

# 14. SENTENCIAS DE CONTROL DE LOS PROGRAMAS

Las sentencias de control que dispone la programación en lenguaje de alto nivel, se pueden agrupar como:

\* **Sentencias de Programación**, formadas por:

- Sentencias de asignación
- Sentencias de visualización
- Sentencias de habilitación-deshabilitación
- Sentencias de control de flujo
- Sentencias de subrutinas
- Sentencias para generar programas

\* **Sentencias de personalización**

En cada bloque se programará una única sentencia, no permitiéndose programar ninguna otra información adicional en dicho bloque.

## 14.1 SENTENCIAS DE ASIGNACIÓN

Es el tipo de sentencia más simple y se puede definir como:

**( destino = expresión aritmética )**

Como destino puede seleccionarse un parámetro local o global o bien una variable de lectura y escritura. La expresión aritmética puede ser tan compleja como se desee o una simple constante numérica.

(P102 = FZLOY)  
(ORGY 55 = (ORGY 54 + P100))

En el caso particular de realizarse una asignación a parámetro local utilizando su nombre (A en vez de P0 por ejemplo) y siendo la expresión aritmética una constante numérica, la sentencia se puede abreviar de la siguiente forma:

(P0=13.7) ==> (A=13.7) ==> (A13.7)

En un único bloque se pueden realizar hasta 26 asignaciones a destinos distintos, interpretándose como una única asignación el conjunto de asignaciones realizadas a un mismo destino.

(P1=P1+P2, P1=P1+P3, P1=P1\*P4, P1=P1/P5) es lo mismo que (P1=(P1+P2+P3)\*P4/P5).

Las diferentes asignaciones que se realicen en un mismo bloque se separarán con comas “,”.

Capítulo: 14 SENTENCIAS DE CONTROL DE LOS PROGRAMAS	Sección: SENTENCIAS DE ASIGNACION	Página 1
--	--------------------------------------	-------------

## 14.2 SENTENCIAS DE VISUALIZACIÓN

### ( ERROR nº entero, “texto de error” )

Esta sentencia detiene la ejecución del programa y visualiza el error indicado, pudiéndose seleccionar dicho error de los siguientes modos:

(ERROR nº entero). Visualizará el número de error indicado y el texto asociado a dicho número según el código de errores del CNC (si existe).

(ERROR nº entero, “texto de error”). Visualizará el número y el texto de error indicados, debiéndose escribir el texto entre comillas.

(ERROR “texto de error”). Visualizará únicamente el texto de error indicado.

El número de error puede ser definido mediante una constante numérica o mediante un parámetro. Cuando se utiliza un parámetro local debe utilizarse su forma numérica (P0-P25).

Ejemplos de programación:      (ERROR 5)  
   (ERROR P100)  
   (ERROR “Error de usuario”)  
   (ERROR 3, “Error de usuario”)  
   (ERROR P120, “Error de usuario”)

### ( MSG “mensaje” )

Esta sentencia visualiza el mensaje indicado entre comillas.

En la pantalla del CNC se dispone de una zona para visualización de mensajes de DNC o de programa de usuario, visualizándose siempre el último mensaje recibido, independientemente de su procedencia.

Ejemplo:      (MSG “Comprobar herramienta”)

### (DGWZ expresión 1, expresión 2, expresión 3, expresión 4, expresión 5, expresión 6)

La sentencia DGWZ (Define Graphic Work Zone) permite definir la zona de representación gráfica.

Cada una de las expresiones que componen la sintaxis de la instrucción corresponden a uno de los límites y se deben definir en milímetros o pulgadas.

expresión 1	X mínimo
expresión 2	X máximo
expresión 3	Y mínimo
expresión 4	Y máximo
expresión 5	Z mínimo
expresión 6	Z máximo

### 14.3 SENTENCIAS DE HABILITACION-DESHABILITACION

#### ( ESBLK y DSBLK )

A partir de la ejecución de la sentencia **ESBLK**, el CNC ejecuta todos los bloques que vienen a continuación como si se tratara de un único bloque.

Este tratamiento de bloque único, se mantiene activo hasta que se anule mediante la ejecución de la sentencia **DSBLK**.

De esta manera, si se ejecuta el programa en el modo de operación BLOQUE a BLOQUE, el grupo de bloques que se encuentran entre las sentencias ESBLK y DSBLK se ejecutarán en ciclo continuo, es decir, no se detendrá la ejecución al finalizar un bloque sino que continuará con la ejecución del siguiente.

Ejemplo:

```
G01 X10 Y10 F8000 T1 D1
(ESBLK)                                ; Comienzo de bloque único
G02 X20 Y20 I20 J-10
G01 X40 Y20
G01 X40 Y40 F10000
G01 X20 Y40 F8000
(DSBLK)                                ; Anulación de bloque único
G01 X10 Y10
M30
```

#### ( ESTOP y DSTOP )

A partir de la ejecución de la sentencia **DSTOP**, el CNC inhabilita la tecla de Stop, así como la señal de Stop proveniente del PLC.

Esta inhabilitación permanecerá activa hasta que vuelva a ser habilitada mediante la sentencia **ESTOP**.

#### ( EFHOLD y DFHOLD )

A partir de la ejecución de la sentencia **DFHOLD**, el CNC inhabilita la entrada de Feed-Hold proveniente del PLC.

Esta inhabilitación permanecerá activa hasta que vuelva a ser habilitada mediante la sentencia **EFHOLD**.

## 14.4 SENTENCIAS DE CONTROL DE FLUJO

Las sentencias **GOTO** y **RPT** no pueden ser utilizadas en programas que se ejecutan desde un PC conectado a través de una de las líneas serie.

