Macro B
Máquinas ISO

Edición N°4
Versión 1.0/MC 0703

06.06.2017

Índice

1 ¡¡Convierta sus programas en inteligentes!! 3
1.1 ¿Qué es "Macro B"? 3
1.2 ¿Sabía que...? 3
2 Instrucciones básicas de macro B 4
2.1 Variables 4
2.2 Operaciones aritméticas y lógicas 5
2.3 Conexiones y repeticiones 6
2.4 Subprogramas 8
2.5 Visualización de una alarma 9
2.6 Visualización de un mensaje 9
3 Ejemplos concretos de uso 10
3.1 Muestreo de piezas 10
3.2 Familia de piezas 11
3.3 Cómo crear su propia macro de mecanizado 12
3.4 Limpieza discontinua de la máquina 14
3.5 Parada de máquina predefinida 15
4 Conviene saber 16
4.1 Tiempo de ciclo 16
4.2 Cursos de formación de Tornos 16
4.3 Instrucción FANUC 16

1 ¡¡Convierta sus programas en inteligentes!!

1.1 ¿Qué es "Macro B"?

Macro B es un lenguaje de programación que se utiliza en los controles numéricos FANUC.
Todas nuestras máquinas ISO de última generación disponen de este lenguaje gratuitamente.
Su función es dotar de inteligencia a los programas. En el capítulo 3 veremos ejemplos de uso concretos.
Pero ante todo no se fíe de las apariencias: al principio este lenguaje puede parecerle complicado, pero es increíblemente fácil de utilizar en los programas que emplea a diario.

1.2 ¿Sabía que...?

Macro B es un lenguaje de programación muy sencillo que ofrece multitud de posibilidades.
Por ejemplo, ¿sabía que la mayoría de las macros Tornos de su máquina ISO comprenden más de 20.000 líneas de código que utilizan este mismo lenguaje?

2 Instrucciones básicas de macro B

2.1 Variables

Para convertir un programa en inteligente es necesario poder sustituir los valores por variables.
Una variable es un dato al que se puede asignar un valor.
La sintaxis de una variable se identifica mediante un "#" y un número de identificación.

Ejemplo de variable: #153
Ejemplo de asignación de un valor a una variable: #153 = 12,4 *(así, la variable #153 contiene el valor 12,4)*
Cuando las variables contienen un valor, se pueden sumar, multiplicar, utilizar como posición, como velocidad, como avance o incluso en una expresión condicional.

Veamos pues cuáles son las variables disponibles.

Variable nula:
La variable nula es aquella que nunca contiene ningún valor.

Es la siguiente: #**0**

Variables locales:
Las variables locales son aquellas que se restablecen en "nulas" cuando se cierra el programa en el que se les ha asignado un valor.

Se trata de las variables: #**1 - #33**

Variables globales por canal:
Las variables globales por canal son aquellas que se restablecen en "nulas" cuando se cierra el programa en el que se les asignó un valor.
Se dice que son "por canal" porque cuando se asigna un valor a una variable, esta contendrá el valor solo en el canal en el que se hizo la asignación.

Se trata de las variables: #**150 - #199**

Variables globales comunes:
Las variables globales comunes son aquellas que se restablecen en "nulas" cuando se cierra el programa en el que se les asignó un valor.
Se dice que son "comunes" porque cuando se asigna un valor a una variable, esta contendrá el valor en todos los canales de la máquina.

Se trata de las variables: #**600 - #699**

Variables de sistemas:
Las variables de sistemas pueden utilizarse para leer y escribir los datos del control numérico, como por ejemplo, variables de compensación de herramienta, datos de posición de ejes, etc.
Para más información sobre las variables de sistemas, consulte la instrucción FANUC.

Se trata de las variables: > **#1000**

2.2 Operaciones aritméticas y lógicas

Operaciones aritméticas:

Las operaciones aritméticas más utilizadas son:

Suma	"+"
Resta	"-"
Multiplicación	"*"
División	"/"

Ejemplo:
#603 = [#601 + #602] / 4

Funciones:

Las funciones más utilizadas son:

Seno	"SIN[#...]"
Coseno	"COS[#...]"
Tangente	"TAN[#...]"
Raíz cuadrada	"SQRT[#...]"
Valor absoluto	"ABS[#...]"
Potencia	"POW[#..., #...]"
Redondeo al número entero inferior	"FIX[#...]"
Redondeo del valor	"ROUND[#...]"

