four different tool magazines. 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 only limitation will be the one imposed by the mechanics of the machine; in other words, by the physical accessibility of the machine to the tool 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.

# 1.

## INTRODUCTION

Basic notions about channels

FAGOR

CNC 8070

(SOFT V02.0X)

CNC 8070

# CONFIGURATION

I

FAGOR

# CONFIGURATION OF A MULTI-CHANNEL SYSTEM 2

## Machine parameters

A multi-channel system is configured mainly through the machine parameter tables. These tables are the only ones for the whole CNC table; it may be accessed from any channel and it is possible to set all the machine parameters.

**General machine parameters and for the channels.**

Some of these parameters are common to the whole CNC system whereas the rest are only for each channel. These parameters must be set first, because they define the number of channels, axes and spindles of the CNC. This creates the parameter tables for these elements.

For each channel defined, a sub-table is shown with its parameters. They set which are the axes and spindles that configure the channel.

**Machine parameter for axes (spindle).**

These parameters determine, for each axis and spindle, whether it is possible to change channels or not. The configuration of axes and spindles of a channel may be modified via part-program or MDI..

**Machine parameter HMI.**

It determines how to handle the operation and display of the different channels.

## PLC program

There is only one PLC program for the whole CNC system. The particularities of each channel will be dealt with in the program itself.

The PLC may be accessed from any channel.

# 2.1 Setting the machine parameters

## Setting the number of channels and their behavior

The first step for configuring a multi-channel system is to define the number of channels and the possible channel groups. The characteristics of each channel must also be defined, such as the type of channel and whether it is a hidden channel or not.

| Parameter | Meaning |
|---|---|
| NCHANNEL | Number of channels. |
| GROUPID | Group the channel belongs to. |
| CHTYPE | Channel type. |
| HIDDENCH | Hidden channel. |

### Channel type

The channel type determines whether the channel is governed from the CNC, from the PLC or from both. A PLC channel may be interesting, for example, for a system to load and unload a tool magazine that is controlled like an axis.

PLC channels cannot be set in jog mode or execute part-programs or blocks in MDI; however the axes that make it up may be displayed in the tables using the corresponding softkey. If during setup, a PLC channel must be displayed, it must be defined as CNC+PLC type during setup and once the setup is completed, it should be set back as PLC type.

### Group

A group of channels is set whenever the number defined in `GROUPID` is other than zero. All the channels with the same `GROUPID` (other than 0) make up a group.

### Hidden channels

A hidden channel cannot be displayed nor selected in the various work modes. Sometimes, it may be interesting to define a channel as hidden once the setup has been completed. It is also a good idea to define as hidden an exclusive PLC channel, once the setup is completed.

## Distributing the channels and spindles between the different channels

First, define the number and the name of the axes and spindles that make up the system.

| Parameter | Meaning |
|---|---|
| NAXIS | Number of axes of the system. |
| AXISNAME | Name the axes of the system. |
| NSPDL | Number of spindles of the system |
| SPDLNAME | Name of the spindles of the system. |

Once the axes and spindles of the system have been defined, they must be distributed between the different channels. The axes and spindles of each channel must be chosen among the ones available in the system. An axis or spindle cannot be in several channels at the same time; although it could happen that it is not assigned to any channel at first.

Likewise, a channel may have at first, one, several, or no axis or no spindle associated with it.

| Parameter | Meaning |
|---|---|
| CHNAXIS | Number of axes of the channel. |
| CHAXISNAME | Name the axes of the channel. |
| CHNSPDL | Number of spindles of the channel. |
| CHSPDLNAME | Name of the spindles of the channel. |

## Allowing to change channel axes and spindles

The configuration of a channel may be modified via part-program or MDI, either adding or removing axes as well as spindles. Doing this requires defining for each axis and spindle whether it may change channels or not and whether the change is permanent or not.

A permanent change is maintained after the end of the program, after a reset and on power-up. The original configuration may be restored either by validating the general parameters and restarting or by a part-program that undoes the changes.

*It also restore the machine parameter settings if a checksum error occurs when powering up the CNC. .*

| Parameter | Meaning |
|---|---|
| AXISEXCH | Channel change permission. |

## Arithmetic parameter configuration

The CNC has three types of arithmetic parameters, namely local, global and common.

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

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

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

| Parameter | Meaning |
|---|---|
| MAXLOCP | Maximum local arithmetic parameter. |
| MINLOCP | Minimum local arithmetic parameter. |

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

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

| Parameter | Meaning |
|---|---|
| MAXGLBP | Maximum global arithmetic parameter. |
| MINGLBP | Minimum global arithmetic parameter. |

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

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

| Parameter | Meaning |
|---|---|
| MAXCOMP | Maximum common arithmetic parameter. |
| MINCOMP | Minimum common arithmetic parameter. |

## Channel operation and display

The way to access the different channels is handled through the change key. This key may be configured either to access the channels sequentially or to show the list of available channels on the softkey menu.

| Parameter | Meaning |
|---|---|
| CHANGEKEY | Customizing the change key. |
| FUNCTION | Function of the change key. Next page of the current mode or next channel. |
| MENU | Set up the system menu. |

## Subroutines related to the –M– functions

The –M– function defining table is general for all the channels. To have different procedures in the subroutines associated with certain –M– functions (for example M06), the code of each channel may be differentiated within the subroutine using the variable (V.)G.CNCHANNEL.

| Parameter | Meaning |
|---|---|
| MPROGNAME | Subroutine associated with function M |

## Channel kinematics

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.

## 2.1.1 Setting the number of channels and their behavior

Some of these parameters correspond to the common general parameters whereas others belong to the general parameters of each channel.

| Parameter | Table |
|---|---|
| NCHANNEL | General machine parameter. |
| GROUPID | General machine parameter of the channel. |
| CHTYPE | General machine parameter of the channel. |
| HIDDENCH | General machine parameter of the channel. |

### NCHANNEL — Number of channels

*Possible values:* *from 1 to 4.*
*Default value:* *1.*

Number of system's channels.

### GROUPID — Group the channel belongs to

*Possible values:* *from 1 to 2.*
*By default:* *0 (it does not belong to any group).*

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.

### CHTYPE — Type of channel

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

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

PLC channels cannot be set in jog mode or execute part-programs or blocks in MDI; however the axes that make it up may be displayed in the tables using the corresponding softkey. If during setup, this type of channel must be displayed, it must be defined as CNC+PLC type during setup and once the setup is completed, it should be set back as PLC type. This type of channel is usually hidden once the setup is completed.

A PLC channel may be interesting, for example, for a system to load and unload a tool magazine that is controlled like an axis.

## HIDDENCH Hidden channel

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

Hidden channels are not displayed and cannot be selected. The exclusive channels of the PLC are usually hidden.

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`.

## 2.1.2 Distributing the channels and spindles between the different channels

Some of these parameters correspond to the common general parameters whereas others belong to the general parameters of each channel.

| Parameter | Table |
|---|---|
| NAXIS | General machine parameter. |
| AXISNAME | General machine parameter. |
| NSPDL | General machine parameter. |
| SPDLNAME | General machine parameter. |
| CHNAXIS | General machine parameter of the channel. |
| CHAXISNAME | General machine parameter of the channel. |
| CHNSPDL | General machine parameter of the channel. |
| CHSPDLNAME | General machine parameter of the channel. |



### NAXIS — Number of axes governed by the CNC

*Possible values:* *From 0 to 28.*
*By default:* *3.*

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

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.

### AXISNAME — Name of each axis

*By default:* *starting with AXISNAME1: X, Y, Z...*

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...

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.

### NSPDL — Number of spindles governed by the CNC

*Possible values:* *From 0 to 4.*
*By default:* *1.*

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

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.

## SPDLNAME — Spindle name

*By default:* *starting with SPDLNAME1: S,S1..S9*

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.

## CHNAXIS — Number of axes of the channel

*Possible values:* *From 0 to 28.*
*By default:* *3.*

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.

It is possible to define the configuration of the axes again from the part-program, add or remove axes, using #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 general machine parameter AXISNAME may belong to the channel.

## CHNSPDL — Number of spindles of the channel

*Possible values:* *From 0 to 4.*
*By default:* *1.*

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.

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 general machine parameter SPDLNAME may belong to the channel.

### 2.1.3 Allowing to change channel axes and spindles

The following parameter corresponds to the axis parameters.

| Parameter | Table |
|---|---|
| AXISEXCH | Axis machine parameters. |

**AXISEXCH** **Channel change permission**

*For axis type:* *Linear / Rotary / Spindle.*
*For drive type:* *Analog / Sercos / Simulated.*
*Possible values:* *No / Temporary / Maintained.*
*By default:* *No.*

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.

**AXISEXCH=NO**

The axis cannot be exchanged nor freed (released) by its channel. This type of axis must be assigned to a channel on CNC power-up.

**AXISEXCH=TEMPORARY**

The axis may be released to another channel, but it is recovered after a RESET or when starting the next program in its channel.

**AXISEXCH=MAINTAINED**

The axis, once released, stays in its new channel after ending the program, after a reset and on power-up. The original configuration may be restored either by validating the general parameters and restarting or by a part-program that undoes the changes.