### ( **GOTO N(expresión)** )

La sentencia **GOTO** provoca un salto dentro del mismo programa, al bloque definido mediante la etiqueta **N(expresión)**.

La ejecución del programa continuará tras el salto, a partir del bloque indicado.

La etiqueta de salto puede ser direccionada mediante un número o mediante cualquier expresión que tenga como resultado un número.

Ejemplo:

```
G00 X0 Y0 Z0 T2 D4
      X10
(GOTO N22)           ; Sentencia de salto
      X15 Y20          ; No se ejecuta
      Y22 Z50          ; No se ejecuta
N22  G01 X30 Y40 Z40 F10000 ; La ejecución continúa en este bloque.
      G02 X20 Y40 I-5 J-5
      .....
      .....
```

### ( **RPT N(expresión)** , **N(expresión)** )

La sentencia **RPT** ejecuta dentro del mismo programa, la parte de programa existente entre los dos bloques definidos mediante las etiquetas **N(expresión)**.

Ambas etiquetas podrán ser indicadas mediante un número o mediante cualquier expresión que tenga como resultado un número.

La parte de programa seleccionado mediante las dos etiquetas debe pertenecer al mismo programa, definiéndose primero el bloque inicial y luego el bloque final.

La ejecución del programa continuará en el bloque siguiente al que se programó la sentencia **RPT**, una vez ejecutada la parte de programa seleccionada.

Ejemplo:

```
N10 G00 X10
      Z20
      G01 X5
      G00 Z0
N20      X0
N30 (RPT N10, N20) N3
N40 G01 X20
      M30
```

Al llegar al bloque N30, el programa ejecutará 3 veces la sección N10-N20. Una vez finalizada, continuará la ejecución en el bloque N40.

**( IF condición <acción1> ELSE <acción2> )**

Esta sentencia analiza la **condición** dada, que deberá ser una expresión relacional. Si la condición es cierta (resultado igual a 1), se ejecutará la **<acción1>**, y en caso contrario (resultado igual a 0) se ejecutará la **<acción2>**.

Ejemplo:

(IF (P8 EQ 12.8) CALL 3 ELSE PCALL 5, A2, B5, D8)

Si P8 =12.8 ejecuta la sentencia (CALL3)

Si P8 <>12.8 ejecuta la sentencia (PCALL 5, A2, B5, D8)

La sentencia puede carecer de la parte ELSE, es decir, bastará con programar IF condición <acción1>.

Ejemplo:

(IF (P8 EQ 12.8) CALL 3)

Tanto <acción1> como <acción2> podrán ser expresiones o sentencias, a excepción de las sentencias IF y SUB.

Debido a que en un bloque de alto nivel los parámetros locales pueden ser denominados mediante letras, se pueden obtener expresiones de este tipo:

(IF (E EQ 10) M10)

Si se cumple la condición de que el parámetro P5 (E) tenga el valor 10, no se ejecutará la función auxiliar M10, ya que un bloque de alto nivel no puede disponer de comandos en código ISO. En éste caso M10 representa la asignación del valor 10 al parámetro P12, es decir, que es lo mismo programar:

(IF (E EQ 10) M10)                      o                      (IF (P5 EQ 10) P12=10)

## 14.5 SENTENCIAS DE SUBROUTINAS

Se llama subrutina a una parte de programa que, convenientemente identificada, puede ser llamada desde cualquier posición de un programa para su ejecución.

Una subrutina puede estar almacenada como un programa independiente o como parte de un programa, y puede ser llamada una o varias veces, desde diferentes posiciones de un programa o desde diferentes programas

Únicamente se pueden ejecutar subrutinas existentes en la memoria RAM del CNC. Por ello, si se desea ejecutar una subrutina almacenada en la "Memkey Card", HD o en un PC conectado a través de una de las líneas serie, copiarla a la memoria RAM del CNC.

Si la subrutina es demasiado grande para pasarla a memoria RAM, convertir la subrutina en programa y utilizar la sentencia EXEC que se detalla en el apartado 14.6

### ( SUB nº entero )

La sentencia **SUB** define como subrutina el conjunto de bloques de programa que se encuentran programados a continuación de éste bloque, identificando dicha subrutina mediante el número entero, comprendido entre 0 y 9999, que se especifica a continuación.

En la memoria del CNC no pueden existir a la vez dos subrutinas con el mismo número de identificación, aunque pertenezcan a programas diferentes.

### ( RET )

La sentencia **RET** indica que la subrutina que se definió mediante la sentencia SUB, finaliza en dicho bloque.

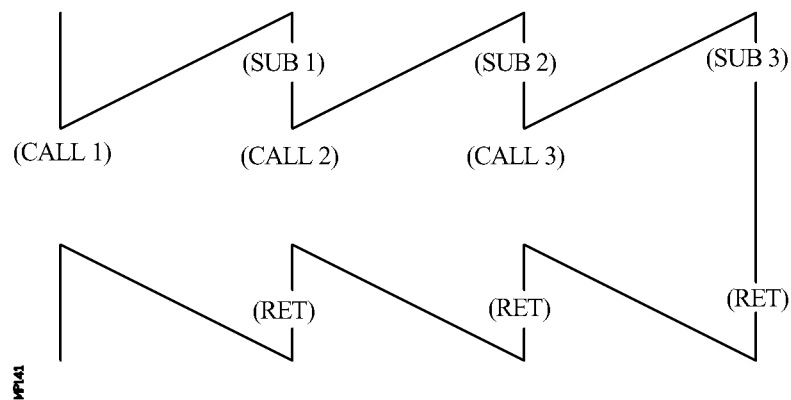
Ejemplo:

```
(SUB 12)                                ; Definición de la subrutina 12
G91 G01 XP0 F5000
      YP1
      X-P0
      Y-P1
(RET)                                    ; Fin de subrutina
```