Ejemplo:
#603 = COS[#602]

Nota: la unidad de medida de ángulo empleada es el grado. Por ejemplo: 90 grados 30 minutos se escribe 90,5 grados.

Operadores relacionales:

Los operadores relacionales permiten comparar dos variables en una expresión condicional.

Los operadores relacionales son:

Igual a	"EQ"
Diferente de	"NE"
Superior a	"GT"
Superior o igual a	"GE"
Inferior a	"LT"
Inferior o igual a	"LE"

Operaciones lógicas:

Las operaciones lógicas permiten probar varias condiciones en una sola expresión condicional.

Las operaciones lógicas más utilizadas son:

Y lógico	"AND"
O lógico	"OR"

2.3 Conexiones y repeticiones

Instrucción incondicional "GOTO":

La instrucción "N*n*" al principio de la línea permite indicar el número de bloque.
La instrucción "GOTO*n*" es muy fácil de comprender: por ejemplo, GOTO 5 significa saltar al bloque N5.

Ejemplo:

```
...
GOTO 5
G0 X2 T35
M103 S200
G0 Z6
N5 G0 X3 T26
...
```

El CN no ejecuta esta parte del programa

Instrucción condicional "IF" - "THEN":

La instrucción "IF" significa: solo **si** la expresión condicional se cumple, se ejecuta lo que sigue.
La instrucción "THEN" significa **entonces**.

Ejemplo:

IF [#600 EQ #601] THEN #602 = 18 *Si los valores de #600 y de #601 son idénticos, entonces #602 toma valor 18*

Nota: "EQ" puede sustituirse por otros operadores relacionales

---

Instrucción condicional "IF" - "GOTO":

La instrucción "IF" significa: solo **si** la expresión condicional se cumple, se ejecuta lo que sigue.
La instrucción "GOTO" significa **ir a**.

Ejemplo:

```
IF [#600 EQ #601] GOTO 10
G0 X2 T35
M103 S200
G0 Z6
...
...
N10 G0 X3 T26
```

Si los valores de #600 y de #601 son idénticos, entonces se salta al bloque N10

El CN solo ejecutará esta parte del programa si #600 es diferente de #601

Nota: "EQ" puede sustituirse por otros operadores relacionales

---

Instrucción de repetición "WHILE":

La instrucción "WHILE" significa **bucle**.
La instrucción "DO" significa **hacer**.

Ejemplo:

```
WHILE [#600 GT #601] DO1
G0 X2 T35
M103 S200
G0 Z6
...
...
#601 = #601 + 1
END1
```

El CN ejecutará esta parte del programa siempre que el valor de #600 sea mayor que el de #601

(Incrementando #601 se sale del bucle)

Nota: "GT" puede sustituirse por otros operadores relacionales

Ejemplo de bucles anidados:

```
WHILE[...] DO1
...
...
    WHILE[...] DO2
    ...
    ...
        WHILE[...] DO3
        ...
        ...
        ...
        END3
    ...
    ...
    END2
...
...
END1
```

Nota: es posible anidar hasta tres bucles unos dentro de otros

2.4 Subprogramas

Ventajas de los subprogramas:

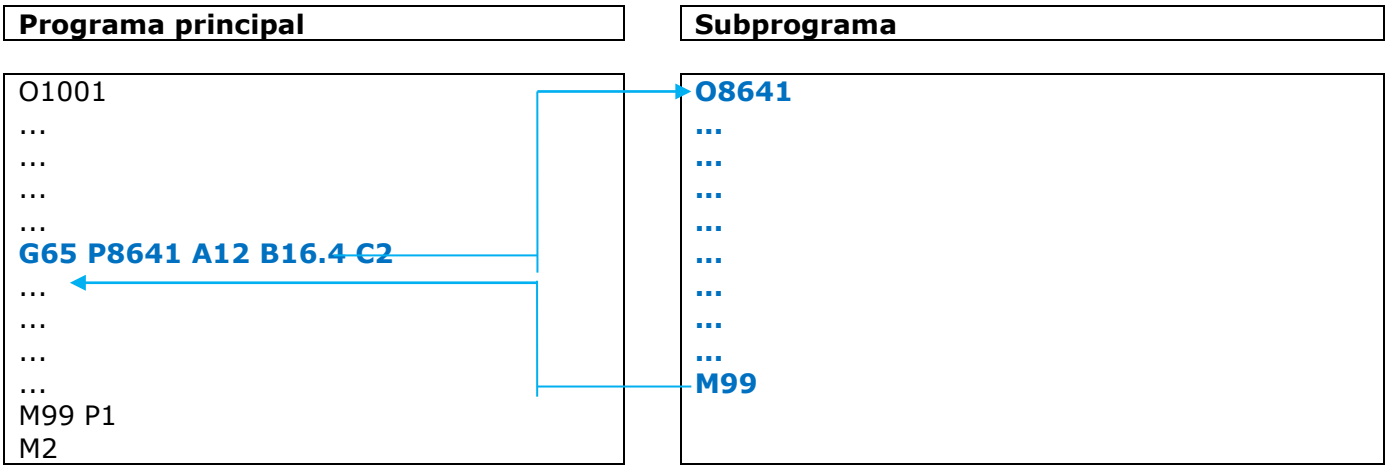
La principal ventaja de un subprograma es que puede recuperarse en varias ocasiones sin necesidad de recodificarlo.
Se puede recuperar varias veces desde un solo programa y también desde varios programas distintos.