*It also restore the machine parameter settings if a checksum error occurs when powering up the CNC. .*

## 2.1.4 Arithmetic parameter configuration

The following parameters correspond to the general parameters.

**MAXLOCP** **Maximum local arithmetic parameter**

**MINLOCP** **Minimum local arithmetic parameter**

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

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 or levels of local parameters in each channel.

**MAXGLBP** **Maximum global arithmetic parameter**

**MINGLBP** **Minimum global arithmetic parameter**

*Possible values:* *from 100 to 9999.*
*By default:* *MAXGLBP=299 and MINGLBP=100.*

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.

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

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

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

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.

## 2.1.5 Channel operation and display

The following parameters correspond to the HMI parameters.

**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**

*Possible values:* *Next page / Next channel / Menu.*
*By default:* *Next Page.*

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.

When selecting to show a menu, parameter MENU must be set with the options to be displayed on each sofkey menu.

**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**

*Possible values:* *Volatile / Fixed.*
*By default:* *Volatile.*

It determines when the system menu is disabled.

- If it is set as –Volatile–, it is disabled when pressing the [ESC] key, the previous menu key, when selecting one of its options or when changing the active component.
- If it is defined as –Fixed–, the softkey menu stays on until the change key is pressed again.

**SYSHMENU** **Horizontal system-menu**

**SYSVMENU** **Vertical system-menu**

*Possible values:* *Disabled / Pages / Channels / Components.*
*By default:* *Disabled.*

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.

## 2.1.6 Subroutines related to the –M– functions

The following parameters correspond to the –M– function table.

The –M– function defining table is common to all the channels. To have different procedures in the subroutines associated with certain –M– functions (for example M06), the code of each channel may be differentiated within the subroutine using the variable (V.)G.CNCHANNEL.

**MPROGNAME** **Name of subroutine associated with M function**

*Possible values:* *any text with up to 64 characters.*

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.

M functions that have a subroutine associated with them may have any type of synchronization. However, if it is programmed in a motion block, the type of synchronization must be "M without synchronization" or "it is sent after - it is synchronized after". The CNC executes the associated subroutine after executing the programmed movement (if any).

## 2.2 The PLC program

There is a group of marks and registers for each channel to create the PLC program. See chapter *"3 Logic CNC inputs and outputs"*.

### Transferring –M– and –H– functions

The M and H functions are exchanged by channel. When using channels, the marks and registers of these functions must indicate the channel number they refer to.

### Transferring -S- functions

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.

### Multiple spindles

The CNC may have up to four spindles. When using channels, the spindles may be distributed indistinctly between them.

All the spindles of a channel may be controlled independently; in other words, each spindle may be given a different command.

In order to be able to manage spindles from the PLC regardless of the channel it belongs to, there is a group of PLC marks and registers for each spindle.

### Spindle synchronization

Via PLC it is possible to synchronize the machining operation in a channel with a particular spindle even if it is in another channel. For example, to program the feedrate depending on the speed of a particular spindle.

From the PLC, it is possible to make a channel synchronize with a spindle that belongs to another channel to carry out electronic threading operation (G33) and set the feedrate per revolution (G95).

# LOGIC CNC INPUTS AND OUTPUTS

# 3

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 |

**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 |

## Type of signals available

The specific logic signals for configuring a system with channels are grouped as follows:

- General signals.
- Signals related to the –M– auxiliary functions.
- Signals related to the –H– auxiliary functions.
- Channel synchronizing signals.

The specific logic signals for configuring a system with several spindles are grouped as follows:

- Signals related to the –S– auxiliary functions.
- Spindle signals.

The specific logic signals for configuring a system with several tool magazines are grouped as follows:

- Tool manager signals.

The rest of the signals that are not affected by the number of channels or spindles are the following:

- Signals related to the axes and spindles in closed loop.
- Signals related to the keys.

# 3.1 General signals

The general signals are specific for each channel. Refer to the installation manual for further detail about each one of them.

The following tables show the mnemonics for each mark (M) or register (R) in each channel.



## Consultation signals

The CNCREADY mark is common to all the channels.

| M/R | Channel ·1· | Channel ·2· | Channel ·3· | Channel ·4· |
|---|---|---|---|---|
| M | STARTC1<br>START | STARTC2 | STARTC3 | STARTC4 |
| M | FHOUTC1<br>FHOUT | FHOUTC2 | FHOUTC3 | FHOUTC4 |
| M | RESETOUTC1<br>RESETOUT | RESETOUTC2 | RESETOUTC3 | RESETOUTC4 |
| M | _ALARMC1<br>_ALARM | _ALARMC2 | _ALARMC3 | _ALARMC4 |
| M | MANUALC1<br>MANUAL | MANUALC2 | MANUALC3 | MANUALC4 |
| M | AUTOMATC1<br>AUTOMAT | AUTOMATC2 | AUTOMATC3 | AUTOMATC4 |
| M | MDIC1<br>MDI | MDIC2 | MDIC3 | MDIC4 |
| M | SBOUTC1<br>SBOUT | SBOUTC2 | SBOUTC3 | SBOUTC4 |
| M | INCYCEC1<br>INCYCE | INCYCEC2 | INCYCEC3 | INCYCEC4 |
| M | RAPIDC1<br>RAPID | RAPIDC2 | RAPIDC3 | RAPIDC4 |
| M | ZEROC1<br>ZERO | ZEROC2 | ZEROC3 | ZEROC4 |
| M | PROBEC1<br>PROBE | PROBEC2 | PROBEC3 | PROBEC4 |
| M | THREADC1<br>THREAD | THREADC2 | THREADC3 | THREADC4 |
| M | TAPPINGC1<br>TAPPING | TAPPINGC2 | TAPPINGC3 | TAPPINGC4 |
| M | RIGIDC1<br>RIGID | RIGIDC2 | RIGIDC3 | RIGIDC4 |
| M | CSSC1<br>CSS | CSSC2 | CSSC3 | CSSC4 |
| M | INTERENDC1<br>INTEREND | INTERENDC2 | INTERENDC3 | INTERENDC4 |

| M/R | Channel ·1· | Channel ·2· | Channel ·3· | Channel ·4· |
|---|---|---|---|---|
| M | INPOSC1<br>INPOS | INPOSC2 | INPOSC3 | INPOSC4 |
| M | BLKSEARCHC1<br>BLKSEARCH | BLKSEARCHC2 | BLKSEARCHC3 | BLKSEARCHC4 |
| M | ADVINPOSC1<br>ADVINPOS | ADVINPOSC2 | ADVINPOSC3 | ADVINPOSC4 |

## General modifiable signals

The following marks are common to all the channels.

MANRAPID LATCHM TIMERON

PLCREADY DISCROSS1..9

| M/R | Channel ·1· | Channel ·2· | Channel ·3· | Channel ·4· |
|---|---|---|---|---|
| M | _EMERGENC1<br>_EMERGEN | _EMERGENC2 | _EMERGENC3 | _EMERGENC4 |
| M | _STOPC1<br>_STOP | _STOPC2 | _STOPC3 | _STOPC4 |
| M | _XFERINHC1<br>_XFERINH | _XFERINHC2 | _XFERINHC3 | _XFERINHC4 |
| M | _FEEDHOLC1<br>_FEEDHOL | _FEEDHOLC2 | _FEEDHOLC3 | _FEEDHOLC4 |
| M | CYSTARTC1<br>CYSTART | CYSTARTC2 | CYSTARTC3 | CYSTARTC4 |
| M | SBLOCKC1<br>SBLOCK | SBLOCKC2 | SBLOCKC3 | SBLOCKC4 |
| M | OVRCANC1<br>OVRCAN | OVRCANC2 | OVRCANC3 | OVRCANC4 |
| M | RESETINC1<br>RESETIN | RESETINC2 | RESETINC3 | RESETINC4 |
| M | AUXENDC1<br>AUXEND | AUXENDC2 | AUXENDC3 | AUXENDC4 |
| M | BLKSKIP1C1<br>BLKSKIP1 | BLKSKIP1C2 | BLKSKIP1C3 | BLKSKIP1C4 |
| M | M01STOPC1<br>M01STOP | M01STOPC2 | M01STOPC3 | M01STOPC4 |

# 3.2 Signals related to the –M– auxiliary functions

The –M– functions are transferred per channel. See chapter *"4 CNC-PLC communication"*.

## 3.2.1 Consultation signals

**MFUN1···MFUN7**

There is one register for each channel. The mnemonics for each channel are the following. Here is an example of the mnemonics for MFUN1; it is the same for the rest of the registers.

`MFUN1C1` (can also be programmed as `MFUN1`)

`MFUN1C2` `MFUN1C3` `MFUN1C4`

The channel uses these registers to indicate to the PLC the `M` auxiliary functions selected for execution. Each one of them indicates the number of one of the `M` functions programmed in the block.

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 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 |

**Checking if a function has been programmed in the channel. Commands `MFUNC1*` - `MFUNC4*`.**

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*` For channel 1. They can also be programmed as `MFUN*`.

`MFUNC2*` For channel 2.

`MFUNC3*` For channel 3.