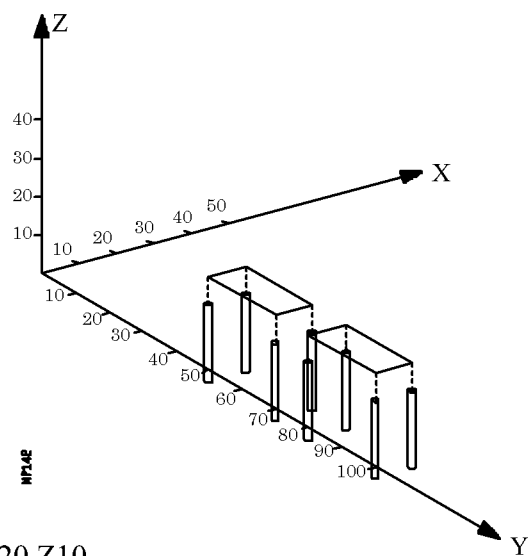
### ( CALL (expresión) )

La sentencia **CALL** realiza una llamada a la subrutina indicada mediante un número o mediante cualquier **expresión** que tenga como resultado un número.

Dado que de un programa principal, o de una subrutina se puede llamar a una subrutina, de ésta a una segunda, de la segunda a una tercera, etc..., el CNC limita estas llamadas hasta un máximo de 15 niveles de imbricación, pudiéndose repetir cada uno de los niveles 9999 veces.



Ejemplo:



G90 G00 X30 Y20 Z10  
**(CALL 10)**  
 G90 G00 X60 Y20 Z10  
**(CALL 10)**  
 M30

(SUB 10)  
 G91 G01 X20 F5000 ; Taladrado y roscado  
**(CALL 11)**  
 G91 G01 Y10 ; Taladrado y roscado  
**(CALL 11)**  
 G91 G01 X-20 ; Taladrado y roscado  
**(CALL 11)**  
 G91 G01 Y-10 ; Taladrado y roscado  
**(CALL 11)**  
 (RET)

(SUB 11)  
 G81 G98 G91 Z-8 I-22 F1000 S5000 T1 D1 ; Ciclo fijo de taladrado  
 G84 Z-8 I-22 K15 F500 S2000 T2 D2 ; Ciclo fijo de roscado  
 G80  
 (RET)

( **PCALL (expresión), (sentencia de asignación), (sentencia de asignación), ...** )

La sentencia **PCALL** realiza una llamada a la subrutina indicada mediante un número o mediante cualquier **expresión** que tenga como resultado un número. Además permite inicializar hasta un máximo de 26 parámetros locales de dicha subrutina

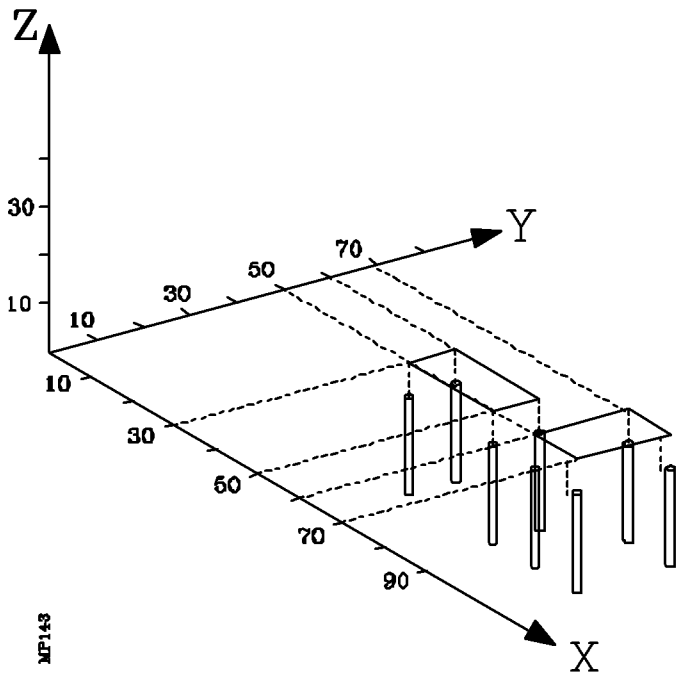
Estos parámetros locales se inicializan mediante las **sentencias de asignación**.

Ejemplo: (PCALL 52, A3, B5, C4, P10=20)

En este caso, además de generar un nuevo nivel de imbricación de subrutinas, se generará un nuevo nivel de imbricación de parámetros locales, existiendo un máximo de 6 niveles de imbricación de parámetros locales, dentro de los 15 niveles de imbricación de subrutinas.

Tanto el programa principal, como cada subrutina que se encuentre en un nivel de imbricación de parámetros, dispondrá de 26 parámetros locales (P0-P25).

