La sección B se compone de 2 niveles de información:

- la información que permite realizar funciones simples. En ese caso, no será necesario leer la totalidad de la documentación, consúltese únicamente los apartados sombreados.

- la información que permite realizar las funciones ofrecidas por el autómata TSX Nano, en ese caso consúltese la totalidad del manual.

1.1 Generalidades

El desarrollo de una aplicación destinada al autómata TSX Nano puede realizarse mediante dos herramientas de programación:

- El terminal de programación FTX 117 que propone el lenguaje de lista de instrucciones (Lista o IL), es un lenguaje booleano que permite la escritura de tratamientos lógico y numérico.

- El programa PL7-07 para IBM PC o compatible que ofrece el lenguaje Lista y de contactos (Ladder o LD), lenguaje gráfico que permite la transcripción de esquemas de relés mediante símbolos (contactos, bobinas), la escritura de cálculos numéricos, puede realizarse desde bloques de operaciones.

El PL707 permite revertir lenguaje: paso del lenguaje Ladder al lenguaje Lista y viceversa.

El TSX Nano soporta además las instrucciones del GRAFCET.

1.2 Lenguaje Lista de instrucciones (Lista o IL)

Estructura del programa
Un programa en lenguaje Lista consta de una serie de instrucciones (hasta 1000 instrucciones) de diversos tipos.

Cada fila de programa tiene un número generado de forma automática, un código de instrucción y un operando tipo bit o palabra

Ejemplo de instrucción: 003 LD %I0.1

Número — Operando

Código de instrucción

El programa en lenguaje Lista es una serie de expresiones lógicas escritas en forma de secuencias de instrucciones booleanas. Cada instrucción booleana, salvo LOAD, STORE y NOT, actúan en dos operandos (uno explícito y otro implícito).

El operando implícito es el acumulador booleano y consta o bien del contenido de la primera instrucción de una secuencia de instrucciones (ej: LD %I0,0), o bien para las siguientes instrucciones, del resultado de la instrucción anterior (ej: AND %I1,2),

```
Ejemplo:   001   LD    %i0,1
           002   AND   %I1.2
           003   ST    %Q1,0
```

La operación AND %I1.2, ejecutará una Y lógica entre el contenido del acumulador (%I0,1) y la entrada %I1.2, y remplazará el contenido del acumulador con este resultado.

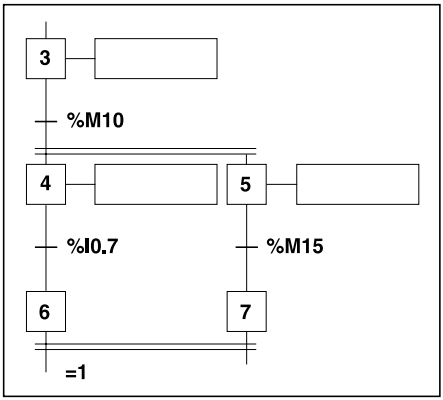
Las instrucciones LOAD y STORE cargan respectivamente el acumulador con el valor del operando o bien almacenan el acumulador en el operando. La instrucción NOT no dispone de operando explícito, inversa simplemente el estado del acumulador.

Instrucciones

Tipo de instrucciones	**Ejemplo**	
• Instrucciones en bit	004 LD %M10	Lee el bit interno %M10
• Instrucciones en bloque	008 IN %TM0	Lanza la temporización %TM0
• Instrucciones en palabra	010 [%MW10:= %MW50+100]	Suma
• Instrucciones en programa	015 SR5	Llama al subprograma n° 5
• Instrucciones en Grafcet	020 -*-8	Etapa n° 8

1.3 Grafcet

Grafcet es un método de análisis que consiste en descomponer un automatismo secuencial en una sucesión de etapas, a las que se asocian acciones, transiciones y condiciones.
El programa PL7-07 al no soportar el Grafcet gráfico posee instrucciones específicas Grafcet.



Grafcet gráfico

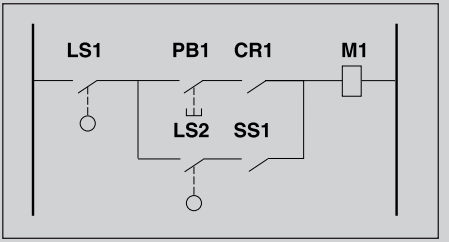
```
-*-     3
LD      %M10
#       4
#       5
-*-     4
LD      %I0.7
#       6
-*-     5
LD      %M15
#       7
...
```

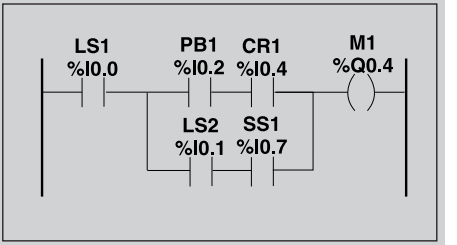
Grafcet Lista de instrucciones

Grafcet lenguaje de contactos

1.4 Lenguaje de contactos (Ladder o LD)

Un programa escrito en lenguaje de contactos se compone de una serie de circuitos ejecutados secuencialmente por el autómata. La representación de un circuito se asemeja a la de un esquema eléctrico de relés. Elementos gráficos de tests simbolizan los contactos (botón pulsador, contactos fin de recorrido, etc...), así como elementos gráficos de acciones simbolizan las bobinas.



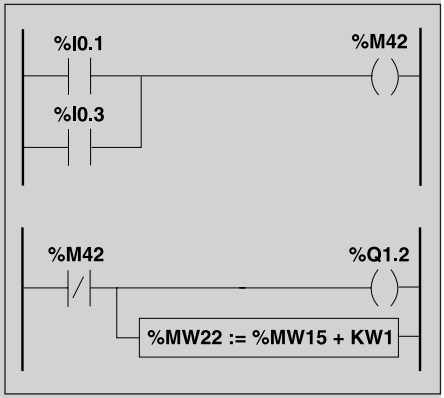


En la figura anterior se ilustra el esquema de cableado simplificado de un circuito de lógica de relés y su equivalente en esquema de contactos. Obsérvese que en el esquema de contactos, todas las entradas asociadas con un dispositivo de conmutación en el esquema de lógica de relés se muestran en forma de contactos, la bobina M1 queda representada por un símbolo de bobina. Las referencias que aparecen encima de cada símbolo de contacto/bobina indican la ubicación de las conexiones de entrada/salida externas en el autómata.
Un circuito de contactos se compone de una serie de instrucciones gráficas específicas, relacionadas entre sí, y situadas entre las dos barras verticales que representan el potencial.

El juego de instrucciones gráficas representa:

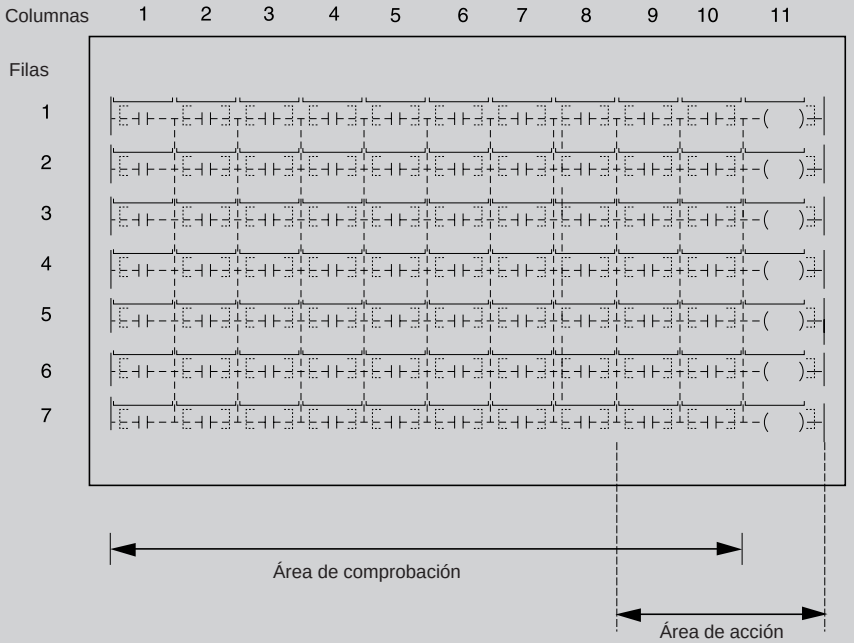
- las entradas/salidas del autómata (botones pulsadores, sensores, relés, indicadores de funcionamiento,...)
- las funciones del autómata (temporizadores, contadores...),
- las operaciones matemáticas y lógicas (suma, división, y, o exclusiva...),
- los operadores de comparación y otras operaciones numéricas (A < B, A = B, desplazamiento, circular...),
- las variables internas del autómata (bits, palabras ...).

Estas instrucciones gráficas se asocian entre sí mediante conexiones horizontales y verticales que conducen a una o varias salidas y/o acciones.
Un circuito no deberá soportar más de un grupo de instrucciones asociadas.
Por lo tanto, el programa de la derecha se compone de dos circuitos distintos.



1.4-1 Principios de programación

Cada circuito de contactos se compone de 7 filas y de 11 columnas y se divide en dos áreas:



- un área de comprobación que contiene las condiciones que deberán reunirse para la ejecución de una acción,
- un área de acción que contiene la salida u operación resultante de las comprobaciones que se le asocian.

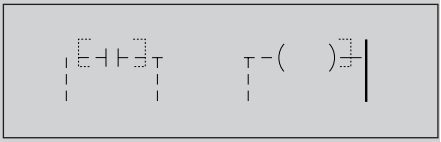
El circuito está representado por un cuadrícula de programación de 7 filas por 11 columnas y que se inicia en la celda superior izquierda. Las instrucciones, comparaciones y funciones asociadas a las comprobaciones se introducen en el área de prueba. Luego, estas instrucciones están justificadas a la izquierda, garantizando así la continuidad del área de acción en la cual se introducen las instrucciones referentes a bobinas, operaciones numéricas y gestión del programa. Estas instrucciones están justificadas a la derecha. El circuito se resuelve o ejecuta (ejecución de las comprobaciones y asignación de las salidas) de arriba abajo y de izquierda a derecha.

Además, aparece un encabezado de circuito justo encima de éste. Este encabezado permite especificar la intención lógica del circuito. Contiene el número de circuito, todas las etiquetas (%Li) o las declaraciones de subprogramas (SRi), el título del circuito así como observaciones referentes al circuito. Para cualquier información adicional acerca del encabezado de circuito y de cómo se corresponde con los comentarios de fila de tipo Lista, consúltese el apartado 1.4-3 de la sección B.

B

- **Contactos, bobinas e instrucciones referentes al desarrollo del programa**

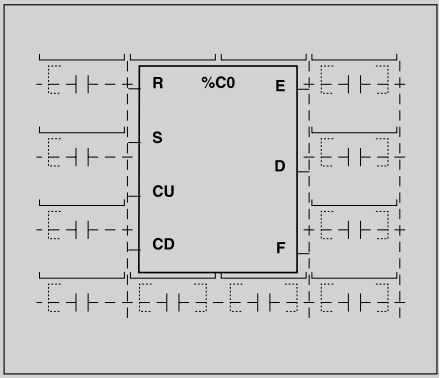
Las instrucciones de contactos, bobinas y desarrollo del programa (salto y llamada) ocupan una sola celda de la cuadrícula de programación. Los bloques de función, comparación y operación ocupan varias celdas.



- **Bloques de función**

Los bloques de función están situados en el área de comprobación de la cuadrícula de programación. El bloque deberá aparecer en la primera fila, no pueden aparecer instrucciones en lenguaje de contactos ni filas de continuidad encima o debajo de este bloque. Las instrucciones de comprobación de lenguaje de contactos llegan a la entrada del bloque de función y las instrucciones de comprobación y/o de acción parten de la salida del bloque.

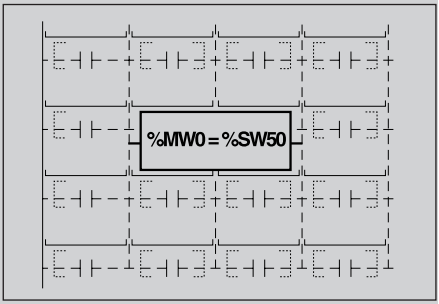
Los bloques de función se organizan verticalmente y ocupan 2 columnas en 4 filas de la cuadrícula de programación.



- **Bloques de comparación**

Los bloques de comparación están situados en el área de comprobación de la cuadrícula de programación. El bloque puede aparecer en cualquier fila o columna de este área siempre y cuando la totalidad de la instrucción se encuentre en dicha área.

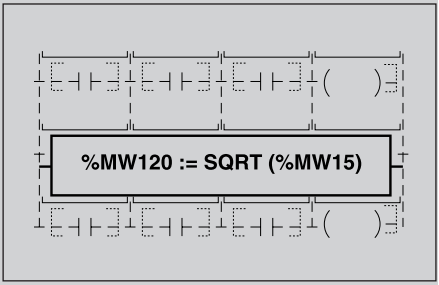
Los bloques de comparación se organizan horizontalmente y ocupan 2 columnas en 1 fila de la cuadrícula de programación.



- **Bloques de operación**

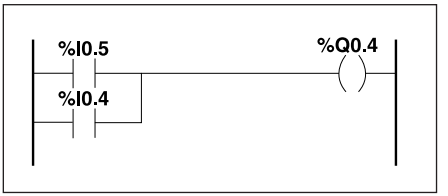
Los bloques de operación se encuentran en el área de acción de la cuadrícula de programación. El bloque puede aparecer en cualquier fila de dicha área. La instrucción está justificada a la derecha. Por lo tanto, aparecerá a la derecha y finalizará en la última columna.

Los bloques de operación se organizan horizontalmente y ocupan 4 columnas en 1 fila de la cuadrícula de programación.



1.4-2 Reversibilidad

En el presente manual, el término "reversibilidad" se refiere a la capacidad del programa PL7-07 del TSX Nano para convertir en lenguaje Lista los programas de aplicación del TSX Nano escritos en lenguaje Ladder y viceversa. Los programas PL7-07 pueden visualizarse en el formato seleccionado, estableciendo una preferencia por uno u otro. El programa PL7-07 también puede convertir un circuito Ladder individual en lenguaje Lista y restablecerlo a continuación seleccionando simplemente la opción Ladder/Lista del menú Herramientas del editor Ladder/Lista.




```
LD    %I0.5
OR    %I0.4
ST    %Q0.4
```

Para entender el concepto de reversibilidad es importante conocer la relación que existe entre el "circuito", conjunto de instrucciones de programación en lenguaje de contactos que constituyen una expresión lógica, y la "sentencia", el conjunto de instrucciones de programación en lenguaje Lista que realizan la misma función. En la figura anterior se muestra un circuito usual en lenguaje de contactos de un programa de usuario. Junto a esta figura se representa la lógica equivalente expresada en forma de sentencia en lenguaje Lista.

Una aplicación escrita en lenguaje PL7-07 (Ladder o Lista) se almacena en la memoria en lenguaje Lista.
La estructura del lenguaje Ladder del programa PL7-07 permite convertir un programa Ladder en programa Lista sin necesidad de adoptar previamente cualquier disposición especial.
La conversión de una aplicación desarrollada en Lista requiere una serie de convenciones de reversibilidad que deberán respetarse y que se detallan en el apartado 1.4-3 de esta sección. Sin embargo, no siempre será posible revertir un programa Lista pero esto no afectará al funcionamiento de una aplicación.

1.4-3 Convenciones de programación en lenguaje Lista reversible

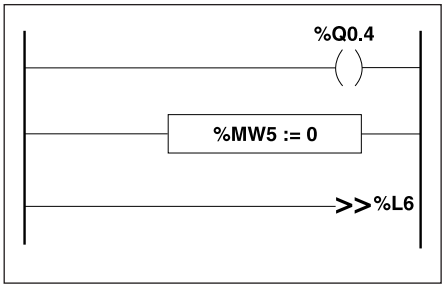
La estructura de un bloque de función reversible en lenguaje Lista requiere emplear ciertas instrucciones específicas. Se trata de:
- BLK que señala el inicio del bloque y define el comienzo del circuito y el inicio de la porción de entrada en el bloque
- OUT_BLK (señala el inicio de la porción de salida del bloque
- END_BLK (señala el fin del bloque y del circuito).

No es indispensable para el buen funcionamiento del programa en lenguaje Lista la utilización de instrucciones reversibles. Para algunas instrucciones, se puede programar en lenguaje Lista no reversible. La programación en lenguaje Lista no reversible de los bloques de función se detalla en el apartado 2.2- sección B.

Otra convención importante es que se debe procurar no utilizar determinadas instrucciones, o combinaciones de instrucciones y operandos, en lenguaje de lista que no tengan equivalente en lenguaje de contactos. Por ejemplo, la instrucción N (que invierte el contenido del acumulador), no tiene equivalente en lenguaje de contactos. En la siguiente tabla se enumeran todas las instrucciones de programación en lenguaje Lista no convertible en lenguaje de contactos.

Instrucción (lista)	Operando	Descripción
JMPCN	%Li	Salto de programa si prueba anterior = 0
N	ninguno	Negación (NOT)
ENDCN	ninguno	Fin de programa si prueba anterior = 0
or'd XORN	cualquiera	XORN precedido por O lógica

Asimismo los circuitos incondicionales deben cumplir una convención de programación en lenguaje Lista para permitir la reversibilidad lenguaje Ladder/Lista. Un circuito incondicional es un circuito en el que no hay ni comprobaciones ni condiciones; la(s) instrucción(es) relativa(s) a la salida y/o la acción se activa(n) o ejecuta(n) constantemente. La siguiente figura presenta circuitos incondicionales y las fases equivalentes en lenguaje Lista.




```
LD    1
ST    %Q0.4
LD    1
[%MW5:= 0]
JMP   %L6
```

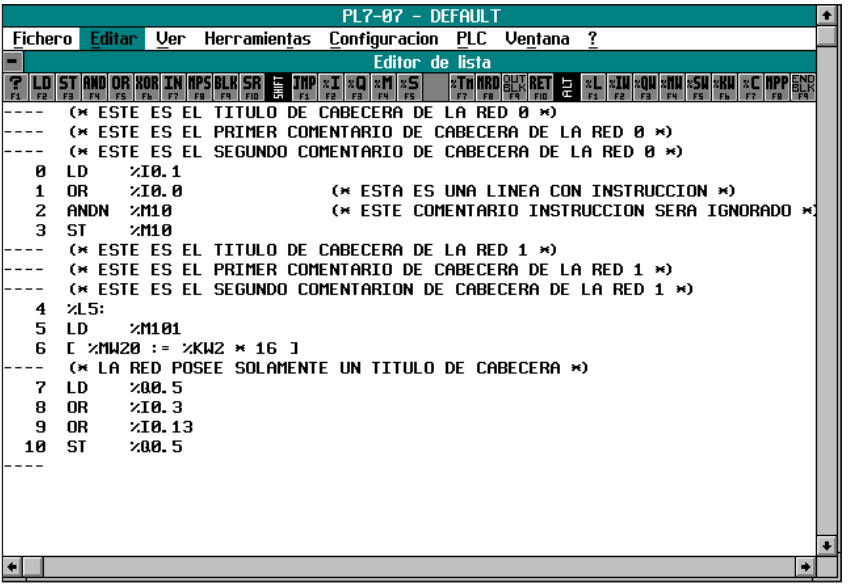
Obsérvese que cada una de las sentencias incondicionales en lenguaje Lista, salvo una, empiezan por la instrucción LD (Load) seguida del número 1. Esta combinación pone el acumulador booleano a 1 y por consiguiente, la bobina (instrucción de memorización) a 1 y pone %MW5 a 0 en cada ciclo de exploración del programa. La instrucción de salto incondicional es una excepción. La instrucción en lenguaje Lista se ejecutará sea cual sea el valor del acumulador y por lo tanto no requiere que el acumulador se haya puesto a 1, mientras que los dos ejemplos anteriores sí lo precisaban.

Se puede revertir un programa en Lista que no sea completamente reversible, las partes reversibles se visualizarán en lenguaje Ladder, y las porciones no reversibles permanecerán en Lista. El programa así convertido mantiene el orden inicial de escritura. Los "circuitos" de instrucciones Lista no reversibles podrán visualizarse y modificarse desde el editor de circuito Lista al que se podrá acceder haciendo doble clic en el circuito correspondiente.

Descripción del programa

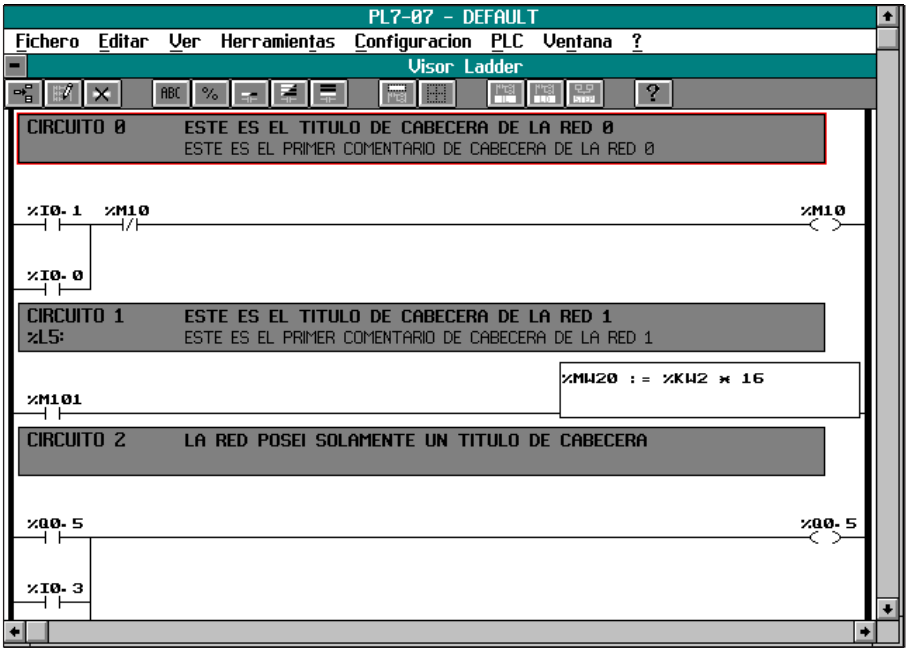
El editor Lista permite insertar en el programa líneas de comentario. Estos comentarios pueden aparecer aislados o en la misma fila que las instrucciones de programación. El editor Ladder le permitirá documentar el programa con encabezados de circuitos situados justo encima de estos últimos.

El programa PL7-07 tendrá en cuenta estos comentarios durante la conversión. Cuando revierte un programa Lista en Ladder, el PL7-07 utiliza líneas de comentarios aisladas, situadas encima de las sentencias en Lista para elaborar los encabezados de los circuitos correspondientes.



La primera líneas de comentarios aislados se utiliza para documentar la primera fila de encabezado del circuito Ladder correspondiente. Por lo tanto el encabezado de un circuito se documenta utilizando las líneas de comentarios aislados en el orden de escritura. Cuando las filas de encabezado están llenas, las restantes líneas de comentarios aislados se ignoran, así como todos los comentarios que se encuentran en las filas de instrucciones en Lista.

Cuando se convierte en lenguaje Lista un circuito Ladder que contiene un encabezado de circuito, las descripción de dicho encabezado se inserta entre las sentencias en Lista. Cualquier etiqueta o declaración de subprograma (%Li o SRi) se situará en la fila siguiente al encabezado, justo antes del principio de la sentencia en Lista. Si el circuito revertido estaba escrito originariamente en Lista y si se ignoraron algunos comentarios durante la conversión de Lista a Ladder, estos comentarios volverán a aparecer en el editor Lista.



B

2.1 Tratamiento booleano

2.1-1 Definición de los principales objetos de bits

• Bits de entradas/salidas

El direccionamiento de estos bits se detalla en el apartado 1.5 sección A. Estos bits son las "imágenes lógicas" de los estados eléctricos de las entradas/salidas. Están almacenados en la memoria de datos y se actualizan en cada exploración del programa.

• Bits internos

Los bits internos memorizan los estados intermedios durante la ejecución del programa.

Nota: los bits de entrada/salida no utilizados no pueden ser empleados como bits internos.

• Bits de sistema

Los bits de sistema de %S0 a %S127 controlan el buen funcionamiento del autómata así como el desarrollo del programa de aplicación. La función y la utilización de estos bits se describe en el capítulo 6 de la presente sección.

• Bits de etapas

Los bits de %X1 a %X62 son los bits asociados a las etapas Grafcet. El bit de etapa Xi está a 1 cuando la etapa correspondiente está activa y a 0 cuando esta etapa está inactiva.

• Bits extraídos de palabras: véase el apartado 3.1-1

Lista de operandos de bits

La siguiente tabla muestra la lista de todos los tipos de operandos de bits

Tipo	Dirección (o valor)	Número máximo	Acceso en escritura(1)	Ver apart.
Valor inmediato	0 ó 1	-	-	-
Bits de entrada de salida	%I0.i o %I1.i (2) %Q0.i o %Q1.i (2)	28 20	no sí	1.5 Sec.A
Bits internos	%Mi	128 (3)	sí	
Bits de sistema	%Si	128	según i	5.1
Bits de etapa Grafcet	%Xi	62	sí	2.3-1
Bits de bloques función	%TMi.Q %DRi.F.....		no (4)	2.2-1
Bits bloques función reversible	E,D,F,Q,TH0,TH1		no	3.3-1
Bits extraídos palabr.				3.1-1

(1) escritura por programa o en modo de ajuste por terminal.
(2) con i = 0 para un autómata de base o una extensión de autómata, i = 1 para una extensión de E/S, y j = n° de la vía. Los bits de entradas/salidas pueden forzarse a 0 ó 1 en modo de ajuste de datos.
(3) los 64 primeros se salvaguardan si se produce un corte de la alimentación.
(4) salvo %SBRi.j y %SCi.j estos bits pueden leerse y escribirse.

2.1-2 Presentación de instrucciones booleanas

Las instrucciones booleanas pueden ser comparadas con los elementos de lenguaje de contactos.

Elementos de comprobación, ej.: la instrucción LD equivale a un contacto abierto.

LD %I0.0

Conduce cuando el objeto que lo controla se encuentra en el estado 1.

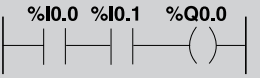
Elementos de acción, ej.: la instrucción ST equivale a una bobina directa.

ST %Q0.0

El objeto asociado toma el valor lógico del resultado lógico del elemento de test.

Ecuación booleana:

LD %I0.0
AND %I0.1
ST %Q0.0



El resultado booleano de los elementos de test se aplica al elemento de acción.

Flancos ascendente y descendente

Las instrucciones de comprobación permiten detectar los flancos ascendente y descendente en las entradas del autómata. Se detecta un flanco cuando el estado de una entrada ha cambiado entre el ciclo n-1 y el ciclo n en curso, y permanece detectado durante el ciclo en curso.

La instrucción LDR (R: Rising edge) equivale a un contacto de detección de flanco ascendente:

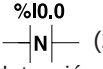
LDR %I0.0

(1)

Flanco ascendente: detección del paso de 0 a 1 de la entrada que lo controla (3).

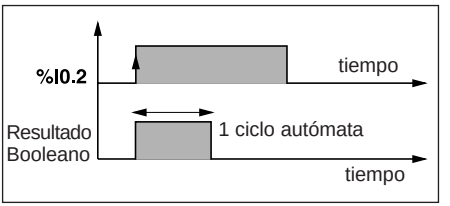
La instrucción LDF (F: Falling edge) equivale a un contacto de detección de flanco descendente:

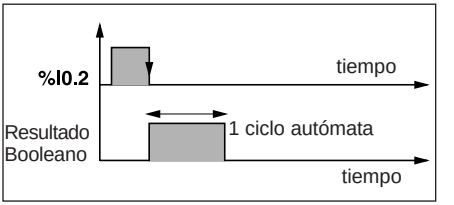
LDF %I0.0



(2)

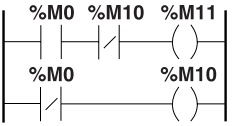
Flanco descendente: detección del paso de 1 a 0 de la entrada que lo controla.





Las instrucciones de flanco llevan a las entradas %I, pero es posible detectar flancos en otros bits (o resultados booleanos) utilizando 2 bits internos.
En el ejemplo, el bit %M11 memoriza el flanco ascendente del bit %M0.

LD %M0
ANDN %M10
ST %M11
LD %M0
ST %M10



(1) Contacto de detección de transición **P**ositiva
(2) Contacto de detección de transición **N**egativa
(3) Por rearranque en frío y caliente, la aplicación detectará un flanco ascendente incluso cuando la entrada permanezca en 1. Se puede enmascarar este fenómeno iniciando un programa con las instrucciones LD %S1 y ENDC.

B

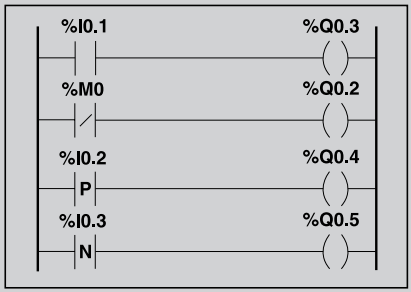
Descripción de instrucciones

La descripción de las instrucciones se realiza de la siguiente manera:

La instrucción booleana descrita estará sombreada. Cada ecuación está acompañada por su correspondiente esquema de contactos.

Instrucciones de carga LD, LDN, LDR, LDF

Las instrucciones LD, LDN, LDR y LDF corresponden respectivamente a los contactos abierto, cerrado de flanco ascendente y flanco descendente.




```
LD   %I0.1
ST   %Q0.3
LDN  %M0
ST   %Q0.2
LDR  %I0.2
ST   %Q0.4
LDF  %I0.3
ST   %Q0.5
```

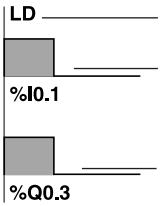
Códig.	Operando
LD	0/1,%I,%Q,%M,%S,%X,%BLK.x,%•:Xk,[
LDN	%I,%Q,%M,%S,%X,%BLK.x,%•:Xk,[
LDR	%I
LDF	%I

Lista de operandos

0/1	valor inmediato 0 ó 1
%I	entrada autómata %Ii.j
%Q	salida autómata %Qi.j
%M	bit interno %Mi
%S	bit de sistema %Si
%X	bit de etapa %Xi
%BLK.x	bit de bloque función, ej: %TMi.Q
%•:Xk	bit de palabra, ej: %MWi:Xk
[	Expresión de comparación ej: [%MWi<1000]

Cronograma

Se han agrupado los 4 cronogramas.



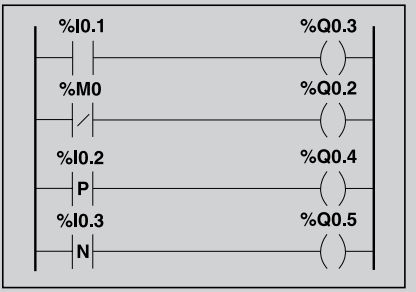
Cronograma de la instrucción LD

Estado de la entrada

Estado de la salida

2.1-3 Instrucciones de carga LD, LDN, LDR, LDF

Las instrucciones LD, LDN, LDR y LDF corresponden respectivamente a contactos abierto, cerrado, de flanco ascendente y de flanco descendente (LDR y LDF únicamente en entradas de autómata).

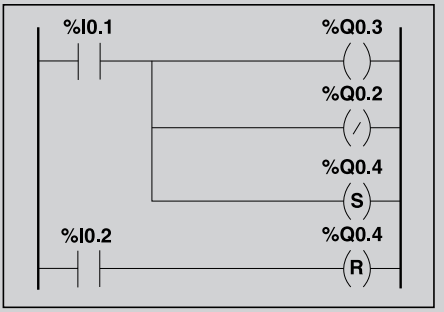



```
LD   %I0.1
ST   %Q0.3
LDN  %M0
ST   %Q0.2
LDR  %I0.2
ST   %Q0.4
LDF  %I0.3
ST   %Q0.5
```

Código	Operando
LD	0/1,%I,%Q,%M,%S,%X,%BLK.x,%•:Xk,[
LDN	%I,%Q,%M,%S,%X,%BLK.x,%•:Xk,[
LDR	%I
LDF	%I

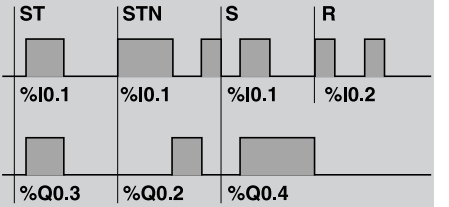
2.1-4 Instrucciones de asignación ST, STN, S, R

Las instrucciones ST, STN, S y R corresponden respectivamente a las bobinas directa, inversa, en la conexión y desconexión.



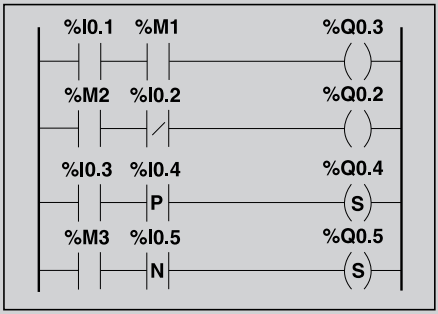

```
LD   %I0.1
ST   %Q0.3
STN  %Q0.2
S    %Q0.4

LD   %I0.2
R    %Q0.4
```

Código	Operando
ST	%Q,%M,%S,%BLK.x,%•:Xk
STN	%Q,%M,%S,%BLK.x,%•:Xk
S	%Q,%M,%S,%X,%BLK.x,%•:Xk
R	%Q,%M,%S,%X,%BLK.x,%•:Xk



2.1-5 Instrucciones Y Lógica: AND, ANDN, ANDR, ANDF

Estas instrucciones realizan un Y lógica entre el operando (o su inverso, o frente ascendente o frente descendente) y el resultado booleano de la instrucción anterior.