`MFUNC4*` 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.

**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".

**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.

# 3.3 Signals related to the –H– auxiliary functions

The –H– functions are transferred per channel. See chapter *"4 CNC-PLC communication"*.

## 3.3.1 Consultation signals

**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 HFUN1; it is the same for the rest of the registers.

HFUN1C1 (can also be programmed as HFUN1)

HFUN1C2 HFUN1C3 HFUN1C4

The channel uses these registers to indicate to the PLC the H auxiliary functions selected for execution. Each one of them indicates the number of one of the H functions programmed in the block.

Each channel can have up to 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 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 |

**Checking if a function has been programmed in the channel. Commands HFUNC1* - HFUNC4*.**

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.

HFUNC1* For channel 1. They can also be programmed as HFUN*.

HFUNC2* For channel 2.

HFUNC3* For channel 3.

HFUNC4* For channel 4.

**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".

```
CPS HFUNC1* EQ 77 = ...
```

Example for detecting H77 in channel 1. If programmed, it will return a "1" and a "0" if otherwise.

## 3.4 Signals related to the –S– auxiliary functions

The –S– function transfer is independent from the channel. See chapter *"4 CNC-PLC communication"*.

### 3.4.1 Consultation signals

**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 |

**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 .

**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.

# 3.5 Spindle signals

## 3.5.1 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** **(REVolutions OK)**

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.

## 3.5.2 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.

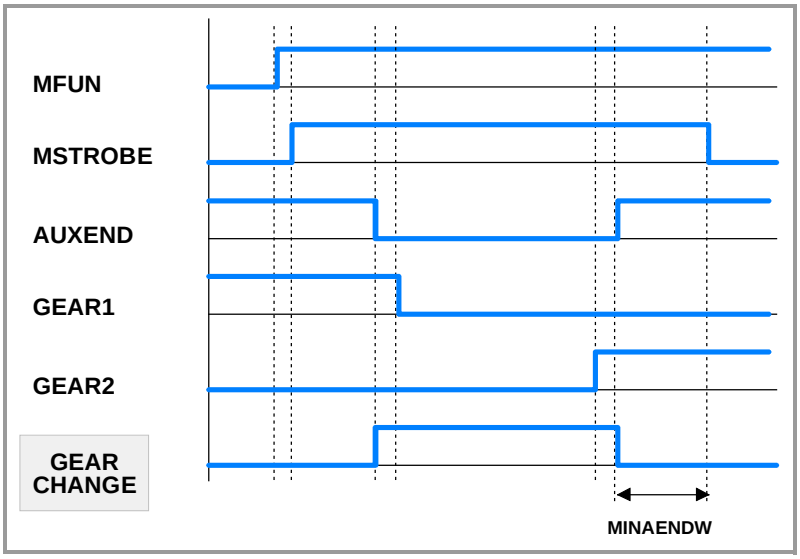
GEAR1SP1 (can also be programmed as GEAR1)

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, programSANALOG = (4x32767)/10 = 13107

  For -4V, programSANALOG = (-4x32767)/10 = -13107

- Sercos spindle in velocity.

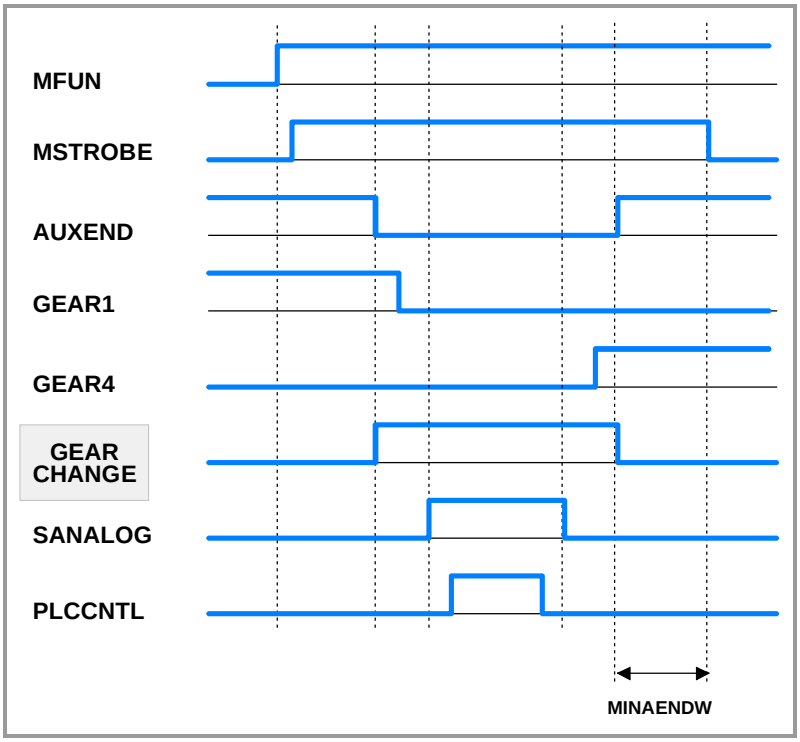
  The command in SANALOG will be given in 0.0001 rpm.

- 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".

# 3.6 Synchronization of channels

## 3.6.1 Consultation signals

**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.

### 3.6.2 Modifiable signals

**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 signal 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.

# 3.7 Tool manager signals

## 3.7.1 Communication signals

### Consultation signals. Manager --> PLC communication

| M/R | Channel ·1· | Channel ·2· | Channel ·3· | Channel ·4· |
|---|---|---|---|---|
| M | TMOPERATIONC1<br>TMOPERATION | TMOPERATIONC2 | TMOPERATIONC3 | TMOPERATIONC4 |
| M | TMOPSTROBEC1<br>TMOPSTROBE | TMOPSTROBEC2 | TMOPSTROBEC3 | TMOPSTROBEC4 |
| R | MZIDC1 | MZIDC2 | MZIDC3 | MZIDC4 |

| M/R | Magazine ·1· | Magazine ·2· | Magazine ·3· | Magazine ·4· |
|---|---|---|---|---|
| R | LEAVEPOSMZ1<br>LEAVEPOS | LEAVEPOSMZ2 | LEAVEPOSMZ3 | LEAVEPOSMZ4 |
| R | TAKEPOSMZ1<br>TAKEPOS | TAKEPOSMZ2 | TAKEPOSMZ3 | TAKEPOSMZ4 |
| R | NEXTPOSMZ1<br>NEXTPOS | NEXTPOSMZ2 | NEXTPOSMZ3 | NEXTPOSMZ4 |

### Modifiable signals. PLC --> Manager communication

| M/R | Magazine ·1· | Magazine ·2· | Magazine ·3· | Magazine ·4· |
|---|---|---|---|---|
| M | MZTOCH1MZ1<br>MZTOCH1 | MZTOCH1MZ2 | MZTOCH1MZ3 | MZTOCH1MZ4 |
| M | CH1TOSPDLMZ1<br>CH1TOSPDL | CH1TOSPDLZ2 | CH1TOSPDLMZ3 | CH1TOSPDLMZ4 |
| M | SPDLTOCH1MZ1<br>SPDLTOCH1 | SPDLTOCH1MZ2 | SPDLTOCH1MZ3 | SPDLTOCH1MZ4 |
| M | SPDLTOCH2MZ1<br>SPDLTOCH2 | SPDLTOCH2MZ2 | SPDLTOCH2MZ3 | SPDLTOCH2MZ4 |
| M | CH1TOMZ1<br>CH1TOMZ | CH1TOMZ2 | CH1TOMZ3 | CH1TOMZ4 |
| M | CH2TOMZ1<br>CH2TOMZ | CH2TOMZ2 | CH2TOMZ3 | CH2TOMZ4 |
| M | SPDLTOGRMZ1<br>SPDLTOGR | SPDLTOGRMZ2 | SPDLTOGRMZ3 | SPDLTOGRMZ4 |
| M | GRTOSPDLMZ1<br>GRTOSPDL | GRTOSPDLMZ2 | GRTOSPDLMZ3 | GRTOSPDLMZ4 |
| M | MZTOSPDLMZ1<br>MZTOSPDL | MZTOSPDLMZ2 | MZTOSPDLMZ3 | MZTOSPDLMZ4 |
| M | MZTOSPDLMZ1<br>MZTOSPDL | MZTOSPDLMZ2 | MZTOSPDLMZ3 | MZTOSPDLMZ4 |

| M/R | Magazine ·1· | Magazine ·2· | Magazine ·3· | Magazine ·4· |
|---|---|---|---|---|
| M | MZROTMZ1<br>MZROT | MZROTMZ2 | MZROTMZ3 | MZROTMZ4 |
| M | TCHANGEOKMZ1<br>TCHANGEOK | TCHANGEOKMZ2 | TCHANGEOKMZ3 | TCHANGEOKMZ4 |
| M | MZPOSMZ1<br>MZPOS | MZPOSZ2 | MZPOSMZ3 | MZPOSMZ4 |



## 3.7.2 Manager Emergency

### Consultation signals

| M/R | Magazine ·1· | Magazine ·2· | Magazine ·3· | Magazine ·4· |
|---|---|---|---|---|
| M | TMINEMZ1<br>TMINEM | TMINEMZ2 | TMINEMZ3 | TMINEMZ4 |