Ejemplo:





G90 G00 X30 Y50 Z0  
**(PCALL 10, P0=20, P1=10)** ;o también (PCALL 10, A20, B10)  
 G90 G00 X60 Y50 Z0  
**(PCALL 10, P0=10, P1=20)** ;o también (PCALL 10, A10, B20)  
 M30

(SUB 10)  
 G91 G01 XP0 F5000  
 (CALL 11)  
 G91 G01 YP1  
 (CALL 11)  
 G91 G01 X-P0  
 (CALL 11)  
 G91 G01 Y-P1  
 (CALL 11)  
 (RET)

(SUB 11)  
 G81 G98 G91 Z-8 I-22 F1000 S5000 T1 D1 ; Ciclo fijo de taladrado  
 G84 Z-8 I-22 K15 F500 S2000 T2 D2 ; Ciclo fijo de roscado  
 G80  
 (RET)

#### ( MCALL (expresión), (sentencia de asignación), (sentencia de asignación), ... )

Por medio de la sentencia **MCALL**, cualquier subrutina definida por el usuario (SUB n° entero) adquiere la categoría de ciclo fijo.

La ejecución de esta sentencia es igual a la sentencia PCALL, pero la llamada es modal, es decir, si a continuación de este bloque, se programa algún otro con movimiento de los ejes, tras dicho movimiento se ejecutará la subrutina indicada y con los mismos parámetros de llamada.

Si estando seleccionada una subrutina modal se ejecuta un bloque de movimiento con número de repeticiones, por ejemplo X10 N3, el CNC ejecutará una única vez el desplazamiento (X10), y tras él la subrutina modal, tantas veces como indique el número de repeticiones.

En caso de seleccionarse repeticiones de bloque, la primera ejecución de la subrutina modal se realizará con los parámetros de llamada actualizados, pero no así el resto de las veces, que se ejecutarán con los valores que en ese momento dispongan dichos parámetros.

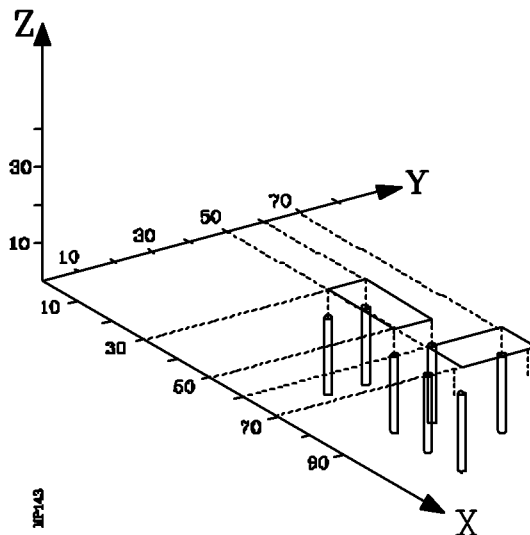
Si estando seleccionada una subrutina como modal se ejecuta un bloque que contenga la sentencia MCALL, la subrutina actual perderá su modalidad y la nueva subrutina seleccionada se convertirá en modal.

## ( MDOFF )

La sentencia **MDOFF** indica que la modalidad que adquirió la subrutina mediante la sentencia **MCALL**, finaliza en dicho bloque.

La utilización de subrutinas modales simplifica la programación.

Ejemplo:



```
G90 G00 X30 Y50 Z0
(PCALL 10, P0=20, P1=10)
G90 G00 X60 Y50 Z0
(PCALL 10, P0=10, P1=20)
M30
```

```
(SUB 10)
G91 G01 XP0 F5000
(MCALL 11)
G91 G01 YP1
G91 G01 X-P0
G91 G01 Y-P1
(MDOFF)
(RET)
```

```
(SUB 11)
G81 G98 G91 Z-8 I-22 F1000 S5000 T1 D1
G84 Z-8 I-22 K15 F500 S2000 T2 D2
G80
(RET)
```

## ( PROBE (expresión), (sentencia de asignación), (sentencia de asignación), ... )

La sentencia **PROBE** realiza una llamada al ciclo de palpador indicado mediante un número o mediante cualquier **expresión** que tenga como resultado un número. Además permite inicializar los parámetros locales de dicho ciclo, mediante las **sentencias de asignación**.

Esta sentencia, también genera un nuevo nivel de imbricación de subrutinas.

**( DIGIT (expresión), (sentencia de asignación), (sentencia de asignación), ... )**

La sentencia **DIGIT** realiza una llamada al ciclo de digitalizado indicado mediante un número o mediante cualquier **expresión** que tenga como resultado un número. Además permite inicializar los parámetros locales de dicho ciclo, mediante las **sentencias de asignación**.

Los puntos digitalizados se enviarán al programa (en memoria o vía DNC) abierto previamente con la sentencia:

(OPEN P (expresión), (directorio destino), A/D, "comentario de programa")

Esta sentencia, también genera un nuevo nivel de imbricación de subrutinas.

**( TRACE (expresión), (sentencia de asignación), (sentencia de asignación), ... )**

La sentencia **TRACE** realiza una llamada al ciclo de copiado indicado mediante un número o mediante cualquier **expresión** que tenga como resultado un número. Además permite inicializar los parámetros locales de dicho ciclo, mediante las **sentencias de asignación**.

Los puntos digitalizados se enviarán al programa (en memoria o vía DNC) abierto previamente con la sentencia:

(OPEN P (expresión), (directorio destino), A/D, "comentario de programa")

Esta sentencia, también genera un nuevo nivel de imbricación de subrutinas.

### 14.5.1 SENTENCIAS DE SUBROUTINAS DE INTERRUPCIÓN

Siempre que se activa una de las entradas lógicas generales de interrupción "INT1" (M5024), "INT2" (M5025), "INT3" (M5026) o "INT4" (M5027), el CNC suspende temporalmente la ejecución del programa en curso y pasa a ejecutar la subrutina de interrupción cuyo número se indica en el parámetro máquina general correspondiente.