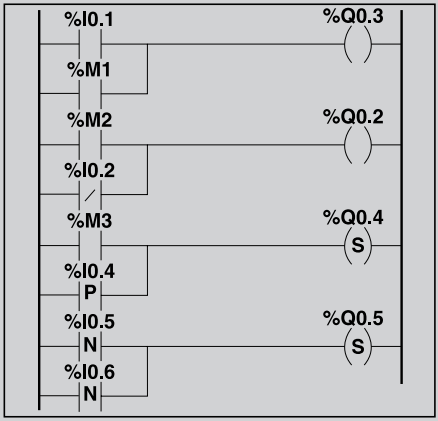



```
LD    %I0.1
AND   %M1
ST    %Q0.3
LD    %M2
ANDN  %I0.2
ST    %Q0.2
LD    %I0.3
ANDR  %I0.4
S     %Q0.4
LD    %M3
ANDF  %I0.5
S     %Q0.5
```

Código	Operando
AND	0/1,%I,%Q,%M,%S,%X,%BLK.x,%•:Xk
ANDN	%I,%Q,%M,%S,%X,%BLK.x,%•:Xk
ANDR	%I
ANDF	%I

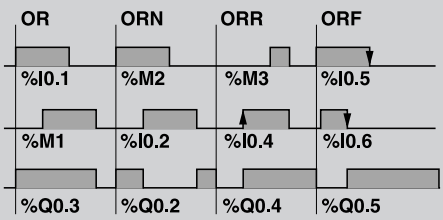
2.1-6 Instrucciones O Lógica: OR, ORN, ORR, ORF

Estas instrucciones realizan un O entre el operando (o su inverso, o frente ascendente, o frente descendente) y el resultado booleano de la instrucción anterior.



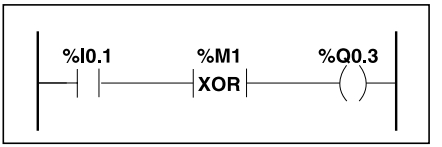

```
LD    %I0.1
OR    %M1
ST    %Q0.3
LD    %M2
ORN   %I0.2
ST    %Q0.2
LD    %M3
ORR   %I0.4
S     %Q0.4
LDF   %I0.5
ORF   %I0.6
S     %Q0.5
```

Código	Operando
OR	0/1, %I,%Q,%M,%S,%X,%BLK.x,%•:Xk
ORN	%I,%Q,%M,%S,%X,%BLK.x,%•:Xk
ORR	%I
ORF	%I

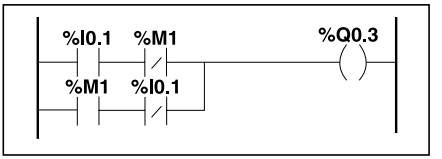


2.1-7 Instrucciones O Exclusiva: XOR, XORN, XORR, XORF

Estas instrucciones realizan un O exclusivo entre el operando (o su inverso, o frente ascendente, o frente descendente) y el resultado booleano de la instrucción anterior.

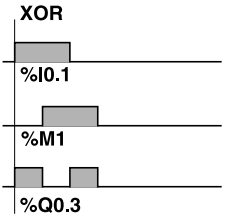



```
LD    %I0.1
XOR   %M1
ST    %Q0.3
```




```
LD    %I0.1
ANDN  %M1
OR(   %M1
ANDN  %I0.1
)
ST    %Q0.3
```

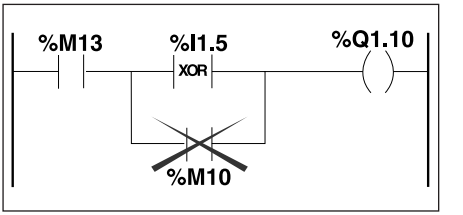
Código	Operando
XOR	%I,%Q,%M,%S,%X,%BLK.x,%•:Xk
XORN	%I,%Q,%M,%S,%X,%BLK.x,%•:Xk
XORR	%I
XORF	%I



- **Caso específico**

  En lenguaje de contactos, la instrucción XOR no deberá:

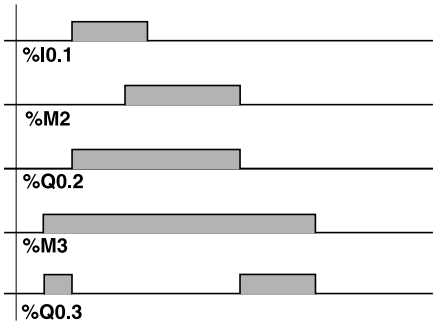
- situarse a la izquierda del circuito de contactos (primera posición),
- disponerse en paralelo.



2.1-8 Instrucción Negación: N

Esta instrucción realiza la negación del resultado de la instrucción anterior.

```
LD    %I0.1
OR    %M2
ST    %Q0.2
N
AND   %M3
ST    %Q0.3
```



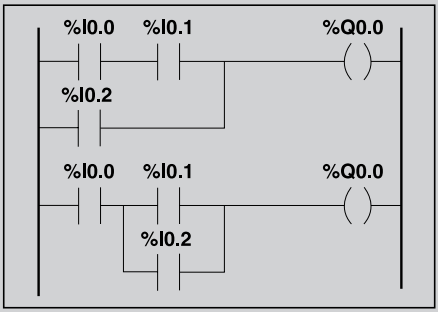
Código	Operando
N	-

Nota: la instrucción N no es reversible.

2.1-9 Utilización de paréntesis

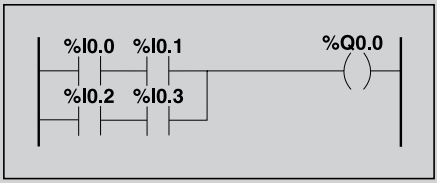
Las instrucciones AND y OR pueden utilizar paréntesis. Estas paréntesis permiten realizar esquemas de contactos de forma sencilla. El signo de abertura de paréntesis se asocia a la instrucción AND u OR. El paréntesis de cierre es una instrucción que es obligatoria para cada paréntesis abierta.

Ejemplo: AND(...)



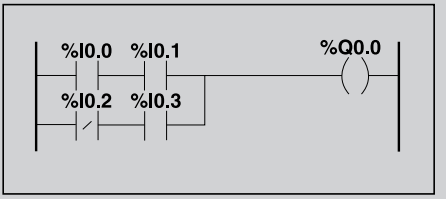

```
LD    %I0.0
AND   %I0.1
OR    %I0.2
ST    %Q0.0

LD    %I0.0
AND(  %I0.1
OR    %I0.2
)
ST    %Q0.0
```

Ejemplo: OR(...)




```
LD    %I0.0
AND   %I0.1
OR(   %I0.2
AND   %I0.3
)
ST    %Q0.0
```

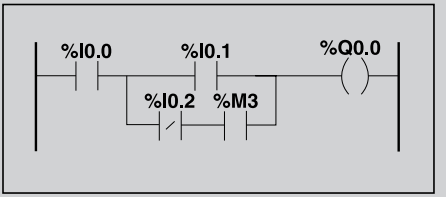
A los paréntesis pueden asociarse los modificadores N, F, R o [:
- N negación, ej: AND(N u OR(N
- F frente ascendente, ej: AND(F u OR(F
- R frente descendente, ej: AND(R u OR(R
- [comparación, véase el apartado 3.1-5




```
LD    %I0.0
AND   %I0.1
OR(N  %I0.2
AND   %I0.3
)
ST    %Q0.0
```

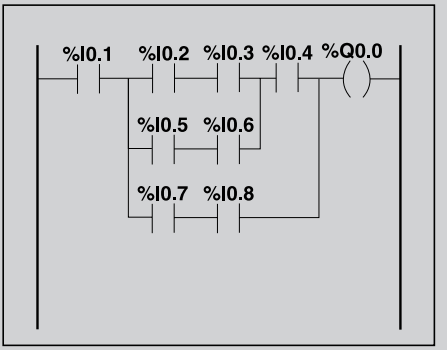
Imbricación de paréntesis
Se pueden imbricar hasta 8 niveles de paréntesis.

Ejemplo




```
LD    %I0.0
AND(  %I0.1
OR(N  %I0.2
AND   %M3
)
)
ST    %Q0.0
```

Ejemplo



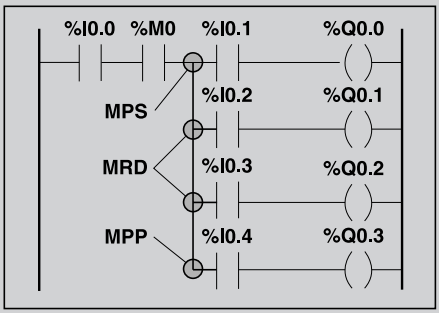

```
LD    %I0.1
AND(  %I0.2
AND   %I0.3
OR(   %I0.5
AND   %I0.6
)
AND   %I0.4
OR(   %I0.7
AND   %I0.8
)
)
ST    %Q0.0
```

Notas:
- cada paréntesis abierta deberá cerrarse obligatoriamente.
- las etiquetas %Li: y los subprogramas %SRi: no deberán colocarse en expresiones entre paréntesis, así como las instrucciones de salto JMP y de llamada a subprograma SRi,
- las instrucciones de asignación ST, STN, S y R no deberán programarse entre paréntesis.

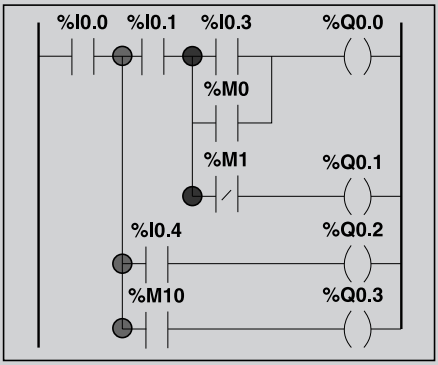
2.1-10 Instrucciones MPS, MRD, MPP

Los tres tipos de instrucciones permiten tratar las transferencias hacia las bobinas. Estas instrucciones utilizan una memoria intermedia llamada pila que puede almacenar hasta 8 informaciones booleanas.
La instrucción MPS almacena el resultado de la última instrucción de comprobación en la parte superior de la pila y desplaza los otros valores hacia el fondo de la pila.
La instrucción MRD lee la cima de la pila.
La instrucción MPP lee, desocupa la cima de la pila y desplaza los otros valores hacia la cima de la pila.

Ejemplos:




```
LD    %I0.0
AND   %M0
MPS
AND   %I0.1
ST    %Q0.0
MRD
AND   %I0.2
ST    %Q0.1
MRD
AND   %I0.3
ST    %Q0.2
MPP
AND   %I0.4
ST    %Q0.3
```




```
LD    %I0.0
MPS
AND   %I0.1
MPS
AND(  %I0.3
OR    %M0
)
ST    %Q0.0
MPP
ANDN  %M1
ST    %Q0.1
MRD
AND   %I0.4
ST    %Q0.2
MPP
AND   %M10
ST    %Q0.3
```

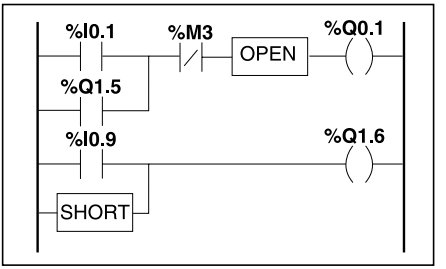
Nota: estas instrucciones no pueden utilizarse en una expresión entre paréntesis.

2.1-11 Instrucciones OPEN y SHORT específicas del lenguaje de contactos

Para facilitar la puesta a punto de un programa se pueden utilizar dos instrucciones específicas del lenguaje de contactos.

La instrucción OPEN interrumpe la continuidad (lógica 0) sea cual sea el resultado de la anterior operación lógica (equivalente en lenguaje Lista: instrucción AND 0).

La instrucción SHORT garantiza la continuidad (lógica 1) sea cual sea el resultado de la anterior operación lógica (equivalente en lenguaje Lista: OR 1).



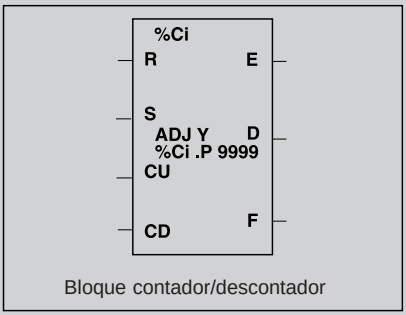

```
LD    %I0.1
OR    %Q1.5
ANDN  %M3
AND   0
ST    %Q0.1
LD    %I0.9
OR    1
ST    %Q1.6
```

2.2 Bloques de función estándares

2.2-1 Objetos de bits y palabras asociadas a bloques de función estándares

Los bloques de función aplican objetos de bits y palabras específicos.

- **Objetos de bits:**
  Corresponden a salidas de bloques. Se accede a estos bits mediante las instrucciones booleanas de comprobación.
  Pueden direccionarse:
  - directamente (ej.: LD E) si están cableados al bloque en programación reversible (véase apartado 2.2-2), o
  - especificando el tipo de bloque (ej.: LD %Ci.E),
  Las instrucciones permiten acceder a las entradas.



Bloque contador/descontador

- **Objetos de palabras:**
  Corresponden:
  - a parámetros de configuración de bloque; el programa permite acceder a estos parámetros (ej.: parámetro de preselección) o no (ej.: base de tiempo),
  - a valores actuales (ej.: %Ci.V valor de contaje en curso).

Lista de objetos bits y palabras de bloques de función accesibles por programa

Bloques función estándares	Palabras y bits asociados		Dirección	Acceso escritura	Ver Ap.
Temporizador %TMi (i = 0 a 31)	Palabra	Valor actual	%TMi.V	no	2.2-3
		Valor de preselección	%TMi.P	sí	
	Bit	Salida temporizador	%TMi.Q	no	
Contador/ descontador %Ci (i = 0 a 15)	Palabra	Valor actual	%Ci.V	no	2.2-4
		Valor de preselección	%Ci.P	sí	
	Bit	Salida desbordam.(vacío)	%Ci.E	no	
		Salida preselec. alcanzada	%Ci.D	no	
		Salida desbordam. (lleno)	%Ci.F	no	
Registro palabra %Ri (i = 0 a 3)	Palabra	Acceso al registro	%Ri.I	sí	2.2-5
		Salida del registro	%Ri.O	sí	
	Bit	Salida del registro lleno	%Ri.F	no	
		Salida del registro vacío	%Ri.E	no	
Programador cíclico %DRi (i = 0 a 3)	Palabra	Nº de paso en curso	%DRi.S	sí	2.2-6
	Bit	Último paso definido en curso		%DRi.F	no

2.2-2 Principios de programación

Los bloques de función estándar pueden programarse de 2 formas distintas:

- con instrucciones de bloque de función (ej.: BLK %TM2); esta forma reversible en lenguaje de contactos autoriza las operaciones en el bloque en un solo lugar del programa,
- con instrucciones específicas (ej.: CU %Ci); esta forma no reversible permite efectuar operaciones en las entradas del bloque en distintos lugares del programa (ej.: línea 100 CU %C1, línea 174 CD %C1, línea 209 LD %C1.D).

Principios de programación reversible de bloques de función estándares

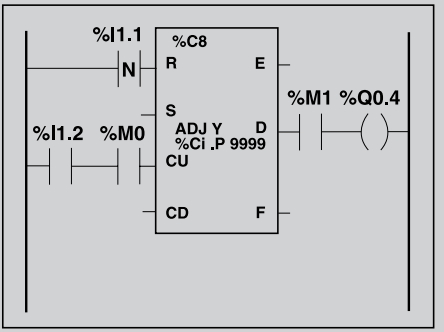
Este tipo de programación utiliza las instrucciones de bloque BLK, OUT_BLK y END_BLK.

BLK indica el inicio del bloque de función

OUT_BLK opcional, permite "cablear" directamente las salidas del bloque.

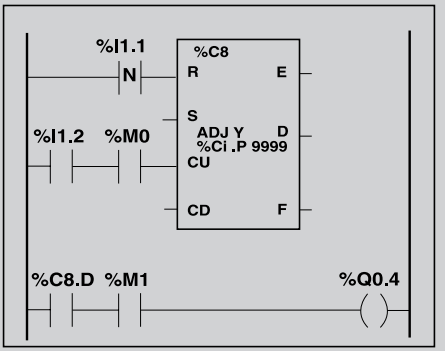
END_BLK indica el final del bloque.

Ejemplo reversible con salidas cableadas




```
BLK   %C8
LDF   %I1.1
R                    Tratamiento de
LD    %I1.2          las entradas
AND   %M0
CU
OUT_BLK
LD    D              Tratamiento de
AND   %M1            las salidas
ST    %Q0.4
END_BLK
```

Ejemplo reversible sin cableado de las salidas

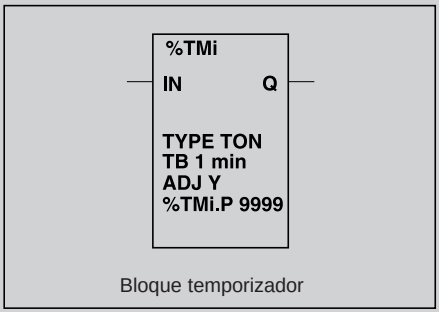



```
BLK   %C8
LDF   %I1.1
R                    Tratamiento de
LD    %I1.2          las entradas
AND   %M0
CU
END_BLK
.......
LD    %C8.D          Tratamiento de
AND   %M1            las salidas
ST    %Q0.4
```

Nota:

Sólo se autorizan las instrucciones de comprobación y de entradas en el bloque implicado entre las instrucciones BLK y OUT_BLK (o entre BLK y END_BLK, cuando OUT_BLK no se ha programado).

2.2-3 Bloques de función de temporizador %TMi



Bloque temporizador

Se proponen 3 tipos de temporizadores:

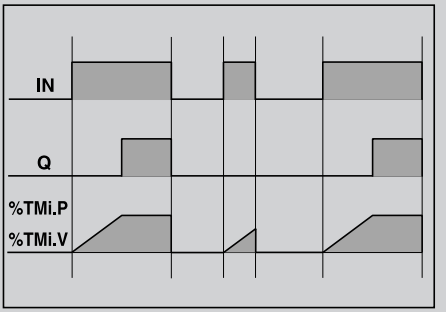
- **TON**: este tipo de temporizador permite gestionar retardos en la conexión. Este retardo es programable y puede ser modificado o no por el terminal.
- **TOF**: este tipo de temporizador permite gestionar los retardos de desconexión. Este retardo es programable y puede ser modificado o no por el terminal.
- **TP**: este tipo de temporizador permite elaborar un impulso de duración precisa. Esta duración es programable y puede ser modificado o no por el terminal.

Características

Número temporizador	**%TMi**	de 0 a 31
Tipo	**TON** **TOF** **TP**	• retardo en la conexión (por defecto) • retardo en la desconexión • monoestable
Base de tiempo	**BT**	1 mn (por defecto), 1 s, 100 ms, 10 ms, 1 ms (para TM0 y TM1). Cuanto más corta es la base de tiempo, mayor es la precisión del temporizador
Valor actual	**%TMi.V**	Palabra que crece de 0 a %TMi.P en el transcurso del temporizador. El programa puede leer y comprobarlo pero no escribirlo(1).
Valor de preselección	**%TMi.P**	0≤%TMi.P≤9999. Palabra que el programa puede leer, comprobar y escribir. Por defecto su valor es 9999. La duración o retardo elaborado es igual a %TMi.P x BT.
Ajuste por terminal	**O/N**	O: posibilidad de modificación del valor de preselección %TMi.P en modo de ajuste. N: sin acceso en modo de ajuste.
Entrada (o instrucción) "Activación"	**IN**	En flanco ascendente (tipo TON o TP) o flanco descendente (tipo TOF), arranca el temporizador.
Salida "Temporizador"	**Q**	Bit asociado %TMi.Q, su puesta a 1 depende de la función realizada TON, TOF O TP.

(1) El terminal en modo Ajuste puede modificar %TMi.V.

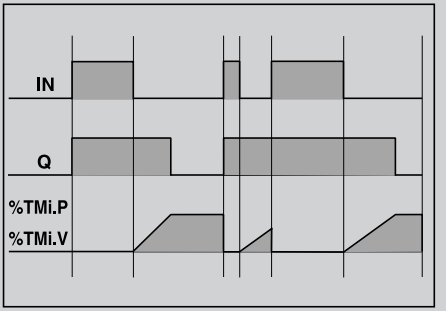
Utilización como temporizador de retardo en la conexión: tipo TON



Tras un flanco ascendente en la entrada IN (1), se activa el temporizador: su valor actual %TMi.V crece desde 0 hacia %TMi.P en una unidad a cada impulso de la base de tiempo BT. El bit de salida %TMi.Q pasa a 1 cuando el valor actual alcanza %TMi.P Luego se mantiene a 1 mientras no se detecte un flanco descendente en la entrada IN.

Al detectar un flanco descendente en la entrada IN (2), el temporizador se detendrá incluso si no ha alcanzado su valor de preselección %TMi.P.

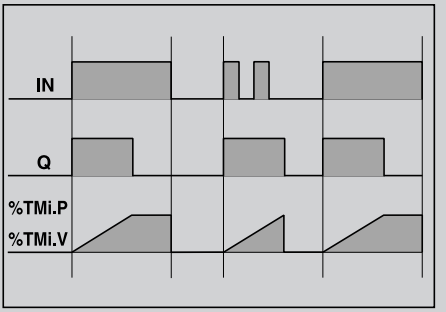
Utilización como temporizador de retardo en la desconexión: tipo TOF



El valor actual %TMi.V toma el valor 0, tras un flanco ascendente de la entrada IN (1) (aunque el temporizador esté en curso de evolución). El temporizador se activa en el flanco descendente de la entrada IN.

El valor crece hacia %TMi.P en una unidad a cada impulso de la base de tiempo BT. El bit de salida %TMi.Q pasa a 1 al detectar un flanco ascendente en la entrada IN y pasa de nuevo a 0 cuando el valor actual alcanza %TMi.P.

Utilización como monoestable: tipo TP



Tras un flanco ascendente en la entrada IN (1), se activa el temporizador actual (si el temporizador ya no se encuentra en curso de evolución).

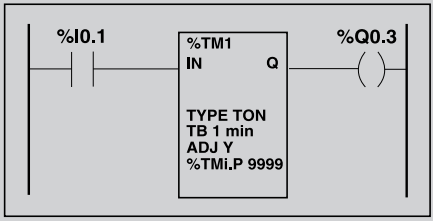
%TMi.V crece de 0 hacia %TMi.P en una unidad a cada impulso de la base de tiempo BT.El bit de salida %TMi.Q pasa a 1 cuando el temporizador se activa y pasa de nuevo a 0 cuando el valor actual alcanza %TMi.P.

Cuando el valor actual %TMi.V alcanza el valor de preselección %TMi.P, %TMi.V toma el valor 0 si la entrada IN está a 0. Este monoestable no puede reactivarse.

(1) O la activación de la instrucción IN.
(2) O la desactivación de la instrucción IN.

Programación y configuración

La programación de los bloques de función del temporizador es idéntica sea cual sea su tipo de utilización. La selección del funcionamiento TON, TOF o TP se efectúa en la configuración.



Programación reversible

```
BLK   %TM1
LD    %I0.1
IN
OUT_BLK
LD    Q
ST    %Q0.3
END_BLK
```

Configuración

Los parámetros siguientes deben completarse en la configuración.

- Tipo: TON, TOF o TP
- BT: 1 min, 1 s, 100 ms, 10 ms o 1 ms
- %TMi.P: 0 a 9999
- Ajuste: S o N

Programación no reversible

```
LD    %I0.1
IN    %TM1
LD    %TM1.Q
ST    %Q0.3
```

Casos específicos

- **Incidencia de un rearranque en frío:** (%S0=1) provoca la puesta a 0 del valor actual y la puesta a 0 de la salida %TMi.Q. El valor de preselección se reinicializa al valor definido en la configuración.
- **Incidencia de un rearranque en caliente:** (%S1=1) no tiene incidencia en el valor actual del temporizador, ni en el valor de preselección. El valor actual no aumenta durante el corte de alimentación.
- **Incidencia de un paso a STOP:** el paso a STOP del autómata no inmoviliza el valor actual.
- **Incidencia de un salto de programa:** el hecho de no explorar las instrucciones en que está programado el bloque del temporizador no inmoviliza el valor actual %TMi.V que continúa creciendo hacia %TMi.P. Incluso el bit %TMi.Q asociado a la salida Q del bloque temporizador sigue funcionando normalmente y puede ser comprobado por otra instrucción. En cambio, la salida directamente cableada a la salida del bloque no se activa, ya que el autómata no la explora.
- **Comprobación del bit %TMi.Q:** es aconsejable comprobar el bit %TMi.Q solamente una vez en el programa.
- **Incidencia de las instrucciones de los relés principales MCS/MCR:** un bloque de temporización programado entre 2 instrucciones MCS/MCR se reinicializa cuando la instrucción MCS está activa.
- **Incidencia de la modificación de la preselección %TMi.P:** la modificación del valor de preselección por instrucción o ajuste sólo se tiene en cuenta en la próxima activación del temporizador.

Temporizadores con base de tiempo de 1 ms (TSX 07 3• ••••)

La base de tiempo de 1 ms sólo está disponible en los temporizadores %TM0 y %TM1. Si el usuario los necesitase, podrá utilizar las cuatro palabras de sistema %SW76, %SW77, %SW78 y %SW79 como "relojes de arena".

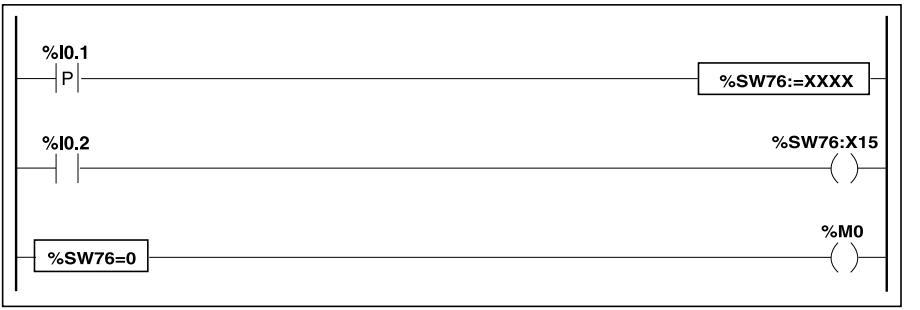
Si el valor es positivo, el sistema disminuirá individualmente estas cuatro palabras de sistema cada milisegundo.

Se puede realizar una temporización múltiple por carga sucesiva de una de estas palabras o comprobación de los valores intermedios.

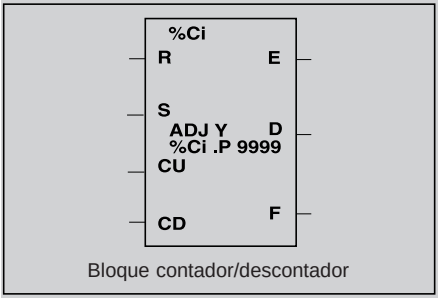
Si una de estas cuatro palabras es inferior a 0, ésta no se modificará. Por lo tanto, se puede "inmovilizar" un temporizador poniendo a 1 el bit 15 correspondiente y luego "movilizarlo" volviéndolo a poner a 0.

Ejemplo de programación:

```
LDR %I0.1          (inicio del temporizador en el flanco ascendente de %I0.1)
[%SW76:=XXXX]  (XXXX= valor seleccionado)
LD   %I0.2          (gestión opcional de inmovilización, la entrada I0.2 sirve de
inmovilización)
ST   %SW76:X15
LD   [%SW76=0]   (comprobación de fin del temporizador)
ST       %M0
.....
```



2.2-4 Bloques de función de contador/descontador %Ci



Bloque contador/descontador

El bloque de función de contador/descontador realiza el contaje o descontaje de eventos, estas dos operaciones pueden ser simultáneas.

Características

Número de contador	**%Ci**	de 0 a 15
Valor actual	**%Ci.V**	Palabra aumentada o disminuida en función de las entradas (o de las instrucciones) CU y CD. El programa puede leerla, comprobarla pero no escribirla (1).
Valor de preselección	**%Ci.P**	0≤%Ci.P≤9999. La palabra puede leerse, comprobarse y escribirse (valor 9999 por def.).
Ajuste por terminal	**O/N**	O: posibilidad de modificar el valor de preselección en modo Ajuste. N: no se puede acceder en modo Ajuste.
Entrada (o instrucción) reinicialización a 0	**R**	En estado 1: %Ci.V = 0.
Entrada (o instrucción) preselección	**S**	En estado 1: %Ci.V = %Ci.P.
Entrada (o instrucción) contaje	**CU**	Aumenta %Ci.V en flanco ascendente.
Entrada (o instrucción) descontaje	**CD**	Disminuye %Ci.V en flanco ascendente.
Salida desbordamiento	**E** (Empty)	El bit asociado %Ci.E=1, cuando el descontaje %Ci.V pasa de 0 a 9999 (puesta a 1 cuando %Ci.V es igual a 9999, y de nuevo a 0 si el contador sigue descontando).
Salida preselección alcanzada	**D** (Done)	El bit asociado %Ci.D=1, cuando %Ci.V=%Ci.P.
Salida desbordamiento	**F** (Full)	El bit asociado %Ci.F =1 cuando %Ci.V pasa de 9999 a 0 (puesta a 1 cuando %Ci.V es igual a 0, y de nuevo a 0 si el contador continúa contando).

(1) El terminal en modo Ajuste puede modificar %Ci.V.

B

Funcionamiento

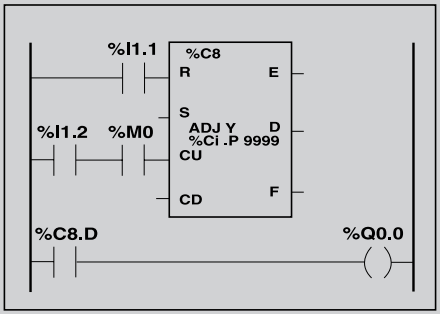
- **Contaje:** con la aparición de un flanco ascendente en la entrada de contaje CU (o activación de la instrucción CU), el valor actual aumenta en una unidad. Cuando este valor es igual al valor de preselección %Ci.P, el bit de salida %Ci.D "preselección alcanzada" asociado a la salida D pasa al estado 1. El bit de salida %Ci.F (desbordamiento del contaje) pasa al estado 1 cuando %Ci.V pasa de 9999 a 0; vuelve a cero si el contador sigue contando.
- **Descontaje:** con la aparición de un flanco ascendente en la entrada de "descontaje" CD (o activación de la instrucción CD), el valor actual %Ci.V disminuye en una unidad. El bit de salida %Ci.E (desbordamiento del contaje de decrementos) pasa al estado 1 cuando %Ci.V pasa de 0 a 9999; vuelve a 0 si el contador sigue descontando.
- **Contaje/descontaje:** Para utilizar de forma simultánea las funciones de contaje de incrementos y decrementos (o activar las instrucciones CD y CU), es necesario controlar las dos entradas correspondientes CU y CD; estas dos entradas se exploran sucesivamente. Si las dos entradas están a 1 simultáneamente, el valor actual no cambia (o si las 2 instrucciones se activan de forma simultánea).
- **Puesta a cero:** cuando se pone a 1 la entrada R (o se activa la instrucción), el valor actual %Ci.V se fuerza a 0, las salidas %Ci.E, %Ci.D y %Ci.F están a 0. La entrada "puesta a cero" es prioritaria.
- **Preselección:** si la entrada S "preselección" se encuentra en el estado 1 (o la instrucción S activa) y la entrada R "puesta a cero" en el estado 0 (o la instrucción R no activa), el valor actual %Ci.V toma el valor %Ci.P y la salida %Ci.D el valor 1.

Casos específicos

- **Incidencia de un rearranque en frío:** (%S0=1)
  - puesta a cero del valor actual %Ci.V.
  - puesta a 0 de los bits de salida %Ci.E, %Ci.D y %Ci.F.
  - la inicialización del valor de preselección por el definido en la configuración.
- **Incidencia de un rearranque en caliente (%S1=1), de un paso en STOP:** no tiene incidencia en el valor actual del contador (%Ci.V).
- **Incidencia de la modificación de la preselección %Ci.P:** la modificación del valor de preselección mediante una instrucción o el ajuste se valida cuando la aplicación gestiona el bloque (activación de una de las entradas).

Configuración y programación

Contaje de un número de piezas = 5000. Cada impulso en la entrada %I1.2 (cuando el bit interno %M0 está a 1) provoca el aumento del contador %C8 hasta el valor de preselección final %C8 (bit %C8.D=1). La entrada %I1.1 pone el contador a 0.



Configuración

Los parámetros que se deben introducir en configuración son:

- %Ci.P, fijado a 5000 en este ejemplo
- Ajuste: O

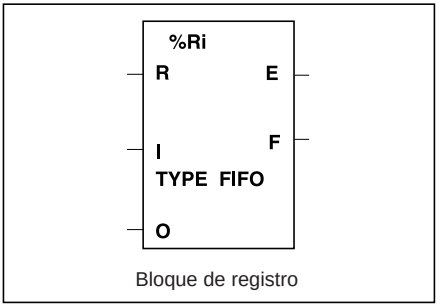
Programación reversible

```
BLK    %C8
LD     %I1.1
R
LD     %I1.2
AND    %M0
CU
END_BLK
LD     %C8.D
ST     %Q0.0
```

Programación no reversible

```
LD     %I1.1
R      %C8
LD     %I1.2
AND    %M0
CU     %C8
LD     %C8.D
ST     %Q0.0
```

2.2-5 Bloques de función de registro %Ri



Bloque de registro

Un registro es un bloque de memoria que permite almacenar hasta 16 palabras de 16 bits de dos maneras diferentes:

- fila de espera (primero en entrar, primero en salir) denominada pila FIFO (First In, First Out),
- pila (último en entrar, primero en salir) denominada pila LIFO (Last In, First Out).