### Modifiable signals

| M/R | Magazine ·1· | Magazine ·2· | Magazine ·3· | Magazine ·4· |
|---|---|---|---|---|
| M | SETTMEMZ1<br>SETTMEM | SETTMEMZ2 | SETTMEMZ3 | SETTMEMZ4 |
| M | RESTMEMZ1<br>RESTMEM | RESTMEMZ2 | RESTMEMZ3 | RESTMEMZ4 |

## 3.7.3 Tool monitoring

### Consultation signals

| M/R | Channel ·1· | Channel ·2· | Channel ·3· | Channel ·4· |
|---|---|---|---|---|
| M | TWORNOUTC1<br>TWORNOUT | TWORNOUTC2 | TWORNOUTC3 | TWORNOUTC4 |

### Modifiable signals

| M/R | Channel ·1· | Channel ·2· | Channel ·3· | Channel ·4· |
|---|---|---|---|---|
| M | CUTTINGONC1<br>CUTTINGON | CUTTINGONC2 | CUTTINGONC3 | CUTTINGONC4 |
| M | TREJECTC1<br>TREJECT | TREJECTC2 | TREJECTC3 | TREJECTC4 |

# 3.8 Signals related to the axes and spindles in closed loop

The signals related to the axes or spindles working in closed loop (M19 or G63) are global for the whole system and do not depend on the channel.

Refer to the installation manual for further detail about each one of them.

**Name of the signals.**

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.

## Consultation signals

| M/R | System |
|---|---|
| M | ENABLE(axis) |
| M | DIR(axis) |
| M | REFPOIN(axis) |
| M | DRSTAF(axis) |
| M | DRSTAS(axis) |
| M | INPOS(axis) |

| M/R | System |
|---|---|
| M | LUBR(axis) |
| M | HIRTHON(axis) |
| M | MATCH(axis) |
| M | PARK(axis) |
| M | UNPARK(axis) |

## Modifiable signals

| M/R | System |
|---|---|
| M | LIMITPOS(axis) |
| M | LIMITNEG(axis) |
| M | DECEL(axis) |
| M | INHIBIT(axis) |
| M | AXISPOS(axis) |
| M | AXISNEG(axis) |
| M | SERVO(axis)ON |
| M | DRO(axis) |

| M/R | System |
|---|---|
| M | SPENA(axis) |
| M | DRENA(axis) |
| M | LIM(axis)OFF |
| M | PARKED(axis) |
| M | LUBRENA(axis) |
| M | LUBROK(axis) |
| M | DIFFCOMP(axis) |

## 3.9 Signals related to the keys

The signals related to the keys are global for the whole system and do not depend on the channel.

Refer to the installation manual for further detail about each one of them.

### Consultation signals

| M/R | System |
|---|---|
| M | KEYBD1 |
| M | KEYBD2 |

### Modifiable signals

| M/R | System |
|---|---|
| M | KEYLED1 |
| M | KEYLED2 |
| M | KEYDIS1 |
| M | KEYDIS2 |
| M | KEYDIS3 |

3.

LOGIC CNC INPUTS AND OUTPUTS

Signals related to the keys

FAGOR

CNC 8070

(SOFT V02.0X)

# CNC-PLC COMMUNICATION 4

## 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 channel number is determined by the order (sequence) in which it has been defined in machine parameter SPDLNAME.

# 4.1 Auxiliary -M- functions



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.

## Registers used in the communication

Each channel has 32-bit registers MFUN1 to 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 |

**Checking if a function has been programmed in the channel. Commands MFUNC1* - MFUNC4*.**

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 = ...
```

## 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.

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. See *"4.4 Transferring auxiliary functions -M-, -H-, -S-"* on page 47.

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.

## 4.2 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 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.



### Registers used in the communication

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 |

**Checking if a function has been programmed in the channel. Commands HFUNC1* - HFUNC4*.**

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. |

Example for detecting H77 in channel 1. If programmed, it will return a "1" and a "0" if otherwise.

```
CPS HFUNC1* 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 *"4.4 Transferring auxiliary functions -M-, -H-, -S-"* on page 47.

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.

# 4.3 Auxiliary –S– function

The auxiliary S function indicates the spindle turning speed with M03 and M04 or the angular position with M19. All the spindles of a channel may be controlled independently; in other words, each spindle may be given a different command.

The CNC may have up to four spindles. When using channels, the spindles may be distributed indistinctly between them.

The channel number is determined by the order (sequence) in which it has been defined in machine parameter SPDLNAME. The marks and registers refer to the spindle regardless of the channel they belong to.



