CNC 8037 -M-

New features

Ref.1507
Soft: V02.2x

FAGOR

FAGOR AUTOMATION

This product uses the following source code, subject to the terms of the GPL license. The applications *busybox* V0.60.2; *dosfstools* V2.9; *linux-ftpd* V0.17; *ppp* V2.4.0; *utelnet* V0.1.1. The library *grx* V2.4.4. The linux kernel V2.4.4. The linux boot *ppcboot* V1.1.3. If you would like to have a CD copy of this source code sent to you, send 10 Euros to Fagor Automation for shipping and handling.

INDEX

VERSION V02.00

VERSION V02.03

VERSION V02.10

VERSION V02.21

FAGOR

CNC 8037

·M· MODEL

VERSION V02.00

1 Calls to subroutines using G functions

When the OEM needs to implement special cycles on his machines, these cycles would usually be implemented using subroutines. Subroutines are used among other things for defining a bunch of operations or movements that are repeated several times throughout the program. These subroutines may be called upon once or several times from another subroutine or from another program.

This feature may be used to call subroutines using specific G functions instead of CALL and PCALL that were used until this version. This way, the calls to subroutines are more similar to machine tool language.

1.1 Functionality

Functions G180-G189 and G380-G399 can call the OEM and user subroutine as long as they are global subroutines. G functions cannot be used to call local subroutines.

Up to 30 subroutines may be defined and associated with functions G180-G189, G380-G399, and it is also possible to initialize local parameters for each subroutine.

When executing one of these functions, its associated subroutine will be executed.

Functions G180-G189 and G380-G399 are not modal.

Programming format

The programming format is the following:

```
G180 <P0..Pn>
<P0..Pn> Optional. Initializing parameters.
```

Example:

```
G183 P1=12.3 P2=6
G187 A12.3 B45.3 P10=6
```

Defining local parameters.

The values of the parameters are defined after the call function and they may be defined using the name of the parameter (P0-P25) or using letters (A-Z) so "A" is the same as P0 and "Z" is the same as P25.

Plus, parameters may also be programmed as follows:

- S=P100
- SP100

In either case, local parameter P18(S) would assume the value of global parameter P100 set.

The definitions described here may be combined in the same block.

Nesting levels

If the functions initialize local parameters, this instruction generates a new nesting level .

The maximum parameter nesting level is 6, within the 15 nesting level of the subroutines, just like PCALL instructions.

Identification via PLC

All the G functions are identified through GGS* read-only variables . GGSH and GGSP read-only variables are used to identify the new G functions via PLC; these variables return the status of the G functions.

The status of each one of the functions will be given in the 25 least significant bits and it will be indicated by a 1 when active and a 0 when not active or when not available in the current software version.

GGSH:

G199	G198	G197	G196	G195	…	G179	G178	G177	G176	G175

G180: Bit 5
G181: Bit 6
...
G189: Bit 14

GGSP:

G399	G398	G397	G396	G395	…	G379	G378	G377	G376	G375

G380: Bit 5
G381: Bit 6
...
G399: Bit 24

Executing a call

Each function G180-G189 and G380-G399, has its corresponding subroutine associated with it. Calling a G function means calling the subroutine of the same name.

2 Anticipated tool management

This feature is used to optimize the time required to change tools on machines that have a tool magazine with a tool changer arm. This optimization consists in preparing the magazine while machining to pick up the tool that will be used in the next operation.

Until now, in the consecutive execution of conversational tools, when a cycle execution was completed, it would prepare the tool required for the next cycle. In other words, the whole tool change process was done after the previous cycle was completed.

Using this feature, the CNC knows well ahead of time which tool will be used next and it requests it some time before it needs it. The magazine moves to the position of the tool and it stays waiting for the tool change. This reduces part production time.

This optimization will also be noticed when programming in ISO code without having to program the next T ahead of time.

2.1 PLC machine parameters, variables and marks

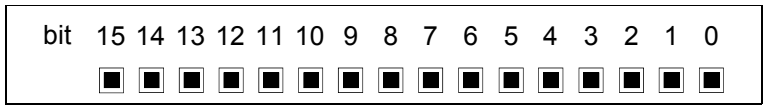
For this feature to work, the machine must have a tool magazine with automatic tool changer and this requires the following:

- g.m.p. TOFFM06 (P28) = YES. Indicates that the machine is a machining center.
- g.m.p. NPOCKET (P24) other than 0. Indicates the number of pockets in the tool magazine.

TOOLTYPE (P167)

Bit 12 of general machine parameter TOOLTYPE (P167) enables the new management of tools and tool magazine. This parameter defines the behavior of the tool or of the tool offset.

It has 16 bits counted from right to left.



Each bit has a function or work mode associated with it. By default, all the bits will be assigned the value of ·0·. Assigning the value of ·1· activates the corresponding function.

Bit	Meaning
0 - 11	Not being used.
12	Enables/disables anticipated tool management.
13	The STOP signal is always executed after the "T" function.
14	Machining in round corner mode when changing the tool offset.
15	Stop block preparation when executing a new "T".

Default value in all the bits: 0

/XINHMZ

The PLC uses this mark to indicate that it is operating to returning a tool to the tool magazine. Using this mark, it is possible to reduce the time the AUXEND signal is kept low when executing an M6, so as not to block the execution of the program.

The PLC must set logic input /XINHMZ low with the M6 to indicate that it is running to return the previous tool to the magazine. When all the steps of the the execution of the M6 are completed and the magazine is ready to make another tool change, the PLC must set logic input /XINHMZ high.

Logic input /XINHMZ (Xfer INHibit MagaZine) is used to make the tool change safer and more efficient.

2.2 Operation

Advanced (anticipated) execution of the next T

Anticipated tool management consists in executing the T in advance; i.e. it may take place up to 1000 blocks before executing the corresponding M6. Using this feature does not require programming anything special.

If two T's have been programmed without an M6 between them, the first one does not allow the tool change.

The anticipated tool management does not begin until the last M, S or T operation is completed. Also, while anticipated tool management is in execution, no PLC operation for M, S or T will be executed until the tool management is completed.

When using anticipated tool management, the new T appears in the history as a pending tool in advance and the tool magazine is also refreshed in advance.

When going into tool inspection mode, when the management of the next T has been anticipated, even if the tool is changed during inspection, the next T will be prepared again after the inspection.

Since the anticipated T management begins when the preparation finds the corresponding M6, if there is a program containing a T and then an M30, without an M6 in between, in this case, the M30 will give way to the anticipated management of that T. This way, it may prepare the T required by the next program to be executed.

Example:

In the following example, the tool change is programmed in two consecutive blocks (Tn and M6). In this case, the execution of the T will be advanced (anticipated) as much as possible. If there is any program block that interrupts block preparation, the anticipation will not be of 1000 blocks, but it will begin once that block is executed.

In the following program:

```
N1 ……..
N2 ……..
……..
N500 (P100 = TPOSX)
N501…….
……..
N600 T2
N601 M6
……..
```

The execution of block T2 can only begin after block (P100 = TPOSX), that interrupts preparation, is executed. If that block is eliminated, the execution of T2 could begin at the same time as block N1.

Anticipation when returning the tool to the magazine

This management may be added in the PLC to improve tool change execution time while making sure that the magazine cycle is completed before starting the next tool change.

For the tool change to be safer and more efficient, the PLC must set logic input /XINHMZ low with the M6 to indicate that it is running to return the previous tool to the magazine. When all the steps of the the execution of the M6 are completed and the magazine is ready to make another tool change, the PLC must set logic input /XINHMZ high. This way, as soon the PLC receives the data of the M6, it can raise the AUXEND signal and the part program may keep running while the tool is being returned to the magazine.

With this management, is an error occurs in the last part of the PLC operation, when the history and the magazine table have already changed, it prevents the real status of the magazine from not matching the one assumed at the CNC.

This management will only take place if the T management is anticipated; i.e. if it is being executed in automatic mode and with bit 12 of g.m.p. TOOLTYPE (P167) = 1.