Otra ventaja de un subprograma es que se le pueden transferir argumentos de configuración.

Recuperar un subprograma:

Los subprogramas se codifican, se nombran y se transfieren a la máquina de la misma forma que los programas principales.
Ejemplo de nombre "O8641".
Los subprogramas se recuperan mediante la función G65 P*n* {A*n* B*n* C*n* ...}.

Ejemplo:
G65 P8641 A12 B16.4 C2 *Recupera el subprograma O8641 y se le transfieren los argumentos A, B, C.*



Argumentos de los subprogramas:

La transferencia de argumentos de configuración a un subprograma es opcional.
Si desea hacerlo, los valores de los argumentos se transfieren automáticamente a las variables locales según la tabla de abajo:

Direcció	Número variable		Direcció	Número variable		Direcció	Número variable
A	#1		I	#4		T	#20
B	#2		J	#5		U	#21
C	#3		K	#6		V	#22
D	#7		M	#13		W	#23
E	#8		Q	#17		X	#24
F	#9		R	#18		Y	#25
H	#11		S	#19		Z	#26

2.5 Visualización de una alarma

También es posible hacer que aparezca una alarma en el CN de la forma siguiente:

#3000 = 1 (ALARMA) *Cuando el CN llegue a este bloque, se mostrará la alarma "MC3001 ALARMA" y bloqueará la interpretación.*

Ejemplo:

```
IF [#600 GE 0] GOTO 10
      #3000 = 2 (ERROR VALOR NEG
N10
```

Si #600 es menor que 0, aparecerá la alarma "MC3002 ERROR VALOR NEGATIVO" y bloqueará la interpretación.

2.6 Visualización de un mensaje

Es posible hacer que aparezca un mensaje en el CN de la forma siguiente:

#3006 = 1 (MENSAJE) *Cuando el CN llegue a este bloque, aparecerá el mensaje "MENSAJE" pero no bloqueará la interpretación.*

Ejemplo:

```
IF [#600 GE 0] GOTO 10
      #3006 = 2 (ATENCIÓN VALOR NEGA
N10
```

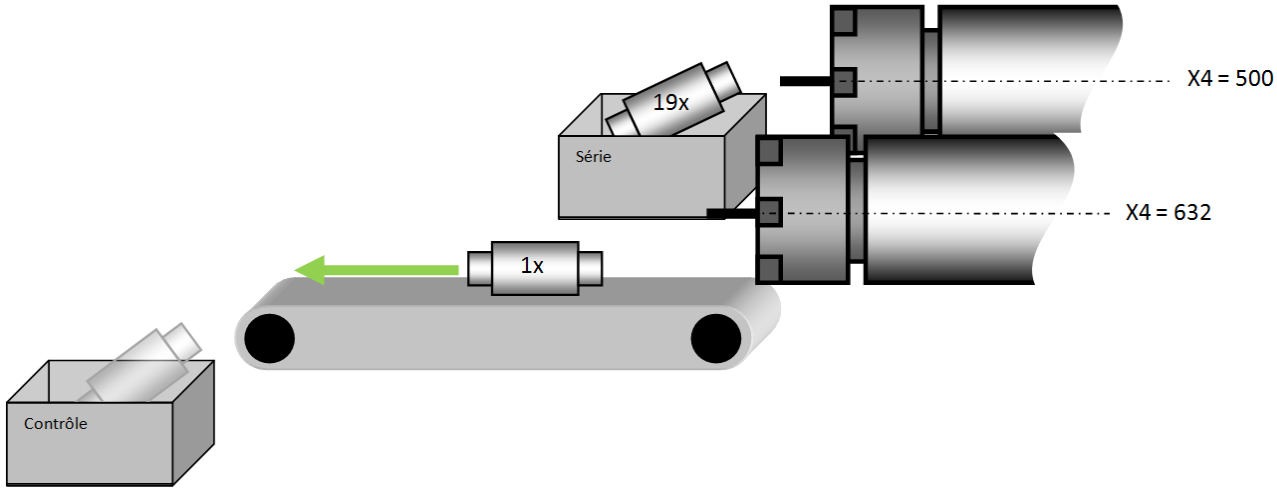
Si #600 es menor que 0, aparecerá el mensaje "ATENCIÓN VALOR NEGATIVO", pero no bloqueará la interpretación.