## Registers used in the communication

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 |

**Check if an auxiliary function has been programmed for a spindle. Commands SP1FUN* - SP4FUN*.**

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 = ...
```

## Sending and synchronizing the function

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 transfer of the S functions is described later on in this chapter. See *"4.4 Transferring auxiliary functions -M-, -H-, -S-"* on page 47.

## 4.4 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

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

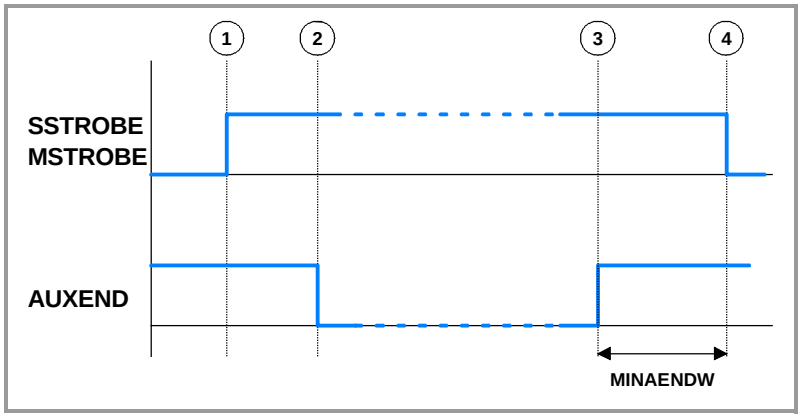
The CNC assigns the values of the S programmed in each spindle to registers SFUN1 through SFUN4.

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.

## 4.4.1 Synchronized transfer

This type of transfer takes place with the S function and with the M functions set with synchronization.




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.

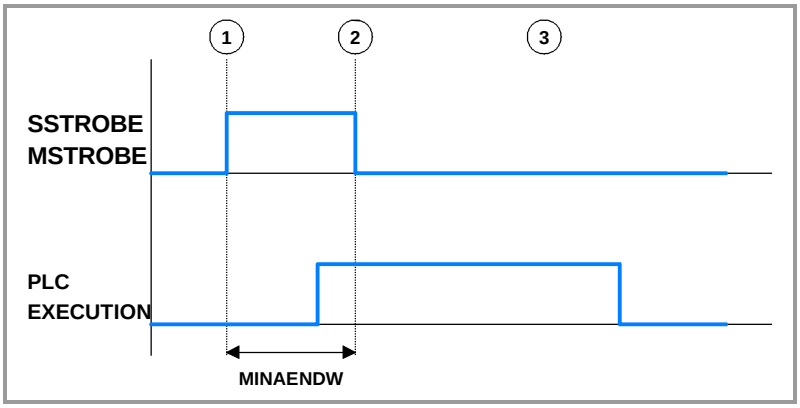
### Transferring -S- functions

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.

### 4.4.2 Non-synchronized transfer

This type of transfer takes place with the H function and with the M functions set without synchronization.




#### Transferring -M- functions

1. The CNC indicates in registers MFUN1 to 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.

# 4.5 Displaying PLC errors and messages

The PLC has 256 marks for displaying messages and another 256 marks for displaying errors at the CNC. These errors and messages are common to all the channels.

When one of the previous marks is high (=1) the message or the error is active.

| | |
|---|---|
| MSG1 - MSG256 | for displaying messages. |
| ERR1 - ERR256 | for displaying errors. |

The PLC has a table for defining the messages and the errors. 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. It also shows the file for additional information, when there is one.

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.

4.

CNC-PLC COMMUNICATION

Displaying PLC errors and messages

FAGOR

CNC 8070

(SOFT V02.0X)

CNC 8070

# OPERATION AND PROGRAMMING

II

FAGOR

The channels are managed from the interface like with a channel selector switch. There is an active channel at all times and it is the one being displayed. All the actions directed to a channel through the keyboard or operator panel will be directed to the active channel.

When changing channels, it displays it and it becomes the active channel.

## Channels that make up a group

Two or more channels may be configured forming a group that will have the following characteristics.

- All the channels are in the same work mode, JOG or automatic.
- The reset command 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.

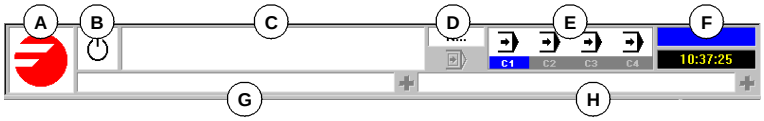
## Operation of the operator panel

By default, the actions directed to a channel through the operator panel will be directed to the active channel. However, this behavior may be different when set accordingly via PLC program.

- The feedrate override switch acts upon all the axes of the system at the same time; in other words, the feedrate override percentage change affects all the channels of the system.
- The spindle control keys (override, stop, etc.) act upon the master spindle of the active channel.
- The [START] or [STOP] keys only affect the active channel.
- The [RESET] key is only applied to the active channel and to the channels that are grouped with it.

# 5.1 The general status bar

As mentioned earlier, the CNC can have four channels. The general status bar at the top of the screen shows the number of channels, which one is the active channel and the operating mode of each of them.




**A.** Icon (customizable) identifying the manufacturer.

**B.** Icon showing the status of the program of the active channel:

The background color will be different depending on the status of the program.

Programmed stopped.

Program in execution.

Background color: Green.

Program interrupted.

Background color: Dark green.

Program in error.

Background color: Red.

**C.** Program selected in the active channel for execution.

The background color will be different depending on the status of the program.

**D.** Number of the block in execution. The bottom icon indicates that the Single-block execution mode is active.

**E.** Information about the channels.

Number of channels available and active channel (indicated in blue). Icons show which operating mode each channel is in.

Execution mode

Manual mode (jog).

MDI mode.

The channel synchronization window may be expanded using the [ALT]+[S] keys. See *"5.3 Channel synchronization window."* on page 59.

**F.** Active work mode (automatic, manual, etc.) selected screen number and total number of screens available.

System clock.

**G.** Active CNC message.

**H.** PLC messages.

## Messages active at the CNC

For each channel, it shows the last message activated by the program that is running. The window shows the last message of the active channel. If there are messages in other channels, it will highlight the "+" sign next to the message window. To display the list of active messages, press the key combination [CTRL]+[O].

The list of messages shows, next to each message, the channel where it is active.

## PLC messages

If the PLC activates two or more messages, the CNC displays the message with the highest priority and it will show the "+" sign indicating that there are more messages activated by the PLC. To display the list of active messages, press the key combination [CTRL]+[M].

On the message list and next to each message, a symbol will appear to indicate whether the message has an additional information file associated with it or not. To display a message, select it with the cursor and press [ENTER]. If the message has an additional information file, it will be displayed on the screen.

## 5.2 Changing channels. The channel selector switch



There is a change key for changing channels. This key may be configured either to change channels directly or to show the system menu.

**Channel change.**

Every time the key is pressed, it shows the next channel. It is a rotating change, so when pressed at the last channel, it shows the first one.

**System menu.**

The system menu shows, on one of the softkey menus, the list of available channels. Pressing the corresponding softkey accesses the desired menu.

The other softkey menu may be disabled or show one of the following options:

- The menu shows the various pages or screens of the active work mode.
- The menu shows the components or work modes of the CNC.

Depending on how the system menu is configured, the system menu will be disabled in one of the following ways.

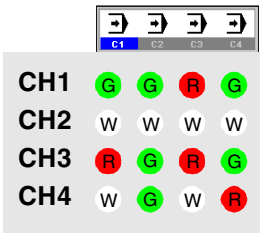
- It is disabled when pressing the [ESC] key, the previous menu key, when selecting one of its options or when changing the active component.
- The softkey menu remains until the change key is pressed again.

## 5.3 Channel synchronization window.

The channel synchronization window may be expanded using the [ALT]+[S] keys. The synchronization is carried out using marks in the programs. The window shows for each channel whether it is waiting for synchronization marks or not and the status of those marks in the channel that originates them.

An LED of various colors of the window show the status of the synchronization marks of each channel. On the left, the channels waiting for the marks and on top the channels that originate them.

| Led | Meaning |
|---|---|
| White | No synchronization mark expected. |
| Green | Synchronization mark expected. The mark is set to ·1· in the channel that originates it. |
| Red | Synchronization mark expected. The mark is set to ·0· in the channel that originates it. |




*(In the graphic, the white LED's are identified with the letter -W-, the green ones with the letter -G- and the red ones with the letter -R-).*

The previous graph shows for example:

- Channel 1 (CH1) is expecting synchronism marks from the rest of the channels. The marks of channels 2 and 4 are set to ·1·. The mark of channel 3 is set to ·0·.
- Channel 2 (CH2) is not expecting any synchronism mark.

# 5.4 The user tables

Some tables are displayed per channel; in other words, they show the data of the active channel. These tables have a softkey for displaying the same table in each channel.



## Zero offset and fixture offset tables.

The tables are common to the whole system. Each zero offset contains all the axes of the system; however, in channel, only the zero offsets of its axes are displayed. The corresponding vertical softkey may be used to display the axes of the rest of the channels.

When applying an offset from a channel, it is only applied to the axes that belong to the channel at the time.

## Arithmetic parameter table

### Global arithmetic parameters.

There is one table per channel; by default, it shows the table of the active channel. The corresponding vertical softkey may be used to display the table of the rest of the channels.

### Local arithmetic parameters.

There are seven tables per channel. By default, it shows the tables of the active channel. The corresponding vertical softkey may be used to display the table of the rest of the channels.

### Common arithmetic parameters.

There is one table for the whole system.

# PROGRAMMING IN A SYSTEM WITH CHANNELS

# 6

This manual only shows the programming functions that are directly related to the multi-channel version. The rest of the functions, that are also valid both in a multi-channel CNC and in a single channel CNC, are described in the programming manual.

## Executing a program in the specified channel

From the automatic mode, each channel can execute its own program. Via part-program or MDI, it is possible to command the execution of a program in a particular channel. The program to be executed may be stored in any folder; and when programming, it will be possible to indicate its location.

## Executing a block in the indicated channel

Via part-program or MDI, it is possible to command the execution of a block in a particular channel.

## Axis swapping

Initially, each channel has some axes assigned to it as set by the machine parameters. While executing a program, a channel may release its axes, request new axes or reorder the existing ones.

## Spindle swapping

Initially, each channel has some spindles assigned to it as set by the machine parameters. While executing a program, a channel may release its spindles or request new spindles.

## Communication and synchronization

These functions deal with such matters as swapping axes or spindles, synchronizing channels, etc.

# 6.1 Executing a program in the specified channel

## #EXEC

### Executes a program in the indicated channel

With this instruction, it is possible to execute a program in the indicated channel. The execution of the program starts in the indicated channel in parallel (at the same time) with the block following the #EXEC instruction.

If the channel where it is to be executed is busy, it will issue the relevant error message.

The programming format is:

```
#EXEC [<path><prg>,<channel>]
```

| Parameter | Meaning |
|---|---|
| <path> | File location |
| <prg> | Program to be executed. |
| <channel> | Channel where the block is to be executed. |

```
#EXEC [PRG1.NC,2]
```
(It executes in channel 2 the indicated program)

```
#EXEC [C:\CNC8070\USERS\PRG\EXAMPLE.NC,3]
```
(It executes in channel 3 the indicated program)

### Program location

The program to be executed may be defined by either writing the full path or without it. When a call indicates the full path, it will only look for it in the indicated directory. If the path is not indicated, the search is carried out in this order and in these directories:

1. Directory selected with the #PATH instruction.
2. Directory of the program that executes the #EXEC instruction.
3. Directory defined by machine parameter SUBPATH.

### Considerations

If the channel is not indicated or it coincides with the channel where the #EXEC instruction is executed, the indicated program will be executed as a subroutine. In this case, functions M02 and M30 will carry out all the associated actions (initialization, sending to the PLC, etc.) except the one for finishing the program. After executing function M02 or M30, it goes on executing the blocks programmed after the #EXEC instruction.

A program containing the #EXEC instruction may be executed, simulated, syntax checked or searched for a particular block. In all the cases, programs called upon using the #EXEC instruction are executed in the same conditions as the original program

## 6.2 Executing a block in the indicated channel

### #EXBLK

**Executes a block in the indicated channel**

With this instruction, it is possible to execute a block in the indicated channel. If the channel where it is to be executed is busy, it will issue the relevant error message. After executing the block, the channel goes back to the previous work mode.

The programming format is:

```
#EXBLK [<block>,<channel>]
```

| Parameter | Meaning |
|---|---|
| <block> | Block to be executed. |
| <channel> | Optional. Channel where the block is to be executed. |