The part program is not considered completed until the /XINHMZ mark is set to 1 indicating that the tool has been put in the magazine. This is good for M30, M2 and the last line of the program.

While the /XINHMZ mark is low, the T will be highlighted in reverse video both in ISO mode and in conversational mode.

Ground tools

A ground tool is one that is in the tool table but not in the magazine table.

A ground tool does not occupy any position in the magazine and is identified with the value -4 in TBCD (when requesting the tool) and in T2BCD (when returning it to the magazine).

Ground tool processing is activated with the same bit of anticipated T.

2.3 Improved tool magazine management

From this version on, additional information will be provided to make tool and magazine management more comfortable.

The idea is that when executing a T function or an M6, the PLC can have all the necessary information abut the new tool that has been requested and about the tool to be returned to the magazine.

This way, on one hand, there is no need to save the information sent out with the T (TBCD) in PLC registers in order to use it when executing the M6. And on the other, between the execution of the T and that of the M6 the tool magazine may have been changed by hand or variables TOOL or NXTOOL may have been changed from the PLC; therefore, the information sent out with the T is no longer correct and it must be updated (refreshed) with the M6.

The following information is applied to all types of magazine with an automatic tool changer. These magazines may be the following:

- Non-random magazine with g.m.p. TOOLMATY(P164)=0.
- Non-random magazine with g.m.p. TOOLMATY(P164)=1.
- Random magazine.

The PLC mark "T2STROBE" (M5535) and the corresponding PLC register "T2BCD" (R559) indicate the magazine position where the active tool is to be left. Until now, they were provided with M6 in the following cases:

- In non-random magazines.
- In random magazines when one of the tools involved in the change is special.

In the rest of the cases, it did not send this information because it was assumed that the active tool was to be left in the magazine position where the new one was going to be picked up.

From this version on, marks "T2STROBE" (M5535) and "T2BCD" (R559) will be sent out with the M6 for any type of magazine with an automatic tool changer as long as bit 12 of g.m.p. TOOLTYPE (P167) is active.

New PLC register "NT2BCD" and mark "NT2STROBE"

In order to make it easier to manage the magazine from the PLC point of view, a new improvement has been implemented that consists in that the CNC provides the PLC with information on the tool that is going to be returned to the magazine as soon as possible . Until this version, this information was sent out in "T2BCD" (R559) with the M6 about the tool that is going to be returned to the magazine in the following cases:

- If the magazine is non-random with tool changer.
- If the magazine is random, when the tool being picked up or the one being returned is special.

From this version on, this information is sent out before executing the T. When a new T is executed, the CNC outputs in the new PLC register "NT2BCD" (R572) the magazine position where the active tool will be put and it also indicates with the new PLC mark "NT2STROBE" (M5573) that there is new information in "NT2BCD". When executing the next M6, the value of register "NT2BCD" will be transferred to register "T2BCD".

This information will be sent out for any type of magazine with an automatic tool changer as long as bit 12 of g.m.p TOOLTYPE (P167) is active.

Duplicate the information given in TBCD and TSTROBE with M6

This improvement consists in duplicating the information provided with the execution of the T(TBCD register and TSTROBE mark) when executing the next M6. This is done for safety reasons, just in case between the execution of the T and that of the M6 an interruption has occurred for manual intervention or the machine has been shut down.

This improvement works when bit 12 of g.m.p. TOOLTYPE (P167) is active.

New CNC variables to be used in the subroutine associated with M6

Inside the M6 subroutine, it is useful to know whether the tools involved in the change are ground tools or not in order to move the machine to the various tool change positions. Two new CNC variables are now available to make programming the subroutine easier.

Variable PNXTOOL:

Magazine position to pick up the next tool. It matches the value that will be received later on in the register "TBCD" (R558) with the M6, but the latter will be in BCD format.

Variable PTOOL:

Magazine position to leave the current tool. It matches the value that will be received later on in the register "T2BCD" (R559) with the M6, but the latter will be in BCD format.

Both variables are read-only, they interrupt block preparation can only be accessed from the CNC. This improvement works when bit 12 of g.m.p. TOOLTYPE (P167) is active.

3 Improved screen customization

Some CNC screens may be customized by the manufacturer of the machine to show more information or more screens.

From this CNC version on, together with V3 of WINDRAW55, it will be possible to manage a 32-color palette as well as "PNG" and "JPG" graphic elements.

32-color palette

The fact that WINDRAW55 can handle a 32-color palette makes it possible to create new screens and adapt old screens or cycles to the current look. More colors may be selected on the palette of the WINDRAW55 when drawing or editing texts.

Compatibilities:

A 32-color graphic file cannot be opened in older CNC versions than this one. When trying to open a ".WGD" an error message will be issued indicating incompatibility between versions. On the other hand, when trying to open a ".PAN" or ".SIM" file, the 32-color graphic elements will not be displayed.

Managing "PNG" and "JPG" graphic elements

From this version on, PNG and JPG/JPEG type graphic elements will be handled. These graphic elements may be used on standard screens of the CNC such as icons or customized using version V3 of the WINDRAW55 program.

This type of graphic elements cannot be handled directly with the CNC screen customizing tools on its graphic editor, nor can they be displayed on the CNC customized screen because only ".SIM" and ".PAN" items can be displayed.

There is now an option to define the customized screen (page) zero with PNG/JPEG type images. When having several images, the order of priorities will be the following:

1. 000.jpg, 000.jpeg, 000.JPG, 000.JPEG
2. 000.png, 000.PNG
3. 000.pan
4. Fagor standard start-up (home) screen.

Loading graphic elements:

PNG and JPG type graphic elements of the WINDRAW55 are loaded into the CNC via FTP or using a pendrive through the USB port. When using a pendrive, it will do a general BACKUP / RESTORE and when selecting the "pages and symbols" option, it will automatically include the PNG/JPG files in that BACKUP / RESTORE.

4 Modifications to several spindle parameters

The maximum values and decimals of several spindle parameters have been changed.

The new values are:

MAXGEAR1 (P2) MAXGEAR2 (P3) MAXGEAR3 (P4) MAXGEAR4 (P5)

They indicate the maximum spindle speed assigned to each gear.

On large lathes where 1 rpm may be a very high speed, it is important to be able to set the maximum rpm with decimals for the gears. Also, on some machines, it is interesting to be able to use spindles with more than 65535 rpm.

From this version, its value will be up to 99999,999 with 3 decimals.

When using an automatic gear change, these values will be used to make the change.

MAXGEAR1	for gear 1 (M41).
MAXGEAR2	for gear 2 (M42).
MAXGEAR3	for gear 3 (M43).
MAXGEAR4	for gear 4 (M44).

Possible values
Values with 3 decimals between 0.000 and 99999.999 rpm.

Default value: For MAXGEAR1 (P2) = 1000 rpm.
For MAXGEAR2 (P3) = 2000 rpm.
For MAXGEAR3 (P4) = 3000 rpm.
For MAXGEAR4 (P5) = 4000 rpm.

When not using all 4 gears, use the lower ones and set the unused ones to the same value as the highest one used.

SLIMIT (P66)

Maximum safety limit for the spindle speed. From this version, its value will be up to 99999,999 with 3 decimals. This limit is activated from the PLC and is applied in all the work modes, including the PLC channel. When the spindle is controlled by the PLC by means of the PLCCNTL mark, this limit is ignored.

Possible values
Values with 3 decimals between 0.000 and 99999.999 rpm.

Default value: 0

This limit is activated using the mark SLIMITAC (M5059). When this limit is canceled, the CNC recovers the programmed speed.

This limit permits clearing the spindle speed temporarily via PLC, e.g. when opening the doors, etc.

When upgrading from an older version to a V02.0x, the parameters indicated above will then have 3 decimals with a 0 value. The integer part will keep its previous value.
When downgrading from a version V02.0x to an older one, the parameters indicated earlier will lose their decimals but they will keep their integer values.

5 Retracing function of 2000 blocks

The retracing function will be handled like in previous versions except that the number of blocks executed that can be retraced will now be 2000.

Since it is now possible to retrace up to 2000 blocks, the distance that may be retraced will also be much greater.