Características

Número registro	**%Ri**	de 0 a 3
Tipo	**FIFO** **LIFO**	Fila de espera (selección por defecto). Pila
Palabra de entrada	**%Ri.I**	Palabra de entrada al registro. Puede leerse, comprobarse y escribirse.
Palabra de salida	**%Ri.O**	Palabra de salida del registro. Puede leerse, comprobarse y escribirse
Entrada (o instrucción) "Almacenamiento"	**I** (In)	En un flanco ascendente, almacena el contenido de la palabra %Ri.I en el registro.
Entrada (o instrucción) "Desalmacenamiento"	**O** (Out)	En un flanco ascendente, coloca una palabra de información en la palabra %Ri.O.
Entrada (o instrucción) "Puesta a cero"	**R** (Reset)	En el estado 1, inicializa el registro.
Salida "Vacío"	**E** (Empty)	El bit %Ri.E asociado indica que el registro está vacío. Puede comprobarse.
Salida "Lleno"	**F** (Full)	El bit %Ri.F asociado indica que el registro está lleno. Puede comprobarse.

Funcionamiento

FIFO (First In, First Out)

La primera información introducida es la primera en salir.
Cuando se tiene en cuenta una petición de entrada (flanco ascendente en la entrada I o activación de la instrucción I), el contenido de la palabra de entrada %Ri.I previamente cargada se almacena en el punto más alto de la pila (figura a).

Cuando la pila está llena (salida F=1), es imposible almacenar.

Cuando se tiene en cuenta una petición de salida (flanco ascendente en la entrada O o activación de la instrucción O), la palabra de información más baja de la pila se coloca en la palabra de salida %Ri.O y el contenido del registro se desplaza un paso hacia abajo (figura b).

Cuando el registro está vacío (salida E=1), es imposible desalmacenar; la palabra de salida %Ri.O ya no cambia y conserva su valor.

La pila puede reinicializarse en todo momento (estado 1 en la entrada R o activación de la instrucción R).

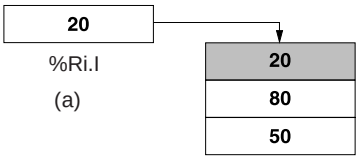
LIFO (Last In, First Out)

La última información introducida es la primera en salir.
Cuando se tiene en cuenta una petición de entrada (flanco ascendente en la entrada o activación de la instrucción I), el contenido de la palabra de entrada %Ri.I previamente cargada se almacena en el lugar más alto de la pila (figura c).

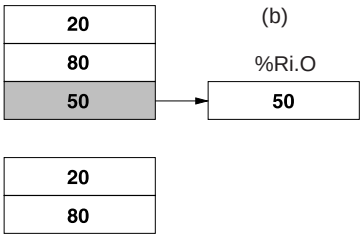
Cuando la pila está llena (salida F a 1), es imposible almacenar.
Cuando se tiene en cuenta una petición de salida (flanco ascendente en la entrada O o activación de la instrucción O), la palabra de información más alta (última información entrada) se coloca en la palabra %Ri.O (figura d).

Cuando el registro está vacío (salida E= 1), es imposible desalmacenar. La palabra de salida %Ri.O ya no cambia y conserva su último valor. La pila puede ser reinicializada en cualquier momento (estado 1 en la entrada R o activación de la instrucción R). El elemento indicado es el más alto de la pila.

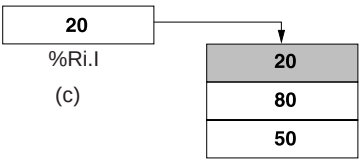
Ejemplo:
Almacenar el contenido de %Ri.I en el punto más alto de la pila.



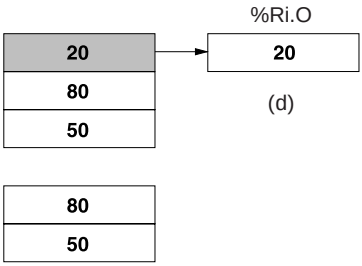
Desalmacenar la primera información y ubicarla en %Ri.O.



Ejemplo:
Almacenar el contenido de %Ri.I en el punto más alto de la pila.



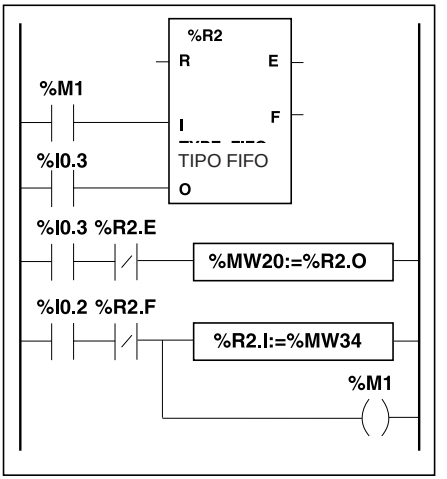
Desalmacenar la palabra de información del lugar más alto de la pila.



Casos específicos

- **Incidencia de un rearranque en frío:** (%S0=1) provoca la inicialización del contenido del registro. El bit de salida %Ri.E asociado a la salida E se pone a 1. Las palabras %Ri.I y de %Ri.O se ponen a 0.
- **Incidencia de un rearranque en caliente:** (%S1=1) no tiene incidencia en el contenido del registro ni en el estado de los bits de salida.

Programación y configuración



El ejemplo de programa muestra la carga de %R2.I para la palabra %MW34 en petición de la entrada %I0.2, si el registro R2 no está lleno (%R2.F=0). %M1 efectúa la petición de entrada en el registro. La petición de salida se realiza por la entrada %I0.3 y la ubicación de %R2.O en %MW20 se efectúa si el registro no está vacío (%R2.E=0).

Configuración

El único parámetro que se debe introducir en la configuración es el tipo de registro FIFO (por defecto) o LIFO.

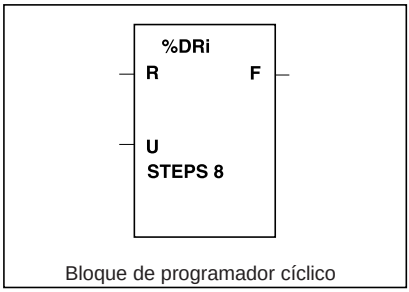
Programa reversible

```
BLK    %R2
LD     %M1
I
LD     %I0.3
O
END_BLK
LD     %I0.3
ANDN %R2.E
[%MW20:=%R2.O]
LD     %I0.2
ANDN %R2.F
[%R2.I:=%MW34]
ST     %M1
```

Programa no reversible

```
LD     %M1
I      %R2
LD     %I0.3
O      %R2
LD     %I0.3
ANDN %R2.E
[%MW20:=%R2.O]
LD     %I0.2
ANDN %R2.F
[%R2.I:=%MW34]
ST     %M1
```

2.2-6 Bloques de función de programador cíclico %DRi



Bloque de programador cíclico

Con un principio de funcionamiento similar al programador de levas, el programador cíclico cambia de paso en función de eventos exteriores. A cada paso, el punto alto de una leva da una orden ejecutada por el automatismo. En el caso de un programador cíclico, estos puntos altos se simbolizan por un estado 1 al nivel de cada paso y son asignados a bits de salida %Qi.j o a bits internos %Mi llamados bits de orden.

Características

Número	**%DRi**	0 a 3
Número del paso en curso	**%DRi.S**	0≤%DRi.S≤7. Palabra que puede leerse y comprobarse. Sólo puede escribirse en el programa a partir de un valor decimal inmediato.
Número de pasos		1 a 8 (por defecto)
Entrada (o instrucción) "regreso al paso 0"	**R** (RESET)	En el estado 1, inicializa el programador al paso 0.
Entrada (o instrucción) "avance"	**U** (UP)	En un flanco ascendente, avanza de un paso el programador y actualiza los bits de orden.
Salida	**F** (FULL)	Indica que el último paso definido está en curso. El bit %DRi.F asociado puede ser comprobado (%DRi.F=1 si %DRi.S=número de pasos configurados - 1).
Bits de orden		Salidas o bits internos asociados al paso (16 bits de orden).

B

Funcionamiento

El programador cíclico se compone de:

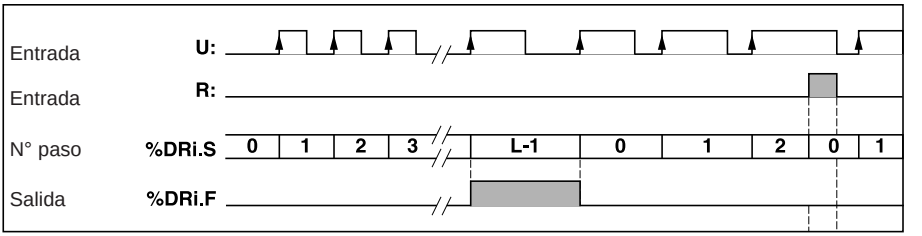
- una matriz de datos constantes (levas) organizada en 8 pasos de 0 a 7 y de 16 informaciones binarias (estados de paso) ordenadas en columnas y referenciadas de 0 a F.
- una lista de bits de orden (1 por columna) que corresponden a salidas %Q0.i o %Q1.i o a bits internos %Mi. En el paso en curso, los bits de orden toman los estados binarios definidos para este paso.

La tabla siguiente resume las características principales del programador cíclico.

Columna	0	1	2		D	E	F
Bits de orden	%Q0.1	%Q0.3	%Q1.5		%Q0.6	%Q0.5	%Q1.0
Paso 0	0	0	1		1	1	0
Paso 1	1	0	1		1	0	0
Paso 5	1	1	1		0	0	0
Paso 6	0	1	1		0	1	0
Paso 7	1	1	1		1	0	0

En el ejemplo anterior, como está en curso el paso 5, los bits de orden %Q0.1, %Q0.3 y %Q1.5 se ponen a 1; los bits de orden %Q0.6, %Q0.5 y %Q1.0 se ponen en 0.
El número del paso en curso aumenta en cada flanco ascendente de la entrada U (o activación de la instrucción U). Este número puede ser modificado por el programa.

Diagrama de funcionamiento

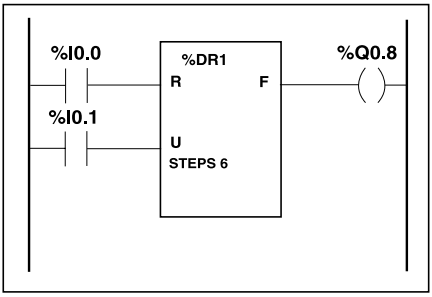


Casos específicos

- **Incidencia de un rearranque en frío:** (%S0=1) provoca la reinicialización del programador al paso 0 (con la actualización de los bits de orden).
- **Incidencia de un rearranque en caliente:** (%S1=1) provoca la actualización de los bits de orden, siguiendo el paso en curso.
- **Incidencia de un salto de programa:** el hecho de no explorar el programador cíclico no provoca la puesta a cero de los bits de orden.
- **Actualización de los bits de orden:** sólo se efectúa en un cambio de paso o en un rearranque en caliente o en frío.
- **Incidencia de las instrucciones de relé maestro MCS/MCR:** cuando se utiliza un programador cíclico entre dos instrucciones MCS/MCR, los bits de orden se ponen a 0 si la condición que precede a MCS es 0.

Programación y configuración

En este ejemplo, las 5 primeras salidas %Q0.0 a %Q0.4 se activan una tras otra cada vez que la entrada %I0.1 se pone a 1.
La entrada I0.0 reinicializa las salidas a 0.



Programación reversible

```
BLK   %DR1
LD    %I0.0
R
LD    %I0.1
U
OUT_BLK
LD    F
ST    %Q0.8
END_BLK
```

Configuración
La información siguiente se define en la configuración:
• número de pasos: 5

• los estados de las salidas (bits de orden) para cada paso del programador

	Q 0.	0	1	2	3	4
Paso 0:		0	0	0	0	0
Paso 1:		1	0	0	0	0
Paso 2:		0	1	0	0	0
Paso 3:		0	0	1	0	0
Paso 4:		0	0	0	1	0
Paso 5:		0	0	0	0	1

• asignación de los bit de orden
0: %Q0.0 1: %Q0.1
2: %Q0.2 3: %Q0.3
4: %Q0.4

Programación no reversible

```
LD    %I0.0
R     %DR1
LD    %I0.1
U     %DR1
LD    %DR1.F
ST    %Q0.8
```

2.3 Instrucciones Grafcet

2.3-1 Descripción

B

Las instrucciones Grafcet del lenguaje PL7 permiten traducir un Grafcet gráfico de forma simple.
El lenguaje PL7 comprende un máximo de 62 etapas incluyendo la o las etapas iniciales. El número de etapas activas simultáneas sólo está limitado por el número de etapas.
El cuadro siguiente agrupa todas las instrucciones y objetos que permiten programar un Grafcet.

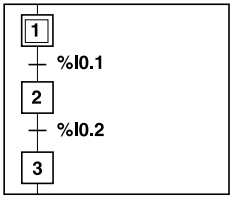
Representación gráfica	Transcripción en lenguaje PL7	Función
 Etapa inicial Transición Etapa	**=*= i**	Comienzo de la etapa inicial (1)
	# i	Activación de la etapa i después de desactivación de la etapa en curso
	-*- i	Comienzo de etapa y validación de la transición asociada (1)
	#	Desactivación de la etapa en curso sin activación de otra etapa
	#Di	Desactivación de la etapa en curso y de la etapa i especificada
	=*= POST	Comienzo del tratamiento posterior y fin del tratamiento secuencial
	%Xi	El bit asociado a la etapa i, puede comprobarse en cualquier lugar del programa pero sólo puede escribirse en el tratamiento preliminar (número máx. de etapas: 62).
 **Xi**	LD %Xi,LDN %Xi, AND %Xi, ANDN %Xi OR %Xi, ORN %Xi XOR %Xi, XORN %Xi	Comprob. de la actividad de la etapa i
 **Xi**	S **%Xi**	Activación de la etapa i
 **Xi**	R **%Xi**	Desactivación de la etapa i

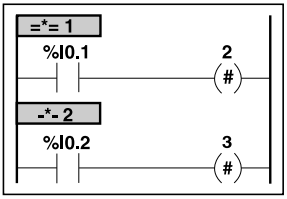
(1) La primera etapa =*=i ó -*-i escrita indica el comienzo del tratamiento secuencial; es decir el fin del tratamiento preliminar.

B

Ejemplos

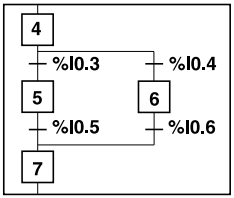
Secuencia lineal

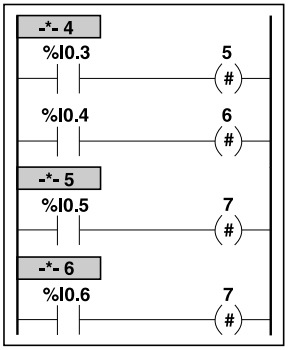





```
=*=1
LD %I0.1
#  2
-*- 2
LD %I0.2
#  3
```

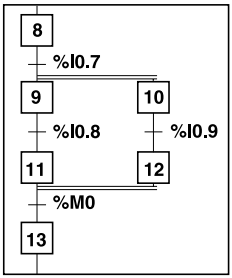
Derivación



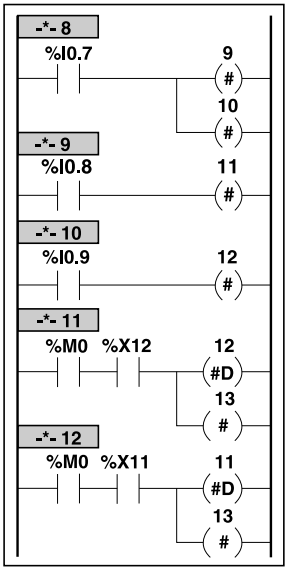



```
-*- 4
LD %I0.3
#  5
LD %I0.4
#  6
-*- 5
LD %I0.5
#  7
-*- 6
LD %I0.6
#  7
```

Secuencias simultáneas



Nota :
para que un Grafcet sea operacional, es necesario declarar al mínimo una etapa inicial =*=i o posicionar previamente el gráfico en el tratamiento preliminar mediante el bit de sistema %S23. (véase el anexo A.10 Sección G)




```
-*- 8
LD %I0.7
#  9
#  10
-*- 9
LD %I0.8
#  11
-*- 10
LD %I0.9
#  12
-*- 11
LD %M0
AND %X12
#D 12
#  13
-*- 12
LD %M0
AND %X11
#D 11
#  13
```

2.3-2 Estructura de un programa

Un programa Grafcet PL7 consta de 3 partes, cada una con una función específica.

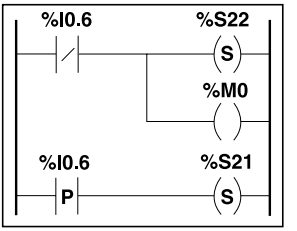
Tratamiento	Ejemplo

Tratamiento preliminar

Consta de instrucciones que aseguran el tratamiento de:

- restablecimiento alimentación
- fallas
- cambios de modos
- ubicación previa del gráfico
- lógicas de entradas

Finaliza con la primera instrucción =*= o -*- encontrada.



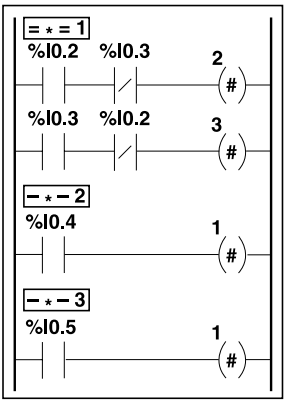

```
000 LDN   %I0.6
001 S     %S22
002 ST    %M0
003 LDR   %I0.6
004 S     %S21
```

Tratamiento secuencial

Formado por el gráfico (instrucciones que representan el gráfico):

- etapas
- acciones que se asocian con la etapa (véase el anexo A.11, sección G)
- transiciones
- receptividades

Finaliza con la ejecución de la instrucción =*=POST.

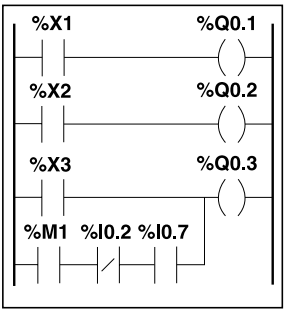



```
005 =*=   1
006 LD    %I0.2
007 ANDN  %I0.3
008 #     2
009 LD    %I0.3
009 LD    %I0.3
010 ANDN  %I0.2
011 #     3
012 -*-   2
013 LD    %I0.4
014 #     1
015 -*-   3
016 LD    %I0.5
017 #     1
```

Tratamiento posterior

Formado por un grupo de instrucciones que aseguran el tratamiento:

- ordenes del tratamiento secuencial para el control de las salidas,
- seguridades indirectas especiales para las salidas.




```
018 =*=   POST
019 LD    %X1
020 ST    %Q0.1
021 LD    %X2
022 ST    %Q0.2
023 LD    %X3
024 OR(   %M1
025 ANDN  %I0.2
026 AND   %I0.7
027 )
028 ST    %Q0.3
```

Observación:

El ciclo de exploración es el definido en el apartado 1.3, sección A. En el tratamiento secuencial, sólo se ejecutan las etapas activas al inicio del ciclo y las instrucciones asociadas.

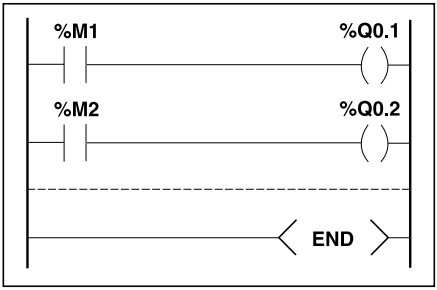
2.4 Instrucciones de programa

2.4-1 Instrucciones de fin de programa END, ENDC, ENDCN

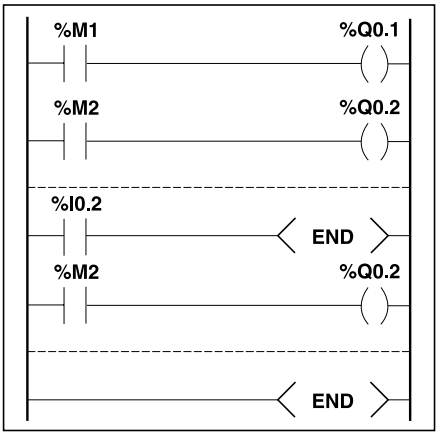
Las instrucciones END, ENDC y ENDCN permiten definir el fin de ejecución del ciclo del programa:
END: fin de programa incondicional.
ENDC: fin de programa si el resultado booleano de la instrucción de comprobación precedente es 1.
ENDCN: fin de programa si el resultado booleano de la instrucción de comprobación precedente es 0.

Por defecto (modo normal), cuando está activado el fin de programa, se actualizan las salidas y se pasa al ciclo siguiente.
Si la exploración es periódica, se espera el fin de período, la actualización de las salidas y el paso al ciclo siguiente.

Ejemplo:




```
LD  %M1
ST  %Q0.1
LD  %M2
ST  %Q0.2
.....
END
```




```
LD  %M1
ST  %Q0.1
LD  %M2      → Si %I0.2 =1, fin de
ST  %Q0.2      exploración del
.....          programa
LD  %I0.2    Si %I0.2 =0, continúa
ENDC           la exploración del
LD  %M2        programa hasta la
ST  %Q0.2      siguiente instrucción
.....          END.
END
```

2.4-2 Instrucción NOP

La instrucción **NOP** no efectúa ninguna acción. Permite reservar filas en un programa y así escribir instrucciones sin modificar los números de fila.

B

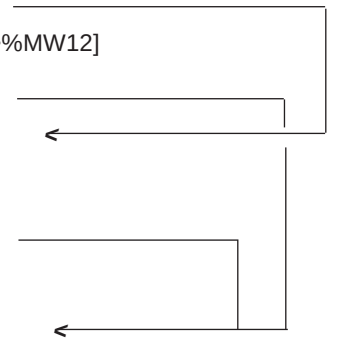
2.4-3 Instrucciones de salto JMP, JMPC, JMPCN a una etiqueta %Li:

Las instrucciones JMP, JMPC y JMPCN provocan la interrupción inmediata de la ejecución y la continuación del programa a partir de la línea de programa que comporta la etiqueta %Li: (i = 0 a 15).

JMP: salto de programa incondicional
JMPC: salto de programa si el resultado booleano de la instrucción de comprobación precedente es 1.
JMPCN: salto de programa si el resultado booleano de la instrucción de comprobación precedente es 0.

Ejemplos:

```
000   LD     %M15
001   JMPC   %L8
002   LD     [%MW24>%MW12]
003   ST     %M15
004   JMP    %L12
005   %L8:
006   LD     %M12
007   AND    %M13
008   ST     %M2
009   JMPCN  %L12
010   OR     %M11
011   S      %Q0.0
012   %L12:
013   LD     %I0.0
.....
```



Salto a la etiqueta %L8: si %M15 está a 1

Salto incondicional a la etiqueta %L12:

Salto a la etiqueta %L12: si %M2 está a 0

Notas:

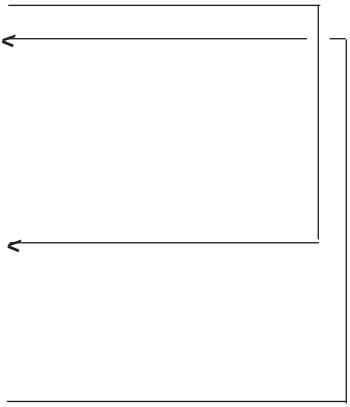
- esta instrucción está prohibida entre paréntesis, por tanto no debe figurar entre las instrucciones AND(, OR(y una instrucción de cierre de paréntesis ")".
- la etiqueta sólo puede figurar delante una instrucción LD, LDN, LDR, LDF o BLK.
- el número i de una etiqueta %Li puede declararse una sola vez en un programa.
- el salto de programa se efectúa hacia una línea de programación anterior o posterior. En el caso de un salto arriba, es necesario prestar atención al tiempo de ejecución del programa: se alarga y puede implicar el desbordamiento del período o del ciclo de autómata, lo que activará el control de secuencia.

2.4-4 Instrucciones de subprograma SRn, SRn:, RET

La instrucción **SRn** efectúa una llamada al subprograma referenciado por la etiqueta **SRn:** si el resultado de la instrucción booleana precedente es 1.
La instrucción **RET** colocada al final del subprograma controla el retorno al programa principal.
El subprograma está referenciado por una etiqueta SRn: con n = 0 a 15.

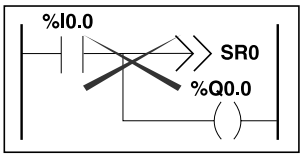
Ejemplo:

```
000   LD     %M15
001   AND    %M5
002   ST     %Q0.0
003   LD      [%MW24>%MW12]
004   SR8
005   LD     %I0.4
006   AND    %M13
007   .
008   .
009   .
010   END

011   SR8:
012   LD  1
013   IN     %TM0
014   LD     %TM0.Q
015   ST     %M10
016   RET
.....
```

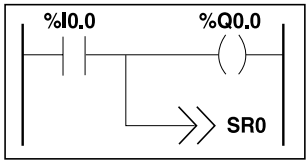


Notas:

- un subprograma no debe llamar a otro subprograma.
- esta instrucción está prohibida entre paréntesis, por tanto no debe figurar entre las instrucciones AND(, OR(y una instrucción de cierre de paréntesis ")".
- la etiqueta sólo debe figurar delante de una instrucción LD o BLK que marca el comienzo de una ecuación o de una red de contactos.
- una llamada de subprograma no debe preceder a una instrucción de asignación, por ejemplo:



~~LD %I0.0~~
~~SR0~~
~~ST %Q0.0~~



LD %I0.0
ST %Q0.0
SR0

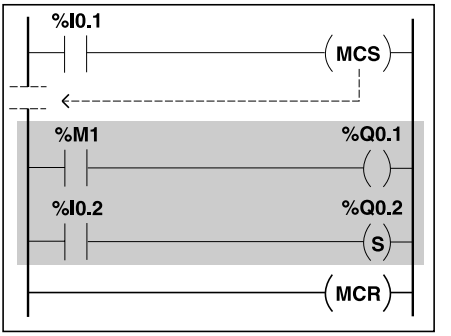
2.4-5 Instrucciones relé maestro MCS y MCR

Cuando el resultado booleano de la instrucción que precede a la MCS es 0, la ejecución de las líneas de programa que siguen a esta instrucción se modifican según la tabla siguiente hasta que se ejecute la instrucción MCR (no condicional).

Instrucciones/bloques	Comportamiento
ST, STN	objeto asociado puesto a 0
S, R	instrucciones no ejecutadas
SRi, JMP, JMPC, JMPCN	no ejecutadas
%TMi	reinicializado
%DRi	bits de orden puesto a 0
%FC	contador inmovilizado y salidas directas a 0
%PWM, %PLS	paro de generación de señales de salida
Otros bloques de función	no ejecutados (conservado en el estado)
Bloques de operaciones	no ejecutados

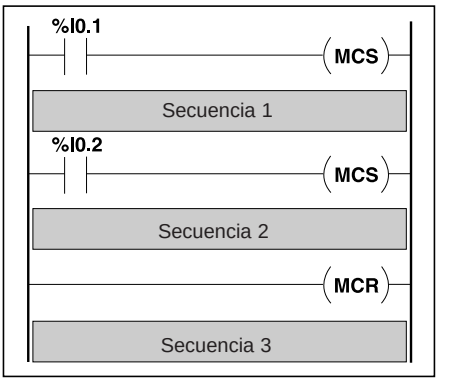
Ejemplo:

```
.....
002   LD    %I0.1
003   MCS
004   LD    %M1
005   ST    %Q0.1
006   LD    %I0.2
007   S     %Q0.2
008   MCR
.....
```



Cuando %I0.1 está a 0, la instrucción MCS se activa, %Q0.1 se fuerza a 0 y la salida %Q0.2 se conserva.

Se pueden utilizar varias instrucciones MCS para una sola instrucción MCR. Todas las instrucciones MCS se desactivan con una sola instrucción MCR.
Cuando %I0.1 está a 0, las secuencias 1 y 2 se modifican y la secuencia 3 se ejecuta de forma normal.
Cuando %I0.1 está a 1 y %I0.2 a 0, la secuencia 2 se modifica y las secuencias 1 y 3 se ejecutan de forma normal.



Si no se programa ninguna instrucción MCR después de una instrucción MCS, la instrucción MCS es efectiva hasta la instrucción END o hasta el final del programa.

Importante

La utilización de las instrucciones MCS y MCR no está permitida en los subprogramas, las receptividades y las acciones Grafcet.

B

3.1 Tratamiento numérico

3.1-1 Definición de los principales objetos de palabra

Los objetos de palabra, situados en la memoria de datos, se direccionan bajo el formato palabra de longitud 16 bits. Contienen un valor algebraico comprendido entre -32768 y 32767 (salvo el contador rápido que evoluciona entre 0 y 65535).

Valores inmediatos
Son valores algebraicos de formato homogéneo al de las palabras de 16 bits, que permiten asignar valores a dichas palabras. Se almacenan en la memoria programa y están comprendidas entre -32768 y 32767.

Formato de las palabras
Se almacena el contenido de las palabras o valores en la memoria de usuario en código binario, en 16 bits, según la convención ilustrada a continuación:

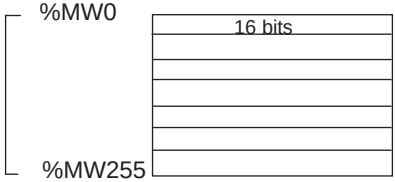
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	**Rango de los bits**
0	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1	**Estado de los bits**
+	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	**Significado de los bits**

En binario con signo, el bit de rango "F" se atribuye según la convención al signo del valor codificado:
- bit "F" a 0: el contenido de la palabra es un valor positivo,
- bit "F" a 1: el contenido de la palabra es un valor negativo (los valores negativos se expresan en lógica complemento a 2).

Las palabras y valores inmediatos pueden ser introducidos o restituidos bajo la forma:
- **decimal** 1579 (máximo: 32767, mínimo:-32768)
- **hexadecimal** 16#A536 (máximo: 16#FFFF, mínimo: 16#0000)
  otra sintaxis posible: #A536.

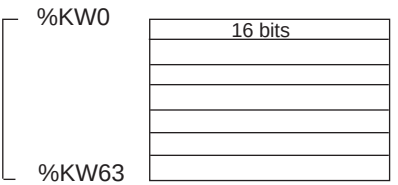
Palabras internas
Las palabras internas están destinadas a almacenar valores.

Se accede a las palabras %MW0 a %MW255 directamente desde el programa (en lectura/escritura). Se utilizan como palabras de trabajo.



Palabras constantes

Las palabras constantes memorizan valores constantes o mensajes alfanuméricos. Su contenido sólo puede escribirse o modificarse mediante el terminal (en modo CONFIGURACIÓN). Estas palabras se almacenan en la memoria de programa. Se accede a las palabras constantes %KW0 a %KW63 directamente desde el programa (sólo en lectura).



Palabras de intercambio de entradas/salidas

Las palabras de intercambio %IW/QW están asociadas a los autómatas conectados al cable de extensión. Permiten los intercambios entre los autómatas (véase el apartado 3-5).

Palabras de sistema

Estas palabras de 16 bits aseguran varias funciones: dan acceso a informaciones que provienen directamente del autómata mediante la lectura de las palabras %SWi (ej: valores de los puntos de ajuste analógico) y permiten actuar sobre la aplicación (ej: ajuste del reloj-calendario). El capítulo 6 explica la función de cada palabra.

Objetos de bits extraídos de palabras

Es posible extraer de una palabra uno de sus16 bits. La referencia de la palabra se completa entonces por medio del rango del bit extraído separado por dos puntos.
Sintaxis: % Objeto Palabra:Xk con k = 0 a 15 rango del bit del objeto palabra.
Ejemplo: %MW5:X6 bit de rango 6 de la palabra interna %MW5.

Lista de los operandos palabras

Tipo	Dirección (o valor)	Número máximo	Acceso en escritura	Ver Apar.
Valores inmediatos • base 10 • base 16	 ej: 2103 ej: 16#AF0D		no	
Palabras internas	%MWi	256	sí	-
Palabras constantes	%KWi	64	no (1)	-
Palabras de sistema	%SWi	128	según i	5.2
Palabras de bloques de función	%TMi.P %Ci.P...			2.2-1 3.3
Pal. de intercambio de entradas de salida	 %IWi.j %QWi.j	 8 8	 no sí	3.5
Bits extraídos de pal. • internas • sistema • constantes • de entrada • de salida	 %MWi:Xk %SWi:Xk %KWi:Xk %IWi.j:Xk %QWi.j:Xk	 256 x 16 128 x 16 64 x 16 8 x 16 8 x 16	 sí según i no no sí	

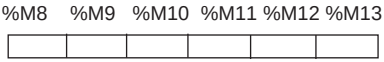
(1) La introducción de las palabras constantes se efectúa en el modo de configuración.