```
#EXBLK [G01 X100 F550, 2]
   (The block is executed in channel 2)
#EXBLK [T1 M6]
   (The block is executed in the current channel)
```

If the channel is not indicated and the instruction is executed from the program, the block is executed in its own channel. If the channel is not indicated and the instruction is executed in MDI, the block is executed in the active channel.

# 6.3 Axis swapping



Initially, each channel has some axes assigned to it as set by the machine parameters. While executing a program, a channel may release its axes or request new axes. This possibility is determined by machine parameter AXISEXCH, which establishes whether an axis can change channels or whether this change is permanent or not.

A permanent change is maintained after the end of the program, after a reset and on power-up. The original configuration may be restored either by validating the general parameters and restarting or by a part-program that undoes the changes.

*It also restore the machine parameter settings if a checksum error occurs when powering up the CNC. .*

### Knowing if an axis can change channels

Machine parameter AXISEXCH may be consulted using the following variable.

V.MPA.AXISEXCH.Xn

Replace "Xn" with the name or logic number of the axis.

| Value | Meaning |
|---|---|
| 0 | It cannot change channels. |
| 1 | The change is temporary. |
| 2 | The change is permanent. |

### Knowing in which channel the axis is

It is possible to know in which channel the axis is by using the following variable.

V.[n].A.ACTCH.Xn

Replace "Xn" with the name or logic number of the axis.

Replace the "n" letter with the channel number.

| Value | Meaning |
|---|---|
| 0 | It is not in any channel. |
| 1-4 | Channel number. |

### Commands for modifying the axis configuration via program

The following instructions are used to modify the configuration of the axes. It is possible to add or remove axes, change their names and even redefine the main axes of the channel by swapping their names.

Changing the configuration of the axes cancels the active polar origin, the pattern rotation, the mirror image and the scaling factor.

In the configuration of the axes (if G17 is active), the axis that occupies the first position must be the abscissa axis, the second will be the ordinate axis, the third will be the axis perpendicular to the work plane, the forth will be the first auxiliary axis and so on.

## #SET AX

### Sets the axis configuration

Defines a new axis configuration in the channel. The channel axes not programmed in the instruction and the nonexistent programmed ones will be added. The axes are placed in the channel in the positions as they are programmed in the instruction #SET AX. Optionally, one or several offsets may be applied to the defined axes.

It is the same as programming a #FREE AX of all the axes and then a #CALL AX of all the new axes.

The instruction #SET AX may also be used only to order the existing axes in the channel differently.

The programming format is:

```
#SET AX [<Xn>,...] <offset> <...>
```

| Parameter | Meaning |
|---|---|
| <Xn> | Axes that make up the new configuration. If instead of defining an axis, a zero is written, an empty space (without an axis) appears in this position. |
| <offset> | Optional. It sets which offset is applied to the axes. Several offsets may be applied. |

```
#SET AX [X,Y,Z]
#SET AX [X,Y,V1,0,A]
```

### Offset setting

The offsets that may be applied to the axes are identified with the following commands. To apply several offsets, program the relevant commands separated by a blank space.

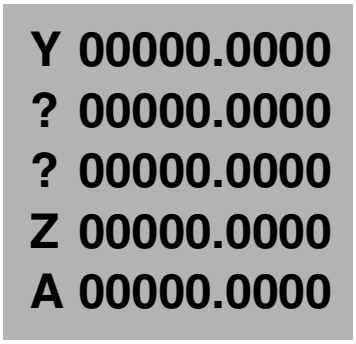
| Command | Meaning |
|---|---|
| ALL | Include all the offsets. |
| LOCOF | Include the offset of the reference search. |
| FIXOF | Include the fixture offset. |
| TOOLOF | Include the tool offset. |
| ORGOF | Include zero offset. |
| MEASOF | Include measurement offset. |
| MANOF | Include the offset of the manual operations. |

```
#SET AX [X,Y,Z] ALL
#SET AX [X,Y,V1,0,A] ORGOF TOOLOF
```

If when defining a new configuration only the order of the axes in the channel is swapped, the offsets are ignored.

### Screen display

At first, the axes appear ordered as they have been defined in the general machine parameter table (by channels) and then as the swapping is defined.




#SET AX [Y, 0, 0, Z, A] #SET AX [X, Y, Z] FIXOF ORGOF

*Screen display of the different configurations. Let us suppose a machine with 5 axes X-Y-Z-A-W.*

## #CALL AX

### Adds an axis to the configuration

it adds one or more axes to the preset configuration and it also allows defining its position. If the axis already exists in the configuration, it is placed in the new position. Optionally, one or several offsets may be applied to the defined axes.

The programming format is:

```
#CALL AX [<Xn>,<pos>...] <offset> <...>
```

| Parameter | Meaning |
|---|---|
| <Xn> | Axes to be added to the configuration. If the axis already exists, it is placed in the new position. |
| <pos> | Optional. Position of the axis in the new configuration. If not programmed, the axis is placed after the last one. If the position is occupied, the relevant error message will be issued. |
| <offset> | Optional. It sets which offset is applied to the axes. Several offsets may be applied. |

```
#CALL AX [X,A]
```
(It adds the X and A axes to the configuration, after the last existing axis)

```
#CALL AX [V,4,C]
```
(It adds the V axis to position 4 and the C axis after the last one)

### Offset setting

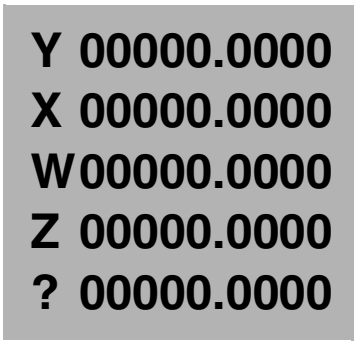
The offsets that may be applied to the axes are identified with the following commands. To apply several offsets, program the relevant commands separated by a blank space.

| Command | Meaning |
|---|---|
| ALL | Include all the offsets. |
| LOCOF | Include the offset of the reference search. |
| FIXOF | Include the fixture offset. |
| TOOLOF | Include the tool offset. |
| ORGOF | Include zero offset. |
| MEASOF | Include measurement offset. |
| MANOF | Include the offset of the manual operations. |

```
#CALL AX [X] ALL
#CALL AX [V1,4,Y] ORGOF TOOLOF
```

### Screen display

At first, the axes appear ordered as they have been defined in the general machine parameter table (by channels) and then as the swapping is defined.




**Axis configuration**

#SET AX [Y, 0, 0, Z]

Y: Abscissa axis.

Z: First auxiliary axis.

#CALL AX [X,2, W, 3]

Y: Abscissa axis.

X: Ordinate axis.

W: Axis perpendicular to the plane.

Z: First auxiliary axis.

## #FREE AX

### Frees an axis from the configuration

Removes the programmed axes from the current configuration. After removing an axis, the position is free, but the order of the axes that remain in the channel does not change.

The programming format is:

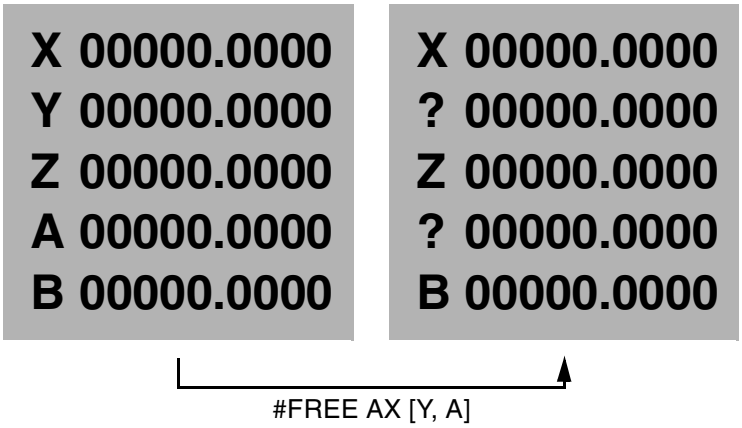
```
#FREE AX [<Xn>,...]
```

| Parameter | Meaning |
|---|---|
| <Xn> | Axis to be removed from the configuration |

```
#FREE AX [X,A]
    (It removes the X and A axes from the configuration)
#FREE AX ALL
    (Removes all the axes from the channel)
```

### Screen display

At first, the axes appear ordered as they have been defined in the general machine parameter table (by channels) and then as the swapping is defined.




*Screen display of the different configurations. Let us suppose a machine with 5 axes X-Y-Z-A-W.*

## #RENAME AX

### Renames the axes

It changes the name of the axes. For each programmed axis pair, the first axis takes the name of the second one. If the second axis is present in the configuration, it takes the name of the first one.

The change of the name of the axes only remains during the execution of the program. The original names of the axes are restored when starting the next program.

The programming format is:

```
#RENAME AX [<Xn1>,<Xn2>][...]
```

| Parameter | Meaning |
|---|---|
| <Xn1> | Axis whose name is to be changed |
| <Xn2> | new axis name. |

```
#RENAME AX [X,X1]
```
(The X axis is now called X1. If X1 already exists in the channel, it is called X)

```
#RENAME AX [X1,Y][Z,V2]
```

## 6.4 Spindle swapping

The CNC can have up to four spindles distributed between the various channels of the system. A channel may have one, several or no spindles associated with it.

Initially, each channel has some spindles assigned to it as set by the machine parameters. While executing a program, a channel may release its spindles or request new spindles. This possibility is determined by machine parameter AXISEXCH, which establishes whether a spindle can change channels or whether this change is permanent or not.

A permanent change is maintained after the end of the program, after a reset and on power-up. The original configuration may be restored either by validating the general parameters and restarting or by a part-program that undoes the changes.

*It also restore the machine parameter settings if a checksum error occurs when powering up the CNC. .*

### Knowing if a spindle can change channels

Machine parameter AXISEXCH may be consulted using the following variable.

V.MPA.AXISEXCH.Sn

Replace "Sn" with the spindle name.

| Value | Meaning |
|---|---|
| 0 | It cannot change channels. |
| 1 | The change is temporary. |
| 2 | The change is permanent. |

### Knowing in which channel the spindle is

It is possible to know in which channel the spindle is by using the following variable.

V.[n].A.ACTCH.Sn

Replace "Sn" with the spindle name.

Replace the "n" letter with the channel number.

| Value | Meaning |
|---|---|
| 0 | It is not in any channel. |
| 1-4 | Channel number. |

### Commands for modifying the spindle configuration via program

The following instructions are used to modify the configuration of the spindles of the channel. It is possible to add or remove spindles, change the name of the spindles and define which one is the master spindle of the channel.

## #FREE SP — Frees a spindle from the configuration

Removes the defined spindles from the current configuration.

The programming format is:

```
#FREE SP [<Sn>,...]
#FREE SP ALL
```

| Parameter | Meaning |
|---|---|
| <Sn> | Spindle name. |
| ALL | Frees all the spindles of the channel. |

```
#FREE SP [S]
    (It removes the spindle S from the configuration)
#FREE SP [S1,S4]
    (It removes spindles S1 and S4 from the configuration)
#FREE SP ALL
    (It removes all the spindles from the configuration)
```

## #CALL SP — Add a spindle to the configuration

It adds one or several spindles to the current configuration. The position of the spindles in the channel is not relevant. To add a spindle to the channel, the spindle must be free; it must not be in another channel.

The programming format is:

```
#CALL SP [<Sn>,...]
```

| Parameter | Meaning |
|---|---|
| <Sn> | Spindle name. |

```
#CALL SP [S1]
    (It adds spindle S1 to the configuration)
#CALL SP [S,S2]
    (It adds spindles S and S2 to the configuration)
```

## #SET SP — Sets the spindle configuration

Defines a new spindle configuration. The spindles existing in the channel and not programmed in #SET SP are removed and those programmed that are not already in the channel will be added.

It is the same as programming a #FREE SP of all the spindles and then a #CALL SP of all the new spindles. The programming format is:

```
#SET SP [<Sn>,...]
```

| Parameter | Meaning |
|---|---|
| <Sn> | Spindle name. |

```
#SET SP [S]
    (Configuring one spindle)
#SET SP [S1,S2]
    (Configuring two spindles)
```

## #RENAME SP — Rename the spindles

It changes the name of the spindles. For each programmed spindle pair, the first spindle takes the name of the second one. If the second spindle is present in the configuration, it takes the name of the first one.

The change of the name of the spindles only remains during the execution of the program. The original names of the spindles are restored when starting the next program.

The programming format is:

```
#RENAME SP [<Sn>,<Sn>][...]
```

| Parameter | Meaning |
|---|---|
| <Sn> | Spindle name. |

```
#RENAME SP [S,S1]
#RENAME SP [S1,S2][S3,S]
```

### 6.4.1 Selecting the master spindle of a channel

**#MASTER** — **Establishes the master spindle of a channel**

The master spindle is the main spindle of the channel. It is the spindle that receives the commands when no specific spindle is mentioned.

The programming format is:

```
#MASTER <Sn>
```

| Parameter | Meaning |
|---|---|
| <Sn> | Spindle name. |

```
#MASTER S
#MASTER S2
```

If no master spindle is indicated, it assumes one according to the following criteria. In general, whenever a channel has a single spindle, it will be its master spindle.

- If the whole system only has one spindle, it will be the master spindle of the current channel.
- If a spindle is added to a channel that does not have one, it will be the master spindle.
- If a channel releases its master spindle and it has only one spindle left, this one will be its new master spindle.
- If a channel having two spindles but no master spindle releases one of them, the remaining one will be its master spindle.
- At first, in a channel with several spindles, the master spindle will be the one configured by machine parameters.
- If two or more spindles remain in a channel and none of the previous rules may be applied, the master spindle must be defined using the #MASTER instruction.

The same treatment described for adding or removing spindles is applied for parking and unparking spindles.

On startup, it follows the same criteria to decide which is the master spindle of the channel. If this spindle is parked, it will assume the next spindle, if there is one, as master spindle of the channel.

## 6.5 Communication and synchronization between channels

Each channel may execute its own program simultaneously and independently from other channels. But, besides this, it can also communicate with other channels, transfer information or synchronize in specific points.

The communication takes place on the basis of a number of marks managed by the part-programs of each channel. These marks establish whether the channel is waiting to be synchronized or it may be synchronized, etc.

There are two different ways to synchronize, each offers a different solution.

- Using the #MEET instruction.

  The easiest way to synchronize. It stops the execution in all the channels involved in the synchronization.

  The set of marks being used are initialized after executing an M02 or an M30, after a reset or on power-up.

- Using the instructions #WAIT - #SIGNAL - #CLEAR.

  This method is somewhat more complicated than the previous one, but more versatile. It does not stop the execution in all the channels in order to synchronize.

  The set of marks being used are maintained after executing an M02 or an M30, after a reset or on power-up.

The synchronism marks of the two methods are independent from each other. The marks managed by the #MEET instruction neither affect nor are affected by the rest of the instructions.

### Other ways to synchronize channels

The common arithmetic parameters can also be used to communicate and synchronize channels. By writing a certain value from a channel and later reading it from another channel, it is possible to set the condition to follow up on the execution of a program.

Accessing the variables of a channel from another channel can also be used as a way to communicate.

Swapping axes between channels also makes it possible to synchronize processes, because a channel cannot grab an axis until it has been released by another one.

| CHANNEL 1 | CHANNEL 2 | CHANNEL 3 |
|---|---|---|
| G1 F1000 | X1=0 Y1=0 Z1=0 | G1 F1000 |
| S3000 M3 | G1 F1000 | X2=20 Z2=10 |
| #FREE AX [Z] | #FREE AX[Z1] | #FREE AX[Z2] |
| (Frees the Z axis) | (Frees the Z1 axis) | (Frees the Z2 axis) |
| X30 Y0 | G2 X1=-50 Y1=0 I-25 | X2=100 Y2=50 |
| #CALL AX [Z1,Z2] | #CALL AX [Z] | #CALL AX[Z2] |
| (It adds the Z1 and Z2 axes) | (Adds the Z axis) | (Recovers the Z2 axis) |
| X90 Y70 Z1=-30 Z2=-50 | G1 X1=50 Z20 | G0 X2=0 Y2=0 Z2=0 |
| #FREE AX [Z1,Z2] | #FREE AX[Z] | M30 |
| (Frees the Z1 and Z2 axes) | (Frees the Z axis) | |
| X0 | X1=20 | |
| #CALL AX [Z] | #CALL AX [Z1] | |
| (Recovers the Z axis) | (Recovers the Z1 axis) | |
| G0 X0 Y0 Z0 | G0 X1=0 Y1=0 Z1=0 | |
| M30 | M30 | |

### Consultation variables

The information about the status of the synchronization marks may be consulted using the following variables.

- MEET or WAIT type mark expected by the "n" channel from the "m" channel

  `V.[n].G.MEETCH[m]`

  `V.[n].G.WAITCH[m]`

  Replace the letters "n" and "m" with the channel number.

- Status of the MEET or WAIT type "m" mark in the "n" channel

  `V.[n].G.MEETST[m]`

  `V.[n].G.WAITST[m]`

## #MEET

**It activates the mark indicated in the channel and waits for it to be activated in the rest of the programmed channels.**

This instruction, after activating the mark in its own channel, waits for it to also be active in the programmed channels before resuming the execution. Each channel has 10 marks that are numbered from 1 to 10.

Programming the same instruction in several channels, all of them stop and wait for the rest to reach the indicated point before they all resume the execution at the same time from that point on.

The programming format is:

```
#MEET [<mark>, <channel>,...]
```

| Parameter | Meaning |
|---|---|
| <mark> | Synchronization mark that is activated in the channel itself and must be activated in the rest of the channels before going on. |
| <channel> | Channel or channels where the same mark must be activated. |

There is no need to include the number of its own channel in each instruction because the mark is activated when executing the #MEET instruction. However, it is recommended to program it in order to make the program more understandable.

### Operation

Programming the same instruction in each channel, all of them are synchronized at that point and the execution resumes from there on. It works as follows.

1. It activates the mark selected in its own channel.
2. It waits for the mark to be activated in all the indicated channels.
3. After synchronizing the channels, it deletes the mark from its own channel and goes on executing the program.

Each channel stops on its #MEET. When the last one of them reaches the command and checks that all the marks are active, the process unlocks for all of them at the same time.

In the following example, it waits for mark ·5· to be active in channels ·1·, ·2· and ·3· to synchronize the channels and resume the execution.

| CHANNEL 1 | CHANNEL 2 | CHANNEL 3 |
|---|---|---|
| %PRG_1 | %PRG_2 | %PRG_3 |
| ... | ... | ... |
| ... | #MEET [5,1,2,3] | ... |
| #MEET [5,1,2,3] | ... | ... |
| ... | ... | ... |
| ... | ... | #MEET [5,1,2,3] |
| M30 | M30 | M30 |

## #WAIT — It waits for the mark to be activated in the indicated channel

The #WAIT instruction waits for the indicated mark to be active in the specified channels. If the mark is already active when executing the command, the execution is not interrupted and the program keeps running.

Each channel has 10 marks that are numbered from 1 to 10.

The programming format is:

```
#WAIT [<mark>, <channel>,...]
```

| Parameter | Meaning |
|---|---|
| <mark> | Synchronization mark waited for to be activated. |
| <channel> | Channel or channels that must activate the mark. |

As opposed to the #MEET instruction, it does not activate the indicated mark of its own channel. The marks of the channel are activated using the instruction #SIGNAL.

## #SIGNAL

**It activates the mark in its own channel**

The #SIGNAL instruction activates the indicated marks in its own channel. Each channel has 10 marks that are numbered from 1 to 10. These marks correspond to the #WAIT instructions.

This instruction does not perform any wait; it goes on executing. Once synchronized, the marks are deactivated, if so wished, using the #CLEAR instruction.

The programming format is:

```
#SIGNAL [<mark>,...]
```

| Parameter | Meaning |
|---|---|
| <mark> | Synchronization marks that is activated in the channel. |

## #CLEAR

**It clears the synchronism marks of the channel**

This instruction activates the indicated marks in its own channel. If no marks are programmed, it deletes all of them.

The programming format is:

```
#CLEAR
#CLEAR [<mark>,...]
```

| Parameter | Meaning |
|---|---|
| <mark> | Synchronization marks that is deleted in the channel. |

In the following example, channels ·1· and ·2· wait for mark ·5· to be active in channel ·3· to synchronize. When mark ·5· is activated in channel ·3·, it resumes the execution in all three channels.

| CHANNEL 1 | CHANNEL 2 | CHANNEL 3 |
|---|---|---|
| %PRG_1 | %PRG_2 | %PRG_3 |
| ... | ... | ... |
| ... | #WAIT [5,3] | ... |
| #WAIT [5,3] | ... | ... |
| ... | ... | #SIGNAL [5] |
| ... | ... | ... |
| ... | ... | #CLEAR [5] |
| M30 | M30 | M30 |

# MULTIPLE SPINDLES. SPINDLE CONTROL

7

The CNC can have up to four spindles distributed between the various channels of the system. A channel may have one, several or no spindles associated with it. Via part-program or MDI, it is possible to indicate which spindle the commands are directed to; when not indicated, the commands are directed to the master spindle of the channel.

All the spindles of the channel may be running at the same time. Also, each of them may be in a different mode; turn in different directions, be in positioning mode, etc.

## Master spindle of the channel

The master spindle is the main spindle of the channel. It is the spindle that receives the commands when no specific spindle is mentioned.

When starting the CNC up, it assumes the first spindle defined in the channel as master spindle of the channel. If this spindle is parked, it assumes the next one, if there is one.

### Master spindle after modifying the configuration of the channel.

If no master spindle is indicated, it assumes one according to the following criteria. In general, whenever a channel has a single spindle, it will be its master spindle.

- If the whole system only has one spindle, it will be the master spindle of the current channel.
- If a spindle is added to a channel that does not have one, it will be the master spindle.
- If a channel releases its master spindle and it has only one spindle left, this one will be its new master spindle.
- If a channel having two spindles but no master spindle releases one of them, the remaining one will be its master spindle.
- At first, in a channel with several spindles, the master spindle will be the one configured by machine parameters.
- If two or more spindles remain in a channel and none of the previous rules may be applied, the master spindle must be defined using the #MASTER instruction.

### Master spindle after parking or unparking spindles

The same treatment is applied as when modifying the configuration of the channel.

# 7.1 Define the master spindle of a channel

The master spindle is the main spindle of the channel. It is the spindle that receives the commands when no specific spindle is mentioned.

When starting the CNC up, it assumes the first spindle defined in the channel as master spindle of the channel. If this spindle is parked, it assumes the next one, if there is one.

If after modifying the configuration of the channel's spindles there are two or more spindles in it, it must be defined which one is the master spindle. To select or modify the master spindle of a channel, use the #MASTER instruction.

## #MASTER Establishes the master spindle of a channel

This instruction may be used to define or change the master spindle of a channel. The master spindle may be selected at any time.

The programming format is:

```
#MASTER <Sn>
```

| Parameter | Meaning |
|---|---|
| <Sn> | Spindle name. |

```
#MASTER S
#MASTER S2
```

# 7.2 Spindle speed

## Spindle speed programming

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

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

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

### Spindle start and stop

Defining a speed does not imply starting the spindle. The startup is defined using the following auxiliary functions. See *"7.4 Spindle start and stop"* on page 82.

M03 - Starts the spindle clockwise.

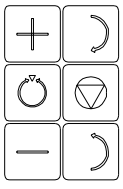
M04 - Starts the spindle counterclockwise.

M05 - Stops the spindle.

## Maximum speed

The maximum turning speed in each range (gear) is limited by the machine parameter G00FEED. When programming a higher turning speed, the CNC limits its value to the maximum allowed by the active range (gear). The same thing occurs when trying to exceed the maximum limits using the "+" and "-" keys of the operator panel or doing it via PLC or by program.

## Speed override



The programmed "S" speed may be varied between 50% and 120% using the "+" and "-" of the operator panel or via PLC. However, the maximum and minimum variation may be different depending on how the machine parameters MINOVR and MAXOVR have been set.

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

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

# 7.3 Constant Surface Speed mode selection



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

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

*The constant surface speed mode is oriented to lathe type machines. In order for it to be available, the machine manufacturer must have set one of the axis -face axis- (FACEAXIS), usually axis perpendicular to the shaft of the part.*

## Speed mode selection

The functions to select the spindle work mode are the following:

G96 - Constant surface speed.

G97 - Constant turning speed.

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

### G96 Constant surface speed

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

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

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

### G97 Constant turning speed

The G97 function affects all the spindles of the channel.

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

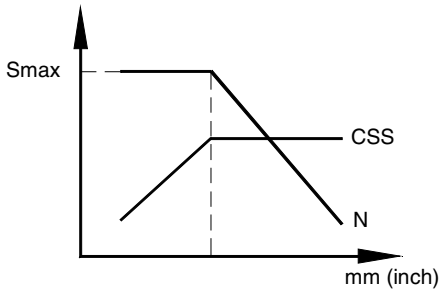
It is recommended to program the speed in the same block as the G97 function; if not programmed, the CNC assumes as programmed speed the one the spindle is currently turning at. The spindle gear (range) (M41, M42, M43, M44) may be selected at any time.

## Properties of the functions

Functions G96 and G97 are modal and incompatible with each other. On power-up, after executing an M02 or M30, and after an EMERGENCY or RESET, the CNC assumes function G97.

## 7.3.1 Turning speed limitation in constant surface speed mode

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




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

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

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

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

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

```
G192 S2500
    (Maximum turning speed = 2500 rpm)
G96 S180
    (Constant surface speed. =180 m/min.)

G97 S1000 M3
    (Constant turning speed. = 1000 rpm.)

G96
S230
    (It activates constant surface speed mode.)
    (The turning speed limit stays active at 2500 rpm.)
```

# 7.4 Spindle start and stop

Defining a speed does not imply starting the spindle. The startup is defined using the following auxiliary functions.

M03 - Starts the spindle clockwise.

M04 - Starts the spindle counterclockwise.

M05 - Stops the spindle.

## Start the spindle clockwise or counterclockwise

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

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

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

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

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

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

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

## Stop the spindle

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

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

M5.S Function M5 associated with the spindle S.

```
S1000 S2=456 M5
    (Stops the master spindle)
M5.S M5.S2 S1=1000 M3.S1
    (Stops the spindles "S" and "S2")
    (Spindle "S1" turns clockwise)
```

# 7.5 Spindle orientation

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

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

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

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

```
M19 S0
    (Positioning of spindle S at 0º)
M19 S2=120
    (Positioning of spindle S2 at 120º)
M19 S1=10 S2=34
    (Positioning of spindle S1 at 10º and S2 at 34º)
```

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

M19.S1 Positioning of spindle S1 at 0º.

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

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

## How is positioning carried out

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

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

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

## Setting the turning direction for spindle orientation

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

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

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

  M19.POS Positioning in the positive direction.

  M19.NEG Positioning in the negative direction.

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

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

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

## Positioning speed

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

Sn.POS Positioning speed of spindle Sn.

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

The positioning speed is given in rpm.

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

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

## 7.6 Gear change

The CNC may have up to 4 different spindle gears. When defining a manual gear change, one must define the gear to be used and the spindle speed. When using an automatic gear change, the CNC itself selects the proper gear. The type of gear change is set by machine parameter AUTOGEAR.

| | |
|---|---|
| AUTOGEAR=YES | The change is automatic. |
| AUTOGEAR=NO | The change is manual. |

The spindle gear to be used is defined with the following auxiliary functions.

| | |
|---|---|
| M41 | - Selects gear ·1·. |
| M42 | - Selects gear ·2·. |
| M43 | - Selects gear ·3·. |
| M44 | - Selects gear ·4·. |

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

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

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

| | |
|---|---|
| M41.S | Function M41 associated with the spindle S. |

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

FAGOR

CNC 8070

(SOFT V02.0X)