6 Quick block search

The quick block search is oriented to CAD/CAM programs that contain at the beginning the blocks that set the machining conditions and then a number of motion blocks. In this type of programs, the interruption point may be found much faster.

However, if there is a block in the middle of the program that changes the machining conditions set at the beginning of the program, finding the interruption point may be slower.

With this new block search, some of the motion blocks already executed will NOT be simulated. The CNC does not simulate the blocks between the last point that do not contain coordinates, labels or comments and the block of the interruption point of the program.

This way, it skips between blocks stopping at the block just before the interruption block; in other words, at the block executed last. This lets resume the program from a slightly previous position to the interruption point, thus ensuring proper continuity of the machining operation that was being carried out.

When interrupting, the CNC automatically identifies whether it is possible to perform a quick block search or not in the interrupted program. If it is possible, this search may be initiated with the "Automatic Search" softkey.

If in the execution prior to the interruption there are blocks associated with canned cycles and/or calls to blocks that involve a level change, like for example PCALL or RPT, the quick search will not be applied, only the normal search.

7 Local subroutines in a program

From this version on, it is possible to define local subroutines inside a program. These subroutines are executed from RAM or hard disk memory.

The local subroutine is defined as part of a program. This subroutine may only be called upon from the program where it has been defined.

Programming

The local subroutines are located at the beginning of the program, before the actual beginning of the program. Local subroutines are defined by programming (LSUB n), where n indicates the subroutine number. Followed by the contents of the subroutine.

The range of local subroutines is from 0 to 9999.

```
(LSUB 0)
9999)
```

The actual beginning of the program is identified with the % sign. Any text may follow this character.

A local subroutine may be called upon using the commands CALL, PCALL or MCALL. When executing the calls, it first looks for the subroutines defined as local in that program and having matching names. If there aren't any, it will look among the global subroutines.

To execute a local subroutine directly, program (LL n). This way, only the local subroutine will be executed. If this subroutine does not exist, it will not execute anything and it will issue an error message indicating undefined subroutine.

Up to 100 local subroutines may be defined in a program. The maximum local subroutine nesting level is 15.

Examples:

Example 1:

```
(LSUB9505)
X100
(RET)

%**** ; beginning of program
(CALL 9505)
M30
```

Example 2:

```
(LSUB9505)
X100
(RET)

%**** ; beginning of program
(LL9505)
M30
```

Executing programs:

(LL n) Call to a local subroutine.
Parameters cannot be initialized with this command.
(CALL n) Call to a local or global subroutine.
Parameters cannot be initialized with this command.
(PCALL n ...) Call to a global or local subroutine.
Local parameters can be initialized with this command.
(MCALL n ...) Modal call to a local or global subroutine.
Local parameters can be initialized with this command.

Limitations:

A local subroutine can call a global subroutine but a global subroutine cannot call a local subroutine except if that local subroutine is defined in the root program; in other words, in the first program that is executed.

Local subroutines defined inside a program that has been called with the "EXEC" command are ignored. It only takes into account the ones defined in the root program.

It only takes into account local subroutines that are in programs that are executed from the CNC execution channel, either in ISO mode or in conversational mode. Local subroutines cannot be executed from the PLC channel.

8 Avoid spindle stop with M30 or Reset

This feature is especially useful on grinders and large lathes where usually the spindle must not be stopped. It lets end program execution with an M30 or reset and start again without stopping the spindle.

Using spindle machine parameter SPDLSTOP (P87), it is possible not to stop the spindle in this situation.

SPDLSTOP (P87)

Using this spindle machine parameter, it is possible not to stop the spindle with an M30 or a RESET.

Value	Meaning
0	The spindle stops with M2, M30 or RESET.
1	The spindle does not stop with M2, M30 or RESET. Program an M5 to stop the spindle.

Default value: 0

Special cases:

On a lathe, if s.m.p. SPDLSTOP(P87) =1 and there is an active G96 before getting a RESET or an M30, neither the G96 nor the PLC mark "CSS" will be deactivated and the spindle will keep turning at the same speed.

If s.m.p. SPDLSTOP(P87) =1 and a spindle turning speed limit has been programmed with G92 S**, a RESET or an M30 will not stop the spindle and will keep the turning speed limited to the value programmed with G92S.

In the following cases, the 0 or 1 value of s.m.p. SPDLSTOP (P87) will be ignored, the CNC will always act as follows:

- The spindle stops when an error occurs at the spindle or an emergency.
- The spindle does not stop when switching from manual to automatic and vice versa both in ISO mode and in conversational mode.
- If synchronized spindles are running, an M30 or a RESET will cancel the synchronism and stop the spindles.

9 Programming T and M6 with associated with a subroutine in the same line

Until this version, the M6 function could not be programmed in the same line as the T function when the M6 had a subroutine associated with it. From this version on, the T and M6 functions may be programmed in the same block. In these cases, the T function can also have an associated subroutine. Nothing else may be programmed in that block.

10 Variables

10.1 OPMODE Variable

The PLC program, OEM programs and dates may be protected by reading the new values of the OPMODE variable.

New values of the OPMODE variable:

63 = Change of protections.

130 = Change of dates.

131 = Change of passwords.

10.2 DISABMOD variable

The DISABMOD variable is used to disable some actions or modes by setting the corresponding bit to 1. This variable may be written from the PLC and read from the PLC, DNC and CNC.

Bit	Meaning
0	If 1, the PLC program cannot be displayed. The PLC in ladder (contacts) diagram cannot be displayed either.
1	If 1, the date cannot be changed, although it displays the access softkey. It is valid for the explorer and for "UTILITIES".
2	If 1, the passwords cannot be changed. The passwords cannot be seen or changed although it displays the access softkey. It is valid for the explorer and for "UTILITIES".

10.3 Internal CNC variables

GGSN:

It returns the status of functions G325 through G349. The status of each one of the functions will be given in the 25 least significant bits and it will be indicated by a 1 when active and a 0 when not active or when not available in the current software version.

G349	G348	G347	G346	G345	…	G329	G328	G327	G326	G325

GGSO:

It returns the status of functions GG350 through G374. The status of each one of the functions will be given in the 25 least significant bits and it will be indicated by a 1 when active and a 0 when not active or when not available in the current software version.

G374	G373	G372	G371	G370	…	G354	G353	G352	G351	G350

GGSP:

It returns the status of functions G375 through G399. The status of each one of the functions will be given in the 25 least significant bits and it will be indicated by a 1 when active and a 0 when not active or when not available in the current software version.

G399	G398	G397	G396	G395	…	G379	G378	G377	G376	G375

GGSQ:

It returns the status of functions G400 through G424. The status of each one of the functions will be given in the 25 least significant bits and it will be indicated by a 1 when active and a 0 when not active or when not available in the current software version.

G424	G423	G422	G421	G420	…	G404	G403	G402	G401	G400

10.4 CYCCHORDERR variable

The CYCCHORDERR variable lets define the chordal error of the canned cycles. This variable may be read and written from the part program.

The CYCCHORDERR variable lets modify the chordal error of the cycles so the user can increase or decrease it for the parts as needed.

Using this variable is necessary, for example, on parts with curved areas using the 3D pocket cycle. On these parts, if the radius is very large, the segments are noticeable. The parts get better by decreasing the chordal error.

Using this variable, the user can decrease the chordal error on the part as needed. Decreasing the chordal error can increase machining time.

Once the value of this variable has been changed, it remains active until the CNC is turned off.

Default value of the CYCCHORDERR variable (250 tenths of a micron).

Programming example:

```
(CYCCHORDERR = 25)
(PCALL 9986, P200=0)
M30
```

It is recommended to use a CYCCHORDERR value of 25 tenths of a micron. This value improves part finish and it does not increase machining time too much.

11 New option of the WRITE instruction

With this new option of the WRITE instruction, it will be possible to write the parameter number directly. For that, use the "$" character followed by "P" as long is it is preceded by an axis.

For example, programming (WRITE X$P100) the result will be: XP100.