Con INT1 (M5024) la indicada por el parámetro INT1SUB (P35)  
Con INT2 (M5025) la indicada por el parámetro INT2SUB (P36)  
Con INT3 (M5026) la indicada por el parámetro INT3SUB (P37)  
Con INT4 (M5027) la indicada por el parámetro INT4SUB (P38)

Las subrutinas de interrupción se definen como cualquier otra subrutina, utilizando las sentencias "(SUB n° entero)" y "(RET)".

Las subrutinas de interrupción no cambiarán el nivel de parámetros locales, por lo que dentro de ella sólo se permitirá la utilización de los parámetros globales.

Dentro de una subrutina de interrupción se puede utilizar la sentencia "(REPOS X, Y, Z, ....)" que se detalla a continuación.

Una vez finalizada la ejecución de la subrutina, el CNC continuará con la ejecución del programa en curso.

#### ( REPOS X, Y, Z, ... )

La sentencia **REPOS** se utilizará siempre dentro de las subrutinas de interrupción y facilita el reposicionamiento de la máquina en el punto de interrupción.

Cuando se ejecuta esta sentencia el CNC desplaza los ejes hasta el punto en que se interrumpió la ejecución del programa.

Dentro de la sentencia REPOS se debe indicar el orden en que se deben desplazar los ejes hasta el punto de interrupción.

- \* El desplazamiento se realiza eje a eje.
- \* No es necesario definir todos los ejes, sólo los que se desean reposicionar.
- \* El desplazamiento de los ejes que forman el plano principal de la máquina se hará de forma conjunta. No es necesario definir ambos ejes ya que el CNC efectúa dicho desplazamiento con el primero de ellos. No se repite el desplazamiento con la definición del segundo eje, lo ignora.

Ejemplo: El plano principal está formado por los ejes XY, el eje longitudinal es el eje Z y la máquina utiliza los ejes C y W como ejes auxiliares. Se desea reposicionar primero el eje C, luego los ejes XY y por último el Z.

Se puede utilizar cualquiera de estas definiciones:

(REPOS C, X, Y, Z) (REPOS C, X, Z) (REPOS C, Y, Z)

Si durante la ejecución de una subrutina que no ha sido activada mediante una de las entradas de interrupción, se detecta la sentencia REPOS el CNC mostrará el error correspondiente.

Página 12	Capítulo: 14 SENTENCIAS DE CONTROL DE LOS PROGRAMAS	Sección: SENTENCIAS DE SUBROUTINAS
--------------	--	---------------------------------------

## 14.6 SENTENCIAS DE PROGRAMAS

El CNC permite desde un programa en ejecución:

Ejecutar otro programa ..... Sentencia (EXEC P.....)  
Generar un programa nuevo ..... Sentencia (OPEN P.....)  
Añadir bloques a un programa ya existente ..... Sentencia (WRITE.....)

### ( EXEC P(expresión), (directorio) )

La sentencia **EXEC P** ejecuta el programa pieza del directorio indicado.

El programa pieza se puede definir mediante un número o mediante cualquier **expresión** que tenga como resultado un número.

Por defecto el CNC entiende que el programa pieza está en la memoria RAM del CNC. Si se encuentra en otro dispositivo hay que indicarlo en (**directorio**).

CARDA en la "Memkey CAR"  
HD en el Disco Duro  
DNC1 en un PC conectado a través de la línea serie 1  
DNC2 en un PC conectado a través de la línea serie 2

### ( OPEN P(expresión), (directorio destino), A/D, "comentario de programa" )

La sentencia **OPEN** comienza la edición de un programa pieza. El número de dicho programa vendrá indicado mediante un número o mediante cualquier **expresión** que tenga como resultado un número.

Por defecto el nuevo programa pieza editado se guardará en la memoria RAM del CNC. Para almacenarlo en otro dispositivo hay que indicarlo en (**directorio destino**).

CARDA en la "Memkey CAR"  
HD en el Disco Duro  
DNC1 en un PC conectado a través de la línea serie 1  
DNC2 en un PC conectado a través de la línea serie 2

El parámetro **A/D** se utilizará cuando el programa que se desea editar ya existe.

- A El CNC añade los nuevos bloques a continuación de los bloques ya existentes.
- D El CNC borra el programa existente y comenzará la edición de uno nuevo.

También es posible, si se desea, asociarle un **comentario de programa** que posteriormente será visualizado junto a él en el directorio de programas.

La sentencia **OPEN** es muy útil cuando se realizan digitalizados de piezas, ya que permite generar desde un programa en ejecución otro programa, que podrá estar en función de los valores que adquiera el programa en ejecución.

Para editar los bloques se debe utilizar la sentencia **WRITE** que se detalla a continuación.

**Notas:** Si el programa que se desea editar existe y no se definen los parámetros A/D el CNC mostrará un mensaje de error al ejecutar el bloque.

El programa abierto con la sentencia **OPEN** se cierra al ejecutarse M30, otra sentencia **OPEN** y después de una Emergencia o Reset .

Desde un PC sólo se pueden abrir programas en la memoria RAM, en la CARD A o en el Disco Duro (HD).

Capítulo: 14 SENTENCIAS DE CONTROL DE LOS PROGRAMAS	Sección: SENTENCIAS DE PROGRAMAS	Página 13
--	-------------------------------------	--------------

### ( WRITE <texto del bloque> )

La sentencia **WRITE** añade tras el último bloque del programa que se comenzó a editar mediante la sentencia OPEN P, la información contenida en <texto del bloque> como un nuevo bloque del programa.