3.1-2 Objetos estructurados

Cadenas de bits

Las cadenas de bits son series de objetos de bits adyacentes del mismo tipo y de longitud definida: L.
Ejemplo de cadena de bits:

%M8:6 (1)



Tipo	Dirección	Tamaño máximo	Acceso en escritura
Bits de entradas TON	%I0:L o %I1:L	0<L<17	No
Bits de salidas TON	%Q0:L o %Q1:L	0<L<17	Sí
Bits de sistema	%Si:L con i múltiplo de 8	0<L<17 y i+L≤128	Según i
Bits de etapas	%Xi:L con i múltiplo de 8	0<L<17 y i+L≤63	Sí (por programa)
Bits internos	%Mi:L con i múltiplo de 8	0<L<17 y i+L≤128	Sí

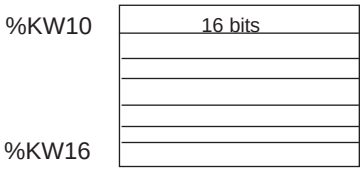
La instrucción de asignación:= permite explotar las cadenas de bits (véase las instrucciones de asignación en el apartado 3.1-4).

Tablas de palabras

Las tablas de palabras son series de palabras adyacentes del mismo tipo y de longitud definida: L.

Ejemplo de tablas de palabras:

%KW10:7



Tipo	Dirección	Tamaño máximo	Acceso en escritura
Palabras internas	%MWi:L	0<L<256 y i+L≤256	Sí
Palabras constantes	%KWi:L	0<L y i+L≤64	No
Palabras de sistema	%SWi:L	0<L y i+L≤128	Según i

La instrucción de asignación:= permite explotar las tablas de palabras (véase las instrucciones de asignación en el apartado 3.1-4).

(1) %M8:6 es correcto (8 es un múltiplo de 8)
%M10:16 es incorrecto (10 no es un múltiplo de 8)

Palabras indexadas

• Direccionamiento directo

Llamamos directo al direccionamiento de objetos, cuando la dirección de dichos objetos está fijada y definida en la escritura del programa.

Ejemplo: %MW26 (palabra interna de dirección 26)

• Direccionamiento indexado

En direccionamiento indexado, un índice completa la dirección directa del objeto: a la dirección del objeto se le añade el contenido del índice. El índice se define por una palabra interna %MWi. El número de "palabras índice" no está limitado.

Ejemplo: %MW108[%MW2]: palabra de dirección directa 108 + contenido de la palabra %MW2. Si la palabra %MW2 tiene por contenido el valor 12, escribir %MW108[%MW2] equivale por lo tanto a escribir %MW120.

Tipo	Dirección	Tamaño máximo	Acceso en escritura
Palabras internas	%MWi[%MWj]	$0 \leq i+\%MWj < 256$	Sí
Palabras constantes	%KWi[%MWj]	$0 \leq i+\%MWj < 64$	No

Las palabras indexadas se explotan mediante la instrucción de asignación:= (véase el apartado 3.1-4) y en las instrucciones de comparación (véase el apartado 3.1-5).

Este tipo de direccionamiento permite recorrer sucesivamente una serie de objetos de la misma naturaleza (palabras internas, palabras constantes...), modificando mediante el programa el contenido de la palabra índice.

• Desbordamiento de índice, bit de sistema %S20

Se produce un desbordamiento de índice cuando la dirección de un objeto indexado excede los límites del área que incluye este mismo tipo de objeto; es decir cuando:
- dirección de objeto + contenido del índice es inferior al valor cero,
- dirección de objeto + contenido del índice es superior a 255 (para las palabras %MWi) o 63 (para las palabras %KWi).

En caso de desbordamiento de índice, el sistema provoca la puesta del bit sistema %S20 al estado 1 y la asignación del objeto se efectúa con un valor de índice igual a 0.

El usuario está encargado de la supervisión del desbordamiento: el programa de usuario deberá leer el bit %S20 para un tratamiento eventual. El usuario deberá ponerlo a cero.

%S20 (estado inicial = 0):
- por desbordamiento de índice: puesta a 1 por el sistema,
- confirmación del desbordamiento: puesta a 0 por el usuario, después de la modificación del índice.

B

3.1-3 Presentación de instrucciones numéricas

Las instrucciones numéricas afectan de una manera general a las palabras de 16 bits (véase el apartado 3.1-1). Se colocan entre corchetes. Se ejecutan si el resultado booleano de la instrucción de comprobación que precede a la instrucción numérica es 1.

Observación: la visualización de las instrucciones numéricas se efectúa en 2 ó 3 filas en el terminal FTX117.

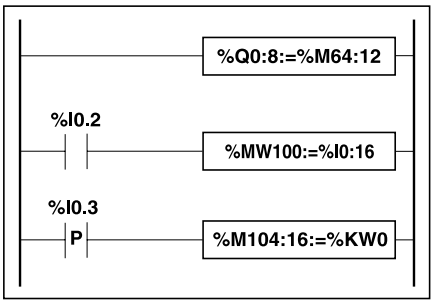
3.1-4 Instrucciones de asignación

Realizan la carga de un operando Op2 en un operando Op1

Sintaxis: [Op1:=Op2] <=> Op2->Op1

Las operaciones de asignación pueden realizarse en:
- cadenas de bits
- palabras
- tablas de palabras

Asignación de cadenas de bits (véase el apartado 3.1-2 Objeto de cadena de bits)
Se pueden realizar las operaciones siguientes en cadenas de bits:
cadena de bits -> cadena de bits (ej. 1)
cadena de bits -> palabra (ej. 2)
palabra -> cadena de bits (ej. 3)
valor inmediato -> cadena de bits
Ejemplos




```
LD    1
[%Q0:8:= %M64:12]          (ej 1)

LD    %I0.2
[%MW100:= %I0:16 ]         (ej 2)

LDR   %I0.3
[%M104:16:=%KW0]           (ej 3)
```

Normas de utilización
- Caso de una asignación de cadena de bits -> palabra: los bits de la cadena se transfieren a la palabra a partir de la derecha (primer bit de la cadena en el bit 0 de la palabra); los bits de la palabra no implicados por la transferencia (longitud<16) se posicionan a 0.
- Caso de una asignación palabra -> cadena de bits: los bits de la palabra se transfieren a partir de la derecha (el bit 0 de la palabra en el primer bit de la cadena).

B

Sintaxis

Operador	Sintaxis	Operando 1 (OP1)	Operando 2 (OP2)
:=	[Op1: = Op2] El operando 1 (Op1) toma el valor del operando 2 (Op 2)	%MWi,%QWi, %SWi %MWi[MWi], **%Mi:L,%Qi:L,%Si:L,** **%Xi:L**	Valor inmedianto, %MWi, %KWi,%IW,%QW,%SWi, %BLK.x,%MWi[MWi], %KWi[MWi]**, %Mi:L,%Qi:L,** **%Si:L,%Xi:L, %Ii:L**

Nota: la abreviación %BLK.x (por ej. %C0.P) se utiliza para designar cualquier palabra del bloque de función.

Asignación de palabras

Pueden realizarse las siguientes operaciones de asignación en las palabras:

palabra -> palabra (ej. 1)
palabra -> palabra indexada
palabra indexada -> palabra
palabra indexada -> palabra indexada (ej. 2)
valor inmediato -> palabra (ej. 3)
valor inmediato -> palabra indexada
cadena de bits -> palabra
palabra -> cadena de bits

Ejemplos


```
LD    1
[%SW112 := %MW100]          (ej. 1)

LD    %I0.2
[%MW0[%MW10] :=
      %KW0[%MW20] ]         (ej. 2)

LDR   %I0.3
[%MW10:=100]                (ej. 3)
```

Sintaxis

Operador	Sintaxis	Operando 1 (Op1)	Operando 2 (Op2)
:=	[Op1: = Op2] El operando 1 (Op1) toma el valor del operando 2 (Op 2)	**%MWi,%QWi, %SWi** **%MWi[MWi],** %Mi:L,%Qi:L,%Si:L, %Xi:L	**Valor inmediato, %MWi**, **%KWi, %IW, %QW, %SWi,** **%BLK.x, %MWi[MWi],** **%KWi[MWi]**, %Mi:L,%Qi:L, %Si:L,%Xi:L,%Ii:L

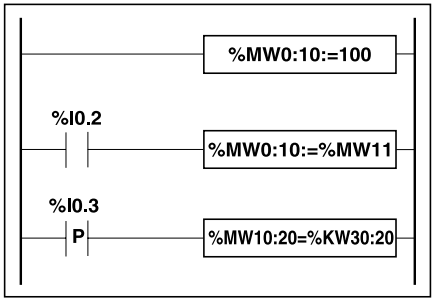
Notas:

- la abreviación %BLK.x (por ej. R3.I) se utiliza para designar cualquier palabra del bloque de función.
- para las cadenas de bits %Mi:L, %Si:L y Xi:L, la dirección del primer bit de la cadena de bits (i) debe ser múltiplo de 8 (0, 8, 16, ..., 96, ...).

Asignación de tablas de palabras (véase el apartado 3.1-2)

Se pueden realizar las siguientes operaciones de asignación de tablas de palabras:
valor inmediato -> tablas de palabras (ej. 1)
palabra -> tabla de palabras (ej. 2)
tabla de palabras -> tabla de palabras (ej. 3)

Ejemplos



LD 1
[%MW0 :10:= 100] (ej. 1)

LD %I0.2
[%MW0:10 := %MW11] (ej. 2)

LDR %I0.3
[%MW10:20=%KW30:20] (ej. 3)

Sintaxis

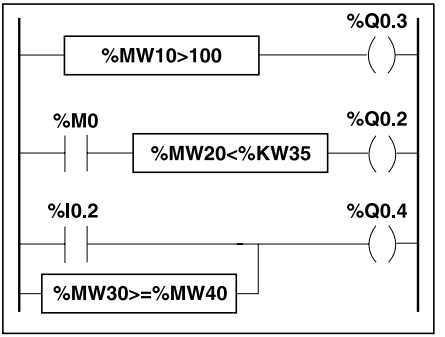
Operador	**Sintaxis**	**Operando 1 (Op 1)**	**Operando 2 (Op 2)**
:=	[Op1: = Op2] El operando 1 (Op1) toma el valor del operando 2 (Op 2),	**%MWi:L,%SWi:L**	**%MWi:L, %KWi:L, %SWi:L** Valor inmediato, %MWi, %KWi, %IW, %QW, %SWi, %BLK.x

3.1-5 Instrucciones de comparación

Las instrucciones de comparación permiten comparar dos operandos.

> : verifica si el operando 1 es superior al operando 2.
>= : verifica si el operando 1 es superior o igual al operando 2.
< : verifica si el operando 1 es inferior al operando 2.
<= : verifica si el operando 1 es inferior o igual al operando 2.
= : verifica si el operando 1 es igual al operando 2.
<> : verifica si el operando 1 es diferente del operando 2.

Estructura
La comparación se realiza entre los corchetes que siguen a las instrucciones LD, AND y OR. El resultado está a 1 cuando la comparación solicitada es verdadera.




```
LD    [%MW10 > 100]
ST    %Q0.3
LD    %M0
AND   [%MW20 < %KW35]
ST    %Q0.2
LD    %I0.2
OR    [%MW30 >= %MW40]
ST    %Q0.4
```

Sintaxis

Operador	Sintaxis	Operando 1 (Op 1)	Operando 2 (Op 2)
>,>=,<,<= =, <>	LD[Op1 Operador Op2] AND[Op1 Operador Op2] OR[Op1 Operador Op2]	%MWi,%KWi,%IW, %QW,%SWi,%BLK.x	Valor inmedianto, %MWi, %KWi,%IW,%QW,%SWi, %BLK.x,%MWi[%MWi], %KWi[%MWi]

Observación
Las instrucciones de comparación pueden utilizarse entre paréntesis.
Ejemplo:

```
LD    %M0
AND(  [%MW20>10]
OR    %I0.0
)
ST    %Q0.1
```

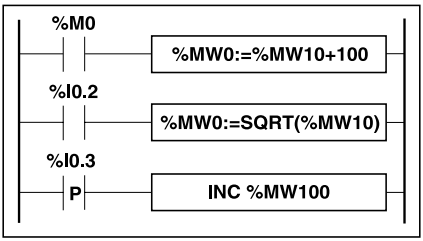
3.1-6 Instrucciones aritméticas

Estas instrucciones permiten realizar una operación aritmética entre dos operandos o en un operando.

+ : suma de dos operandos,
- : resta de dos operandos,
***** : multiplicación de dos operandos,
/ : división de dos operandos,
REM : resto división de 2 operandos,
SQRT: raíz cuadrada de un operando,
INC : incremento de un operando,
DEC : decremento de un operando.

Estructura

Las operaciones aritméticas se realizan de la siguiente manera:




```
LD      %M0
[%MW0 := %MW10 + 100]
LD      %I0.2
[%MW0 := SQRT(%MW10)]
LDR     %I0.3
[INC %MW100]
```

Sintaxis

Depende de los operadores utilizados, véase la siguiente tabla.

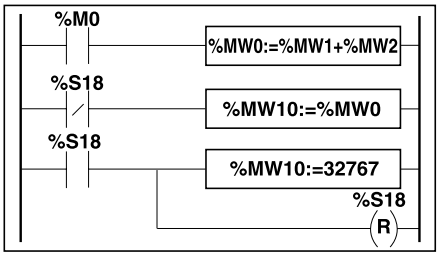
Operador	Sintaxis	Operando 1 (Op 1)	Operandos 2 y 3 (Op 2 y 3)
+,-,*,/,REM (1)	[Op1: = Op2 Operador Op3]	%MWi,%QWi, %SWi	Val. inmed. (2), %MWi,%KWi,
SQRT	[Op1: = SQRT(Op2)]		%IW,%QW,%SWi,%BLK.x
INC, **DEC**	[Operador Op1]		

Normas de utilización

• **Suma: desbordamiento de capacidad durante la operación**

En caso que el resultado exceda los límites -32768 o +32767, el bit %S18 (overflow) se pone a 1. Por lo tanto, el resultado es no significativo. La gestión del bit %S18 se efectúa con el programa de usuario.

Ejemplo:




```
LD      %M0
[%MW0 := %MW1+ %MW2]
LDN     %S18
[%MW10 := %MW0 ]
LD      %S18
[%MW10 := 32767 ]
R       %S18
```

Con %MW1 =23241, %MW2=21853, el resultado real (45094) no puede expresarse en una palabra de 16 bits, el bit %S18 se pone a 1 y el resultado obtenido (-20442) es erróneo. En este ejemplo como el resultado es superior a 32767, su valor es igual a 32767.

(1) Con los TSX 07 de versión inferior o igual a V2.2, el resultado (Op1) de la división (/) o del resto de división (REM) es no significativo cuando el operando 3 (Op3) es superior a 255.

(2) Con el operador SQRT, el operando Op2 no puede ser un valor inmediato.

Desbordamiento de la capacidad absoluta del resultado (aritmética sin signo):
Al hacer algunos cálculos, es interesante interpretar un operando en aritmética sin signo (el bit F representa entonces el valor 32768). El valor máximo para un operando es 65535. La suma de 2 valores absolutos (sin signo) cuyo resultado sea superior a 65535 provoca un desbordamiento. Dicho desbordamiento es señalado por la puesta a 1 del bit sistema %S17 (carry) que representa el valor 65536.

Ejemplo 1: [%MW2:=%MW0 + %MW1] con %MW0 =65086, %MW1=65333
La palabra %MW2 contiene el número 64883 y el bit %S17 se pone a 1 y representa el valor 65536. El resultado aritmético sin signo es por lo tanto igual a:
65536 + 64883 es decir 130419.

Ejemplo 2: [%MW2:=%MW0 + %MW1] con %MW0 =45736 (es decir -19800 en valor con signo), %MW1=38336 (es decir -27200 en valor con signo).
Los dos bits sistema %S17 y %S18 se ponen a 1. El resultado aritmético con signo (+18536) es erróneo. En aritmética sin signo, el resultado (18536 + valor de %S17 es decir 84072) es correcto.

- **Resta:**

Resultado negativo
Si el resultado de la resta es inferior a 0, el bit sistema %S17 se pone a 1.

- **Multiplicación:**

Desbordamiento de capacidad durante la operación
Si el resultado excede la capacidad de la palabra de destino, el bit %S18 (overflow) se pone a 1 y el resultado es no significativo.

- **División/resto de la división:**

División por 0
Si el divisor es igual a 0, la división es imposible y el bit sistema %S18 se pone a 1; el resultado será pues erróneo.

Desbordamiento de capacidad durante la operación
Si el cociente de la división excede la capacidad de la palabra de destino, el bit %S18 se pone a 1.

- **Extracción de la raíz cuadrada:**

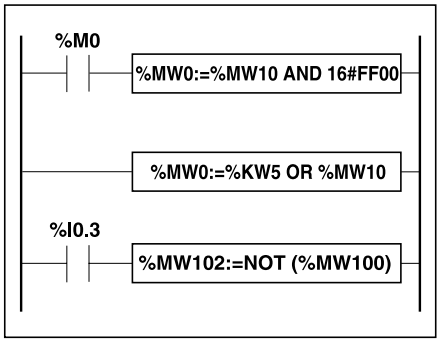
La extracción de raíz cuadrada se efectúa únicamente en valores positivos. Así el resultado es siempre positivo. Si el operando de la raíz cuadrada es negativo, el bit sistema %S18 se pone a 1 y el resultado es erróneo.

Nota: el programa de usuario se encarga de la gestión de los bits sistema %S17 y %S18. El autómata los pone automáticamente a 1; el programa debe reponerlos a cero para que puedan ser reutilizados (véase el ejemplo de la página anterior).

3.1-7 Instrucciones lógicas

Las instrucciones asociadas permiten realizar una operación lógica entre dos operandos o en un operando.
AND : Y (bit a bit) entre dos operandos,
OR : O lógica (bit a bit) entre dos operandos,
XOR : O exclusiva (bit a bit) entre dos operandos,
NOT : complemento lógico (bit a bit) de un operando.

Estructura
Las operaciones lógicas se realizan de la siguiente manera:




```
LD      %M0
[%MW0 := %MW10 AND 16#FF00]

LD      1
[%MW0 := %KW5 OR %MW10]

LD      %I0.3
[%MW102:= NOT (%MW100)]
```

Sintaxis
Depende de los operadores utilizados, véase la siguiente tabla.

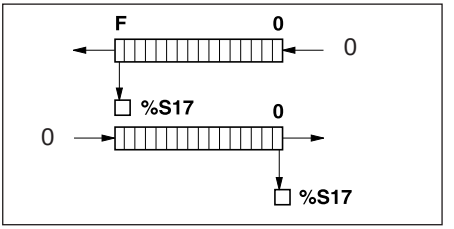
Operador	Sintaxis	Operando 1 (Op 1)	Operandos 2 y 3 (Op 2 y 3)
AND, **OR**, **XOR**	[Op1: = Op2 Operador Op3]	%MWi,%QWi,%SWi	Val. inmed.(1), %MWi,%KWi, %IW,%QW,%SWi,%BLK.x
NOT	[Op1: = NOT(Op2)]		

Ejemplo: [%MW15:=%MW32 AND %MW12]

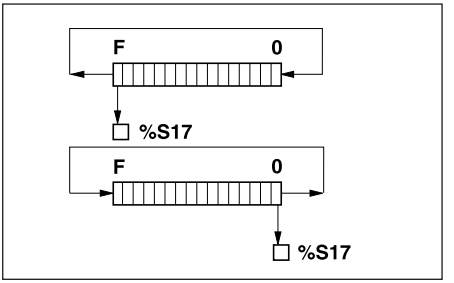
(1) con el operador NOT, el operando Op2 no puede ser un valor inmediato.

3.1-8 Instrucciones de desplazamiento

Las instrucciones de desplazamiento consisten en desplazar los bits de un operando un cierto número de posiciones hacia la derecha o hacia la izquierda.
Se distinguen dos tipos de desplazamientos:



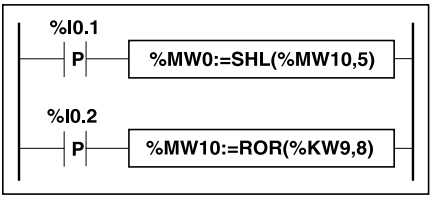
- el **desplazamiento lógico**:
  - **SHL(**op2**,i)** desplazamiento lógico hacia la izquierda de i posiciones.
  - **SHR(**op2**,i)** desplazamiento lógico hacia la derecha de i posiciones.



- el **desplazamiento circular**:
  - **ROL(**op2**,i)** desplazamiento circular hacia la izquierda de i posiciones.
  - **ROR(**op2**,i)** desplazamiento hacia la derecha de i posiciones.

Como el operando a desplazar es un operando de simple longitud, la variable i estará comprendida necesariamente entre 1 y 16. El estado del último bit salido se memoriza en el bit %S17.

Estructura
Las operaciones lógicas se realizan de la manera siguiente:




```
LDR   %I0.1
[%MW0 := SHL(%MW10,5)]

LDR   %I0.2
[%MW10 := ROR(%KW9,8)]
```

Sintaxis
Depende de los operadores utilizados, véase la siguiente tabla.

Operador	Sintaxis	Operando 1 (Op 1)	Operando 2 (Op 2)
SHL, **SHR** **ROL**, **ROR**	[Op1: = Operador(Op2,i)]	%MWi,%QWi,%SWi	%MWi,%KWi,%IW,%QW, %SWi,%BLK.x

3.1-9 Instrucciones de conversión

Se proponen dos instrucciones de conversión:
- **BTI**: conversión BCD --> Binaria
- **ITB**: conversión Binaria --> BCD

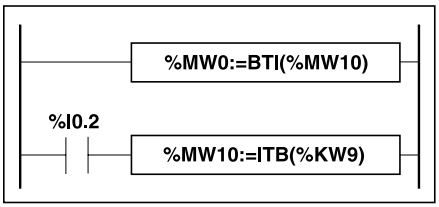
Recordatorio sobre el código BCD
Consideraciones sobre el código BCD:
El código BCD (Binary Coded Decimal) que significa Decimal codificado en binario permite representar una cifra decimal 0 a 9 mediante un conjunto de 4 bits. Un objeto palabra de 16 bits puede así contener un número expresado en 4 cifras ($0 \leq N \leq 9999$). En una conversión, cuando el valor no corresponde a un valor BCD, el bit de sistema %S18 se pone a 1.

Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Ejemplo
- Palabra %MW5, que expresa el valor BCD "2450", corresponde al valor binario 0010 0100 0101 0000
- Palabra %MW12, que expresa el valor decimal "2450", corresponde al valor binario 0000 1001 1001 0010

El paso de la palabra %MW5 a la palabra %MW12 se efectúa con la instrucción BTI.
El paso de la palabra %MW12 a la palabra %MW5 se efectúa con la instrucción ITB.

Estructura
Las operaciones de conversión se realizan de la manera siguiente:




```
LD      1
[%MW0 := BTI (%MW10)]

LD      %I0.2
[%MW10 := ITB (%KW9)]
```

Sintaxis
Depende de los operadores utilizados, véase la siguiente tabla.

Operador	Sintaxis	Operando 1 (Op 1)	Operando 2 (Op 2)
BTI, **ITB**	[Op1: = Operador(Op2)]	%MWi,%QWi,%SWi	%MWi,%KWi,%IW,%QW, %SWi,%BLK.x

Ejemplos de aplicaciones
La instrucción BTI se utiliza para tratar un valor de consigna presente en una entrada del autómata con ruedas codificadas BCD.
La instrucción ITB se utiliza para visualizar valores numéricos (ej: resultado de cálculo, valor actual de bloque de función) en visualizadores codificados BCD.

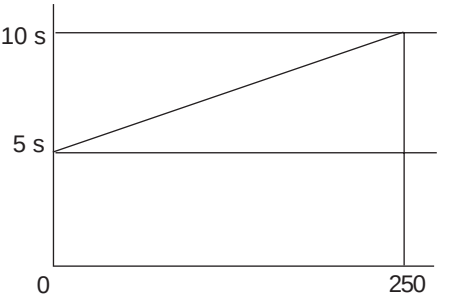
3.2 Puntos de ajuste analógico

Recordatorio del apartado 1.8 sección A:
Los autómatas TSX Nano de base disponen en la parte delantera de:
- un potenciómetro de ajuste analógico para los autómatas TSX Nano de 10, 14 y 20 E/S,
- dos potenciómetros de ajuste analógico para los autómatas TSX Nano de 16 y 24 E/S.

Programación
Los valores numéricos de 0 a 255, que corresponden a los valores analógicos proporcionados por dichos potenciómetros, están disponibles en las palabras sistema:
- %SW112 para el potenciómetro n°0
- %SW113 para el potenciómetro n°1

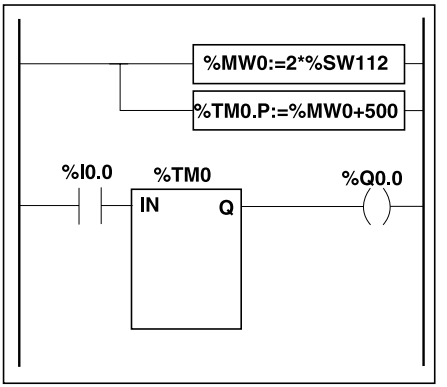
Estas palabras se pueden utilizan mediante las operaciones aritméticas. Pueden utilizarse para cualquier tipo de ajuste: preselección del temporizador, del contador, ajuste de la frecuencia del generador de impulsos, tiempo de precalentamiento de máquinas ...

Ejemplo:
Ajuste de la duración de una temporización de 5 a 10 s con el potenciómetro n°0.
Para este ajuste, se utiliza prácticamente toda la extensión de ajuste del potenciómetro de 0 a 250.



En la configuración, se seleccionan los parámetros siguientes para el bloque de temporización %TM0:
- tipo TON
- base de tiempo BT: 10 ms

El valor de preselección del temporizador se deduce del valor de ajuste del potenciómetro mediante la ecuación %TM0.P:=2*%SW112+500




```
LD      1
[%MW0:=2*%SW112]
[%TM0.P:=%MW0+500]
LD      %I0.0
IN      %TM0
LD      %TM0.Q
ST      %Q0.0
```

3.3 Vía analógica (TSX 07 32/33 •• ••)

Los autómatas TSX Nano **TSX 07 32/33 •• ••** incorporan una vía analógica 0/10 V no aislada.

Principio
Un convertidor analógico/numérico convierte la tensión de entrada 0-10 V en un valor numérico de 0 a 255 que se coloca en la palabra de sistema %SW112.

Tensión de entrada	%SW112
0 V	0
40 mV	1
80 mV	2
•	•
•	•
9,96 V	249
10 V	250
10,2 V	255

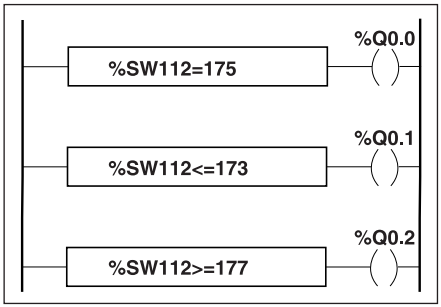
El valor numérico 255 permite detectar un rebasamiento del valor máximo de la señal de entrada.

Puede utilizarse el potenciómetro P0, situado en la parte delantera, para corregir el error que puede producirse debido a la cadena de medida en determinadas aplicaciones.

Ejemplo de programación
Regulación de la temperatura de un horno de cocción.
La temperatura de cocción se establece en 315 °C y una variación de +/- 3,6 °C provoca respectivamente el control de las salidas %Q0.1 y %Q0.2. Se utiliza prácticamente todo el margen de ajuste posible del potenciómetro de 0 a 250 para este ajuste.

0 °C -> 0 V -> %SW112 = 0
311,4 °C -> 6,92 V -> %SW112 = 173
315 °C -> 7 V -> %SW112 = 175
318,6 °C -> 7,08 V -> %SW112 = 177
450 °C -> 10 V -> %SW112 = 250

```
LD   [%SW112 = 175]
ST   %Q0.0
LD   [%SW112 <= 173]
ST   %Q0.1
LD   [%SW112 >= 177]
ST   %Q0.2
```



3.4 Bloques de función específicos

3.4-1 Objetos bits y palabras asociadas a bloques de función específicos

B

Los bloques de función específicos activan objetos de bits y palabras específicos del mismo tipo que los bloques de función estándares (véase el apartado 2.2).

Lista de objetos bits y palabras de bloques de función accesibles desde el programa

Bloques de función específicos		Palabras y bits asociados	Dirección	Acceso escritura	Ver ap.
Salida modulación de amplitud %PWM	Palabra	% del impulso a 1 con relación al período total	%PWM.R	sí	3.4-3
		Valor de preselección del período	%PWM.P	no	
Generador de impulsos %PLS	Palabra	Valor de preselección	%PLS.P	sí	3.4-4
		Nº impulsos para generar	%PLS.N	sí	
	Bit	Salida en curso	%PLS.Q	no	
		Salida generación terminada	%PLS.D	no	
Contador rápido %FC	Palabra	Umbral i (i = 0 ó 1)	%FC.Si	sí	3.4-5
		Valor actual	%FC.V	no	
		Valor de preselección	%FC.P	sí	
	Bit	Salida desbordamiento	%FC.F	no	
		Salida desbordamiento umbral i	%FC.THi	no	
Envío de mensaje %MSG	Bit	Salida falla enlace	%MSG.E	no	3.4-6
		Salida enlace disponible	%MSG.D	no	
Registro bit de desplazamiento %SBRi (i=0 à 7)	Bit	Bit del registro j=0 a 15	%SBRi.j	sí	3.4-7
Paso a paso %SCi (i=0 a 7)	Bit	Bit del paso a paso, j=0 a 255	%SCi.j	no	3.4-8

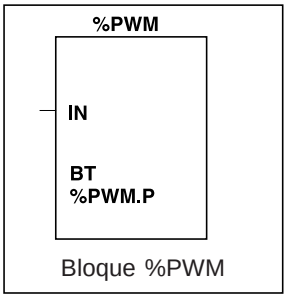
3.4-2 Principios de programación

Así como los bloques de función estándares, los bloques de función específicos pueden programarse de 2 modos diferentes:

- de manera no reversible: mediante instrucciones específicas,
- de manera reversible: simulando los bloques de función del lenguaje de contactos.

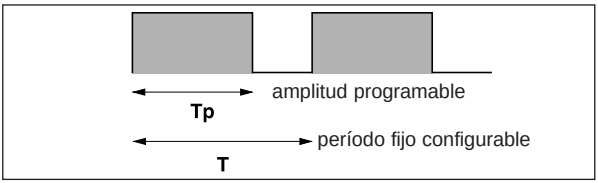
Consúltese el apartado 2.2-2

3.4-3 Salida de modulación de amplitud %PWM



Bloque %PWM

El bloque de función %PWM permite generar en la salida autómata %Q0.0 una señal rectangular, cuya amplitud puede cambiarse (relación cíclica) por programa (véase descripción, apartado 4.6, secc.A).
Además esta función permite controlar un módulo de salida analógica conectado a la salida %Q0.0 (véase el apartado 4.4, sección B).



Características

Base de tiempo	**BT**	0,1 ms (1), 10 ms, 1 s (valor por defecto)
Preselección del período	**%PWM.P**	0<%PMW.P<32767 si base de tiempo 10 ms ó 1 s. 0<%PMW.P<255 si base de tiempo 0,1ms (0 = función inactiva) En configuración, se accede a la preselección y a la base de tiempo en escritura; permiten fijar el período de la señal **T = %PWM.PxBT**. %PWM.P debe seleccionarse tanto mayor cuanto menos elevados sean los coeficientes. **Gama de período obtenida:** • 0,2 a 26 ms con pasos de 0,1 ms, • 20 ms a 5,45 mn con pasos de 10 ms, • 2 s a 9,1 horas con pasos de 1 s.
Intervalo del período	**%PWM.R**	0≤%PMW.R≤100 (2), esta palabra da el porcentaje de la señal en el estado 1 sobre el período (0 = valor por defecto). La "amplitud" Tp es pues igual a: **Tp=T x (%PWM.R/100)** La palabra %PWM.R se escribe mediante el programa de usuario; esta palabra es la que permite efectuar la modulación de amplitud.
Entrada (o instrucción) generación de impulsos	**IN**	En estado 1, genera la señal modulada en amplitud en la salida %Q0.0. En estado 0, pone la salida %Q0.0 a 0.

Atención:

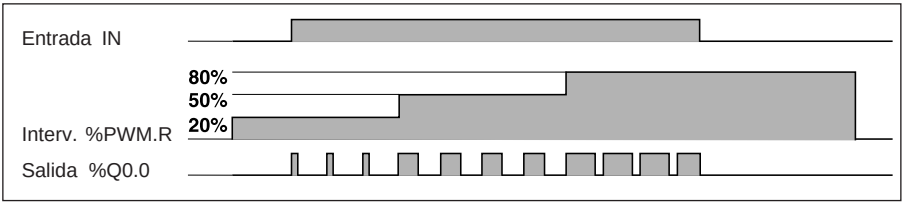
El algoritmo de toma en cuenta de la modulación con amplitud %PWM se ha perfeccionado entre las versiones V3.0 y V3.1 del TSX Nano. Esto puede provocar en la salida una relación cíclica diferente entre las dos versiones para una misma %PWM.R.

(1) Se desaconseja utilizar esta base de tiempo para los TSX Nano con salidas relés.
(2) Los valores superiores a 100 se considerarán iguales a 100.

B

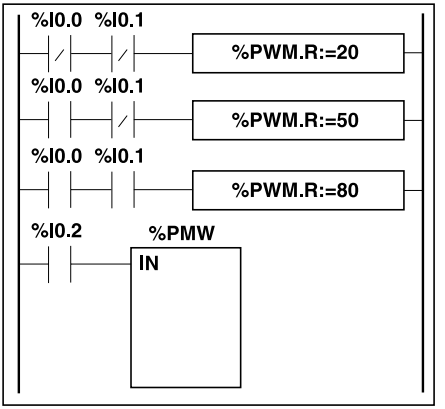
Funcionamiento

La frecuencia de la señal en la salida %Q0.0 se fija en configuración mediante la selección de la base de tiempo BT y de la preselección %PWM.P. La modulación de amplitud de la señal se obtiene modificando el cociente %PWM.R por el programa.