3 Ejemplos concretos de uso

3.1 Muestreo de piezas

Imaginemos que vamos a producir una serie de piezas que necesitan someterse a un control cada 20 ciclos.
Vamos a ver cómo puede ayudarnos Macro B.

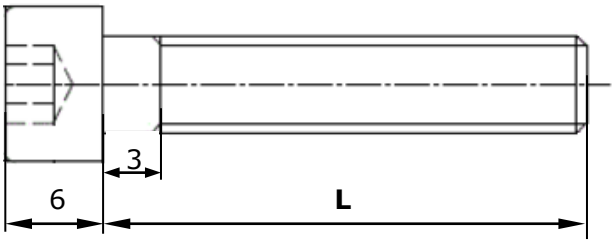
Principio:
Se trataría de expulsar la pieza 19 veces en el recuperador del interior de la máquina y una sola vez en el transportador de piezas, de modo que pueda controlarse fuera de la máquina.



Programación:

Canal del contrahusillo	
#600 = 0	**(INICIALIZACIÓN CONTADOR DE CICLOS)**
#601 = 20	**(PIEZA A CONTROLAR CADA TANTOS CICLOS)**
#602 = 500	**(POSICIÓN X4 EXTRACCIÓN SERIE)**
#603 = 632	**(POSICIÓN X4 EXTRACCIÓN PARA CONTROL)**
…	
…	
N1 M120	**(INICIO DE BUCLE)**
IF [#600 EQ #601] THEN	**(EN EL CICLO 20 PUESTA A 0 DEL CONTADOR)**
#600 = 0	**(INCREMENTO DEL CONTADOR)**
#600 = #600 + 1	
…	
…	
IF [#600 EQ #601] GOTO 10	**(CICLO DISTINTO DEL 20.º SE EXPULSA A X500)**
G53 X500	
GOTO 11	
N10	**(CICLO NÚMERO 20 SE EXPULSA A X630)**
G53 X632	
N11	**(EXPULSIÓN)**
M84	
…	
…	**(FIN DE BUCLE)**
M121	

3.2 Familia de piezas



Imaginemos que fabrica una gama de tornillos.
Todos son idénticos salvo en la longitud "L".
En este caso, sería interesante tener un solo programa para todos los tornillos en lugar de un programa para cada uno.