Si se trata de un bloque paramétrico editado en código ISO todos los parámetros (globales y locales) son sustituidos por el valor numérico que tienen en ese momento.

(WRITE G1 XP100 YP101 F100)                   => G1 X10 Y20 F100

Cuando se trata de un bloque paramétrico editado en alto nivel hay que indicar con el carácter ? que se desea sustituir el parámetro por el valor numérico que tiene en ese momento.

(WRITE (SUB P102))                               => (SUB P102)  
(WRITE (SUB ?P102))                           => (SUB 55)

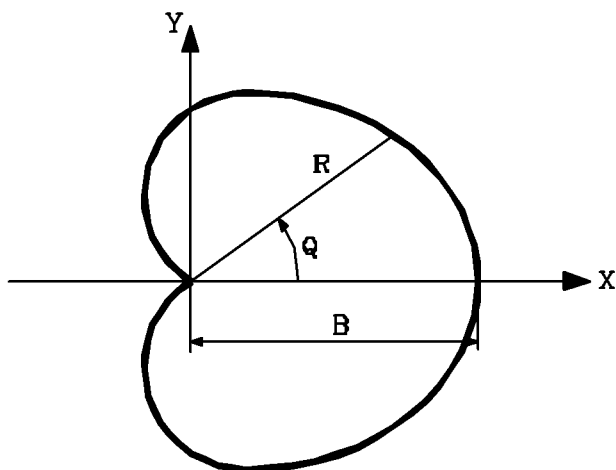
(WRITE (ORGX54=P103))                       => (ORGX54=P103)  
(WRITE (ORGX54=?P103))                   => (ORGX54=222)

(WRITE (PCALL P104))                         => (PCALL P104)  
(WRITE (PCALL ?P104))                     => (PCALL 25)

Si se programa la sentencia WRITE sin haber programado previamente la sentencia OPEN, el CNC mostrará el error correspondiente, excepto al editar un programa de personalización de usuario, en cuyo caso se añade un nuevo bloque al programa en edición.

Ejemplo de creación de un programa que contiene diversos puntos de una cardioide cuya fórmula es:

$$| R = B \cos (Q/2) |$$



Se utiliza la subrutina número 2, teniendo sus parámetros el siguiente significado:

- |        |                                    |
|--------|------------------------------------|
| A o P0 | Valor del ángulo Q                 |
| B o P1 | Valor de B                         |
| C o P2 | Incremento angular para el calculo |
| D o P3 | Avance de los ejes                 |

Una forma de utilizar este ejemplo podría ser:

```
G00 X0 Y0
G93
(PCALL 2, A0, B30, C5, D500)
M30
```

Subrutina de generación del programa:

	(SUB 2)	
	<b>(OPEN P12345)</b>	; Comienza la edición del programa P12345
	<b>(WRITE FP3)</b>	; Selecciona avance mecanizado
N100	(P10=P1 * (ABS(COS(P0/2))))	; Calcula R
	<b>(WRITE G01 G05 RP10 QP0)</b>	; Bloque de movimiento
	(P0=P0+P2)	; Nuevo ángulo
	(IF (P0 LT 365) GOTO N100)	; Si ángulo menor que 365°, calcula nuevo punto
	<b>(WRITE M30)</b>	; Bloque de fin de programa
	(RET)	; Fin de subrutina

## 14.7 SENTENCIAS DE PERSONALIZACIÓN

Las sentencias de personalización podrán utilizarse únicamente en los programas de personalización realizados por el usuario.

Estos programas de personalización, deben estar almacenados en la memoria RAM del CNC, y pueden utilizar las "Sentencias de Programación" y se ejecutarán en el canal especial destinado a tal efecto, indicándose en los siguientes parámetros máquina generales el programa seleccionado en cada caso.

En "USERDPLY" se indicará el programa que se desea ejecutar en el Modo de Ejecución.

En "USEREDIT" se indicará el programa que se desea ejecutar en el Modo de Edición.

En "USERMAN" se indicará el programa que se desea ejecutar en el Modo Manual.

En "USERDIAG" se indicará el programa que se desea ejecutar en el Modo de Diagnóstico.

Los programas de personalización pueden disponer, además del nivel actual, de otros cinco niveles de anidamiento. Además, las sentencias de personalización no admiten parámetros locales, no obstante se permite utilizar todos los parámetros globales en su definición.

### ( PAGE (expresión) )

La sentencia **PAGE** visualiza en la pantalla el número de página indicado mediante un número o mediante cualquier **expresión** que tenga como resultado un número.

Las páginas definidas por el usuario estarán comprendidas entre la página 0 y la página 255 y se definirán desde el teclado del CNC en el modo de personalización tal y como se indica en el Manual de Operación.

Las páginas del sistema se definirán mediante un número superior a 1000. Ver apéndice correspondiente.

### ( SYMBOL (expresión 1), (expresión 2), (expresión 3) )

La sentencia **SYMBOL** visualiza en la pantalla el símbolo cuyo número viene indicado mediante el valor de la **expresión 1** una vez evaluada.

Asimismo su posición en la pantalla viene definida por la **expresión 2** (columna) y por la **expresión 3** (fila).

Tanto expresión 1, como expresión 2 y expresión 3 podrán contener un número o cualquier expresión que tenga como resultado un número.

El CNC permite visualizar cualquier símbolo definido por el usuario (0-255) desde el teclado del CNC en el modo de personalización tal y como se indica en el Manual de Operación.

Para posicionarlo dentro del área de visualización se definirán los pixels de la misma, 0-639 para las columnas (expresión 2) y 0-335 para las filas (expresión 3).