Programación y configuración

En este ejemplo, la amplitud de la señal se modifica por programa en función del estado de las entradas %I0.0 y %I0.1 del autómata.
El período de la señal se fija en 500 ms en configuración.
Si %I0.0 y %I0.1 están a 0, el intervalo %PWM.R se fija en 20%. La duración de la señal en el estado 1 es entonces de 20% x 500 ms = 100 ms.
Si %I0.0 en 0 y %I0.1 está a 1, %PWM.R se fija en 50% (duración 250 ms).
Si %I0.0 y %I0.1 en 1, el intervalo %PWM.R se fija en 80% (duración 400 ms).




```
LDN   %I0.0
ANDN %I0.1
[%PWM.R:=20]
LD     %I0.0
ANDN %I0.1
[%PWM.R:=50]
LD     %I0.0
AND   %I0.1
[%PWM.R:=80]
BLK   %PWM
LD     %I0.2
IN
END_BLK
```

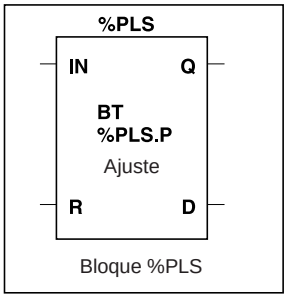
Configuración

Salida %Q0.0 = salida %PWM BT = 10 ms %PWM.P = 50

Casos específicos

- **Incidencia de un arranque en frío:** (%S0=1) causa la puesta a 0 de %PWM.R.
- **Incidencia de un arranque en caliente:** (%S1=1) ninguna incidencia.
- **Incidencia en STOP autómata:**
  La salida %Q0.0 se pone a 0 sea lo que sea el estado del bit sistema %S8. Si la versión del autómata es anterior o igual a V2.2, la salida %Q0.0 se pone a 0 si %S8 = 1 o se mantiene (generación de la señal) si %S8 = 0.
- **Con la base de tiempo 0,1 ms**, el forzado de la salida %Q0.0 por un terminal no interrumpe la generación.

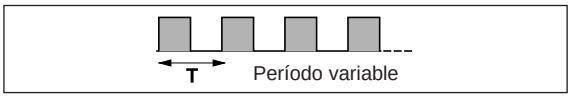
3.4-4 Salida del generador de impulsos %PLS



Bloque %PLS

El bloque de función %PLS permite generar una señal cuadrada (relación cíclica de 50% garantizada si %PLS.P tiene un valor par) en la salida autómata %Q0.0.
Esta señal puede ser:

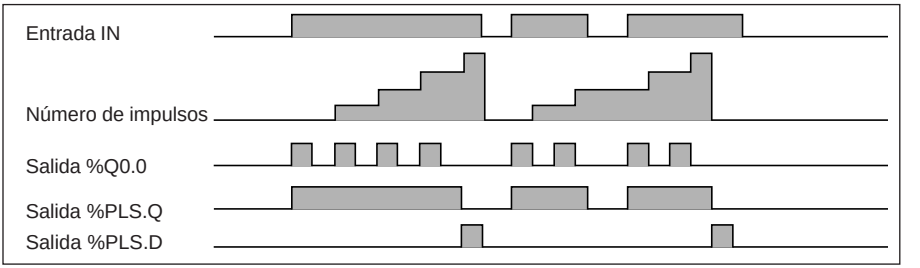
- de duración limitada: el número de impulsos y el período se escriben por el programa (o en configuración).
- de duración ilimitada: el período se escribe por el programa o en la configuración. (Descripción apart. 4.5, sección A).



Características		
Base de tiempo	**BT**	0,1 ms (1), 10 ms, 1 s (valor por defecto)
Preselección	**%PLS.P**	0<%PLS.P<32767 si BT=10 ms o 1 s. 0<%PLS.P<255 si BT= 0,1 ms (1)(2) (0= salida a 0, 1=salida a 1). La preselección permite modular el período de la señal **T = %PLS.PxBT**. **Gama de período obtenida:** • 0,2 a 26 ms con pasos de 0,1 ms, • 20 ms a 5,45 mn con pasos de 10 ms, • 2s a 9,1 horas con pasos de 1s. **Nota: %PLS.P debe ser un número par**
Número de impulsos	**%PLS.N**	0≤%PLS.N≤32767, esta palabra da el número de impulsos del tren de impulsos a generar. 0 = señal cuadrada de duración ilimitada (por defecto). El programa verifica y escribe %PLS.N.
Ajuste por terminal	**O/N**	O: posibilidad de modificar el valor de preselección %PLS.P en modo de ajuste. N: ningún acceso en modo de ajuste.
Entrada (o instrucción) generación de impulsos	**IN**	En estado 1, genera la señal sobre la salida %Q0.0. En estado 0 pone la salida %Q0.0 a 0.
Entrada (o instrucción) reinicialización	**R**	En estado 1, pone a 0 el número de impulsos de las salidas %PLS.Q y %PLS.D.
Salida generación de impulsos en curso	**%PLS.Q**	En estado 1, generación de la señal en la salida %Q0.0 en curso.
Salida generación de impulsos terminada	**%PLS.D**	En estado 1, generación de la señal en la salida %Q0.0 terminada.
Contaje PLS (2)		N=no, S=sí, esta opción permite utilizar la entrada %I0.0 como entrada de contaje.

(1) Se desaconseja utilizar esta base de tiempo para los TSX Nano con salidas relés.
(2) La opción de contaje %PLS es obligatoria cuando se selecciona la base de tiempo de 0,1 ms. En tren de impulsos, requiere el reenlace físico de la salida %Q0.0 en la entrada %I0.0. En este tipo de funcionamiento, el %PLS.P debe ser superior o igual a 6 (frec.máx. =1,6 Khz) para garantizar el buen funcionamiento de la función.

Funcionamiento



Casos específicos

- **Incidencia de un arranque en frío:** (%S0=1) provoca la inicialización del valor de %PLS.P por el definido en la configuración.
- **Incidencia de un arranque en caliente:** (%S1=1) ninguna incidencia.
- **Incidencia en STOP autómata:** véase Incidencia en Stop autómata en la página 3/18.
- **Incidencia de una modificación de la preselección %PLS.P:** se toma en cuenta instantáneamente la modificación de %PLS.P por instrucción o en ajuste.
- **Con la base de tiempo 0,1 ms**: el forzado de la entrada de reenlace %I0.0 no interrumpe la generación.

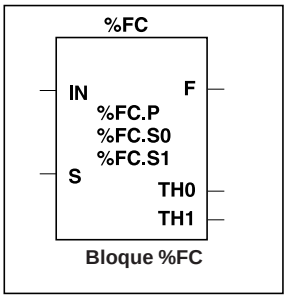
Observación:

El uso de la base de tiempo 0,1 ms puede provocar una imprecisión de un impulso entre el número de impulsos solicitado (%PLS.N) y el número de impulsos que se genera en realidad.
Para corregir esta imprecisión, es necesario provocar una reinicialización (RAZ) de la función al final de la generación:

```
LD   %Mi
IN   %PLS
N
R    %PLS
```

No obstante, esta acción correctora reinicia una generación de %PLS.N impulsos desde el momento en que la entrada IN pasa a 1.

3.4-5 Funciones de contaje rápido, frecuencímetro y contador/descontador %FC



Bloque %FC

El bloque de función %FC se utiliza para realizar una de las siguientes funciones:

- contaje rápido,
- frecuencímetro,
- contaje/descontaje.

(Véase descripción detallada, apartado 4.4, sección A). La selección se realizará en configuración.

Nota: las funciones de contaje rápido pueden realizarse sin ninguna programación, simplemente configurando las entradas/salidas y los parámetros.

Características del bloque %FC

El bloque de función ofrece un conjunto de palabras, bits de entrada y salida, que permiten elaborar las 3 funciones de contaje. Para conocer exactamente el papel de cada objeto en la función que se va a realizar, véase la descripción de la función.

Valor actual	**%FC.V**	Palabra incrementada o disminuida en función de las entradas y de la función seleccionada. Puede ser leída y verificada pero no escrita.
Valor de preselección	**%FC.P**	Utilizado únicamente por la función contaje/descontaje: 0≤%FC.P≤65535. Palabra que puede ser leída, verificada y escrita.
Valor de umbral S0 (1)	**%FC.S0**	0≤%FC.S0≤65535, palabra que contiene el valor del umbral 0, definida en configuración. Puede ser leída y escrita mediante el programa.
Valor de umbral S1 (1)	**%FC.S1**	0≤%FC.S1≤65535, palabra que contiene el valor del umbral 1, definida en configuración. Puede ser leída y escrita mediante el programa.
Entrada (o instrucción) validación	**IN**	En estado 1, valida la función en curso. En estado 0, inhibe la función en curso.
Entrada (o instrucción) preselección	**S**	En estado 1: • inicializa el valor actual al de preselección (función contaje/descontaje) o pone el valor actual a 0, • inicializa el funcionamiento de las salidas directas, • acepta los valores de umbral %FC.S0 y %FC.S1 modificados por el programa.
Salida desbordamiento	**%FC.F**	Estado 1, cuando el valor actual %FC.V excede 65535. Puede ser reinicializado por la entrada de preselección (%I0.1 o instrucción S) o el rearranque en frío
Salida umbral 0 (2)	**%FC.TH0**	Estado 1, cuando el valor actual es superior o igual al valor de umbral %FC.S0
Salida umbral 1 (2)	**%FC.TH1**	Estado 1, cuando el valor actual es superior o igual al valor de umbral %FC.S1

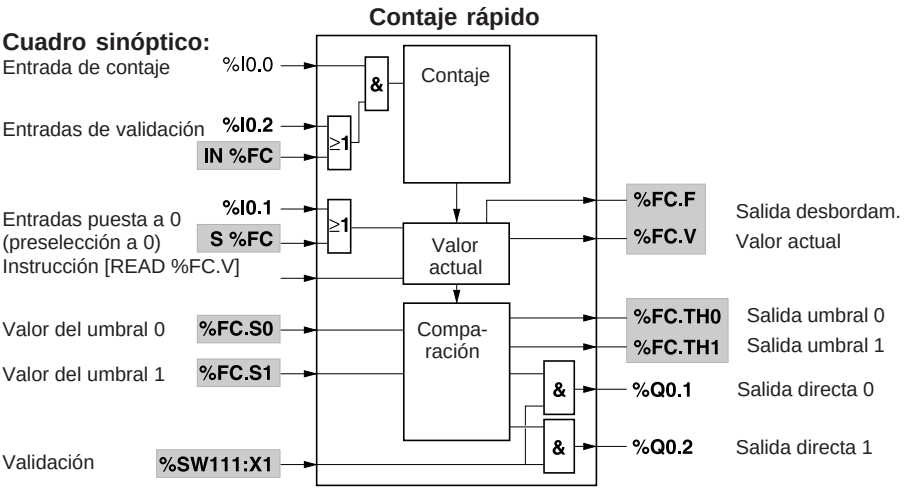
(1) El valor de umbral %FC.S0 debe ser inferior al valor de umbral %FC.S1.
(2) Se aconseja comprobar los bits %FC.THx una sola vez en el programa.

Función de contaje rápido

La función de contaje rápido permite el contaje a una frecuencia máxima de 10 kHz en modo rápido (ó 5 kHz en modo normal, selección en configuración), con una capacidad de contaje máxima de 65535 puntos.

El contador recibe las señales para contar en la entrada autómata %I0.0. Se compara el valor de contaje (valor actual %FC.V) con 2 umbrales %FC.S0 y %FC.S1 definidos en configuración, y modificables por el programa (modificación tomada en cuenta en caso de activar la entrada puesta a 0).

Cuadro sinóptico:



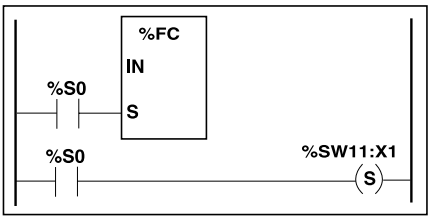
Observación: salvo la entrada contaje %I0.0, las demás E/S TON del bloque de función son opcionales (selección o no en configuración). Existe para cada una de ellas una función equivalente (referenciada con una trama en el cuadro sinóptico).

Salidas directas: el contador rápido controla directamente las salidas directas (no se espera la regeneración de las salidas al final del ciclo) según la matriz de estado que se definirá en la configuración.

Salida	FC.V< umbral 0<umbral 1	umbral 0≤FC.V≤umbral1	umbral 0<umbral 1<FC.V
%Q0.1	0 ó 1	0 ó 1	0 ó 1
%Q0.2	0 ó 1	0 ó 1	0 ó 1

Al inicializar, el funcionamiento de las salidas directas debe ser validado imperativamente por un comando de preselección del contador rápido. Ejemplo de programación:

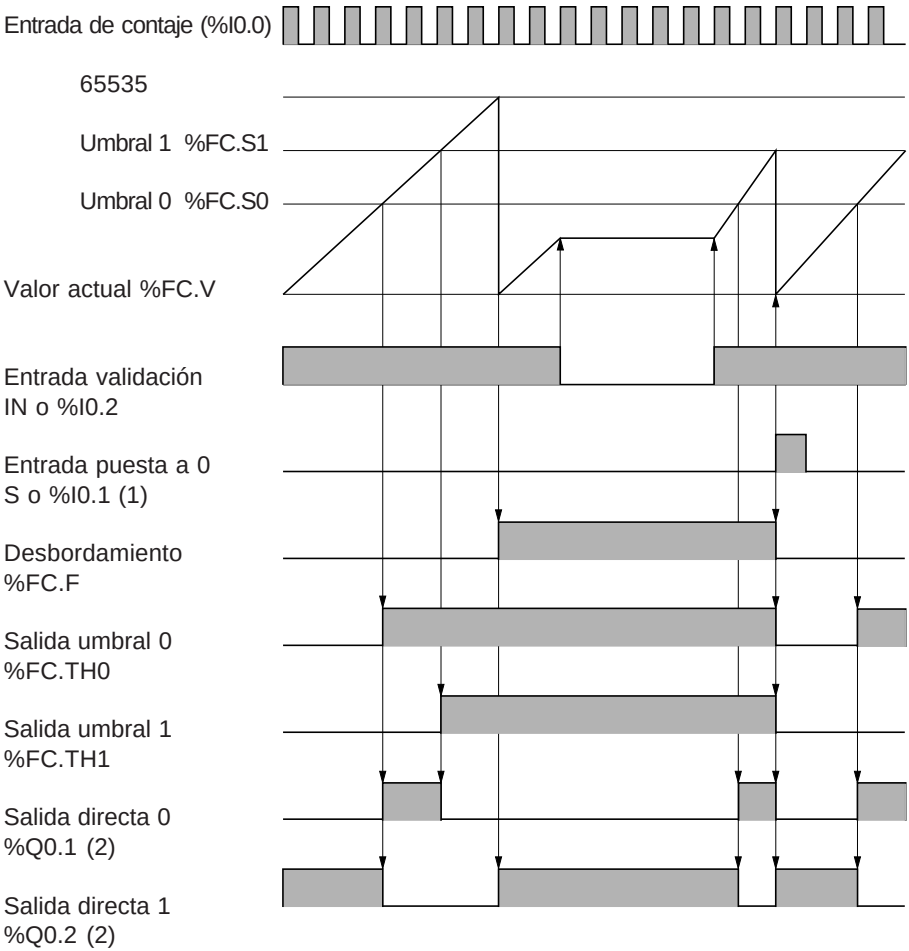
```
BLK  %FC
LD  %S0
S
END_BLK
LD  %S0
S   %SW111:X1     (Puesta a 1 del bit de
                   validación %SW111:X1)
```



Lectura del valor actual

El valor actual %FC.V se actualiza al final de cada ciclo autómata.%FC.V puede actualizarse igualmente mediante la instrucción READ:

sintaxis: **[READ %FC.V] Cronograma**



(1) la entrada %I0.1 funciona en el flanco ascendente (véase el cronograma), al contrario de la entrada S que lo hace en estado.

(2) en este cronograma la matriz de estado es la siguiente:

Salida	FC.V< umbral 0<umbral 1	umbral 0≤FC.V≤umbral 1	umbral 0<umbral 1<FC.V
%Q0.1	0	1	0
%Q0.2	1	0	0

Función de frecuencímetro

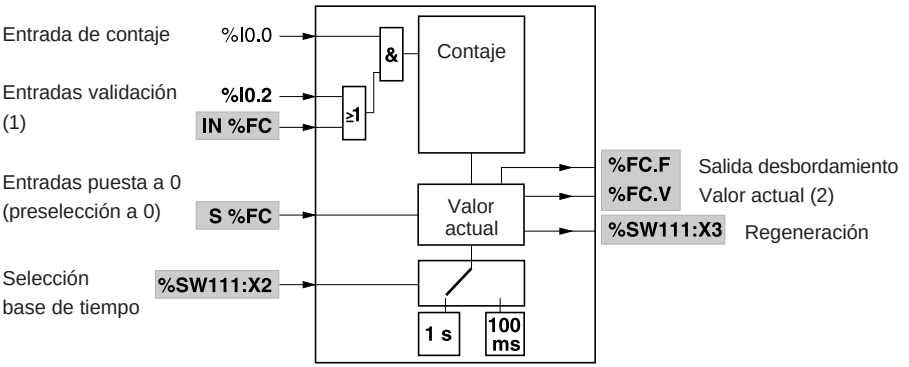
La función de frecuencímetro permite medir la frecuencia en Hz de una señal periódica en la entrada %I0.0. Se proponen dos modos: rápido (filtrado 10 kHz) o normal (filtrado 5 kHz).

La gama de frecuencia que puede ser medida con una precisión admitida se extiende de 1Hz a 10 kHz. El usuario puede seleccionar entre 2 bases de tiempo mediante el bit sistema %SW111:X2 (1 = base de tiempo de 100 ms, 0 = base de tiempo de 1 s).

Base de tiempo	Gama de medida	Precisión	Regeneración
100 ms	100 Hz-10 kHz	0,1% para 10 kHz 10% para 100 Hz	10 veces por segundo
1 s	10 Hz-10 kHz	0,01% para 10 kHz 10% para 10 Hz	1 vez por segundo

El bit sistema %SW111:X3 se pone a 1 cuando se produce una regeneración del valor actual. Su puesta a cero se efectúa mediante programa de usuario.

Cuadro sinóptico:



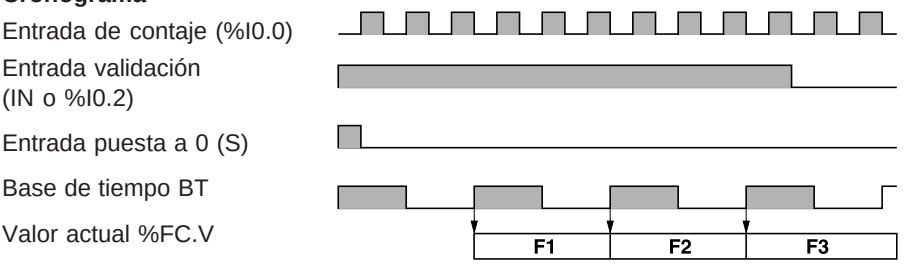
(1) La entrada %I0.2 es opcional y su selección se realiza en configuración

(2) El valor actual FC.V se indica en Hz.

Observación:

Esta función permite además confirmar el valor del módulo analógico conectado a la entrada %I0.0 (véase el apartado 4.3 de la sección B).

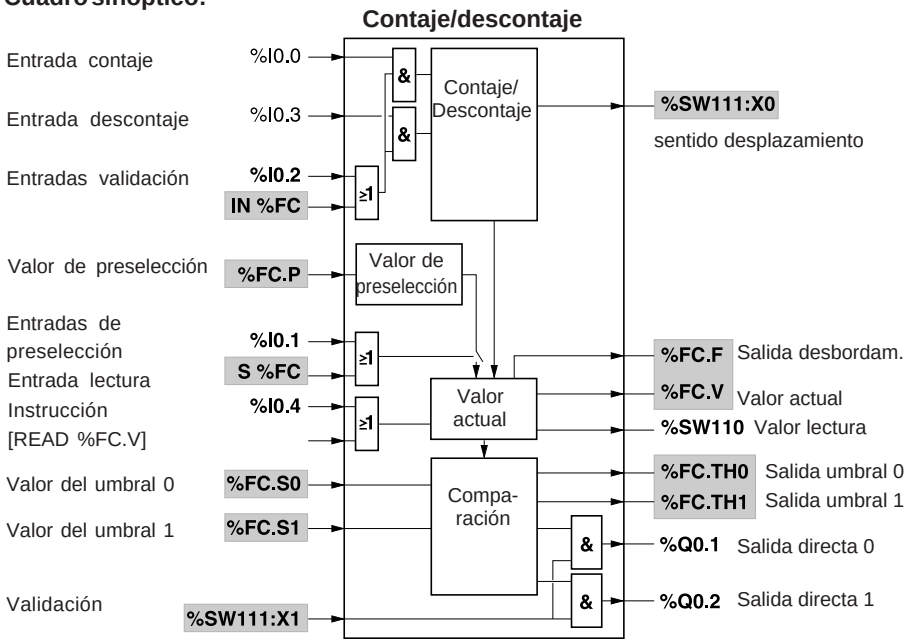
Cronograma



Función de contaje/descontaje

La función de contaje/descontaje permite el contaje/descontaje a una frecuencia máxima de 1 kHz, con un margen de contaje/descontaje de 0 a 65535 puntos.
El contador recibe las señales para contar en la entrada autómata %I0.0 y las señales para descontar en la entrada autómata %I0.3. Proporciona a la salida la indicación de sentido de desplazamiento: si el contador cuenta, %SW111:X0 = 1 o descuenta, %SW111:X0 = 0. Se compara el valor de contaje/descontaje (valor actual %FC.V) con 2 umbrales %FC.S0 y %FC.S1 definidos en configuración y modificables por programa (modificación tomada en cuenta tras activar la entrada de preselección).

Cuadro sinóptico:



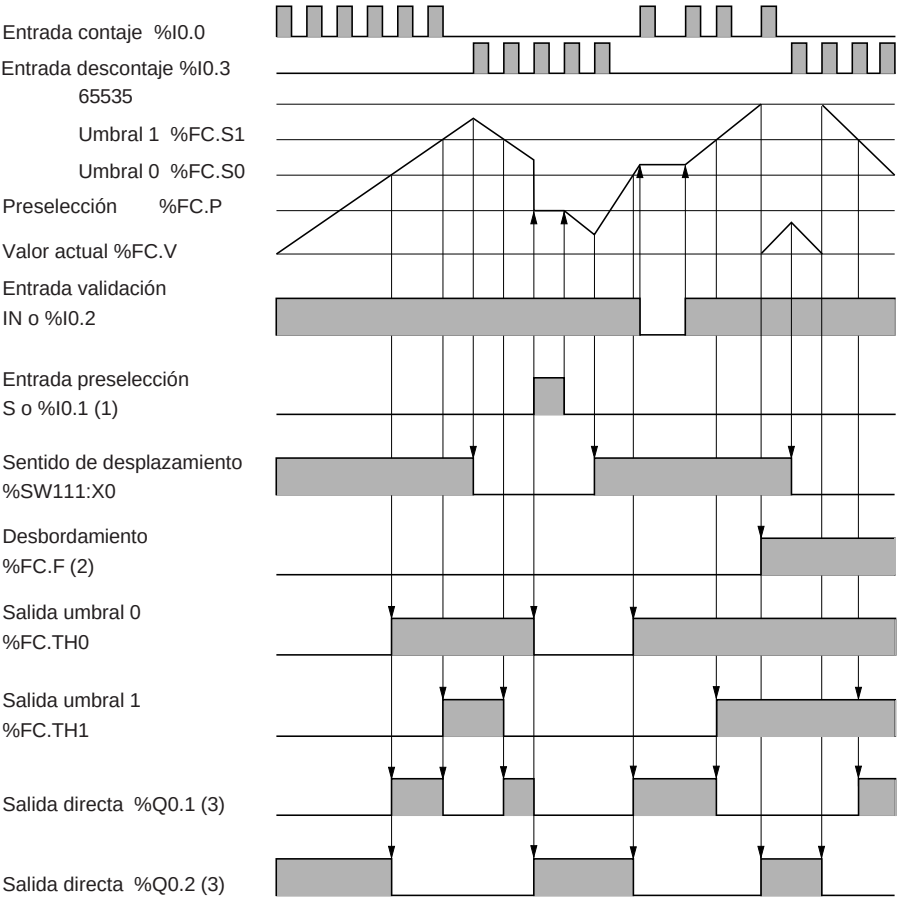
Salvo las entradas contaje %I0.0 y descontaje %I0.3, las demás E/S TON del bloque de función son opcionales (selección o no en configuración). Existe para cada una de ellas una función equivalente (referenciada con una trama en el cuadro sinóptico).

Preselección: se define el valor de preselección de 0 a 65535 en configuración y puede modificarse mediante programa. El valor actual se carga mediante el valor de preselección en el flanco ascendente de la entrada %I0.1 o en el estado 1 de la entrada S.

Salidas directas: véase la función contaje rápido.

Lectura del valor actual

El valor actual %FC.V se actualiza al final de ciclo. También se puede utilizar la instrucción READ, sintaxis: **[READ %FC.V].**
Un flanco ascendente en la entrada lectura %I0.4 escribe el valor actual en la palabra sistema %SW110.



(1) La entrada %I0.1 funciona sobre flanco ascendente, al contrario de la entrada S que funciona sobre estado.
(2) La salida %FC.F de desbordamiento se pone a cero después de reiniciar el contador.
(3) Véase la matriz de estado de la función contaje rápido.

Importante:

Mientras una de las 4 entradas del flanco %I0.0, %I0.1, %I0.3, I0.4 esté en el estado 1, no se ejecutará ninguna acción asociada a una de las tres otras entradas

Casos específicos (contaje rápido, frecuencímetro y contaje/descontaje)

- **Incidencia de un arranque en frío:** (%S0=1) provoca la puesta a 0 del valor actual, de las salidas %FC.F, %FC.TH0, %FC.TH1, del bit de validación de las salidas directas (SW111;X1) y la copia de los valores definidos por configuración en las palabras %FC.P,%FC.S0/S1.

- **Incidencia de un arranque en caliente** (%S1=1) **y STOP autómata:** sin incidencia en el valor actual.

  Después de una instrucción relé maestro MCS activada, las salidas directas pasan a 0 después de la desactivación de MCS, y sólo volverán a posicionarse tras un desbordamiento de umbral en modo contaje o una acción de contaje/descontaje en modo descontaje.

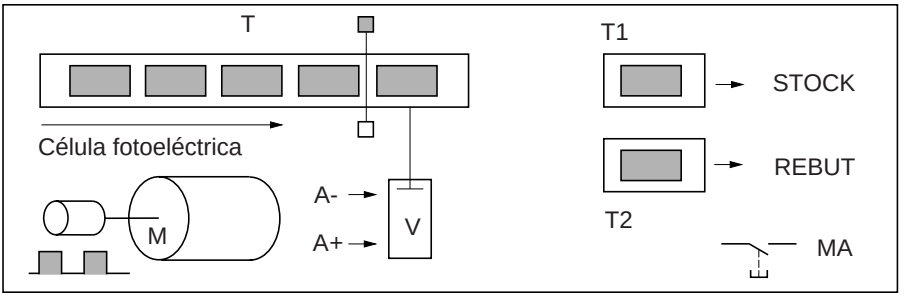
- **Visualización de valores %FC:**
  Para cualquier valor resultante de la función %FC, los valores se visualizarán:
  - por el terminal FTX 117 en valor con signo complemento de 2 (Decimal) o directo (Hexadecimal).
  - por PL7-07 en valor directo (Decimal o Hexadecimal)

  Por ejemplo:

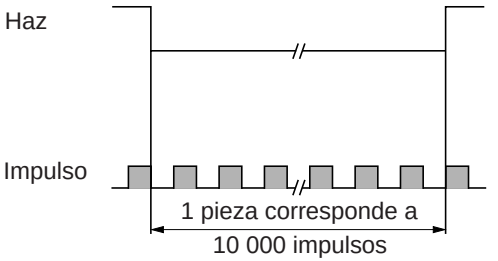
	Valores visualizados		
%FC.V	**(PL7-07)**	**(FTX 117)**	**(PL7-07 - FTX 117)**
0	0	0	0000
1	1	1	0001
2	2	2	0002
•	•	•	•
•	•	•	•
•	•	•	•
32766	32766	32766	7FFE
32767	32767	32767	7FFF
32768		-32768	8000
•	•	•	•
•	•	•	•
•	•	•	•
65533	65533	-3	FFFD
65534	65534	-2	FFFE
65535	65535	-1	FFFF
0	0	0	0

Ejemplo de función de contaje rápido

Descripción de la aplicación



Las piezas que se van a medir se colocan sobre un tapiz arrastrado permanentemente sin posibilidad de escurrirse mediante un motor al que se ha acoplado un codificador incremental rotativo. La medición se efectúa contabilizando el número de impulsos durante el tiempo en que la célula C detecta el paso de la pieza. Un cilindro V controla el desplazamiento lateral del tapiz de transporte T con el fin de situarla en frente del tapiz T1 o T2, según el resultado de la medición.
El número de impulsos medido se compara con 2 valores extremos (tolerancias de la medición de longitud).



- si está comprendida entre estos 2 valores, la pieza se desvía hacia el "stock" (tapiz T1, control A+ del cilindro V)
- en el caso contrario, la pieza es desviada hacia el "rebut" (tapiz T2, control A- del cilindro V)

El botón pulsador MA garantiza la puesta en marcha del conjunto.

Asignación de entradas/salidas

Entradas
- %I0.0 entrada contaje conectada al codificador incremental
- %I0.1 entrada puesta a cero conectada a la célula fotoeléctrica
- %I0.2 entrada validación conectada al botón marcha

Salidas
- %Q0.1 salida control del cilindro A+
- %Q0.2 salida control del cilindro A-
- %Q0.0 salida control del tapiz

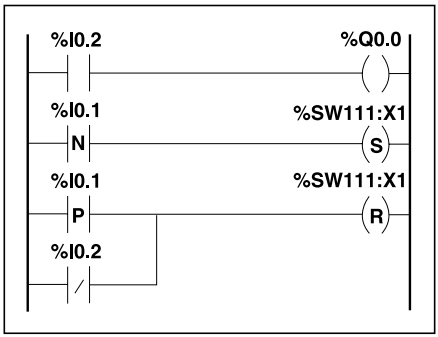
Tratamiento de la aplicación

La función contaje rápido puede ser tratada sin programación del autómata, únicamente por configuración del bloque de función %FC.
%FC: contaje
Modo: rápido
Entrada contaje: %I0.0
Puesta a cero: %I0.1
Entrada validación: %I0.2
%FC.S0: 9950 umbral 0 correspondiente a la tolerancia mínima
%FC.S1: 10 000 umbral 1 correspondiente a la tolerancia máxima

Matriz de salidas

Salida	FC.V< umbral 0<umbral 1	umbral 0≤FC.V≤umbral 1	umbral 0<umbral 1<FC.V
%Q0.1	0	1	0
%Q0.2	1	0	1

Programa




```
LD    %I0.2
ST    %Q0.0
LDF   %I0.1
S     %SW111:X1
LDR   %I0.1
ORN   %I0.2
R     %SW111:X1
```

Funcionamiento

La entrada %I0.0 cuenta el número de impulsos resultante del codificador incremental tan pronto como se activa la entrada validación %I0.2 (conmutador marcha).

Sobre el flanco ascendente de la entrada %I0.1, el valor actual del contador se pone a 0.

Cuando la célula (entrada %I0.1) detecta el final de paso de la pieza sobre el tapiz, las salidas %Q0.1 y %Q0.2 son validadas (por el bit %SW111:X1) y pasan al estado 0 ó 1 en función del valor actual del contador %FC.V (según la matriz de las salidas).
La salida %Q0.1 se pone a 1 cuando la pieza está dentro de las tolerancias. Controla la salida del cilindro A de posicionamiento sobre el tapiz T1 (véase la matriz de las salidas).
La salida %Q0.2 se pone a 1 cuando la pieza está fuera de tolerancia (entra de nuevo el cilindro A sobre el tapiz T2).

3.4-6 Emisión/Recepción de mensajes y control de intercambios

El TSX Nano puede comunicar con un terminal de programación (FTX117 o programa PL7-07) y con otros equipos UNI-TELWAY conectados a la toma terminal. Además el TSX Nano puede configurarse para enviar y/o recibir un mensaje en modo caracteres (protocolo ASCII).

El lenguaje PL7 dispone para ello de:

- la instrucción de emisión/recepción de mensajes **EXCH**,
- el bloque de función de control de intercambios **%MSG**.

El autómata TSX Nano determina el protocolo en función de las patillas del cable utilizado y proporciona esta información en el bit de sistema %S100 (gestión del /DPT).

Los equipos que soportan el protocolo UNI-TELWAY así como su configuración se detallan en la sección F de este manual.

Instrucción EXCH

La instrucción EXCH permite al TSX Nano enviar y/o recibir información hacia un equipo UNI-TELWAY o ASCII. El usuario define una tabla de palabras (%MWi:L o %KWi:L) que contiene los datos que se van a emitir y/o recibir (un máximo de 64 palabras de datos en emisión y/o recepción). El formato de la tabla de palabras se detalla en los apartados referentes a cada protocolo (ASCII y UNITELWAY).

El intercambio de mensajes se realiza mediante la instrucción EXCH.

Sintaxis: [EXCH %MWi:L] (1) **o [EXCH %KWi:L]**

Notas: ciertos equipos que soportan el protocolo UNI-TELWAY (como por ejemplo interfaces hombre-máquina: XBT o CCX17) así como los sistemas de identificación inductivos pueden comunicar con un TSX Nano (envío y/o recepción de información) sin programación específica en el TSXNano.