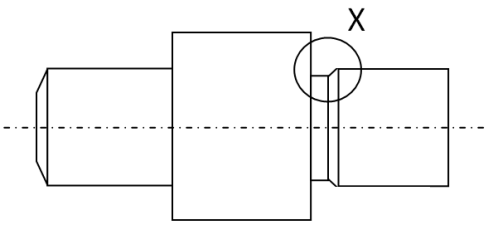
Canal 1

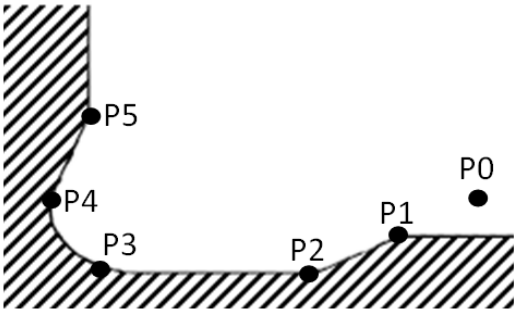
```
#600 = 53427                    (N° IDENTIFICACIÓN DE TORNILLO 53426, 53427,
IF [#600 EQ 53426] GOTO 5       53428, ...)
IF [#600 EQ 53427] GOTO 10
IF [#600 EQ 53428] GOTO 15
IF [#600 EQ 53429] GOTO 20
IF [#600 EQ 53430] GOTO 25
N5 #601 = 10
GOTO 30                         (LONGITUD "L" PARA TORNILLO 53426)
N10 #601 = 12
GOTO 30                         (LONGITUD "L" PARA TORNILLO 53427)
N15 #601 = 15
GOTO 30                         (LONGITUD "L" PARA TORNILLO 53428)
N20 #601 = 20
GOTO 30                         (LONGITUD "L" PARA TORNILLO 53429)
N25 #601 = 22
N30                             (LONGITUD "L" PARA TORNILLO 53430)
G800 A10 B[6+#601] C[#601+2]
...                             (DATO DE CICLO B: LONGITUD DE PIEZA, C: DISTANCIA
...                             TOMA DE PIEZA)
N1 M120
...
...                             (INICIO DE BUCLE)
G0 X5 Z2 T12 D0
G1 Z-#601 F0.06
G1 X12
...                             (TORNEADO DEL ALCANCE)
...
G78 P020060 Q500 R0.02
G78 X4.2 Z-[#601-3] R0 P4300
Q900 F0.7                       (FILETEADO)
...                             (FILETEADO)
...
M121
...                             (FIN DE BUCLE)
```

Como vemos, basta con cambiar el número de identificación del tornillo en la primera línea del programa para cambiar de tornillo.

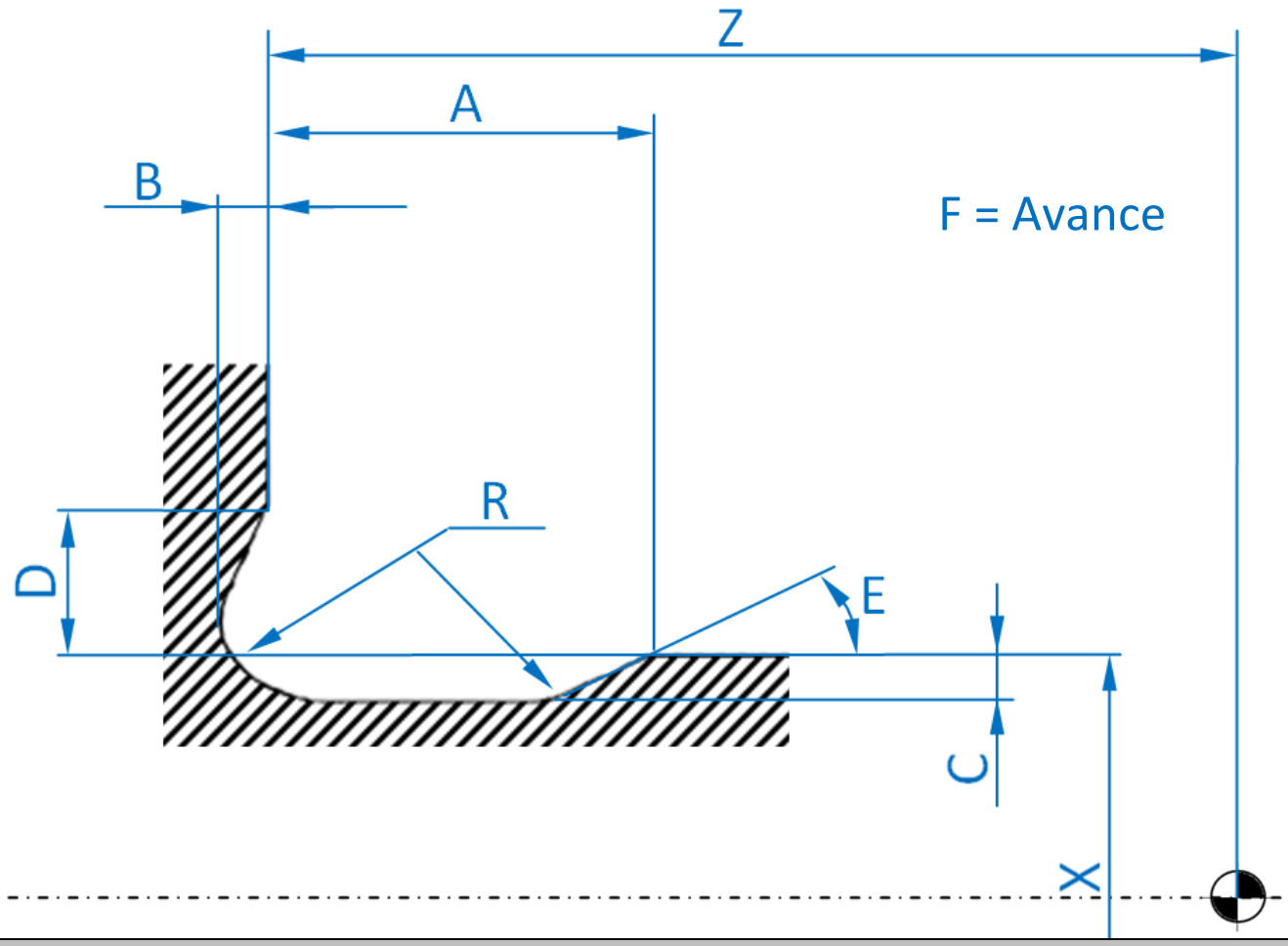
3.3 Cómo crear su propia macro de mecanizado