### ( **IB (expresión) = INPUT “texto”, formato )**

El CNC dispone de 26 variables de entrada de datos (IB0-IB25).

La sentencia **IB** visualiza en la ventana de entrada de datos el **texto** indicado y almacena en la variable de entrada indicada mediante un número o mediante cualquier **expresión** que tenga como resultado un número, el dato introducido por el usuario.

La espera de introducción de datos se realizará únicamente cuando se programe el **formato** del dato solicitado. Este formato podrá tener signo, parte entera y parte decimal.

Si lleva signo “-” admitirá valores positivos y negativos, y si no lleva signo admitirá solo valores positivos.

La parte entera indica el número máximo de dígitos enteros (0-6) que se desean.

La parte decimal indica el número máximo de dígitos decimales (0-5) que se desean.

Si se programa sin formato numérico, por ejemplo (IB1 = INPUT “texto”), la sentencia visualiza el texto indicado y no espera la introducción de datos.

### ( **ODW (expresión 1), (expresión 2), (expresión 3) )**

La sentencia **ODW** define y dibuja en la pantalla una ventana de color blanco y de dimensiones fijas (1 fila x 14 columnas).

Cada ventana lleva asociado un número que viene indicado por el valor de la **expresión 1** una vez evaluada.

Asimismo su posición en la pantalla viene definida por la **expresión 2** (fila) y por la **expresión 3** (columna).

Tanto expresión 1, como expresión 2 y expresión 3 podrán contener un número o cualquier expresión que tenga como resultado un número.

El CNC permite definir 26 ventanas (0-25) y posicionarlas dentro del área de visualización, disponiendo para ello de 21 filas (0-20) y de 80 columnas (0-79).

**( DW (expresión 1) = (expresión 2), DW (expresión 3) = (expresión 4), ... )**

La sentencia **DW** visualiza en la ventana indicada por el valor de la **expresión 1, expresión 3, ..** una vez evaluada, el dato numérico indicado por la **expresión 2, expresión 4, ....**

Expresión 1, expresión 2, expresión 3, .... podrán contener un número o cualquier expresión que tenga como resultado un número.

El ejemplo siguiente muestra una visualización dinámica de variables:

```
(ODW 1, 6, 33)           ; Define la ventana de datos 1
(ODW 2, 14, 33)          ; Define la ventana de datos 2
N10 (DW1=DATE, DW2=TIME) ; Visualiza la fecha en la ventana 1 y la hora en
                           la 2
(GOTO N10)
```

El CNC permite visualizar el dato en formato decimal, hexadecimal y binario, disponiendo para ello de las siguientes instrucciones:

```
(DW1 = 100)   Formato decimal. Visualiza en la ventana 1 el valor "100".
(DWH2 = 100)  Formato hexadecimal. Visualiza en la ventana 2 el valor "64".
(DWB3 = 100)  Formato binario. Visualiza en la ventana 3 el valor "01100100".
```

Cuando se emplea la representación en formato binario (DWB) la visualización se limita a 8 caracteres, mostrándose el valor "11111111" para valores superiores a 255 y el valor "10000000" para valores inferiores a -127.

Además el CNC permite visualizar en la ventana solicitada, el número almacenado en una de las 26 variables de entrada de datos (IB0-IB25).

El ejemplo siguiente muestra una petición y posterior visualización del avance de los ejes:

```
(ODW 3, 4, 60)           ; Define la ventana de datos
                           3
(IB1=INPUT "Avance de los ejes: ", 5.4) ; Petición del avance de
                                           los ejes
(DW3=IB1)                 ; Visualiza el avance en la
                           ventana 3
```

( **SK (expresión 1)** = “texto 1”, (**expresión 2**) = “texto 2”, .... )

La sentencia **SK** define y visualiza el nuevo menú de softkeys indicado.

Cada una de las **expresiones** indicará el número de softkey que se desea modificar (1-7, comenzando por la izquierda) y los **textos** lo que se desea escribir en ellas.

Expresión 1, expresión 2, expresión 3, .... podrán contener un número o cualquier expresión que tenga como resultado un número.

Cada texto admitirá un máximo de 20 caracteres que se representarán en dos líneas de 10 caracteres cada una. Si el texto seleccionado tiene menos de 10 caracteres el CNC lo centrará en la línea superior, pero si tiene más de 10 caracteres el centrado lo realizará el programador.

Ejemplos:

(SK 1=“HELP”, SK 2=“COTA MAXIMA”)

HELP	COTA MAXIM A
------	-----------------

(SK 1=“AVANCE”, SK 2=“ \_\_ \_COTA\_ \_ \_ \_MAXIMA”)

AVANCE	COTA MAXIMA
--------	----------------

### **Atención:**



Si estando activo un menú de softkeys estándar del CNC se selecciona una o varias softkeys mediante la sentencia de alto nivel "SK", el CNC borrará todas las softkeys existentes y mostrará únicamente las que se han seleccionado.

Si estando activo un menú de softkeys de usuario se selecciona una o varias softkeys mediante la sentencia "SK", el CNC sustituirá únicamente las softkeys seleccionadas manteniendo el resto.

### **( WKEY )**

La sentencia **WKEY** detiene la ejecución del programa hasta que se pulse una tecla.

La tecla pulsada quedará registrada en la variable KEY.

Ejemplo:

.....

.....

**(WKEY)**

(IF KEY EQ \$FC00 GOTO N1000); ; Espera tecla  
Si se ha pulsado la tecla F1 continúa en N1000

.....

.....

### ( WBUF “texto”, (expresión) )

La sentencia **WBUF** se puede utilizar únicamente en el programa de personalización que se desea ejecutar en el Modo de Edición.