El TSX07 debe terminar el intercambio de una primera instrucción EXCH antes de activar otra. El bloque %MSG debe utilizarse cuando se envíen varios mensajes.

(1) L: Número de palabras de la tabla de palabras.
Los valores de la tabla de palabras internas %MWi:L son $i+L \leq 255$.

Observación

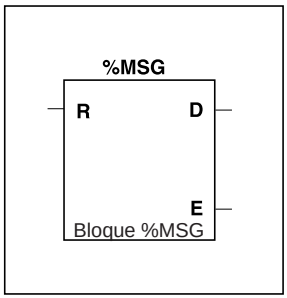
El programa PL7-07 V1 sólo permitía realizar emisiones mediante la instrucción SEND utilizada en los TSX Nano V1 y V2.
Sin embargo, los autómatas TSX Nano V3 (TSX 07 3•••••) permiten emitir y/o recibir mensajes mediante la instrucción EXCH. (En los TSX 07 3 • 10 ••, la instrucción EXCH en modalidad de recepción sólo está disponible en las versiones V3.1.) (En los PL7-07 V1, la instrucción EXCH se denomina SEND).

Bloque de función %MSG

Este bloque se utiliza para gestionar los intercambios de datos (su uso es opcional). Dispone de tres funciones:

- **Control de error**
  Esta función verifica que la longitud de la tabla de palabras de la instrucción EXCH tiene las dimensiones correctas para contener el mensaje que se va a enviar (comparación con la longitud programada en el octeto menos significativo de la primera palabra de la tabla).
  También verifica que se ha recibido un mensaje UNI-TELWAY.
- **Coordinación del envío de varios mensajes**
  Para asegurar la coordinación al emitir varios mensajes, este bloque de función proporciona la información de fin de envío del mensaje anterior.
- **Emisión de un mensaje prioritario**
  El bloque de función %MSG autoriza la parada de la emisión en curso para permitir el envío inmediato de un mensaje urgente.

Bloque de control de intercambios



El bloque de función **%MSG** permite gestionar los intercambios.
Su programación es opcional.

Características del bloque %MSG

Entrada (o instrucción) Reinicialización	**R**	En el estado 1, reinicializa la comunicación, %MSG.E = 0 y %MSG.D =1.
Salida enlace disponible	**%MSG.D**	En el estado 1, enlace disponible si: • Fin de emisión si emisión, • Fin de recepción (carácter de final recibido), • Error, • Reinicialización del bloque. En el estado 0, petición en curso.
Salida falla	**%MSG.E**	En el estado 1 si: • Petición incorrecta, • Tabla mal configurada • Carácter recibido incorrecto (velocidad, par., etc.) • Tabla de recepción llena (sin actualizar) En el estado 0, conexión OK.

Si se produce algún error durante la utilización de la instrucción EXCH, los bits %MSG.D y %MSG.E pasa a 1 y la palabra de sistema %SW69 contiene el código de error. Consúltese el apartado 6.2-2.

Entrada RESET (R): la puesta a 1 de esta entrada provoca la parada inmediata de la emisión en curso, la puesta a 0 de la salida Error y la puesta a 1 del bit Done. Entonces, se puede enviar un nuevo mensaje.

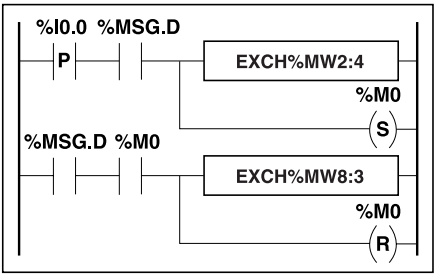
Salida ERROR (%MSG.E): esta salida se pone a 1 tanto al ocurrir un error de programación como uno de transmisión. También se pone a 1, si el número de octetos de datos definidos en la tabla de palabras asociadas a la instrucción EXCH (octeto menos significativo de la primera palabra) es superior a 128 (80 en hexadecimal). Esta salida se pone a 1, si se detecta un problema durante el intercambio. En este caso, el usuario debe verificar el cableado y ver si el equipo de destino soporta el protocolo UNI-TELWAY.

Salida DONE (%MSG.D): cuando esta salida está a 1, el TSX Nano está listo para enviar un nuevo mensaje. Se aconseja utilizar este bit al enviar varios mensajes. Si no se utiliza, se pueden perder mensajes.

Envío de varios mensajes sucesivos

La activación de un bloque de mensajes en el programa de aplicación se efectúa ejecutando la instrucción EXCH. Se emite el mensaje, si el bloque de mensajes no está activo (%MSG.D=1). Si se envían varios mensajes en el mismo ciclo, sólo se emite el primero. El usuario deberá gestionar mediante el programa la transmisión de varios mensajes.

Ejemplo 4: envíos succesivos de 2 mensajes.

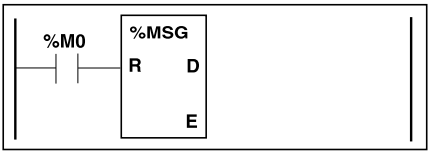



```
LDR   %I0.0
AND   %MSG.D
[EXCH %MW2:4]
S     %M0
LD    %MSG.D
AND   %M0
[EXCH %MW8:3]
R     %M0
```

Reinicialización de intercambios

La anulación de un intercambio se obtiene mediante la activación de la entrada (o de la instrucción) R. Dicha entrada inicializa la comunicación y pone a 0 la salida %MSG.E y a 1 la salida %MSG.D. Es posible reinicializar un intercambio si se detecta una falla.

Ejemplo:
```
BLK %MSG
LD    %M0
R
END_BLK
```



Casos específicos

- **Incidencia de un arranque en frío:** (%S0=1) provoca la reinicialización de la comunicación.

- **Incidencia de un arranque en caliente:** (%S1=1) no tiene incidencia.
- **Incidencia de un paso a STOP:** si un mensaje está en curso de emisión, el autómata termina su transferencia y actualiza las salidas de los bloques %MSG.D y %MSG.E.

Características del modo ASCII:

El modo de comunicación en cadena de caracteres (ASCII) se selecciona enlazando varias patillas de la toma terminal (para una información adicional consúltese la sección F de este manual). El bit de estado %S100 se pone a 1 cuando el TSX Nano está en modo ASCII. También confirma que el cable está conectado.

Las tres utilizaciones posibles de esta instrucción son las siguientes:

- **Emisión**
- **Emisión/Recepción**
- **Recepción**

El tamaño máximo de las tramas emitidas y/o recibidas es de 128 octetos.
La tabla de palabras asociada a la instrucción EXCH se compone de tablas de emisión y recepción.

Octeto más significativo	Octeto menos significativo	
Comando	Longitud LgE / LgR	Control
Octeto emitido 1	Octeto emitido 2	Tabla de emisión
...	...	
...	Octeto emitido n	
Octeto emitido n+1		
Octeto recibido 1	Octeto recibido 2	Tabla de recepción
...	...	
...	Octeto recibido p	
Octeto recibido p+1		

El octeto Longitud (LgE) contiene la longitud que se va a emitir, y luego se substituye por el número de caracteres recibidos (LgR) al final de la recepción.

El octeto de comando debe contener uno de los siguientes valores:

- 0: Emisión
- 1: Emisión seguida de una recepción
- 2: Recepción

La tabla sólo puede ser de tipo %KWi en el caso de una emisión.
La recepción se interrumpirá en cuanto se reciba el octeto de fin de trama (1). El usuario podrá modificar el valor de este octeto (menos significativo de la palabra sistema %SW68). El valor por defecto de esta palabra es H'0D' (retorno de carro).

(1) Atención:
El sistema no gestiona el tiempo de espera de recepción

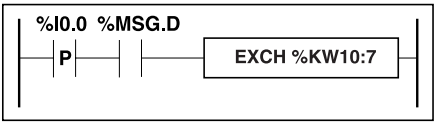
Envío de un mensaje hacia un equipo en ASCII: sólo emisión

El contenido de la tabla de palabras asociada a la instrucción EXCH que se requiere para enviar datos en ASCII (hacia una pantalla de vídeo, una impresora, ...) se detalla a continuación:

Octeto más significativo	Octeto menos significativo
0 (emisión)	Longitud del mensaje (octetos)
Datos en ASCII	

La longitud máxima del mensaje será de 128 octetos.

Ejemplo 1: visualizar el mensaje "FALLO 10" en una impresora que utiliza el protocolo ASCII.




```
LDR   %I0.0
AND   %MSG.D
[EXCH %KW10:7]
```

Contenido de la tabla de palabras:

Palabra	Contenido	Octeto más significativo	Octeto menos significativo
%KW10	12	0	Longitud LgE en octetos
%KW11	'DE'	Texto en ASCII	
%KW12	'FA'		
%KW13	'UT'		
%KW14	' 1'		
%KW15	'0 '		
%KW16	16#0A0D	Salto de línea	Retorno de carro

Emisión/Recepción en ASCII

Al finalizar la emisión, el TSX Nano pasa a espera de recepción y una vez recibida la respuesta, la vuelve a copiar en el área de %MWi contigua a la tabla de emisión si el estado de la recepción es OK y si la longitud de la pregunta (LgE) y de la respuesta (LgR) es inferior al área de %MWi reservada (longitud L). Si no es este el caso, el %MSG.E pasa a 1.
El fin de recepción se realiza cuando se detecta el código de fin (H'0D' por defecto pero se puede modificar en %SW68) o la tabla de recepción está llena.
No hay gestión de tiempo de espera de recepción.

Observación

El TSX Nano V1 o V2 ne puede recibir mensajes en ASCII.

El contenido de la tabla de palabras asociada a la instrucción EXCH que se requiere para emitir/recibir datos en ASCII se detalla a continuación:

Octeto más significativo	**Octeto menos significativo**	
1 (emisión/recepción)	Longitud LgE o LgR	Control
Octeto que se va a emitir 1	Octeto que se va a emitir 2	Tabla de emisión
...	...	
...	Octeto que se va a emitir n	
Octeto que se va a emitir n+1		
Octeto recibido 1	Octeto recibido 2	Tabla de recepción
...	...	
...	Octeto recibido p	
Octeto recibido p+1	Código de fin (H'0D')	

Observaciones

Las palabras de tipo %KWi están prohibidas.

Cuando finaliza el intercambio (carácter de fin de trama recibido), el octeto de longitud de emisión (LgE) contiene el número de caracteres recibidos (LgR). **Por ello, será necesario actualizar el octeto de longitud de emisión antes de cada intercambio.**

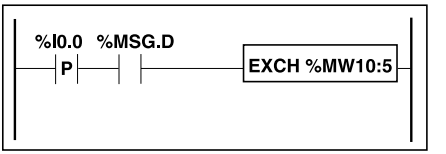
El área de recepción de mensajes está siempre alineada en la palabra dependiendo del área de emisión.

Ejemplo:

Palabra	Más signif.	Menos signif.
%MW10	16#0001	16#0007
%MW11	'V'	'A'
%MW121	'L'	'U'
%MW13	'E'	' '
%MW14	':'	no utilizado

Programa asociado:

LDR %I0.0
AND %MSG.D
[EXCH %MW10:9]



Este programa transmite la siguiente trama: VALUE: es decir 7 octetos y espera la recepción de una respuesta (8 octetos máximo).
Se puede acceder a los caracteres recibidos en las palabras %MW15 a %MW18.

Cálculo de la longitud de la tabla de palabras

LgE = 7
LgR = 8

$$L = 1 + \frac{LgE + LgE\%2}{2} + \frac{LgR + LgR\%2}{2} = 9$$

Observación

Los Nano-autómatas TSX 07 2• ••••, no pueden recibir mensajes en ASCII.

Recepción de un mensaje de un equipo en ASCII

Al ejecutar el bloque EXCH cuya definición de parámetros se realiza en recepción, el TSX Nano pasa a espera de recepción y copia la respuesta en el área de %MWi si el estado de la recepción es OK y si la longitud de la respuesta (LgR) es inferior al área de %MWi reservada (longitud L). Si no es éste el caso, el bit %MSG.E pasa a 1.
El fin de recepción se realiza cuando se detecta el código de fin (16#0D por defecto pero se puede modificar en %SW68) o la tabla de recepción está llena.
No hay gestión de tiempo de espera de recepción.

Observación

El TSX Nano V1 o V2 no puede recibir mensajes en ASCII.

El contenido de la tabla de palabras asociada a la instrucción EXCH que requiere al recibir datos en ASCII se detalla a continuación:

Octeto más significativo	**Octeto menos significativo**	
2 (recepción)	0 (1)	Control
Octeto recibido 1	Octeto recibido 2	Tabla de recepción
...	...	
...	Octeto recibido p	
Octeto recibido p+1	Código de fin (H'0D')	

Observación

Las palabras de tipo %KWi están prohibidas.

El tipo de comunicación se gestiona en la primera palabra de la tabla.

(1) En modo recepción, este octeto menos significativo no se tiene en cuenta.

Control de intercambios

El control de los intercambios se realiza con la ayuda del bloque de función %MSG y de la palabra de sistema %SW69.
Después de cada intercambio, la %SW69 (confirmación del bloque EXCH) se actualiza y toma uno de los siguientes valores:

- 0: Intercambio OK.
- 1: Tabla de emisión demasiado extensa (LgE>128).
- 2: Tabla de emisión demasiado corta (LgE=0).
- 3: Tabla de palabras demasiado corta (1).
- 7: Comando incorrecto en ASCII (octeto de comando <> 0, 1 ó 2).
- 8: No utilizado
- 9: Error de recepción (problema de formato de comunicación (velocidad, paridad)).
- 10: Tabla %KWi prohibida en recepción o emisión/recepción.

$$(1)\quad L < 1 + \frac{LgE + LgE\%2}{2} + \frac{LgR + LgR\%2}{2}$$

con L en palabras
LgE y LgR en octetos

Diálogo con un equipo UNI-TELWAY

La tabla de palabras asociada a la instrucción EXCH utilizada para enviar una petición hacia un equipo UNI-TELWAY, como variadores de velocidad ATV, equipos de interfaz hombre-máquina (CCX 17 o XBT) se compone de tablas de emisión y recepción.

Octeto más significativo	**Octeto menos significativo**	
Dirección destinatario	Longitud LgE / LgR (octetos)	Control
Código de categoría	Código de petición	Tabla de emisión
Primera palabra (PF)	Primera palabra (Pf)	
...	...	
Palabra n-1 (PF)	Palabra n-1 (Pf)	
Palabra n (PF)	Palabra n (Pf)	
00 (forzado)	Código respuesta recibido	Tabla de recepción
Dato recibido 2	Dato recibido 1	
...	...	
Dato recibido p-1	...	
	Dato recibido p	

El tamaño máximo de los mensajes emitidos y recibidos es de 128 octetos.
El área de recepción del mensaje se alinea siempre a la palabra dependiendo del área de emisión.

El bloque EXCH lee la longitud que se va a emitir (LgE) en el campo de longitud de emisión.
Cuando finaliza el intercambio, escribe en dicho campo la longitud del mensaje recibido (LgR).
Por ello, se deberá actualizar el octeto de longitud de emisión antes de cada intercambio.

El código de respuesta recibido se inscribe en el menos significativo de la primera palabra de la tabla de recepción. El más significativo de esta palabra, se fuerza a 0. Los eventuales datos siguientes se alinean en la próxima palabra.

Importante

A partir de los TSX Nano V3, el orden de emisión de los datos de una petición UNI-TE pasa a ser más significativo y luego menos significativo.
Las aplicaciones que funcionan en los TSX Nano V1 o V2 deberán modificarse para que tengan en cuenta esta inversión si se cargan en un TSX Nano V3.

B

Unitelway Maestro

En este modo, el TSX Nano gestiona normalmente dos equipos distribuidos en 5 direcciones esclavas. Se pueden controlar 2 equipos y un puesto de programación PL7 07, si el terminal de programación está configurado en una sola dirección.

El TSX Nano no gestiona el encaminamiento de esclavo a esclavo.

El TSX Nano Maestro puede emitir una petición hacia cualquier esclavo de dirección de 1 a 5, con la ayuda del bloque EXCH. Utiliza la dirección de origen 0.254.16.

La dirección de destino codificada en la tabla de palabras asociada al bloque EXCH, deberá ser una de las siguientes:

- 0: Emisión de una petición hacia el esclavo 4 (compatibilidad TSX07 2.).
- 1 a 5: Emisión y recepción de una petición hacia un esclavo de dirección de 1 a 5.

Si la dirección de destino vale 0, las características del bloque EXCH serán las siguientes:

- La memoria intermedia de recepción está inutilizable,
- La tabla puede situarse en el área %KWi,
- El bit %MSG.D pasa a 1 cuando se recibe la respuesta del esclavo,
- La respuesta del esclavo se ignora.
- Únicamente pueden utilizarse las peticiones Escritura y Datos no solicitados.

Si la dirección de destino está comprendida entre 1 y 5:

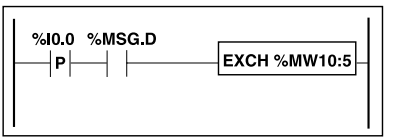
- La tabla de recepción es obligatoria (1 palabra mínimo),
- La tabla de palabras deberá situarse en el área %MWi,
- El bit %MSG.D pasa a 1 cuando se recibe la respuesta del esclavo,
- La respuesta del esclavo se copia de nuevo en la tabla de recepción.

Ejemplo de utilización:

Emisión de la petición " Lectura de palabra " %MW513 (16#0201) hacia un equipo situado en la dirección 2:

Palabras	Más significativo	Menos significativo
%MW10	02	04
%MW11	07	04
%MW12	02	01

Programa asociado:
LDR %I0.0
AND %MSG.D
[EXCH %MW10:5]



El bloque EXCH utiliza %MWi:L como parámetros:

- i indica el número de la primera palabra de la tabla
- L indica el número de palabras de la tabla de palabras.

Una vez que los bits %MSG.D = 1 y %MSG.E = 0, la tabla contiene los siguientes datos:

Palabra	Más signif.	Menos signif.
%MW10	02	**04**
%MW11	07	04
%MW12	02	01
%MW13	**00**	**34**
%MW14	'**AB**'	'**CD**'

Los datos en **negrita** significan:

- 4 octetos recibidos,
- código de respuesta recibido = 16#0034,
- valor de la palabra %MW513 = 16#ABCD.

Unitelway Esclavo

Cualquier equipo (local o remoto) puede interrogar al servidor del sistema del TSX Nano Esclavo utilizando como dirección de destino Ad0 (servidor).

Un TSX Nano esclavo puede emitir (cliente) una petición hacia cualquier equipo Maestro o Esclavo (dirección de 0 a 98) mediante el bloque EXCH (cuando el maestro es un autómata TSX 37/57).

La dirección de destino codificada en la tabla de palabras asociada al bloque EXCH, deberá estar comprendida entre 100 y 198 (cuando el maestro es un autómata TSX 47/67/87/107).
Las características del bloque EXCH son las siguientes:

- La tabla de recepción es obligatoria (mínimo 1 palabra),
- La tabla de palabras deberá estar situada en el área %MWi,
- El bit %MSG.D pasa a 1 cuando se recibe la respuesta del esclavo,
- La respuesta se vuelve a copiar en la tabla de recepción.

Ejemplo de utilización:

Emisión de la petición " Lectura de palabra " %MW513 hacia el maestro:

Palabra	Más signif.	Menos signif.
%MW10	00	04
%MW11	07	04
%MW12	02	01

Programa asociado:

LDR %I0.0
AND %MSG.D
[EXCH %MW10:5]

El bloque EXCH utiliza %MWi:L como parámetros:
- i indica el número de la primera palabra de la tabla
- L indica el número de palabras de la tabla de palabras.

Cuando los bits %MSG.D = 1 y %MSG.E = 0, la tabla contiene los siguientes datos:

Palabras	Más signif.	Menos signif.
%MW10	00	**04**
%MW11	07	04
%MW12	02	01
%MW13	**00**	**34**
%MW14	**'AB'**	**'CD'**

Los datos en **negrita** significan:

- 4 octetos recibidos,
- código de respuesta recibido = 16#0034,
- valor de la palabra %MW513 = 16#ABCD.

Otro ejemplo:

Emisión de la petición " Lectura de palabra " %MW513 hacia el esclavo 32:

Palabra	Más signif.	Menos signif.
%MW10	20	04
%MW11	07	04
%MW12	02	01

Programa asociado:

```
LDR    %I0.0
AND    %MSG.D
[EXCH %MW10:5]
```

El bloque EXCH utiliza %MWi:L como parámetros:

- i indica el número de la primera palabra de la tabla
- L indica el número de palabras de la tabla de palabras.

Cuando los bits %MSG.D = 1 y %MSG.E = 0, la tabla contiene los siguientes datos:

Palabra	Más signif.	Menos signif.
%MW10	20	**04**
%MW11	07	04
%MW12	02	01
%MW13	**00**	**34**
%MW14	'**AB**'	'**CD**'

Los datos en **negrita** significan:

- 4 octetos recibidos,
- código de respuesta recibido = 16#0034,
- valor de la palabra %MW513 = 16#ABCD.

Control de intercambios

El control de los intercambios se realiza con la ayuda del bloque de función %MSG y de la palabra de sistema %SW69.
El bit %MSG.D pasa a 1 en los siguientes casos:

- Al final de la recepción de la respuesta.
- En caso de error de transmisión (recepción negativa)
- En caso de reinicialización del bloque.
- Si no se recibe la respuesta en 7 segundos (tiempo de espera de aplicación).

El bit %MSG.E pasará a 1 en los diferentes casos de error (detallados en la palabra %SW69):

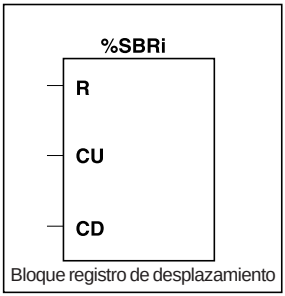
Después de cada intercambio, la palabra %SW69 (confirmación del bloque EXCH) se actualiza y toma uno de los siguientes valores:

- 0: Intercambio OK.
- 1: Tabla de emisión demasiado extensa (LgE>128).
- 2: Tabla de emisión demasiado corta (LgE=0).
- 3: Tabla de palabras demasiado corta (1).
- 4: Dirección Unitelway incorrecta (la dirección de destino no pertenece a [0...98] o [100...198] en modo UNI-TELWAY Esclavo o a [1...5] en UNI-TELWAY Maestro.
- 5: Tiempo de espera transcurrido.
- 6: Error de emisión (el destinatario responde Nach).
- 7: Comando incorrecto en ASCII (octeto de comando <> 0, 1ó 2).
- 8: No utilizado
- 9: Error de recepción (problema de formato de comunicación (velocidad, paridad)).
- 10: Tabla %KWi prohibida en recepción o emisión/recepción.

(1) $L < 1 + \frac{LgE + LgE\%2}{2} + \frac{LgR + LgR\%2}{2}$

con L en palabras
LgE y LgR en octetos

3.4-7 Bloques de función de registro de desplazamiento de bit %SBRi



Bloque registro de desplazamiento

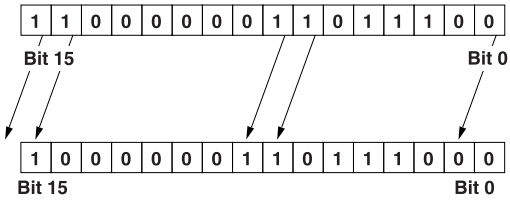
Un registro con desplazamiento de bit permite introducir informaciones binarias (0 ó 1) y hacerlas evolucionar en un sentido u otro.

Características		
Número de registro	**%SBRi**	0 a 7
Bit de registro	**%SBRi.j**	Bits 0 a 15 (j=0 a 15) del registro de despl. Puede verificarse con una instrucción de compr.y escribirse mediante la de asignación.
Entrada (o instrucción) puesta a 0	**R**	Sobre flanco ascendente, puesta a 0 de los bits %SBRi.j del registro.
Entrada (o instrucción)	**CU**	Sobre flanco ascendente, desplazamiento a la izquierda de un bit del registro.
Entrada (o instrucción)	**CD**	Sobre flanco ascendente, desplazamiento a la derecha de un bit del registro.

Funcionamiento
Estado inicial

CU %SBRi efectúa desplazamiento hacia izquierda

El bit 15 se pierde

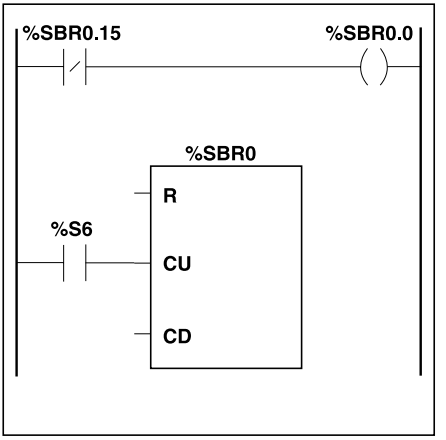


Sucede lo mismo si se solicita el desplazamiento a la derecha de un bit (del bit 15 hacia el bit 0) mediante la instrucción CD. Se pierde el bit 0.

Si la utilización de un registro de 16 bits no es suficiente, es posible mediante el programa poner varios registros en cascada.

Programación

Ejemplo: efectuar cada segundo un desplazamiento de un bit a la izquierda; el bit 0 asume el estado inverso del bit 15.



Programación reversible

```
LDN    %SBR0.15
ST     %SBR0.0
BLK    %SBR0
LD     %S6
CU
END_BLK
```

Programación no reversible

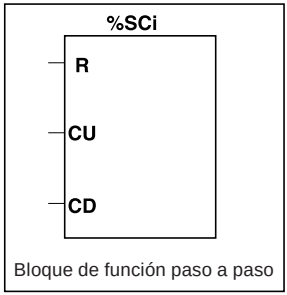
```
LDN    %SBR0.15
ST     %SBR0.0
LD     %S6
CU     %SBR0
```

Casos específicos

- **Incidencia de un arranque en frío:** (%S0=1)
  - provoca la puesta a 0 de todos los bits de la palabra registro.

- **Incidencia de un arranque en caliente:** (%S1=1) sin incidencia sobre los bits de la palabra registro.

3.4-8 Bloques de función paso a paso %SCi

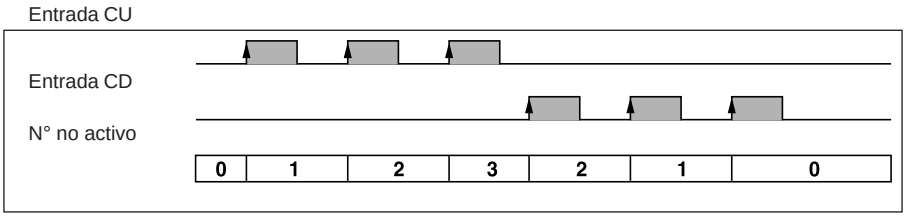


Bloque de función paso a paso

Un paso a paso es una serie de pasos a los que pueden asociarse acciones. El tránsito de un paso a otro se hace en función de eventos externos o internos. Cada vez que un paso está activo, el bit asociado se pone a 1. Sólo puede estar activo un único paso.

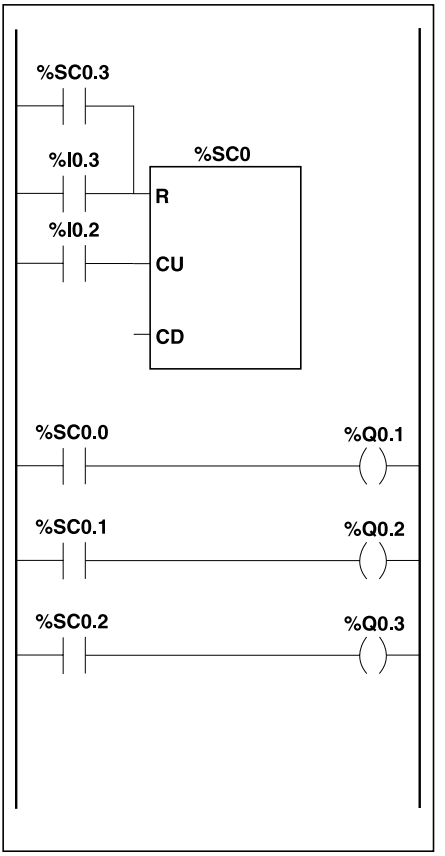
Características		
Número del paso a paso	**%SCi**	0 a 7
Bit de paso a paso	**%SCi.j**	Bits 0 a 255 (j=0 a 255) del paso a paso, puede comprobarse mediante una instrucción LD y escribirse mediante la de asignación.
Entrada (o instrucción) puesta a 0	**R**	Sobre flanco ascendente, puesta a 0 de los bits %SCi.j del paso a paso.
Entrada (o instrucción) incremento	**CU**	Sobre flanco ascendente, incremento de un paso en la función paso a paso.
Entrada (o instrucción) decremento	**CD**	Sobre flanco ascendente, disminución de un paso en la función paso a paso.

Funcionamiento



Programación

Ejemplo: programar el paso a paso 0 incrementado por la entrada %I0.2. Se pone a 0 por la entrada %I0.3 o cuando llega al paso 3.
El paso 0 controla la salida %Q0.1, el 1 la salida %Q0.2 y el 2 la salida %Q0.3.



Programación reversible

```
BLK    %SC0
LD     %SC0.3
OR     %I0.3
R
LD     %I0.2
CU
END_BLK
LD     %SC0.0
ST     %Q0.1
LD     %SC0.1
ST     %Q0.2
LD     %SC0.2
ST     %Q0.3
```

Programación no reversible

```
LD     %SC0.3
OR     %I0.3
R      %SC0
LD     %I0.2
CU     %SC0
LD     %SC0.0
ST     %Q0.1
LD     %SC0.1
ST     %Q0.2
LD     %SC0.2
ST     %Q0.3
```

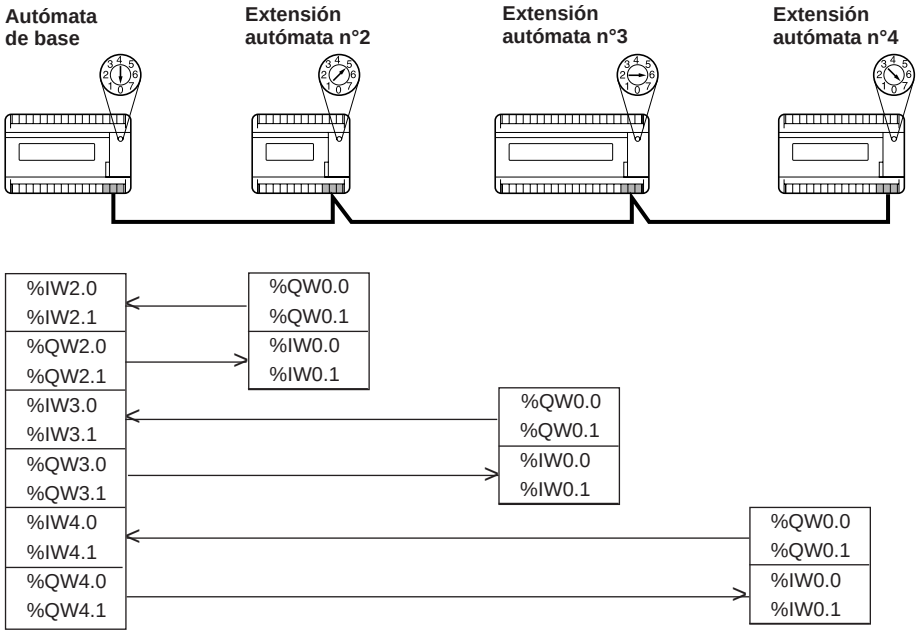
Casos específicos

- **Incidencia de un arranque en frío:** (%S0=1)
  - provoca la inicialización del paso a paso.

- **Incidencia de un arranque en caliente:** (%S1=1) sin incidencia sobre el paso a paso.

3.5 Comunicación entre autómatas

Las palabras %IW y %QW permiten el intercambio de información entre autómatas. La siguiente figura muestra para cada autómata las palabras intercambiadas.



La actualización de estas palabras de intercambio se realiza automáticamente cuando los autómatas están en ejecución (RUN). El programa de usuario se limita para cada autómata a:

- escribir en las palabras de salida %QWi.j
- leer las palabras de entrada %IWi.j

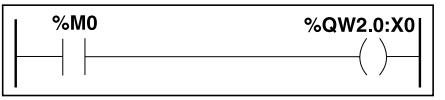
El ciclo de regeneración de las palabras IW/QW es síncrono con el ciclo de los autómatas. El bit sistema %S70 se pone a 1 cuando se realiza un ciclo completo; su puesta a 0 se efectúa desde el programa o terminal.

Los bits %S71 / %S72 y la palabra %SW71 permiten además controlar los intercambios (véase el capítulo 6).

Nota: la dirección de cada autómata se define en función de la posición del seleccionador situado en la parte delantera del autómata, su posición se tendrá en cuenta en cada conexión.

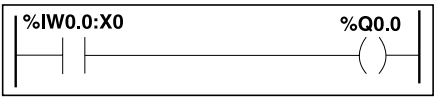
Ejemplo 1:
El autómata de base transmite a la extensión autómata n°2 una información de tipo fin de fabricación (bit %M0=1). Al recibir esta información la extensión de autómata pone en marcha una máquina mediante activación de la salida %Q0.0.

Programación de autómata de base




```
LD    %M0
ST    %QW2.0:X0
```

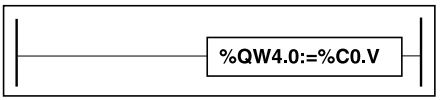
Programación de extensión autómata




```
LD    %IW0.0:X0
ST    %Q0.0
```

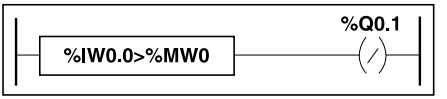
Ejemplo 2:
El autómata de base transmite a la extensión de autómata n°4 el valor actual del contador 0. Cuando este valor actual es superior al umbral contenido en la palabra %MW0 la extensión de autómata detiene una máquina mediante la desactivación de la salida %Q0.1.