To show something in $, the value must be programmed after the $ sign. But, to take the value from a parameter, put a space between the "$" sign and the parameter.

In short, these are the options:

- When programmed $P, it will output $P.
- When programmed $[space]P, it will output $[space] and the content of P.
- When programmed $[number], it will output $[number].

Example:

Being parameter P100=22.

Program	Result
(WRITE XP100)	X22
(WRITE X$P100)	XP100
(WRITE $ P100)	$ 22
(WRITE $3000)	$3000

12 Cancel additive handwheel offset with G04 K0

The CNC reads blocks ahead of the one being executed in order to calculate in advance the path to follow.

Function G04 interrupts block preparation and waits for the block in question to be executed in order to start the preparation of blocks once more.

The function associated with G04 K0 may be used to update the coordinates of the axes of the channel after waiting for all the blocks prior to the G04 K0 block are executed. Until now, the coordinates (position values) were initialized but the offset of the additive handwheel was not eliminated.

From this version on, if bit 10 of g.m p. ADIMPG (P176) =1, the instruction G04 K0 initialize the coordinates and eliminates the offset applied with the additive handwheel on all the axes that had an offset.

The coordinates will be initialized to the real coordinates of the machine and the offset will be deleted without moving any machine axis at all.

13 Ethernet parameter NFSPROTO (P32)

The TCP or UDP protocols may be used when configuring a remote hard disk using NFS.

The TCP protocol is more safer but in some network configurations copying and refreshing folders with lots of programs becomes very slow as compared to the UDP protocol.

Using the new Ethernet protocol NFSPROTO (P32) it is possible to select the desired protocol.

NFSPROTO (P32)

Using this Ethernet parameter, it is possible to select the desired protocol when configuring a remote hard disk using NFS.

Value	Meaning
0	The TCP protocol will be used.
1	The UDP protocol will be used.

Default value: 0

This parameter is validated with SHIFT/RESET or on power-up.

VERSION V02.03

1 Screen customizing instructions PAGE and SYMBOL

Customizing instructions PAGE and SYMBOL are used when customizing programs made by the user.

In order to be able to use the new formats (PNG and JPG/JPEG) in user programs created in versions older than V02.00, the operation of the SYMBOL and PAGE instructions has been changed.

The format of the PNG and JPG/JPEG files must be a 3 digit number. For example "001.jpg" for page 1. The size of the page must be 638x335.

(PAGE (expression))

The mnemonic PAGE displays the page number indicated by means of a number or by means of any expression resulting in a number.

From this version on, the JPG/JPEG formats are also supported. This way, if there is an "n.jpg", "n.jpeg" or "n.pan", the screen will display this file. When having several files, the order of priorities will be the following:

1. "n.jpg".
2. "n.jpeg".
3. "n.pan".

User-defined pages will be from page 0 to page 255 and will be defined from the CNC keyboard in the Graphic Editor mode and as indicated in the Operating Manual. System pages will be defined by a number greater than 1000.

(SYMBOL (expression 1), (expression 2), (expression 3))

The mnemonic SYMBOL displays the symbol whose number is indicated by means of the value of expression 1 once this has been evaluated.

Its position on screen is also defined by expression 2 (column) and by expression 3 (row).

Expression 1, expression 2 and expression 3 may contain a number or any expression resulting in a number.

From this version on, the PNG format is also supported. This way, if there is an "n.png" file, it will be displayed in the position indicated by the expressions 2 and 3. If it is missing, it will display the "n.sim" file.

The CNC allows displaying any user-defined symbol (0-255) defined at the CNC keyboard in the Graphic Editor mode such as is indicated in the Operating Manual.

In order to position it within the display area its pixels must be defined, 0-638 for columns (expression 2) and 0-335 for rows (expression 3).

2 Maximum spindle rpm up to 200,000 rpm

The idea is to increase the S programming range up to 200,000 rpm.

2.1 Modifications to several spindle parameters

The maximum values of several spindle parameters have been changed.

The new values are:

MAXGEAR1 (P2) MAXGEAR2 (P3) MAXGEAR3 (P4) MAXGEAR4 (P5)

They indicate the maximum spindle speed assigned to each gear.

On some machines, it is interesting to be able to use spindles with more than 99999.999 rpm. From this version, its value will be up to 200000.000 rpm.

When using an automatic gear change, these values will be used to make the change.

MAXGEAR1	for gear 1 (M41).
MAXGEAR2	for gear 2 (M42).
MAXGEAR3	for gear 3 (M43).
MAXGEAR4	for gear 4 (M44).

Possible values
Values with 3 decimals between 0.000 and 200000,000 rpm.

Default value: For MAXGEAR1 (P2) = 1000 rpm.
For MAXGEAR2 (P3) = 2000 rpm.
For MAXGEAR3 (P4) = 3000 rpm.
For MAXGEAR4 (P5) = 4000 rpm.

When not using all 4 gears, use the lower ones and set the unused ones to the same value as the highest one used.

SLIMIT (P66)

Maximum safety limit for the spindle speed. From this version, its value will be up to 200000.000 rpm. This limit is activated from the PLC and is applied in all the work modes, including the PLC channel. When the spindle is controlled by the PLC by means of the PLCCNTL mark, this limit is ignored.

Possible values
Values with 3 decimals between 0.000 and 200000,000 rpm.

Default value: 0

This limit is activated using the mark SLIMITAC (M5059). When this limit is canceled, the CNC recovers the programmed speed.

This limit permits clearing the spindle speed temporarily via PLC, e.g. when opening the doors, etc.

When upgrading from an older version to a V02.0x, the parameters indicated above will then have 3 decimals with a 0 value. The integer part will keep its previous value.
When downgrading from a version V02.0x to an older one, the parameters indicated earlier will lose their decimals but they will keep their integer values.

DFORMAT (P1)

Indicates the display format for the spindle. It is not used for the second spindle.

Value	Meaning
0	In 4 digits.
1	In 5 digits.
2	In 4.3 format
3	In 5.3 format
4	It is not displayed.
5	In 6 digits.

Default value: 0

FAGOR

CNC 8037

·M· MODEL
SOFT: V02.0X

VERSION V02.10

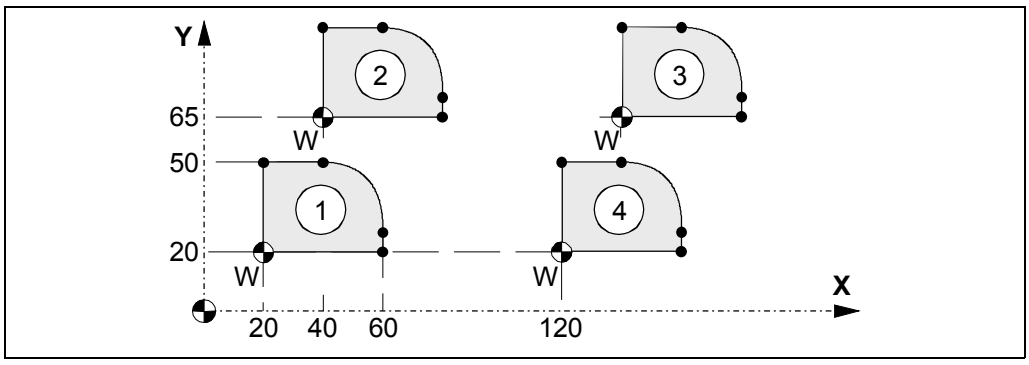
1 Incremental zero offset (G158)

The G158 instruction may be used to program and activate an incremental offset in a program. This feature is used to define new part zeros in the same program without having to set them previously in the offset table or use high level instructions.