Imaginemos que necesita programar regularmente canales en sus piezas, de los que necesita calcular varios puntos. Para hacerle la vida más fácil, puede crear su propia macro de mecanizado.





Programación de la macro:



Programa principal	
...	
G65 P8512 A2 B0.1 C0.25 D1.15 E30 F0.05 R0.6 X12 Z36	**(RECUPERACIÓN DEL SUBPROGRAMA O8512)**
...	

Subprograma O8512 ***(Macro de mecanizado)***

```
(*** MEMORIZACIÓN ARGUMENTOS ***)
#600 = #1                                  (ARGUMENTO A)
#601 = #2                                  (ARGUMENTO B)
#602 = #3                                  (ARGUMENTO C)
#603 = #7                                  (ARGUMENTO D)
#604 = #8                                  (ARGUMENTO E)
#605 = #9                                  (ARGUMENTO F)
#606 = #18                                 (ARGUMENTO R)
#607 = #24                                 (ARGUMENTO X)
#608 = #26                                 (ARGUMENTO Z)
#609 = 2                                   (CONSTANTE DE SEGURIDAD)

(*** CÁLCULO P0 ***)
#611 = #607 + #609                         (P0 X)
#612 = -[ABS[#608] - #600 -                (P0 Z)
[[#609/2]/[TAN[#604]]]]
(*** CÁLCULO P1 ***)
#613 = #607                                (P1 X)
#614 = -[ABS[#608] - #600]                 (P1 Z)

(*** CÁLCULO P2 ***)
#615 = #607 - [#602 * 2]                   (P2 X)
                                           (P2 Z)
#616 = -[ABS[#608] - #600 +
[#602/TAN[#604]]]
(*** CÁLCULO P3 ***)                       (P3 X)
#617 = #615                                (P3 Z)
#618 = -[ABS[#608] + #601 - #606]

(*** CÁLCULO P4 ***)                       (P4 X)
#619 = #615 + [#606 * 2]                   (P4 Z)

#620 = -[ABS[#608] + #601]
                                           (P5 X)
(*** CÁLCULO P5 ***)                       (P5 Z)
#621 = #607 + [#603 * 2]

#622 = -[ABS[#608]]

(*** CÓDIGO ISO ***)                       (P0)
G90 G95                                    (P1)
G0 Y0                                      (P2)
G0 X#611 Z#612                             (P3)
G1 X#613 Z#614 F#605                       (P4)
G1 X#615 Z#616 ,R#606 F#605                (P5)
G1 X#617 Z#618 F#605
G2 X#619 Z#620 I#606 K0 F#605              (SALIDA DEL SUBPROGRAMA)
G1 X#621 Z#622 F#605

M99
```

3.4 Limpieza discontinua de la máquina

Imaginemos que produce una serie de piezas que le obligan a limpiar con cierta frecuencia el interior de la máquina para evacuar las virutas.
Vamos a ver cómo puede ayudarnos Macro B.

Principio:
Se trataría de limpiar regularmente las herramientas de corte mediante una bomba de alta presión, pero sin que funcione de forma continua.
Este procedimiento ofrece la ventaja de reducir el ruido y el consumo eléctrico de su taller.
En el ejemplo de abajo también contamos con otra ventaja: la limpieza de la pinza del contrahusillo con aire.