Programación de autómata de base




```
LD    1
[%QW4.0:=%C0.V]
```

Programación de extensión autómata




```
LD    [%IW0.0>%MW0]
STN   %Q0.1
```

4.1 Presentación

Los autómatas TSX 07 30/31 •• •• de base, con entradas/salidas TON, de versión V3 o superior, pueden controlar módulos de entrada/salida analógicos.

Los autómatas TSX Nano permiten utilizar tres tipos de módulos analógicos:

- **Los módulos de entrada/salida analógicos TSX AMN 4000/4001**

  Estos módulos se comunican con el autómata de base mediante el enlace de extensión de E/S .
  La gestión de estos módulos se realiza mediante las palabras de intercambio %IW y %QW.

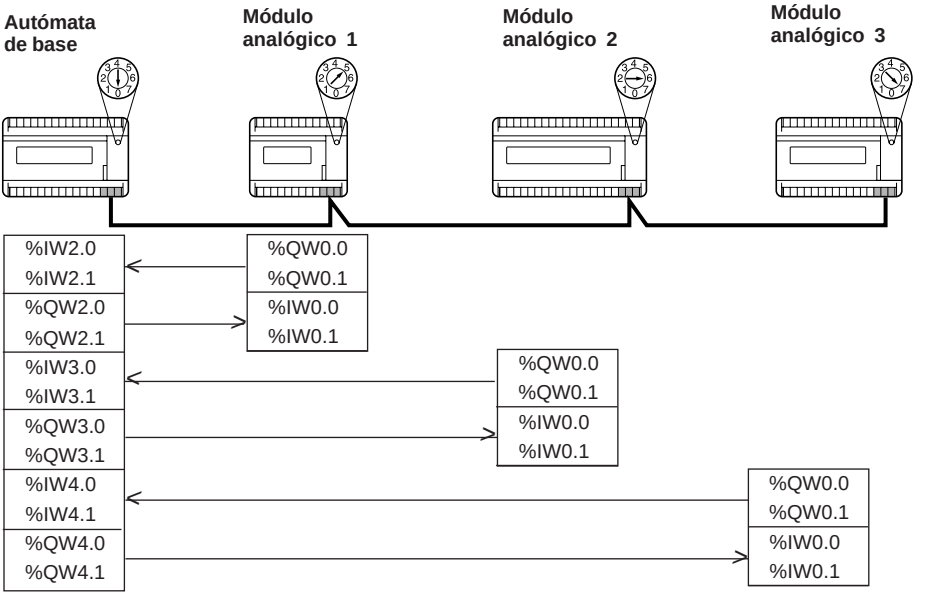
- **Módulo de entrada**:
  Estos módulos realizan una conversión de tensión/frecuencia, por lo que requieren utilizar la entrada %I0.0 en modo frecuencímetro en el autómata. Sólo se puede conectar un módulo de entrada por autómata.
  La gestión del módulo se realiza desde el programa de aplicación mediante las palabras de sistema %SW100 y %SW101.

- **Módulo de salida:**
  Estos módulos realizan una conversión PWM/tensión, mediante la integración de la señal proporcionada en la salida %Q0.0 del autómata (en modulación de amplitud de impulsos). Sólo se puede conectar un módulo por autómata (autómatas de base dotados de salidas estáticas).
  La gestión del módulo se realiza desde el programa de aplicación mediante las palabras de sistema %SW102 y %SW103.

4.2 Módulos analógicos TSX AMN 4000/4001

4.2-1 Principio de funcionamiento de los módulos analógicos

Las palabras %IW y %QW permiten el intercambio de datos de aplicación entre un autómata de base y los módulos analógicos TSX AMN 400•.
Estos datos, limitados a cuatro palabras (dos de lectura y dos de escritura) tanto para el autómata de base como para los módulos analógicos, pueden intercambiarse en los dos sentidos.
La figura siguiente muestra las palabras intercambiadas para cada uno de los autómatas.



La actualización de estas palabras de intercambio se realiza automáticamente cuando el autómata de base está en ejecución (RUN). El programa de usuario se limita a llevar a cabo las acciones siguientes para cada uno de los autómatas:

- escribir en las palabras de salida %QWi.j
- leer las palabras de entrada %IWi.j

El ciclo de renovación de las palabras %IW/%QW tiene lugar de forma síncrona con el ciclo de los autómatas. El bit de sistema %S70 se establece en 1 cuando se ha completado un ciclo; la nueva puesta a 0 de este bit se realiza por programa o terminal.

Nota
La posición del selector que se encuentra en la parte delantera del autómata define la dirección de cada uno de los autómatas. Su posición se tiene en cuenta al conectar la tensión.

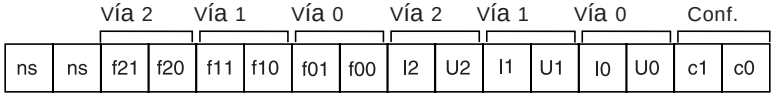
4.2-2 Programación de los módulos analógicos

La programación de un módulo de entrada/salida analógico TSX AMN 400• se realiza mediante 2 palabras de intercambio %QWi.0, %QWi.1 y 2 palabras de intercambio %IWi.0, %IWi.1.

Palabra de intercambio %QWi.0

Esta palabra permite configurar las vías de entradas analógicas de un módulo. Contiene:

- el número de vías de entrada utilizadas,
- el tipo de entrada para cada vía,
- el tipo de filtro para cada vía.



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

C1	C0	Número de vías de entrada utilizadas
0	0	Ninguna
0	1	Vía 0
1	0	Vía 0 y vía1
1	1	Vía 0, vía 1 y vía 2

In	Un	Tipo de entrada para la vía n
0	0	Entrada de tensión +/-10 V
0	1	Entrada de tensión 0..10 V
1	0	Entrada de corriente 0..20 mA
1	1	Entrada de corriente 4..20 mA

fn1	fn0	Tipo de filtro para la vía n (filtro digital de primer orden)
0	0	Filtrado hard
0	1	75 ms
1	0	300 ms
1	1	1,5 s

ns: no significativo

Palabra de intercambio %QWi.1

Esta palabra de 16 bits contiene el valor de la salida analógica 0 de un módulo.

N° de bit	Significado
De x0 a x14	Valor de la salida 0 codif. en 15 bits
x15	Bit de signo

Palabra de intercambio %IWi.0

Esta palabra de 16 bits contiene el valor de la entrada analógica 0 y el estado de un módulo.

N° de bit	Significado
De x0 a x10	Valor de la entrada 0 codif. en 11 bits
x11	Bit de signo de la vía 0
x12 = 1	Falla de autocalibración
x13 = 1	Rebasamiento de topes vía 0
x14 = 1	Rebasamiento de topes vía 1
x15 = 1	Rebasamiento de topes vía 2

Palabra de intercambio %IWi.1

Esta palabra de 16 bits contiene el valor de la entrada analógica 1 o bien los valores de las entradas analógicas 1 y 2 de un módulo.

N° de bit	Significado cuando las entradas 0 y 1 están configuradas	Significado cuando las entradas 0,1 y 2 están configuradas
De 0 a 6	Valor de la entrada 1 codif. en 15 bits	Valor de la entrada 1 codif. en 7 bits
7		Bit de signo de la vía 1
De 8 a 14		Valor de la entrada 2 codif. en 7 bits
15	Bit de signo de la vía 1	Bit de signo de la vía 2

Gama de valores de los módulos de entradas

Tipo entrada	Gama de valores 7 bits + signo	11 bits + signo	Umbral detección rebasamiento de topes inferior y superior
-10 V / +10 V	-125 ..125	-2000 .. 2000	+/- 2,5% de toda la graduación
0 / 10 V	0 .. 125	0 .. 2000	+/- 2,5% de toda la graduación
0 / 20 mA	0 .. 125	0 .. 2000	+/- 2,5% de toda la graduación
4 / 20 mA	25 .. 125	0 .. 2000	+/- 2,5% de toda la graduación

Toda falla de rebasamiento de un tope se indica mediante la puesta a 1 de uno de los bits de estado de la palabra %IWi.0 del módulo. Para la gama 4/20 mA, se declara una falla si la corriente es inferior a 3,5 mA, pero el módulo puede elevar los valores hasta [-0,5 mA].

Gama de valores de los módulos de salida

Tipo de salida	Valores escritos en la palabra %QWi.1	Valores máx. de rebasamiento de los topes inferior y superior
-10 V / +10 V	-2000 .. 2000	+/- 2,5% de toda la graduación
0 / 10 V	0 .. 2000	+/- 2,5% de toda la graduación
0 / 20 mA	0 .. 2000	+/- 2,5% de toda la graduación
4 / 20 mA	400 .. 2000	+/- 2,5% de toda la graduación

En caso de falla interna, el módulo coloca la salida en modalidad de reposición (0 V para la salida de tensión y 0 mA para la salida de corriente, sea cual sea la gama).

Atención

Cuando el autómata está en STOP, la salida analógica conserva el valor anterior a la puesta en STOP. En este caso, no existe reposición automática.

4.2-3 Uso de las palabras %IW en el programa de usuario

Para que las instrucciones del PL7-07 puedan utilizar las palabras de intercambio %IW, éstas deben convertirse, según el caso, al formato 15 bits + signo.

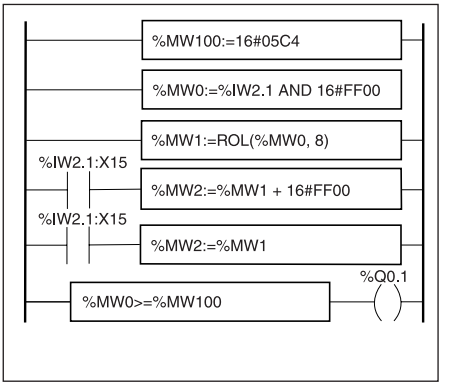
La tabla siguiente muestra las operaciones de conversión que se deben insertar en el programa de usuario:

Valor del bit de signo	Pal. 7 bits + signo	Pal. 11 bits + signo	N° de vía	Conversión al formato 15 bits + signo
0		%IWi.0		%MW0:= %IWi.0 AND 16#0FFF
1		%IWi.0		%MW0:= %IWi.0 OR 16#F000
0	%IWi.1		1	%MW0:= %IWi.1 AND 16#00FF
1	%IWi.1		1	%MW0:= %IWi.1 AND 16#00FF %MW1:= %MW0 OR 16#FF00
0	%IWi.1		2	%MW0:= %IWi.1 AND 16#FF00 %MW1:= ROR (%MW0,8)
1	%IWi.1		2	%MW0:= %IWi.1 AND 16#FF00 %MW1:= ROR (%MW0,8) %MW2:= %MW1 OR 16#FF00

Nota:
Las palabras internas %MW0, %MW1 y %MW2 se utilizan como ejemplos de variables en las operaciones de conversión.

Ejemplo de programa de usuario
Comparación del valor analógico de la vía 2 del módulo 1, todas las entradas del cual están configuradas con un umbral contenido en la palabra %MW100 (valor hexadecimal 16#05C4).
Puesta a 1 de la salida Q0.1 del autómata de base cuando el valor es igual o superior al umbral.

```
LD   1
[%MW100 := 16#05C4]
LD   1
[%MW0 := %IW2.1 AND 16#FF00]
LD   1
[%MW1 := ROR ( %MW0 , 8 )]
LD   %IW2.1:X15
[%MW2 := %MW1 OR 16#FF00]
LDN %IW2.1:X15
[%MW2 := %MW1]
LD   [%MW0 >= %MW100]
ST   %Q0.1
```



4.2-4 Diagnóstico del estado de comunicación con los módulos analógicos

Este diagnóstico es idéntico al de una extensión de autómata y se indica mediante la palabra %SW71 (véase el apartado 6.2-2 de la sección B).

4.3 Módulos de entrada analógica TSX ASN •••

4.3-1 Configuración de las entradas analógicas

La utilización de la entrada %I0.0 en modo frecuencímetro para la conexión del módulo de entrada analógica requiere la siguiente definición de parámetros:

- **Tipo contador rápido:** deberá posicionarse en Frecuencia para utilizar la entrada como frecuencímetro
- **Frecuencia máxima:** deberá posicionarse en 10 KHz

4.3-2 Programación de las entradas analógicas

La programación del módulo de entrada se realiza utilizando las dos palabras de sistema %SW100 y %SW101 y validando el contador rápido %FC (véase ejemplo en apartado 4.4-4).

- **%SW100:** palabra de comando de las funciones de entrada analógica
- **%SW101:** valor de entrada analógica confirmada

La selección del modo de funcionamiento se realiza escribiendo desde el programa la palabra %SW100. El valor confirmado de la entrada analógica se puede leer en la palabra %SW101.

El sistema pondrá estas dos palabras cuando se produzca un rearranque en frío.

El sistema dispone mediante selección del modo de funcionamiento de un servicio de graduaciones. Estas graduaciones estarán comprendidas en una graduación de 0 a +10 000 para los módulos unipolares (módulos de entrada 4/20 mA y 0/10 V) y -10 000 a
+10 000 para los módulos bipolares (-10/+10 V).

%SW100	Funcionamiento	Gama de valor de %SW101
0	Invalidación del servicio entrada analógica en %I0.0	0
1	Funcionamiento sin graduaciones	0...1 000 Período de medida 125 ms
2	Graduaciones para gama unipolar (4/20 mA, 0/10 V)	0...10 000 Período de medida de 125 ms
3	Graduaciones para gama bipolar (-10/+10 V)	-10 000 ... +10 000 Período de medida de 125 ms
4	Graduaciones para gama unipolar (4/20 mA, 0/10 V)	0...10 000 Período de medida de 500 ms
5	Graduaciones para gama bipolar (-10/+10V)	-10 000 ... +10 000 Período de medida de 500 ms

El valor analógico bruto o graduación está disponible en %SW101 si %SW100 está escrito en un valor de 1 a 5. La validez de esta medida puede controlarse mediante el bit de sistema %SW111:X3 (puesta a 1 por el sistema si la medida es válida).
Si la aplicación pone a cero el bit de sistema %SW111:X3, se inicia un servicio de confirmación analógica y las confirmaciones de la medida siguen realizándose con el autómata en STOP.

La medida de frecuencia bruta está disponible en la palabra %FC,V asociada a la entrada %I0.0 pero se trata de la función de medida del período de medida (ej: la graduación completa de 8 KHz dará 1000 para 125 ms y 4000 para 500 ms). Por lo tanto, se recomienda, para simplificar la aplicación, utilizar preferentemente la palabra de sistema %SW101.

Observación

El período de medida puede modificarse en curso de funcionamiento mediante la reescritura de la palabra %SW100 pero se desaconseja este modo de utilización ya que la primera medida después del cambio de período puede ser errónea.

Utilización de módulos de entrada analógica en autómatas V2

Se pueden utilizar módulos de entrada analógica en autómatas V2,
TSX 07 2•••, respetando las siguientes normas:

- utilización de la entrada %I0.0 como frecuencímetro (validación del funcionamiento mediante la instrucción IN %FC)
- configuración del período de medida por escritura mediante la aplicación del bit %SW111:X2.
  %SW111:X2=0 mide cada segundo (por defecto)
  %SW111:X2=1 mide cada 100 ms
- la imagen del valor de la entrada analógica está disponible en el objeto %FC,V; la interpretación del valor se realiza de la siguiente manera:

Gama	Fórmula
0/10 V	$U(V) = 1,25 \times (\%FC,V \times 10^{-3})$
4/20 mA	$I(mA) = 2 \times [(\%FC,V \times 10^{-3}) + 2]$
-10/+10 V	$U(V) = 2,5 \times [(\%FC,V \times 10^{-3}) - 4]$

Nota
En el módulo 4/20 mA, la frecuencia es nula entre 0 y 4 mA.

4.3-3 Tiempo de respuesta de las entradas analógicas

El tiempo de respuesta TRE en confirmación de una entrada analógica, entre la variación efectiva de la intensidad eléctrica en los bornes del módulo y la correspondencia en la palabra %SW101 del valor medido depende esencialmente del período de medida seleccionado (125/500 ms) y en menor manera del tiempo de ciclo del autómata. La variación de la intensidad eléctrica interviene de forma negativa en este tiempo de respuesta.

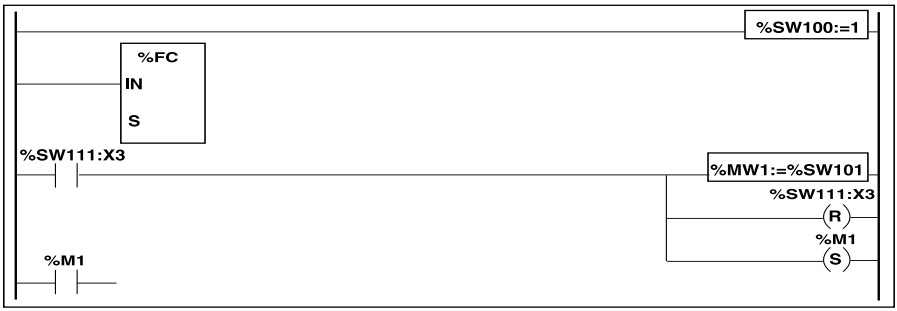
- Para una confirmación en el período de 125 ms: TRE es inferior a 500 ms.
- Para una confirmación en el período de 500 ms: TRE es inferior a 1,2 s.

Nota

El sistema tendrá en cuenta un cambio de modo de funcionamiento (cambio de %SW100) en cada ciclo o inmediatamente en un flanco ascendente del "IN %FC". Las medidas se encadenan en tiempo real unas tras otras en el período seleccionado (125 ms o 500 ms). El resultado de la última medida efectuada se transmite al principio del ciclo de autómata en %SW101. Esta palabra no cambia de valor durante el ciclo del autómata.

4.3-4 Ejemplo de programación de entradas analógicas

```
(* VALIDACIÓN SEVICIOS ENTRADAS ANALÓGICAS *)
LD      1                   (* ENTRADA ANA MODO 0..1000 EN125 MS *)
[%SW100 := 1]               (* CONSIDERACIÓN DEL MODO SELECCIONADO
*)
BLK %FC
LD  1
IN
END_BLK
(* CONFIRMACIÓN MEDIDA *)
LD      %SW111:X3           (* MEDIDA VÁLIDA *)
[%MW1 := %SW101]            (* MEMO MEDIDA *)
R       %SW111:X3           (* INDICACIÓN MEDIDA *)
S       %M1                 (* ÍNDICE MEDIDA VÁLIDO *)
(* EXPLOTACIÓN DE LA MEDIDA, SEGÚN APLICACIÓN *)
LD      %M1
```



4.3-5 Características de las entradas analógicas

Tipo	Valor de entrada	Valor %SW101 período 125 ms	Resolución (1) /incremento	Valor %SW101 período 500 ms	Resolución(1) /incremento
4/20 mA	4 mA 12 mA 20 mA	0 5000 10000	16 µA/10 lsb	0 5000 10000	4 µA/2,5 lsb
0/10 V	0 V 10 V	0 10000	10 mV/10 lsb	0 10000	2,5 mV/2,5 lsb
-10/+10 V	-10 V +10 V	-10000 10000	20 mV/10 lsb	-10000 10000	5 mV/2,5 lsb

Los valores de %SW101 corresponden al modo de funcionamiento con graduaciones.

(1) Resolución: valor mínimo de variación de la entrada para obtener una variación de medida. Dicha variación varía por pasos denominados incremento.

4.4 Módulos de salida analógica TSX AEN •••

4.4-1 Configuración de las salidas analógicas

La utilización de la salida %Q0.0 para la conexión del módulo de salida analógica requiere la siguiente definición de parámetros:

- **Salida %Q0.0:** se utilizará en modulación de amplitud de impulso %PWM.
- **Base de tiempo:** se posicionará en 0,1 ms.
- **Preselección:** se posicionará imperativamente en 249 para que el funcionamiento siga siendo válido después de un arranque en caliente. El ajuste de este parámetro sólo es útil para los autómatas de base TSX Nano V3.0.

4.4-2 Programación de las salidas analógicas

La programación del módulo de salida se realiza mediante dos palabras de sistema %SW102 y %SW103 y validando la salida %PWM (véase ejemplo en apartado 4.3-4).

- **%SW102:** palabra de comando/estado de las funciones de la salida analógica
- **%SW103:** valor de salida analógica que se va a generar

La selección del modo de funcionamiento se realiza escribiendo desde la aplicación la palabra %SW102 y el valor que se va a generar en la salida analógica en la palabra %SW103.

El sistema pondrá a cero estas dos palabras cuando se produzca un rearranque en frío.

El sistema ofrece, gracias a la selección del modo de funcionamiento, un servicio de graduaciones. Estas graduaciones estarán comprendidas en una escala de 0 a +10 000 para los módulos unipolares (módulos de entrada 4/20 mA y 0/10 V) y de -10 000 a +10 000 para los módulos bipolares (-10/+10 V).

%SW102	Funcionamiento	Gama de valores de %SW103
0	Invalidación del servicio de salida analógica en %Q0.0	No utilizado
1	Funcionamiento sin graduaciones	5...249
2	Graduaciones para gama unipolar (4/20 mA, 0/10 V)	0...10 000
3	Graduaciones para gama bipolar (-10/+10 V)	-10 000 ... +10 000

La resolución efectiva de las salidas analógicas es de 245 puntos.

Cuando el valor escrito en %SW103 es inferior al valor mínimo (por ejemplo inferior a 0 en modo unipolar), será el valor mínimo de la gama el que se aplique al módulo de salida.
Cuando el valor escrito en %SW103 es superior al valor máximo (por ejemplo superior a 10000 en modo unipolar), será el valor máximo de la gama el que se aplique al módulo de salida.
Estos dos tipos de error de programación no se señalarán en la aplicación.

Importante

En las condiciones de retorno de las salidas TON, la generación del PWM cesará y la señal no llegará a los módulos de salida.
Por ello, los módulos bipolares tomarán el valor más bajo (-10 V).
El usuario deberá tener en cuenta este modo de retorno.

4.4-3 Tiempo de respuesta de las salidas analógicas

El tiempo de respuesta TRS de una salida analógica, entre la escritura de la consigna en la palabra %SW103 y la espera de la tensión (y/o corriente) que corresponde a los bornes del módulo, depende de la amplitud de variación y del tiempo de ciclo del autómata.

- Para una variación en la graduación completa, TRS será inferior a 500 ms.

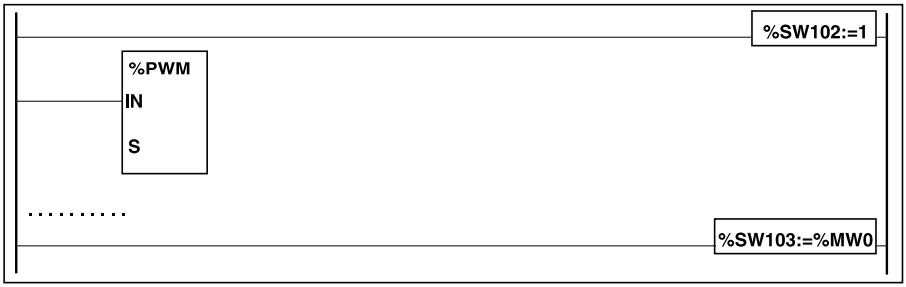
Cuanto más corto sea el tiempo de ciclo del autómata más corto será este tiempo y débil la variación de consigna. Para un tiempo de ciclo de 10 ms y una variación de 1/10 de la graduación completa, este tiempo de respuesta bajará a 50 ms aproximadamente.

Nota

Se tendrá en cuenta un cambio de modo de funcionamiento (cambio de %SW102) por flanco ascendente de la entrada IN del PWM (ejecución de la instrucción "IN %PWM") o por cambio de consigna %SW103. El sistema tendrá en cuenta un cambio de consigna (%SW103) en cada ciclo y será efectivo en el siguiente ciclo de aplicación (tiempo máximo: 3 ms).

4.4-4 Ejemplo de programación de salidas analógicas

```
(* VALIDACIÓN SERVICIOS SALIDAS ANALÓGICAS *)
LD      1
[%SW102 := 1]           (* SALIDA ANA MODO BRUTO 5..249  *)
IN %PWM                 (* CONSIDERACIÓN DEL MODO *)

(* GENERACIÓN CONSIGNA EN PWM *)
....                    (* CÁCULO CONSIGNA SEGÚN APLICACIÓN *)
LD      1
[%SW103 := %MW0]        (* APLICACIÓN CONSIGNA  EN SALIDA *)
```



4.4-5 Características de las salidas analógicas

Tipo	Valor para generar en salida	Valor %SW103 para escribir por aplicación	Resolución módulo de salida	Resolución lsb (1) %SW103
4/20 mA	4 mA	0	65 µA	40
	12 mA	5000		
	20 mA	10000		
0/10 V	0 V	0	40 mV	40
	5 V	5000		
	10 V	10000		
-10/+10 V	-10 V	-10000	81 mV	81
	0 V	0		
	+10 V	10000		

Los valores de %SW103 corresponden al modo de funcionamiento con graduaciones.
(1) Resolución LSB: variación mínima que se va a aplicar en %SW103 para obtener una variación del módulo de salida igual a la resolución.

B

5.1 Presentación

Los autómatas TSX 07 • 1 16/24 •• poseen un reloj mediante el cual se pueden elaborar tres funciones:

- **Programador temporal,** que permite controlar acciones a horas predefinidas o calculadas,
- **Registrador temporal,** que permite el fechado de sucesos y la **medición de la duración.**

El ajuste de la fecha y hora del reloj-calendario del TSX 07 se efectúa o por configuración o por programa. Su funcionamiento queda asegurado durante 30 días aunque el autómata esté desconectado si se ha cargado la batería durante al menos 6 horas ininterrumpidamente antes de la parada del autómata.

El reloj tiene un formato de 24 horas y distingue los años bisiestos.

5.2 Programador temporal

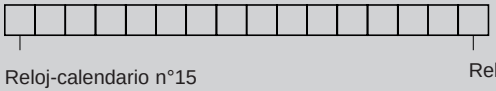
El programador temporal permite controlar las acciones en horas y fechas predefinidas o calculadas.
Se pueden utilizar hasta 16 bloques horarios que realizan cada uno esta función. Estos bloques no necesitan ninguna introducción de programa; son configurables (véase los modos operativos, sección C).

5.2-1 Características

Número de bloque horario	**RTC : n**	n=0 a 15
Salida	**Q**:	Asignación de la salida activada por el reloj-calendario: %Mi o %Qj.k. Esta salida pasa al estado 1 cuando la fecha y hora actuales están comprendidas entre las marcas de inicio de período activo y las de fin de período activo.
Fecha inicio	**JJ:MMM**	Indica el día (1 a 31) y el mes (enero a diciembre) del inicio de validación del reloj-calendario.
Fecha fin	**JJ:MMM**	Indica el día (1 a 31) y el mes (enero a diciembre) del fin de validación del reloj-calendario.
Día	**LMXJVSD**	Indica los días de activación (L: Lunes, ..., D:Domingo).
Hora inicio	**hh:mm**	Indica en horas (0 a 23) y minutos (0 a 59) el inicio de activación del reloj-calendario.
Hora fin	**hh:mm**	Indica en horas (0 a 23) y minutos (0 a 59) el fin de activación del reloj-calendario.

B

La palabra de sistema %SW114 permite validar con los bits (a 1) o inhibir (a 0) el funcionamiento de cada uno de los bloques.



Todos los bits de esta palabra de sistema están a 1 por defecto (o después de un rearranque en frío): **su gestión por programa es opcional**.

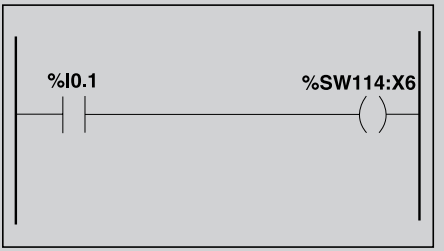
Observación:

• Si se asigna a varios bloques la misma salida (%Mi o %Qj.k), es la "O lógica" de los resultados de cada uno de los bloques la que finalmente se asigna a este objeto (permite tener varios "márgenes de funcionamiento" para una misma salida).

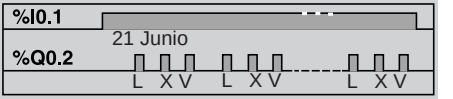
Ejemplo : configuración de un reloj-calendario, riego programado para los meses de verano.

- RTC **6**: reloj-calendario n°6,
- Q : **%Q0.2**: salida activada por el reloj-calendario
- **21 -jun -> 21-sept**: período de validación
- **L•X•V••**: días de validación (lunes, miércoles y viernes),
- **21 : 00 - 22 : 00**: margen horario de activación

```
RTC:6          Q: %Q0.2
 21-Jun  -->  21-Sept
      L•X•V••
 21 : 00  -  22 : 00
```




```
LD    %I0.1
ST    %SW114:X6
```



En este ejemplo, el usuario puede inhibir el reloj-calendario por medio de un interruptor o un detector de humedad cableado en la entrada %I0.1.

Nota:

Es importante controlar el estado del bit %S51 que señala cualquier falla del reloj-calendario.

5.2-2 Control de la fecha y hora por programa

Fecha y hora están igualmente disponibles en las palabras de sistema %SW50 a %SW53 (véase el apartado 6.2).Por lo tanto se puede llevar a cabo la programación a través del autómata realizando comparaciones aritméticas entre la fecha y hora actual y los valores inmediatos o las palabras %MWi (o %KWi) que pueden contener consignas.

5.3 Registrador temporal

La función registrador temporal permite memorizar la fecha y hora de aparición de un evento.

Las palabras de sistema %SW50 a %SW53 (véase el apartado 5.2) contienen la fecha y hora actuales, en formato BCD, útil para la visualización o el envío hacia un periférico.

Para fechar un evento, basta con utilizar las operaciones de asignación para transferir el contenido de las palabras de sistema en palabras internas y, luego, tratar estas palabras internas (por ejemplo: envío de la instrucción EXCH a las pantallas).

Ejemplo:


```
...
LDR    %I0.1
[%MW12:4 :=%SW50:4]
...
```

Una vez detectado el evento, la tabla de palabras contiene:

Código:	**Octeto más significativo**	**Octeto menos significativo**
%MW12	Segundos	Día de la semana (1)
%MW13	Horas	Minutos
%MW14	Meses	Días
%MW15	Siglos	Años

Ejemplo:	lunes 19 abril 1994
En hexa	13 H, 40 mn, 30 s
3000	30 s, 0=lunes
1340	13 H, 40 mn
0419	4=abril, 19
1994	1994

(1) con 0=lunes, 1=martes, 2=miércoles, 3=jueves, 4=viernes, 5=sábado, 6=domingo

Lectura de la fecha y hora de la última parada por palabras de sistema

Las palabras de sistema %SW54 a %SW57 (véase el apartado 6.2) contienen la fecha y hora de la última parada; la palabra %SW58 contiene el código que indica la causa de la última parada, en formato BCD.

5.4 Ajuste del reloj-calendario

B

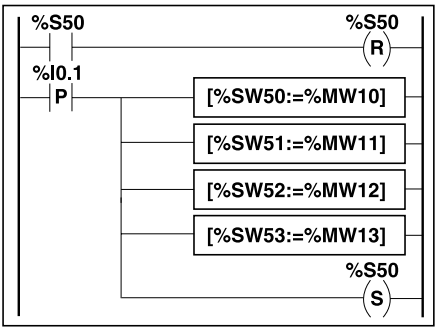
5.4-1 Actualización de la fecha y hora desde el terminal

El modo TSX del terminal de programación permite un acceso simple y rápido a la actualización de la fecha y hora (véase los modos operativos, sección C).

5.4-2 Actualización de la fecha y hora por palabras de sistema

Las palabras de sistema ofrecen dos otras posibilidades de actualización de fecha y hora:

Actualización por palabras de sistema %SW50 a %SW53
(véase el apartado 6.2)
Para ello, el bit %S50 debe ponerse a 1. Este bit:
- anula la actualización de las palabras %SW50 a %SW53 por reloj interno,
- transmite los valores escritos en las palabras %SW50 a %SW53 al reloj interno.




```
LD    %S50
R     %S50
LDR   %I0.1
[%SW50:=%MW10]
[%SW51:=%MW11]
[%SW52:=%MW12]
[%SW53:=%MW13]
S     %S50
```

Las palabras %MW10 a %MW13 deben contener la nueva fecha y hora en el formato BCD y corresponder al código de palabras %SW50 a 53.

La tabla de palabras debe contener las nuevas fecha y hora.