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

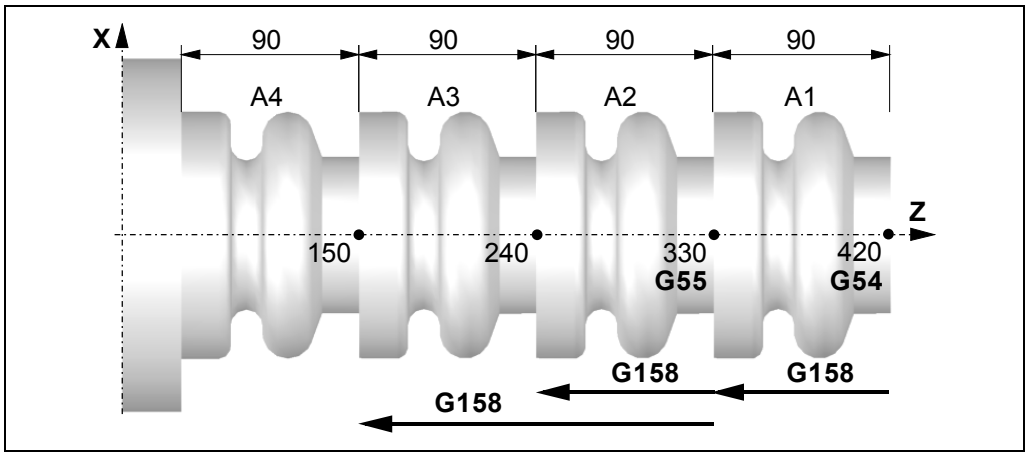
Programming

Incremental zero offset are defined by program using function G158 followed by the values of the zero offset to be applied on each axis. To cancel the incremental zero offset, program function G158 without axes in the block. To only cancel the incremental zero offset on particular axes, program an incremental offset of "0" on each of those axes.



	X	Y
G54 (G159N1)	20	20
G55 (G159N2)	120	20

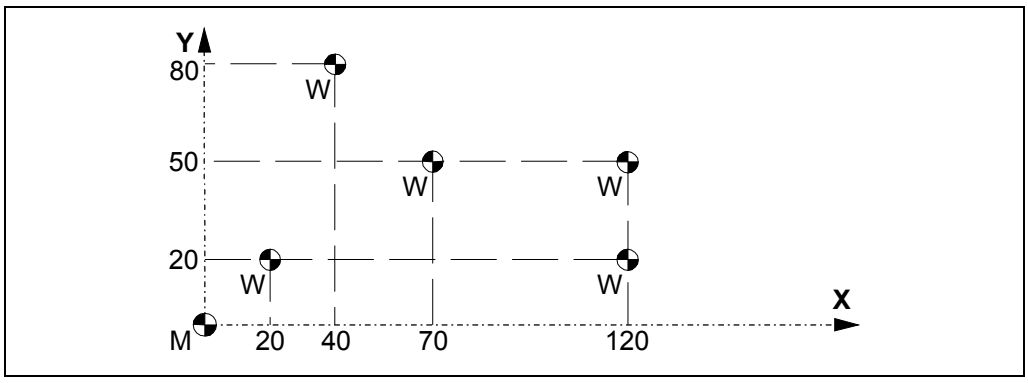
```
N100 G54                (It applies the first zero offset)
…                       (Machining of profile 1)
N200 G158 X20 Y45       (Apply incremental zero offset)
…                       (Machining of profile 2)
N300 G55                (It applies the second zero offset. G158 stays active)
…   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



	X	Z
G54 (G159N1)	0	420
G55 (G159N2)	0	330

```
N100 G54          (It applies the first absolute zero offset)
...               (Machining of profile A1)
N200 G158 Z-90    (Apply incremental zero offset)
...               (Machining of profile A2)
N300 G55          (It applies the second absolute zero offset)
                  (The incremental zero offset stays active)
...               (Machining of profile A3)
N200 G158 Z-180   (It applies the second incremental zero offset)
...               (Machining of profile A4)
```

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



	X	Y
G54 (G159N1)	20	20