Esta sentencia se puede programar de dos formas y en cada caso permite:

#### ( WBUF “texto”, (expresión) )

Añade al bloque que se encuentra en edición y dentro de la ventana de entrada de datos, el **texto** y el valor de la **expresión** una vez evaluada.

(Expresión) podrá contener un número o cualquier expresión que tenga como resultado un número.

La expresión será opcional programarla, pero no así el texto que será obligatorio definirlo, si no se desea texto se programará “”.

Ejemplos para P100=10: (WBUF “X”, P100)      =>    X10  
(WBUF “X P100”)      =>    X P100

### ( WBUF )

Introduce en memoria, añadiendo al programa que se está editando y tras la posición que ocupa el cursor, el bloque que se encuentra en edición (previamente escrito con sentencias "(WBUF “texto”, (expresión))"). Además borra el buffer de edición, inicializándolo para una nueva edición de bloque.

Esto posibilita al usuario editar un programa completo, sin necesidad de abandonar el modo de edición de usuario tras cada bloque y pulsar [ENTER] para introducirlo en memoria.

Ejemplo:

(WBUF “(PCALL 25, “	; Añade al bloque en edición “(PCALL 25, “
(IB1=INPUT “Parámetro A:“, -5.4)	; Petición del parámetro A
(WBUF “A=”, IB1)	; Añade al bloque en edición “A = (valor introducido)”
(IB2=INPUT “Parámetro B: “, -5.4)	; Petición del parámetro B
(WBUF “, B=”, IB2)	; Añade al bloque en edición “B=(valor introducido)”
(WBUF “)”	; Añade al bloque en edición “)”
(WBUF )	; Introduce en memoria el bloque editado

.....  
.....

Tras ejecutar este programa se dispone en memoria de un bloque de este estilo:  
(PCALL 25, A=23.5, B=-2.25)

### ( SYSTEM )

La sentencia **SYSTEM** finaliza la ejecución del programa de personalización de usuario y vuelve al menú estándar correspondiente del CNC.

### Ejemplo de un programa de personalización:

El siguiente programa de personalización debe ser seleccionado como programa de usuario asociado al Modo Editor.

Tras seleccionarse el Modo Editor y pulsar la softkey USUARIO este programa comienza a ejecutarse y permite realizar una edición ayudada de los 2 ciclos de usuario permitidos. Esta edición se realiza ciclo a ciclo y cuantas veces se desea.

*; Visualiza la página inicial de edición*

N0 (PAGE 10)

*; Personaliza las softkeys de acceso a los distintos modos y solicita una opción*

(SK 1="CICLO 1",SK 2="CICLO 2",SK 7="SALIR")

N5 (WKEY)

*;Pedir tecla*

(IF KEY EQ SFC00 GOTO N10)

*;Ciclo 1*

(IF KEY EQ SFC01 GOTO N20)

*;Ciclo 2*

(IF KEY EQ SFC06 SYSTEM ELSE GOTO N5)

*;Salir o pedir tecla*

*; CICLO 1*

*; Visualiza la página 11 y define 2 ventanas de datos*

N10 (PAGE 11)

(ODW 1,10,60)

(ODW 2,15,60)

*;Edición*

(WBUF "( PCALL 1,")

*; Añade al bloque en edición (PCALL 1,*

(IB 1=INPUT "X:",-6.5)

*; Petición del valor de X*

(DW 1=IB1)

*; Visualiza, en la ventana 1, el valor introducido*

(WBUF "X",IB1)

*; Añade al bloque en edición X (valor introducido)*

(WBUF " ,")

*; Añade al bloque en edición ,*

(IB 2=INPUT "Y:",-6.5)

*; Petición del valor de Y*

(DW 2=IB2)

*; Visualiza, en la ventana 2, el valor introducido*

(WBUF "Y",IB2)

*; Añade al bloque en edición Y (valor introducido)*

(WBUF " )")

*; Añade al bloque en edición )*

(WBUF )

*; Introduce en memoria el bloque editado. P.e. (PCALL 1, X2, Y3)*

(GOTO N0)

*; CICLO 2*

*; Visualiza la página 12 y define 3ventanas de datos*

N20 (PAGE 12)

(ODW 1,10,60)

(ODW 2,13,60)

(ODW 3,16,60)

*;Edición*

(WBUF “( PCALL 2,”) ; *Añade al bloque en edición (PCALL 2,*

(IB 1=INPUT “A:”,-6.5) ; *Petición del valor de A*  
(DW 1=IB1) ; *Visualiza, en la ventana 1, el valor introducido*  
(WBUF “A”,IB1) ; *Añade al bloque en edición A (valor introducido)*

(WBUF “,”) ; *Añade al bloque en edición ,*

(IB 2=INPUT “B:”,-6.5) ; *Petición del valor de B*  
(DW 2=IB2) ; *Visualiza, en la ventana 2, el valor introducido*  
(WBUF “B”,IB2) ; *Añade al bloque en edición B (valor introducido)*

(WBUF “,”) ; *Añade al bloque en edición ,*

(IB 3=INPUT “C:”,-6.5) ; *Petición del valor de C*  
(DW 3=IB3) ; *Visualiza, en la ventana 3, el valor introducido*  
(WBUF “C”,IB3) ; *Añade al bloque en edición C (valor introducido)*

(WBUF “)”) ; *Añade al bloque en edición )*

(WBUF ) ; *Introduce en memoria el bloque editado. P.e. (PCALL 2, A3, B1, C3)*

(GOTO N0)