Canal del contrahusillo

```
#600 = 0                    (INICIALIZACIÓN CONTADOR DE CICLOS)
#601 = 100                  (MÁQUINA A LIMPIAR CADA TANTOS CICLOS)
...
N1 M120                     (INICIO DE BUCLE)
IF [#600 EQ #601] THEN      (EN EL CICLO 100 PUESTA A 0 DEL CONTADOR)
#600 = 0                    (INCREMENTO DEL CONTADOR)
#600 = #600 + 1
...
M11                         (EXPULSIÓN DE LA PIEZA DE LA PINZA DEL CONTRA-
M84                         HUSILLO)
...
...
IF [#600 NE #601] GOTO      (CADA 100 CICLOS SE EJECUTA EL CÓDIGO HASTA N10)
10                          (APERTURA VÁLVULA 1)
M532 M11 M1                 (ACTIVACIÓN BOMBA DE ALTA PRESIÓN)
M532 M1 M1                  (LIMPIEZA DEL SISTEMA DE HERRAMIENTAS 1)
G4 X4                       (CIERRE VÁLVULA 1)
M532 M11 M0                 (APERTURA VÁLVULA 2)
M532 M12 M1                 (LIMPIEZA DEL SISTEMA DE HERRAMIENTAS 2)
G4 X4                       (CIERRE VÁLVULA 2)
M532 M12 M0                 (APERTURA VÁLVULA 3)
M532 M13 M1                 (LIMPIEZA DEL SISTEMA DE HERRAMIENTAS 3)
G4 X4                       (CIERRE VÁLVULA 3)
M532 M13 M0                 (DESACTIVACIÓN BOMBA DE ALTA PRESIÓN)
M532 M1 M0                  (SOPLADO CENTRO CONTRA-HUSILLO 3 SEG. PARA LIMPIEZA
M841 M2 M3000               DE PINZA)
N10
...
...
M121                        (FIN DE BUCLE)
...
```

3.5 Parada de máquina predefinida

Imagine que su máquina produce toda la semana pero necesita detenerla el sábado por la tarde para evitar que el desgaste de la herramienta produzca piezas fuera de tolerancia, hasta el lunes por la mañana.
Seguramente agradecería no tener que volver a la fábrica todos los sábados por la tarde para detener la máquina. Vamos a ver cómo puede ayudarnos Macro B.

Principio:
Se trataría de controlar en cada ciclo la fecha y la hora actual del CN y hacer que la máquina se detenga cuando llegue la fecha y la hora indicadas.

En este ejemplo vamos a detener la máquina
el: 24.06.2017 (#600)
a las: 20:35:00 (#601)

Canal 1

```
#600 = 203500                    (HORA DE LA PARADA: HORA - MINUTO - SEGUNDO)
#601 = 20170624                  (FECHA DE PARADA DE LA MÁQUINA: AÑO - MES -
...                               DÍA)
...
N1 M120
...                              (INICIO DE BUCLE)
...
IF [#601 NE #3011] GOTO 10
IF [#600 LT #3012] GOTO 10       (COMPRUEBA SI LA FECHA ACTUAL ES LA MISMA QUE
M105                              LA FECHA DE PARADA DE LA MÁQUINA)
M405                             (COMPRUEBA CUÁNDO LLEGA LA HORA DE PARADA DE
M1105                             LA MÁQUINA)
M9                               (PARADA DE LOS HUSILLOS)
M0
N10
M121                             (PARADA DEL ACEITE)
...                              (PARADA DEL CICLO)

                                 (FIN DE BUCLE)
```

NB: La #3011 es una variable del sistema que corresponde a la fecha actual en el CN (año, mes, día)
La #3012 es una variable del sistema que corresponde a la hora actual en el CN (hora, minuto, segundo)

4 Conviene saber

4.1 Tiempo de ciclo

Le recomendamos codificar en Macro B todo lo que sea posible antes del bucle de mecanizado *(antes de M120)*. De esta forma evitará perder tiempo de ciclo con el tratamiento de las condiciones y de los cálculos.

4.2 Cursos de formación de Tornos

Tornos imparte cursos de programación que le permitirán convertirse en un verdadero especialista y sacar el máximo partido a todas las posibilidades de este lenguaje.

4.3 Instrucción FANUC

En la instrucción FANUC B-63944 se explican con detalle todas las posibilidades del lenguaje.