```
N100 G54          (Apply absolute zero offset)
N200 G158 X20 Y60 (It applies the first incremental zero offset)
N300 G158 X50 Y30 (It applies the second incremental zero offset)
N400 G158 X100    (It applies the third incremental zero offset)
N500 G158 Y0      (It applies the fourth incremental zero offset)
N600 G158 X0      (Cancel incremental zero offset)
```

The incremental zero offset is not canceled after applying a new absolute zero offset (G54-G57 or G159Nx).

As described earlier, only one incremental zero offset may be active; therefore, instructions G58 and G59 are incompatible with G158. This way, the last incremental zero offset programmed cancels the incremental zero offset that is currently active.

Programming the G158 function alone in the block or G158 with a 0 value in the axes cancels the incremental zero offset G158 activated earlier. Those instructions also cancel the incremental zero offsets G58/G59 that are currently active.

Considerations

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

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

Properties of the function

G158 is modal and incompatible with G53.

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

Display in the zero offset table

In ISO mode and conversational mode, the zero offset table is one line over the the G54 position where it identifies the G158 with its values X, Y, Z, etc.

This line cannot be modified from the table, it can only be modified by programming G158.

Variables related to the incremental zero offset G158

Variable ADDORG (X-C):

The ADDORG variable returns the value of the active incremental zero offset corresponding to the axis selected at the time.

It is read-only variable that can be read from the CNC, PLC and DNC.

2 Programs with letters

From this version on, letters and numbers may be used to identify the part-programs of the CNC.

Using this feature requires the new version of WinDNC (V6.01).

Functional requirements

Programs may be identified with up to 24 characters that may be letters and numbers. Blank spaces can also be used.

It is NOT case sensitive. Identifying a program as "PROGRAM1", will be the same as identifying it as "program1".

A program name cannot have only blank spaces. The "ñ" letter is not allowed.

The allowed special characters are:

!	#	$	%	&	(	)	+	-
;	=	@	[	]	_	{	|	}

The following programs will still be identified with 6 digits:

- The special programs like OEM programs that are created by Fagor (999997, 999998, etc).
- Programs referred to in the general machine parameters (USERPDLY, USEREDIT, USERMAN, USERDIAG, CFGFILE, MSGFILE, STPFILE, ISOSIMUL).
- The programs where the black box is saved as well as the SERCOS and CAN logs.
- The program that keeps the log of the communication open via serial line and the CAN servo log.
- The programs for the cycles of the MC/TC mode. In this case, each cycle will still correspond to a program number.
- The profile programs that are used in the cycles.

Compatibilities

When having programs with letters and installing a version older than V02.1x, these programs will not appear on the list, but they will still be in memory. When installing a version V02.1x, the programs with letters will appear on the list.

We recommend doing a backup of the program that are in user memory before installing version V02.1x.

There are two kinds of programs when it comes to keep compatibility with older versions:

1. Programs that have 6 digits and are equal to or smaller than 999999 and plc-error programs, plc-message programs and plc programs.
2. Programs that have a number with more than 6 digits or programs with letters.

This way, users who do not wish to use the new way to identify programs with letters can keep using just the 6 digits, like up to this version.

Programs identified with numbers must have 6 digits.

Example: Program "17.pim" would be wrong.

Program "000017.pim" would be right.

Program search on the explorer

To search for a program, press F before writing the name of the program. This is valid for numbers and letters.

The information bar of the explorer located down to the left, will keep update the search string as it is typed in.

Example: To search for the program "EXAMPLE" on the program list, type "F", "E", "X", "E", etc.

Execution of high level instruction EXEC

A program may be executed using the number or the name of the program.

Program execution using the number:

When using the number, it will be done as in previous versions.

Example:

(EXEC P1234, HD)

Program execution using the name of the program:

When using the name of the program, the program needs NOT be in the same PRG folder of the hard disk. In this case, the path must be specified. Also, the name of the program (including the path) must be between quotation marks.

If the path is not specified, it will be HD:/PRG by default.
The high level instruction RPT cannot be used with programs identified with letters.

The following characters may be used to indicate the path:

- ./ It indicates that the program that contains the EXEC is in the same folder.
- ../ It indicates that the program that contains the EXEC is in a folder at a higher level.
- ../../ Writing ../ several times means that it is that number of levels up. In this case, it would be two levels up.

Example:

Considering this folder structure, the high-level instruction EXEC will be used as follows:

- If the folder containing the EXEC is in Memory, but the program to be executed is in HD, the full path must be indicated.
  - To execute the program Soporte#1.pim:

    (EXEC "CHASIS_05J1/Soporte#1",HD)
  - To execute the program Soporte#2.pim:

    (EXEC "CHASIS_05J1/SUBCHASIS_05J1/Soporte#2",HD)

- If the folder containing the EXEC is in HD:/PRG, and program Soporte#1 is to be executed, (the following examples show three ways to indicate the path):

  (EXEC "CHASIS_05J1/Soporte#1,HD)
  (EXEC "/CHASIS_05J1/Soporte#1,HD)
  (EXEC "\CHASIS_05J1\Soporte#1,HD)

- If the program that contains the EXEC is in the folder CHASIS_05J1 and program Soporte#1 is to be executed:

  (EXEC "./Soporte#1",HD)

- If the program that contains the EXEC is in the folder CHASIS_05J1 and program Soporte#2 located in the folder SUBCHASIS_05J1 is to be executed:

  (EXEC "./SUBCHASIS_05J1/Soporte#2",HD)

- If the program that contains the EXEC is in the folder SUBCHASIS_05J1 and program Soporte#1 located in the folder CHASIS_05J1 is to be executed:

  (EXEC "../Soporte#1",HD)

Displaying the program list

The list of programs may be displayed as follows:

1. The whole program name.

   In the MC/TC mode, up to 16 characters will be displayed.

2. The first six characters of the program name with the comment. This is the way by default.

To toggle from one display method to the other, press "OPTIONS / SHOW COMMENT" on the explorer.

On power-up, the CNC will remember the display method chosen last.

Variables

PRGN:

Returns the program number being executed. If none is selected, a value of -1 is returned.

With the programs with letters, it returns a number calculated by the CNC for each program.

EXECLEV:

New variable that indicates the EXEC level that is in execution at the time.

- If the value is 0, it indicates that it is not inside an EXEC.
- If the value is 1, it indicates that it is inside an EXEC.
- If the value is 2, it indicates that it is inside two calls.

This variable is used on machines with automatic shutdown with M30. This way, it distinguishes whether it is the last M30 or it is the M30 of an EXEC and the execution continues.

3 Korean language

From this version on, the CNC may be set in Korean. For that, set g.m.p. LANGUAGE (P122) to 13.

LANGUAGE (P122)

Defines the work language.

Value	Meaning	Value	Meaning
0	English	7	Czech
1	Spanish	8	Polish
2	French	9	Mainland Chinese
3	Italian	10	Basque
4	German	11	Russian
5	Dutch	12	Turkish
6	Portuguese	13	Korean

Default value: 0

OEM texts in Korean.

The machine manufacturer can set the CNC in Korean and include his own messages. For that, just follow the instructions of the installation manual.

4 Variable for reading the active zero offset

The new EXTORG variable may be used to read the active absolute zero offset. This read-only variable interrupts block preparation and may be read from the CNC, from the PLC and from DNC.

The values returned by the variable are identical for both possible expressions of absolute zero offsets.

The values of the EXTORG variables that correspond to the absolute zero offsets are:

Value of EXTORG	Active zero offset
0	G53 (there is no zero offset)
1	G54 or G159N1
2	G55 or G159N2
3	G56 or G159N3
4	G57 or G159N4
5	G159N5
6	G159N6
7	G159N7
8	G159N8
9	G159N9
10	G159N10
11	G159N11
12	G159N12
13	G159N13
14	G159N14
15	G159N15
16	G159N16
17	G159N17
18	G159N18
19	G159N19
20	G159N20

Considerations:

- When programming only an incremental zero offset (G58 or G59), the value of the EXTORG variable will be 0.
- When programming an absolute and an incremental zero offset, the EXTORG variable will keep the value of the absolute zero offset.

  Example: If G54 + G58, EXTORG = 1 has been programmed.

5 Image handling via DNC

From this version on, it is possible to send and receive PNG, JPG/JPEG and BMP type images via DNC.

Using this feature requires the new version of WinDNC (V6.01).

WinDNC software

Version V6.01 of WinDNC supports files with extension bmp, png, jpg and jpeg. The maximum length or the file name is 16 characters (including the dot and the extension).

The application scans all image type files in the work folder. When sending the files, if the name of any file is too long, it will ask the user to enter a new shorter name (up to 16 characters). It must also keep the original extension.

6 Saving/restoring an oscilloscope trace.

This feature may be used to analyze the data of a trace of the oscilloscope captured earlier, be it from the same CNC or from another one.

Use the new ANALYSIS softkey to access the softkeys SAVE TRACE and RESTORE TRACE. The ANALYSIS softkey will only be displayed there is no capture in progress.

SAVE TRACE:	It saves the data of a trace in the file "oscillo_trace". Saves the configuration and the data for each channel and variable.
RESTORE TRACE:	It restores the data of a trace that was saved earlier in the file "oscillo_trace". It restores the configuration and the data for each channel and variable.

The file "oscillo_trace" may be sent or received via WinDNC like a regular part-program.

FAGOR

CNC 8037

·M· MODEL
SOFT: V02.1X

VERSION V02.21

1 PLC library

This feature may be used to create a PLC library with subroutines. Once the library has been created, any OEM can use it to run a specific control of the machine. For that, the OEM's PLC program will call the subroutines that are contained in the library.

It will not be possible to read the instructions of the PLC library, only call its functions to use them.

When compiling the PLC program, it will take into account whether there is a library or not. If there is a PLC library, the PLC program and the library will be compiled together. The library must be located in the "PRG" folder of the hard disk.

Operation

The PLC menu offers the [PLC LIBRARY] option. When pressing this softkey, it requests the PLC password (if there is one).

After entering the right password, it displays the following softkeys:

Softkey [MAKE PLC_LIB]:

Press this softkey to create a library (plc_lib file) from the existing PLC program. The library will be created in the "PRG" folder of the hard disk.

The PLC must be compiled before creating the library. The library will be created if there are no compiling errors.

When creating a library, the comment that the PLC program has at the time will be copied to the library. Once the library has been generated, it will not be possible to change the comment. A version number should be given to identify the library.

Softkey [DELETE PLC_LIB]:

Press this softkey to delete a library. A library cannot be erased from the utilities menu, only from the PLC menu using this softkey.

Considerations

The symbols defined in the library may be used in the following cases:

- On the screen [MONITORING / OSCILLOSCOPE].
- When editing the PLC program.

The resources used in the library will appear as used in the PLC statistics.

The labels used in the library cannot be repeated in the PLC program. Likewise, the PRG and PE instructions cannot be defined in the library.

How to create a PLC Library

The following example shows how to create a PLC library:

1. Edit a PLC program.

   Use only subroutines and symbols in the PLC program.

2. Modify the comment of the PLC program.

   This can be used to identify the PLC library.

   Example: Punching library V1.1

   This is a good way to identify the library versions because the comment of a library cannot be changed.

   The PLC program comment is copied to the library comment when MAKE PLC_LIB is selected.

3. Go to the PLC library, following this keystroke sequence:

   <MAIN MENU> <F7=+> <F2=PLC> <F7=+> <F4=PLC LIBRARY>

   If there is a PLC password, it will be requested. Enter the password.

4. Create a PLC library just pressing <F1=MAKE PLC_LIB>.

   It will generate a PLC library automatically from the current PLC program.

   The library file is either "plc_lib.pim" or "plc_lib.pit".

   If the PLC program has PRG, or CY1, or PE modules, the PLC library cannot be generated.

5. Make a copy of the PLC program, in order to have a backup for future changes.

6. Delete the PLC program used to generate the PLC library.

7. Once this is done, the PLC library can be copied to any CNC. This file cannot be read or written.

How to use the PLC library

The following example shows how to use a PLC library:

1. Copy the PLC library into the CNC you want to use.

   The file must be copied into following directory of the CNC's hard disk:

   HARD DISK / PRG

2. Create a PLC program calling the subroutines edited in the PLC library.

3. Resources used in the PLC library can be used in a PLC program like all DEFINES. Be careful when using the PLC library resources. Hand this info to the OEM when using these resources.

plc_lib source example

```
DEF OUT_SECOND O50
DEF MARK_SECOND M2017
DEF RCOUNTER R300
DEF INCREMENT_PAR R301
;
L1500
MARK_SECOND = OUT_SECOND = M1500
END
;
;
L1501
INCREMENT_PAR = ADS RCOUNTER 1 RCOUNTER
END
```

Example of a PLC program that uses PLC_LIB

```
PRG
() = CAL L1500
;
; PLC library symbols can be used
CPS RCOUNTER GT 1000 = MSG14
END
PE 4
MARK_SECOND = MOV 1 INCREMENT_PAR
 = MOV 0 INCREMENT_PAR
() = CAL L1501
END
```

PLC_LIB documentation example

This document must be provided so the PLC library can be used correctly.

Library version 1.1

Library Symbols :

DEF OUT_SECOND O50
DEF MARK_SECOND M2017
DEF RCOUNTER R300
DEF INCREMENT_PAR R301

Library subroutines :

L1500
Description: Changes the value of OUT_SECOND every second
Input parameters: None
Output parameters: None
Resources modified: OUT_SECOND, M1500

L1501
Description: Increments counter RCOUNTER depending on the value of INCREMENT_PAR.
Input parameters: INCREMENT_PAR 1 value increments, 0 doesn't.
Output parameters: RCOUNTER
Resources modified: RCOUNTER

2 Zero offsets table in ISO mode

The same zero offset table is available in ISO mode and in conversational mode. The access table may be accessed in ISO mode from the jog mode.

It is accessed using the following softkeys:

JOG / DISPLAY / ORIGINS

The table works the same way in ISO mode and in conversational mode.

The zero offset table looks like this. It shows all the offsets, PLC offset included, and their value in each axis.

	X	Y	Z	C
PLC	0.0000	0.0000	0.0000	0.0000
ADI	0.0000	0.0000	0.0000	0.0000
G158	0.0000	0.0000	0.0000	0.0000
G54=G159N1	0.0000	0.0000	0.0000	0.0000
G55=G159N2	0.0000	0.0000	0.0000	0.0000
G56=G159N3	0.0000	0.0000	0.0000	0.0000
G57=G159N4	0.0000	0.0000	0.0000	0.0000
ΔG58(G159N5)	0.0000	0.0000	0.0000	0.0000
ΔG59(G159N6)	0.0000	0.0000	0.0000	0.0000
G159N7	0.0000	0.0000	0.0000	0.0000
G159N8	0.0000	0.0000	0.0000	0.0000
G159N9	0.0000	0.0000	0.0000	0.0000
G159N10	0.0000	0.0000	0.0000	0.0000

When scrolling the focus through the table, the elements appear in different colors as follows.

Color	Meaning
Green background. Text in white.	The real value of the table and the value shown on the screen are the same.
Red background. White text.	The real value of the table and the value shown on the screen are NOT the same. The value on the table has been changed, but it has not been validated. Press [ENTER] to validate the change.
Blue background.	The zero offset is active. There may be two active zero offsets at the same time, one absolute (G54 ... G57, G159N7 ... G159N20) and another one incremental (G58G59).

How to edit the table data

The following operations are possible in the zero offset table. Press [ENTER] to validate any changes.

- Editing a zero offset.

  It is edited one axis at a time. Select a data with the focus and edit its value. If the focus is placed on a zero offset (G54 ... G159N20), the editing begins on the first axis of that zero offset.

- Load the active zero offset into the table.

  Place the focus on the zero offset to be defined (G54 ... G59, G159N7 ... G159N20) and press the [RECALL] key. The active preset is saved in the selected zero offset.

  If instead of placing the focus on a zero offset, it is placed on one of the axes, only that axis will be affected.

- Deleting a zero offset.

  Place the focus on the zero offset to be deleted (G54 ... G59, G159N7 ... G159N20) and press the [CLEAR] key. All the axes of that zero offset are reset to zero.

  If instead of placing the focus on a zero offset, it is placed on one of the axes, only that axis will be affected.

3 Compensation of the elastic deformation in the coupling of an axis

This feature should be applied on machines whose dynamics causes significant elastic deformation on the transmission system (coupling) of each axis generating unacceptable deviations on any path followed by the tool tip in machining processes, cutting processes etc. that cannot be compensated by the control loops because they are out of the measuring system.

With laser cutting machines, a.m.p. DYNDEFRQ (P103) may be used to compensate the deformation of the arm that supports the laser when it is accelerating or decelerating.

Prior considerations

This feature may be applied to any machine that only uses motor feedback regardless of the type of path the tool tip will follow.

The dynamic behavior of the machine should be analyzed when the tool tip follows a circular path (since it is an easy geometric shape for measuring path deviations) in order to obtain the value of the elastic deformation compensation on each axis of the machine.

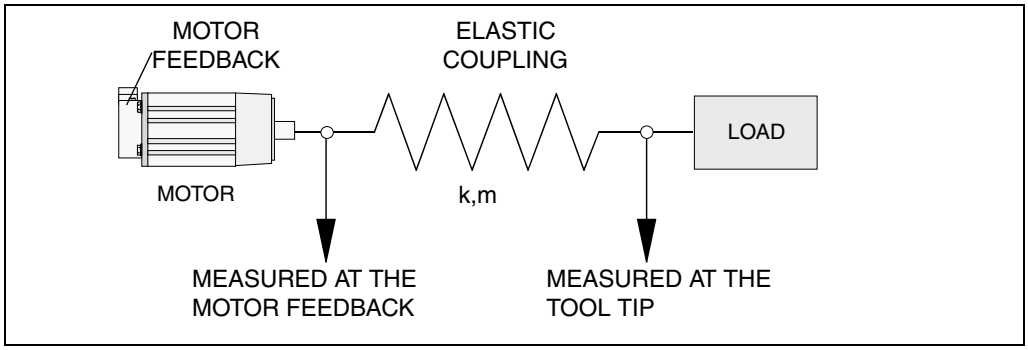
That's why most of the mathematical expressions appearing next are only to be applied to circular paths.

If the user wishes to analyze the elastic deformations on his machine by running some tests with other more complex tool paths, he will not be able to apply some of the expressions shown here.

The numerical data appearing in their examples are merely illustrative. Do not copy this data to run your tests. Remember that each machining operation that the machine carries out requires very specific cutting and working conditions that rarely coincide with the data shown in the examples.

Elastic deformation in the coupling of an axis

Let us suppose a system consisting of a servomotor with a position feedback, an elastic coupling and the tool tip..



When the system starts moving, if the coupling would ideally not be deformed, the position of the tool tip would strictly follow the path that the part program of the CNC commanded and it would be the same as the position given by the feedback device integrated into the motor.

However, assuming that the coupling cannot be deformed is not realistic Therefore, while moving, the coupling suffers an elastic deformation more or less significant depending on its acceleration; i.e. of the relative feedrate between the tool tip and the table that affects, to a greater or lesser degree. the path to follow.

Then, the path demanded by the part-program of the CNC is not truthfully followed by the tool tip, there is a deviation due to the elastic deformation of the axis coupling.

This path deviation is not measured by the motor position feedback device because it is located just before the elastic coupling and, consequently, it is not aware of this deformation. This is why there is no deviation to compensate for the control loop of the CNC.

In this scenario where this deviation is not measured and, even if it is, it cannot be compensated for by increasing the proportional gain of the system in the control loops, this deviation must be compensated using the axis parameter DYNDEFRQ (P103), setting it with the value of the

resonance frequency (in Hz) associated with the elastic coupling. The procedure to obtain this frequency will be described later on.

DYNDEFRQ (P103)

A.m.p. DYNDEFRQ (P103) compensates the deformation of the arm that supports the tool when it is accelerating or decelerating. This parameter is applied in the position loop.

Possible values
Between 0.01 and 9999.99 Hz.

Default value: 0 Hz.

The value of this parameter is updated with a reset.

A low value of a.m.p. DYNDEFRQ (P103) can cause too much movement of the axis issuing the following error message: Too much following error (axis lag).
When having gantry axes, the master axis and the slave axis must be set with the same value.

Dynamic factors that affect elastic deformation.

The deformation suffered by an elastic mechanical system subject to a force is given by the formula:

$$\text{Fuerza} = k \cdot \Delta x$$

where:

- k elastic constant
- Dx amount of deformation

Also knowing that:

$$\text{Fuerza} = m \cdot a$$

where:

- m Mass of all moving elements
- a system acceleration

and replacing its value in the previous formula we get the equation:

$$m \cdot a = k \cdot \Delta x$$

Deformation is, therefore, proportional to acceleration:

$$\Delta x = \frac{m}{k} \cdot a = Cte \cdot a$$

For a circular path, recommended to be used in the machining tests for adjusting the compensation of elastic deformation because it is a comfortable geometrical shape for taking measurements, normal acceleration is given by the formula:

Only applicable to circular paths	$a = \frac{[F/60]^2}{R}$

where:

R Radius of the circular path followed by the tool tip. Enter its value in meters (m).

F Feedrate of the tool tip as long as the table is not moving. If the tool and the table are moving, F will be the relative feedrate between them.. Enter its value in meters per minute (m/min),

Conclusions:

The elastic deformation on the axis of a machine is directly proportional to the acceleration and when its dynamics is a circular path, it is also directly proportional to the square of the feedrate.

Therefore, the higher the relative feedrate between the tool tip and the table, the more significant the elastic deformation will be.

Machining a part following a circular path at low feedrate considering the elastic deformation only on one axis or on both implies practically getting the desired circular path because the path deviation suffered on the axes will be very small. It will not be a perfect circle, but, in practice, it will not be necessary to compensate for elastic deformation.

Machining a part following a circular path at high feedrate considering the elastic deformation only on one axis or on both implies getting an elliptic path because the path deviation suffered on the axes will be considerable. The other main radius of the ellipse will be the same as the radius of the circular path when assuming that there was no elastic deformation on the other axis. Considering elastic deformation on both axes will also result in an elliptic path where none of the two main radii coincides with the radius of the programmed circular path.

To compensate these deformations, set a.m.p. DYNDEFRQ (P103) for each axis with the value of the resonance frequency of its elastic coupling.

Remarks:

Before compensating the elastic deformation, the CNC will always display a circular path, not elliptical as could be expected because there is elastic deformation.. The motor feedback is placed before the elastic coupling and does not register the path deviation due to elastic deformation at all.

After compensating the elastic deformation, the CNC will always display an elliptical path, not circular as could be expected after compensating the elastic deformation..

Resonance frequency of the elastic coupling

From the formula of the oscillation frequency of a simple harmonic movement (s.h.m.) it is possible to obtain its relationship with acceleration and deformation through this equation:

$$f = \frac{1}{2\pi} \cdot \sqrt{\frac{k}{m}} = \frac{1}{2\pi} \cdot \sqrt{\frac{a}{\Delta x}}$$

For a circular path, the Δx deviation with respect to R may be obtained directly by measuring the part that has been previously machined.

The acceleration of the system is calculated from the formula shown earlier:

$$a = \frac{[F/60]^2}{R}$$

With these values, now determined, the formula of the frequency provides the value for setting a.m.p. DYNDEFRQ (P103) to compensate the elastic deformation caused in that axis.

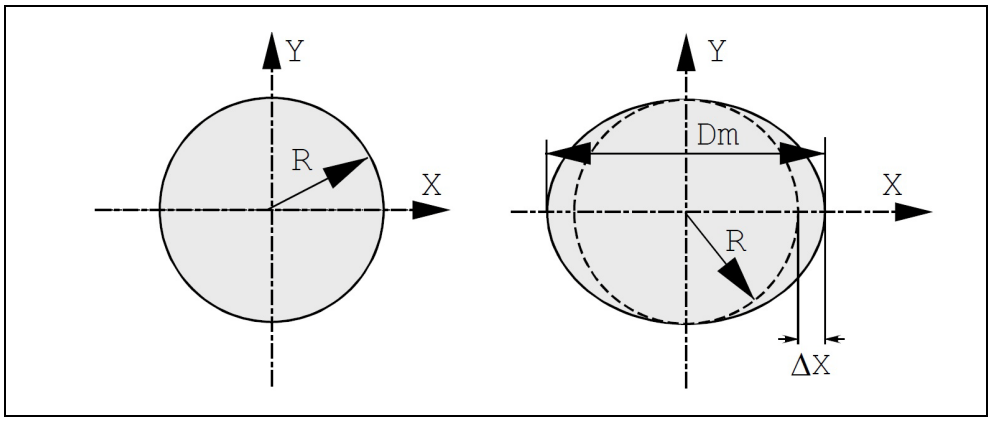
Example

To adjust the compensation of elastic deformation caused in the X axis of a laser cutting machine. The machine has two servomotors with position feedback that move the tool tip in a plane defined by the X and Y axes. Supposing that only the X axis has an elastic coupling.

Its purpose is to drill circular holes with a radius R = 5 mm = 0.005 m at high speed on a sheet metal that rests on fixed frame.

To obtain the frequency value for setting a.m.p. DYNDEFRQ (P103) to compensate the elastic compensation on the X axis, proceed as follows.

1. Make a hole keeping the tool tip at high feedrate, for example, F = 8000 mm/min in order to cause high deviation Δx and generate a clear elliptical path.
2. Once the hole has been made, use a caliper to measure the various diameters of the elliptical hole until obtaining the diameter of the larger axis Dm of the ellipse. Observe that the elliptical shape will not be visually noticeable.



The real shape is elliptical, not circular, when only the X axis suffers elastic deformation.

3. Obtain the value of the elastic deformation from the formula:

$$\Delta x = \frac{D_m}{2} - R$$

Let us suppose (for the sake of using numbers) that the deformation value is: Δx = 90 µm = 90 x 10^{-6} m.

4. Obtain the acceleration value for a circular path from the formula:

$$a = \frac{[F/60]^2}{R} = \frac{[8/60]^2}{0,005} = 3,6 m/s^2$$

5. Obtain the frequency value from the formula:

$$f = \frac{1}{2\pi} \cdot \sqrt{\frac{a}{\Delta x}} = \frac{1}{2\pi} \cdot \sqrt{\frac{3,6}{90 \times 10^{-6}}} = 31,8 Hz$$

6. Set a.m.p. DYNDEFRQ (P103) with the obtained value, i.e. DYNDEFRQ (P103) = 31.8 Hz.
7. Verify that after setting a.m.p DYNDEFRQ (P103), your particular part is machined properly regardless of the path of its profile.

4 Change of maximum value of axis and spindle parameter NPULSES

The maximum value of axis machine parameter NPULSES (P8)and spindle machine parameter NPULSES (P13) has been changed. From this version, the maximum value will be 99999.

Axis machine parameter NPULSES (P8)

Indicates the number or pulses/rev provided by the rotary encoder. When using a linear encoder, it must be set to ·0·.

It must be set when the drive's velocity command is analog; it is sent via Sercos (DRIBUSLE = 0) or via CAN (DRIBUSLE = 0 or 1).

When using gear reduction on the shaft, only the whole assembly must be taken into account when setting one of parameters PITCH or NPULSES.

Possible values
Integer numbers between 0 and 99999.

Default value: 1250

When using CAN servo, if both parameters NPULSES and PITCHB are set to ·0·, the CNC will assume the equivalent values of the drive.

Spindle machine parameter NPULSES (P13)

Indicates the number of pulses per revolution provided by the spindle encoder. 0 means that there is no spindle encoder.

It must be set when the drive's velocity command is analog; it is sent via Sercos (DRIBUSLE = 0) or via CAN (DRIBUSLE = 0 or 1).

When the main spindle does not have an encoder (NPULSES=0), the CNC shows its theoretical rpm (affected by the %).

Possible values
Integer numbers between 0 and 99999.

Default value: 1000

When using a CAN servo system, if parameter NPULSES and parameters INPREV and OUTPREV of all the gears are set with a ·0· value, the CNC will assume the equivalent ones of the drive.

FAGOR

CNC 8037

·M· MODEL
SOFT: V02.2X