Código:	Octeto más significativo	Octeto menos significativo
%MW10	Segundos	Día de la semana(1)
%MW11	Horas	Minutos
%MW12	Meses	Días
%MW13	Siglos	Años

Ejemplo : lunes 19 abril 1994

Hexa	13 H, 40 mn, 30 s
3000	30 s, 0=lunes
1340	13 H, 40 mn
0419	4=abril, 19
1994	1994

(1) con 0=lunes, 1=martes, 2=miércoles, 3=jueves, 4=viernes, 5=sábado, 6=domingo

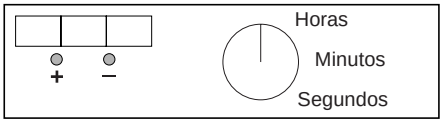
Con el fin de garantizar la actualización del reloj-calendario de un TSX Nano V1, V2 o V3 al cambiar de siglo, el autómata debe permanecer encendido mientras se pasa del 1999 al año 2000. Sin embargo, existe también a posibilidad de actualizar el reloj-calendario desde el programa añadiendo a la aplicación:

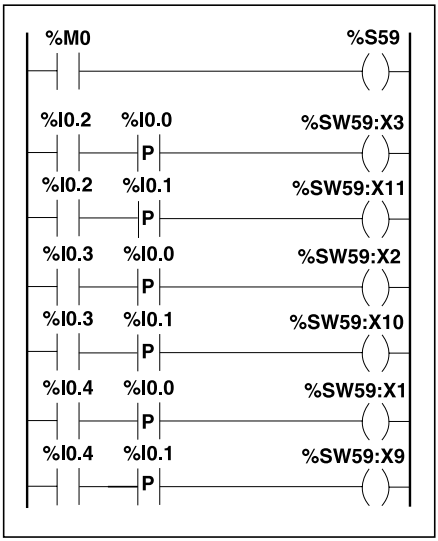
```
LD [%SW53]=16#1900
ST %S50
[%SW53:=16#2000]
```

Actualización por la palabra de sistema %SW59

Otra posibilidad de actualización es el bit de validación %S59 y la palabra de ajuste %SW59.
La puesta a 1 del bit %S59 garantiza la validación del ajuste de la fecha y hora actuales por la palabra %SW59. Esta palabra, que se describe en el apartado 6.2, permite aumentar o reducir cada uno de los componentes de la fecha y hora sobre un flanco ascendente.

Ejemplo : se realiza un flanco ascendente para poder modificar la hora, los minutos y los segundos del reloj interno.






```
LD    %M0
ST    %S59
LD    %I0.2          (hora)
ANDR  %I0.0
ST    %SW59:X3
LD    %I0.2
ANDR  %I0.1
ST    %SW59:X11
LD    %I0.3          (minuto)
ANDR  %I0.0
ST    %SW59:X2
LD    %I0.3
ANDR  %I0.1
ST    %SW59:X10
LD    %I0.4          (segundo)
ANDR  %I0.0
ST    %SW59:X1
LD    %I0.4
ANDR  %I0.1
ST    %SW59:X9
```

- El conmutador Horas/Minutos/Segundos controla las entradas %I0.2, %I0.3 y %I0.4.
- La entrada %I0.0, botón pulsador +, realiza el incremento.
- La entrada %I0.1, botón pulsador -, realiza el decremento.

B

6.1 Bits de sistema

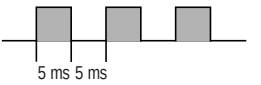
6.1-1 Lista de bits de sistema

Bit	Función	Est. inicial	Gestión
%S0	1 = arranque en frío (restablecimiento alimentación con pérdida de datos)	0	S o U->S
%S1	1 = arranque en caliente (restablecimiento alimentación sin pérdida de datos)	0	S o U->S
%S4, %S5	Base de tiempo 10 ms, 100 ms	-	S
%S6, %S7	Base de tiempo 1 s, 1 mn	-	S
%S8	0 = mantenimiento de salidas en STOP	1	U
%S9	1 = puesta a 0 de salidas del autómata en RUN	0	U
%S10	0 = falla de entradas/salidas	1	S
%S11	1 = desbordamiento del control de secuencia	-	S
%S13	1 = primer ciclo después de puesta en RUN	1	S
%S17	1 = desbordamiento cálculo sin signo o desplazamiento circular	0	S->U
%S18	1 = desbordamiento o error aritmético	0	S->U
%S19	1 = desbordamiento del período exploración	0	S->U
%S20	1 = desbordamiento de índice	0	S->U
%S21	1 = inicialización del Grafcet provoca: puesta a 0 de las etapas y a 1 de las etapas iniciales	0	U->S
%S22	1 = puesta a cero del Grafcet	0	U->S
%S23	1 = validación del posicionamiento previo del GRAFCET, el mantenimiento a 1 provoca la inmovilización del GRAFCET	0	U->S
%S49	1 = petición de arranque cada 10 s de las salidas estáticas desconectadas por sobreintensidad o cortocircuito	0	U
%S50	1 = ajuste del reloj-calendario	0	U
%S51	1 = reloj-calendario no inicializado o en falla 0 = fecha y hora actualizadas	0	S
%S59	1 = ajuste de la fecha actual	0	U
%S69	1 = visualización de bits internos	0	U
%S70	1 = actualización intercambio %IW/%QW en extensión. Tratamiento petición Modbus.	0	S
%S71	1 = intercambio en enlace de extensión	0	S
%S72	0 = exploración de autómatas de extensión	0	U
%S100	Estado del /DPT	-	S
%S118	1 = falla de autómata de base	0	S
%S119	1 = falla de extensión de entradas/salidas	0	S

S = gestión por el sistema,U = gestión por el usuario,U->S = puesta a 1 por el usuario, puesta a 0 por el sistema, S—>U = puesta a 1 por el sistema, puesta a 0 por el usuario.

6.1-2 Descripción detallada de los bits de sistema

Los autómatas TSX Nano disponen de bits de sistema %Si que indican los estados del autómata o permiten controlar su funcionamiento.
Estos bits pueden comprobarse en el programa de usuario para detectar todos los eventos de funcionamiento que deben implicar un procedimiento particular de tratamiento. Algunos deben ponerse en su estado inicial o normal desde el programa. Sin embargo, los bits de sistema que se pone a su estado inicial o normal desde el sistema no deben ponerse desde el programa o el terminal.

Bits de sistema	**Función**	**Designación**
%S0	Rearr. en frío	Normalmente en 0, pasa a 1 por: • restablecimiento de la alimentación con pérdida de datos (falla de batería) • programa de usuario, • terminal (modo Ajuste), Este bit se pone a 1 durante el primer ciclo completo. Volverá a ponerse a 0 antes del ciclo siguiente. Funcionamiento: véase el apartado 7.1 sección A.
%S1	Rearr. en caliente	Normalmente en 0, se pone a 1 por: • restablecimiento de alimentación con salvaguarda de datos, • programa de usuario, • terminal (modo Ajuste). El sistema lo pone a 0 al final del primer ciclo completo y antes de la actualización de las salidas. Funcionamiento: véase el apartado 7.1 sección A .
%S4 **%S5** **%S6** **%S7**	Bases de tiempo 10 ms 100 ms 1 s 1 min	Bits cuyo cambio de estado temporiza un reloj interno. Son asíncronos en relación al ciclo del autómata. Ejemplo: %S4 

Bits de sistema	Función	Designación
%S8	Seguridad de salidas	Inicialmente a 1. Puede ponerse a 0 por el programa o terminal (modo Ajuste): • estado 1: provoca la puesta a cero de las salidas del autómata, en caso de ejecución anormal del programa o de STOP del autómata, • estado 0: mantiene las salidas en el estado definido en caso de ejecución anormal del programa o de autómata en STOP.
%S9	Puesta a cero de salidas	Normalmente a 0. Puede ponerse a 1 por el programa o terminal (modo Ajuste): • estado 1: provoca el forzado al estado 0 de las salidas del autómata en RUN, • estado 0: las salidas se actualizan de forma normal.
%S10	Falla E/S	Normalmente a 1. Se pone a 0 cuando se detecta una falla E/S del autómata de base o de extensión (configuración no conforme, falla de intercambio, falla de hardware, disyunción de salidas estáticas protegidas). Los bits %S118 y %S119 indican la falla del autómata y las palabras %SW118 y %SW119 precisan la naturaleza de la falla (véase apart. 5.2) El bit %S10 pasa a 1 cuando desaparece la falla.
%S11	Desbordamiento del control secuencia	Normalmente a 0. Se pone a 1 por el sistema cuando el tiempo de ejecución del programa excede el tiempo de ciclo máximo (control de secuencia del programa). El desbordamiento del control de secuencia provoca el paso a STOP del autómata.
%S13	Primer ciclo	Normalmente a 0. Se pone a 1 por el sistema en el primer ciclo después de la puesta a RUN del autómata.
%S17	Rebasamiento de la capacidad (carry)	Normalmente a 0. Se pone a 1 por el sistema: • en caso de rebasamiento de la capacidad en una opera ción aritmética sin signo (retenida). • en un desplazamiento circular; señala la salida del bit a 1. Debe ser comprobado por el programa de usuario después de cada operación en que haya riesgo de rebasamiento. Si éste ocurre, el usuario debe ponerlo a 0.

Bits de sistema	Función	Designación
%S18	Desbordamiento o error aritmético "Overflow"	Normalmente a 0. Se pone a 1 en caso de desbordamiento de capacidad en una operación de 16 bits, es decir: • resultado superior a + 32767 o inferior a - 32768, • división por 0, • raíz cuadrada de un número negativo. • conversión BTI o ITB no significativa (valor BCD excede límites) Debe ser comprobado por el programa de usuario, después de cada operación en que haya riesgo de desbordamiento. Si éste ocurre, el usuario debe ponerlo a 0.
%S19	Desbordamiento del período de exploración (explor. periódica)	Normalmente a 0. Se pone a 1 por el sistema en caso de exceder el período de ejecución (tiempo de ejecución de la tarea superior al período definido por el usuario en la configuración o programado en %SW0). Este bit se pone a 0 por el usuario.
%S20	Desbordamiento de índice	Normalmente a 0. Se pone a 1 cuando la dirección del objeto indexado es inferior a 0 o superior a 255. Debe ser comprobado por el programa de usuario después de cada operación en que haya riesgo de desbordamiento. Si éste ocurre, el usuario debe ponerlo a 0.
%S21	Inicialización Grafcet	Normalmente a 0. Se pone a 1 mediante: • el rearranque en frío, %S0=1, • el programa de usuario en el tratamiento preliminar únicamente, cuando se utiliza la instrucción S o la bobina Set, • el terminal. En el estado 1, inicializa GRAFCET. Las etapas activas se desactivan y las iniciales se activan. Se pone a 0 por el sistema después de la inicialización del Grafcet.

Bits de sistema	Función	Designación
%S22	Puesta a cero del Grafcet	Normalmente a 0. Sólo puede ponerse a 1 por el programa en el tratamiento preliminar. A 1, provoca la desactivación de todas las etapas activas del Grafcet. El sistema lo pone a 0 al comienzo de la ejecución del tratamiento secuencial.
%S23	Preposicionamiento e inmovilización del Grafcet	Normalmente a 0. Sólo puede pasar a 1 por el programa de usuario en el tratamiento preliminar. A 1, permite validar el preposicionamiento del Grafcet. Mantenido a 1, provoca la inmovilización del Grafcet (interrupción del gráfico). El sistema lo pone a 0 al comienzo de la ejecución del tratamiento secuencial, con el fin de asegurar la evolución del Grafcet a partir de la situación fijada.
%S49	Reactivación de las salidas estáticas	Normalmente a 0. El usuario lo pone a 1 para solicitar una reactivación cada 10 s desde la detección de la falla de las salidas estáticas desconectadas por sobreintensidad o cortocircuito.
%S50	Actualización de la fecha y hora por palabras %SW50 a 53.	Normalmente a 0. Puede ponerse a 1 o a 0 por el programa o el terminal. • A 0: acceso a la fecha y hora mediante la lectura de las palabras de sistema %SW50 a 53. • A 1: actualización de la fecha y hora mediante la escritura de las palabras de sistema %SW50 a 53.
%S51	Fecha y hora del reloj	• A 0, la fecha y la hora están actualizadas. • A 1, la fecha y hora no están actualizadas. Cuando este bit está a 1, el usuario no ha actualizado la fecha ni la hora o la batería está defectuosa.
%S59	Actualización de la fecha y hora por palabras %SW59	Normalmente a 0. Puede ponerse a 1 o a 0 por el programa o el terminal. • A 0: el sistema no gestiona la palabra de sistema %SW59. • A 1: el sistema gestiona los flancos en la palabra %SW59 para el ajuste de la fecha y hora actuales.
%S69	Visualización de los bits internos en la parte delantera del autómata	Normalmente en estado 0. Puede ponerse a 1 o a 0 por el programa o el terminal. • A 0: los estados de las E/S se visualizan en los indicadores del autómata. • A 1: los estados de 8 bits internos (TSX 07 10, 14 y 16 E/S) o 16 bits internos (TSX 07 20 y 24 E/S) se visualizan en los indicadores del autómata (apartado 1.9, sección A). El indicador derecho parpadea para indicar que se ha seleccionado la visualización de los bits internos.

Bits de sistema	Función	Designación
%S70	Regeneración de las palabras de intercambio Tratamiento petición Modbus	Para el autómata de base, este bit se pone a 1 tras efectuar un ciclo completo de envío de palabras de intercambio %IW/%QW hacia las extensiones de autómata. Para cada extensión de autómata, este bit se pone a 1 cuando la extensión ha recibido y enviado las palabras de intercambio con el autómata de base. Este bit se pone a 0 por programa o terminal. Este bit se pone a 1 por tratamiento de una petición Modbus. Puede ser explotado por el usuario. Este bit se pone a cero por programa o terminal.
%S71	Intercambios en el enlace de extensión	Inicialmente en estado 0. Pasa a 1 cuando se detecta un intercambio en el enlace de extensión. Este bit se pone en estado 0 cuando no se realiza ningún intercambio en el enlace de extensión. La palabra %SW71 del autómata de base presenta la lista y el estado de las extensiones presentes.
%S72	Exploración de las extensiones del autómata	Únicamente en autómatas de versiones anteriores o iguales a V2.2. Normalmente a 0. Puede ponerse a 0 por el programa o el terminal. • Estado 0: exploración de las extensiones de autómata • Estado 1: inhibición de la exploración
%S100	Estado de la señal /DPT	Indicación del estado del fleje INL/DPT en la toma consola: • Fleje ausente: protocolo UNI-TELWAY maestro (%S100 = 0) • Fleje presente: (/DPT al 0 V) protocolo definido en configuración de la aplicación (%S100 = 1).
%S118	Falla de autómata	Normalmente en estado 0. Se pone en 1 cuando el autómata de base detecta una falla de entradas/salidas o la disyunción de las entradas estáticas protegidas. La palabra %SW118 determina la naturaleza de la falla. El bit %S118 se pone a 0 cuando desaparece la falla.
%S119	Falla de autómata	Normalmente en estado 0. Se pone a 1 cuando la extensión de entradas/salidas detecta una falla de entradas/salidas o la disyunción de las salidas estáticas protegidas. La palabra %SW119 determina la naturaleza de la falla. El bit %S119 se pone a 0 cuando desaparece la falla.

6.2 Palabras de sistema

6.2-1 Lista de palabras de sistema

Mot	Función	Gestión
%SW0	Valor del período de exploración del autómata (en periódico)	U
%SW11	Duración del control de secuencia del programa	S
%SW14	Tiempo de espera UNITELWAY	S y U
%SW15	Versión e IE del autómata	S
%SW30	Tiempo del último ciclo de exploración del autómata	S
%SW31	Tiempo de ciclo máximo de exploración del autómata	S
%SW32	Tiempo de ciclo mínimo de exploración del autómata	S
%SW50 **%SW51** **%SW52** **%SW53**	Función reloj-calendario: palabras con los valores actuales de la fecha y hora (en BCD) %SW50 = segundos y días de la semana %SW51= hora y minuto %SW52 = mes y día %SW53= siglo y año	S y U
%SW54 **%SW55** **%SW56** **%SW57**	Función de reloj-calendario: palabras con la fecha y hora del última falla de alimentación o parada del autómata (en BCD) %SW54 = segundos y código de falla %SW55 = hora y minuto %SW56 = mes y día %SW57= siglo y año	S
%SW58	Código de identificación de la última parada	S
%SW59	Ajuste de la fecha actual	U
%SW67	Valor del carácter de fin de trama Modbus modo ASCII	U
%SW68	Valor del carác. de fin de trama (recepción) modo ASCII (toma TER)	U
%SW69	Código de error del bloque EXCH	S
%SW70	Función y tipo de autómata TSX Nano	S
%SW71	Equipos presentes en el enlace de extensión	S
%SW76	Temporizador en 1 ms	S
%SW77	Temporizador en 1 ms	S
%SW78	Temporizador en 1 ms	S
%SW79	Temporizador en 1 ms	S
%SW100	Palabra de control de la función de entrada de módulo analógico	U
%SW101	Valor de entrada del módulo analógico confirmado	S
%SW102	Palabra de control de la función de salida de módulo analógico	U
%SW103	Valor de salida del módulo analógico que se va a generar	U
%SW110	Valor de contaje leído	S
%SW111	Funciones de contaje rápido	S y U
%SW112	Valor de punto de ajuste analógico n°0	S

Palabra	Función	Gestión
%SW113	Valor de punto de ajuste analógico n°1	S
%SW114	Validación de los bloques reloj-calendario	U
%SW118	Palabra de estado del autómata de base	S
%SW119	Palabra de estado de la extensión de entradas/salidas	S

S = controlado por el sistema,

U = controlado por el usuario.

6.2-2 Descripción detallada de las palabras de sistema

Los autómatas TSX Nano disponen de las siguientes palabras de sistema:

Palabras sistema	Función	Designación
%SW0	Período de exploración	Modifica el período de exploración del autómata definido en configuración, por le programa de usuario o por el terminal (modo Ajuste).
%SW11	Duración del control de secuencia	Lee la duración del control de secuencia (150 ms).
%SW14	Tiempo espera Unitelway	Modifica el valor del tiempo de espera UNITELWAY, desde el programa de usuario (véase sección F apartado 1.6)
%SW15	Versión e IE del autómata	Permite, desde las versiones V3.1, conocer la versión del autómata (octeto más significativo) y su IE (octeto menos significativo). Ej.: 0x000: versión anterior a V3.1 0x3119: autómata de versión V3.1 e IE: 25
%SW30	Último tiempo de ejecución (1)	Indica el tiempo de ejecución del último ciclo de exploración del autómata (en ms).
%SW31	Tiempo de exploración máximo (1)	Indica el tiempo de ejecución del ciclo más largo de exploración del autómata desde el último arranque en frío (en ms).
%SW32	Tiempo de exploración mínimo (1)	Indica el tiempo de ejecución del ciclo más corto de exploración del autómata desde el último arranque en frío (en ms).
%SW50 **%SW51** **%SW52** **%SW53**	Función de reloj-calendario	Palabras de sistema que contienen los valores actuales de la fecha y la hora (en BCD): %SW50: SSXN Segundos y día de la semana con (N= 0 para lunes a 6 para domingo) %SW51: HHMM Hora y minutos, %SW52: MMDD Mes y día, %SW53: SSAA Siglo y año. El sistema controla estas palabras cuando el bit %S50 está a 0. Estas palabras pueden escribirse desde el programa de usuario o desde el terminal cuando el bit %S50 está en 1
%SW54 **%SW55** **%SW56** **%SW57**	Función de reloj-calendario	Palabras de sistema que contienen la fecha y hora de la última falla de alimentación o parada del autómata (en BCD): %SW54 : segundos y día de la semana, %SW55 : hora y minutos, %SW56 : mes y día, %SW57 : siglo y año.

(1) Este tiempo corresponde al tiempo transcurrido entre el inicio (confirmación de entradas) y el fin (actualización de salidas) de un ciclo de exploración.

Palabras sistema	Función	Designación
%SW58	Código de la última parada	Menciona el código que muestra la causa de la última parada: 1= paso de RUN a STOP por el terminal 2= parada por falla de programa (desbordamiento de la tarea de autómata) 4= corte de alimentación 5= parada por falla hardware
%SW59	Ajuste de la fecha actual	Contiene dos series de 8 bits para ajustar la fecha actual. La acción se realiza siempre en flanco ascendente del bit. Esta palabra se valida por el bit %S59. **Incremento** / **Decremento** / **Definición de parámetros** bit 0 / bit 8 / día de la semana bit 1 / bit 9 / segundos bit 2 / bit 10 / minutos bit 3 / bit 11 / horas bit 4 / bit 12 / días bit 5 / bit 13 / mes bit 6 / bit 14 / año bit 7 / bit 15 / siglo
%SW67	Fin de trama Modbus	Define los parámetros de 'LF' de fin de trama en Modbus en modo ASCII. El sistema escribe esta palabra en 16#000A por arranque en frío. El usuario puede modificar esta palabra desde el programa o en Ajuste cuando el maestro utilice un carácter de fin de trama diferente de 16#000A.
%SW68	Fin de trama Recepción Modo ASCII	Define los parámetros del valor del octeto de fin de trama en ASCII. La recepción se detiene cuando se recibe este octeto. El valor por defecto es 16#000D.
%SW69	Código error bloque EXCH	En caso de error durante la utilización del bloque EXCH, los bits de salida %MSG.D y %MSG.E pasan a 1. Esta palabra de sistema contiene el código de error. Los valores posibles son: 0: Sin error, intercambio correcto 1: Memoria intermedia de emisión demasiada extensa 2: Memoria intermedia de emisión insuficiente 3: Tabla demasiado reducida 4: Dirección Unitelway incorrecta (modo Unitelway únicamente) 5: Tiempo de espera transcurrido (modo Unitelway únicamente) 6: Error de emisión (modo Unitelway únicamente) 7: Comando ASCII incorrecto (modo ASCII únicamente) 8: No utilizado 9: Error de recepción (modo ASCII únicamente) 10: Tabla %KWi prohibida. Se posiciona a 0 cada vez que se utiliza el bloque EXCH.

Palabras sistema	Función	Designación
%SW70	Dirección y tipo de autómata	Contiene la siguiente información: • bit 0: 0= Modelo TSX 07 3L ••28 1= Otros modelos • bit 2: 1= presencia de reloj-calendario • bit 4 bit 3 Tipo de autómata TSX Nano 0 0 TSX Nano de 6 entradas/4 salidas (10 E/S) 0 1 TSX Nano de 9 entradas/7 salidas (16 E/S) 1 0 TSX Nano de 14 entradas/10 salidas (24 E/S) 1 1 TSX Nano de entradas alternativas (16 E/S) • bits 7,6 y 5: dirección del autómata (copia del selector de código de dirección). Si está presente una extensión de E/S: • bits 12 y 11: tipo de extensión de E/S (mismo código que los bits 3 y 4) • bits 13: 1 = extensión de E/S presente Los bits inutilizados están a 0.
%SW71	Equipos presentes en enlace de extensión	Indica el estado de la comunicación de cada extensión presente con el autómata de base: bit 1: extensión de E/S bit 2: extensión de autómata o módulo analógico n°2 bit 3: extensión de autómata o módulo analógico n°3 bit 4: extensión de autómata o módulo analógico n°4 Bit en 0 si la extensión está ausente, sin alimentación o en falla. Bit en 1 si la extensión está presente y se realizan intercambios con el autómata de base.
%SW76 a %SW78	Palabras descontadores 1 ms	Estas 4 palabras sirven de temporizadores de 1 ms. El sistema las descuenta individualmente cada milisegundo si el valor es positivo. Por lo tanto 4 descontadores de tiempo por milisengundo, es decir un margen de explotación de 1 ms a 32767 ms. La puesta a 1 del bit 15 interrumpe el decremento.
%SW100	Entrada analógica	Palabra de comando de funciones de entrada analógica. Valor: 0 Entrada analógica sin validación Valor: 1 Funcionamiento sin graduaciones Valor: 2 Graduación de la gama unipolar (período de 125 ms) Valor: 3 Graduación de la gama bipolar (período 125 ms) Valor: 4 Graduación de la gama unipolar (período 500 ms) Valor: 5 Graduación de la gama bipolar (período 500 ms) La escritura de esta palabra se realizará desde la aplicación.

Palabras sistema	Función	Designación
%SW101	Entrada analógica	Palabra que contiene el valor de entrada analógica confirmado. La escala de valores del funcionamiento seleccionado en %SW100. %SW100=0 %SW101=0 %SW100=1 %SW101 varía de 0 a 1000 %SW100=2 ó 4 %SW101 varía de 0 a 10000 %SW100=3 ó 5 %SW101 varía de -10000 a 10000
%SW102	Salida analógica	Palabra de comando de las funciones de la salida analógica. Valor: 0 Funcionamiento %PWM normal Valor: 1 Funcionamiento sin graduaciones Valor: 2 Graduaciones de la gama unipolar Valor: 3 Graduaciones de la gama bipolar %PWM analógica La escritura de esta palabra se realiza desde la aplicación.
%SW103	Salida analógica	Palabra que contiene el valor que se aplicará en la salida analógica. La escala de valores depende del funcionamiento seleccionado en %SW102. %SW102=0 %SW103=0 %SW102=1 %SW103 comprendida entre 5 y 249 %SW102=2 %SW103 comprendida entre 0 y 10000 %SW102=3 %SW103 comprendida entre -10000 y 10000 La escritura de esta palabra se realiza desde la aplicación
%SW110	Contaje/ descontaje	Valor leído del contador en flanco ascendente de entrada %I0.4.
%SW111	Contaje rápido	bit 0: sentido de desplazamiento (1=contaje, 0=descontaje) bit 1: 1= validación de las salidas directas bit 2: 1= selección de la base de tiempo del frecuencímetro (1=100 ms, 0=1 s) bit 3: 1= regeneración de %FC en frecuencia (señala también la validez del valor confirmado en el módulo de entrada analógica). La puesta a 0 de este bit correrá a cargo del usuario.
%SW112	Valor punto de ajuste analógico 0	Contiene la conversión en 8 bits (0 a 255) de la posición del potenciómetro n° 0.
%SW113	Valor punto de ajuste analógico 1	Contiene la conversión en 8 bits (0 a 255) de la posición del potenciómetro n°1.
%SW114	Validación reloj-calendario	Valida o cancela el funcionamiento del reloj-calendario desde el programa de usuario o desde el terminal. bit 0: 1 = validación reloj-calendario n°0 …… bit 15: 1 = validación del reloj-calendario n°15 Inicialmente todos los bloques de reloj-calendario están validados.

Palabras sistema	Función	Designación
%SW118	Estado autómata de base	Señala las fallas detectadas en el autómata de base. bit 0: 0 = disyunción de las salidas estáticas (1) bit 3: 0 = falla de alimentación del sensor bit 8: 0 = falla interno o hardware TSX Nano bit 9: 0 = falla externa o de diálogo bit 11: 0 = autómata en autocomprobraciones bit 13: 0 = falla de configuración (extensión de E/S configurada pero ausente o en falla) Los restantes bits de esta palabra están a 1 y reservados. Por ello, en un autómata sin falla, esta palabra tiene el valor: 16#FFFF.
%SW119	Estado autómata extensión de E/S	Señala las falls detectadas en el autómata de extensión de E/S (esta palabra sólo está controlada por el autómata de base). La asignación de los bits de esta palabra es idéntica a la de %SW118 salvo: • bit 13: no significativo • bit 14: ausencia de la extensión aún cuando esta presente en la inicialización.

(1) debido a un cortocircuito o sobrecarga en una de las salidas.

B

7.1 Modos de funcionamiento

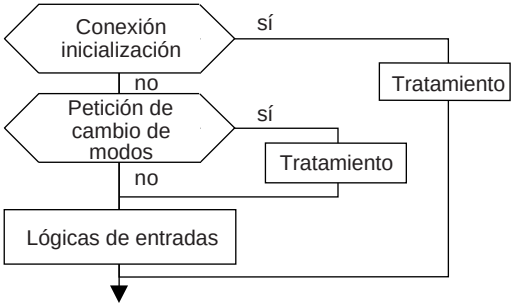
El lenguaje PL7 permite tener en cuenta tres grandes familias de modos de funcionamiento (1):
- verificación,
- funcionamiento o producción,
- parada.

Estos distintos modos de funcionamiento pueden obtenerse alrededor o a partir del Grafcet mediante las siguientes posibilidades:
- inicialización del Grafcet,
- preposicionamiento de etapas,
- mantenimiento de la situación,
- inmovilización de gráficos.

La utilización del tratamiento preliminar y de los bits de sistema asegura la gestión de los modos de funcionamiento sin complicar ni sobrecargar el programa de usuario.

Estructura del tratamiento preliminar
El cuadro sinóptico siguiente muestra la estructura con la que deberá dotarse el tratamiento preliminar con el fin de efectuar por orden de importancia cada tratamiento en caso de:

- Conexión
- Cambio de modos de funcionamiento
- Lógicas de entradas



Bits de sistema Grafcet
La utilización de los bits %S21, %S22 y %S23 queda reservada únicamente para el tratamiento preliminar. El sistema pone estos bits a 0 automáticamente; su escritura debe realizarse únicamente por el código de operación S.

Inicialización del Grafcet, %S21
Causas:
- rearranque en frío,
- ajuste a 1 de %S21 por el programa o el terminal.

Consecuencias: desactivación de todas las etapas activas y activación de todas las etapas iniciales.

(1) Estos modos de funcionamiento están definidos en GEMMA ("Guide pour l'étude des modes de marches et d'arrêts" = "Guía para el estudio de los modos de funcionamiento y parada", propuesto por ADEPA).

Puesta a cero del Grafcet, %S22

Causas: puesta a 1 de %S22 por el programa o el terminal.

Consecuencias:
- desactivación de todas las etapas activas,
- parada de la exploración del tratamiento secuencial.

Preposicionamiento del Grafcet, %S22 y %S23

Procedimiento:
- puesta a cero del Grafcet por puesta a 1 de %S22,
- preposicionamiento de las etapas a activar por un conjunto de instrucciones S Xi,
- validación del preposicionamiento por puesta a 1 de %S23

Inmovilización de una situación:
- en situación inicial: mediante el programa, mantenimiento a 1 de %S21,
- en situación "vacía": mediante el programa, mantenimiento a 1 de %S22,
- en situación determinada: mediante mantenimiento a 1 de %S23.

7.2 Consejos de programación

Gestión de los saltos de programa
Los saltos de programa se utilizarán con precaución con el fin de evitar bucles demasiado largos que pueden aumentar el tiempo del ciclo. Se evitarán los saltos de programa hacia las instrucciones situadas en un punto anterior del programa.

Programación de las salidas
Cada bit de salida o bit interno no debe ser accionado más de una sola vez en el programa. En el caso de los bits de salida, sólo el último valor explorado se tendrá en cuenta en la actualización de las salidas.

Consideración de las seguridades directas
Los sensores de seguridad inmediata no deben ser tratados por el autómata. Deben actuar directamente sobre los preaccionadores correspondientes.

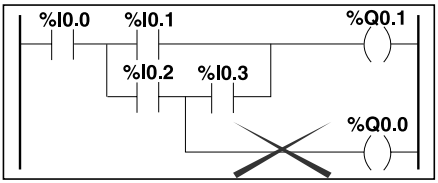
Gestión del restablecimiento de la alimentación
Se condicionará un restablecimiento de la alimentación a una operación manual ya que un rearranque automático de la instalación puede ser peligroso (utilización de los bits de sistema %S0, %S1 y %S9).

Gestión del tiempo y del bloque reloj-calendario
Se aconseja controlar el estado del bit %S51 que señala cualquier falla del reloj-calendario.

Nota: Al introducir el programa, el terminal controla la sintaxis de las instrucciones, los operandos y su asociación. La función de diagnóstico del terminal permite verificar los errores de programación (véase Anexos de la sección G).

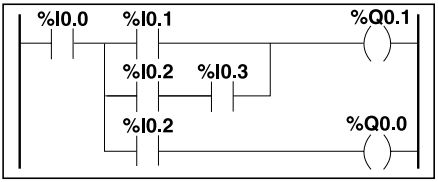
Complemento para la utilización de paréntesis

- Las operaciones de asignación no deben colocarse entre paréntesis.




```
LD    %I0.0
AND   %I0.1
OR(   %I0.2
ST    %Q0.0
AND   %I0.3
)
ST    %Q0.1
```

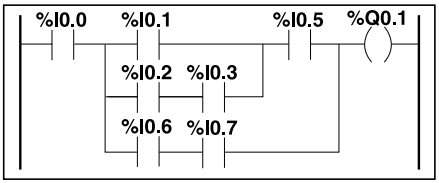
Para realizar la misma función, se programarán las siguientes ecuaciones:



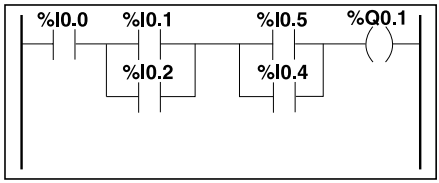

```
LD    %I0.0
MPS
AND(  %I0.1
OR(   %I0.2
AND   %I0.3
)
)
ST    %Q0.1
MPP
AND   %I0.2
ST    %Q0.0
```

- Si se efectúan varias puestas en paralelo de contactos, éstas deberán imbricarse o disociarse completamente.

Ejemplo 1

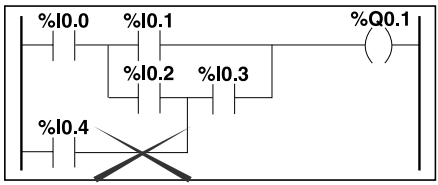


Ejemplo 2

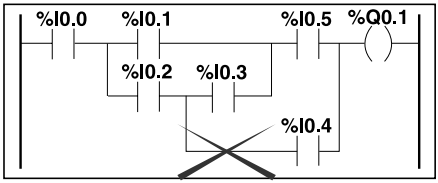


Al contrario, los esquemas siguientes no pueden ser programados.

Ejemplo 3

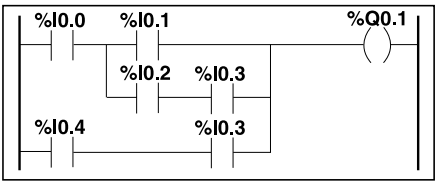


Ejemplo 4



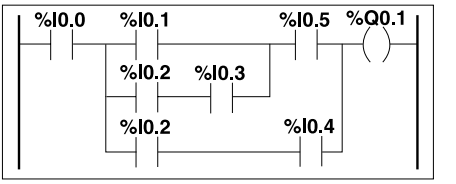
Para realizar esquemas equivalentes a los de la página precedente, es necesario modificarlos de la siguiente forma:

Ejemplo 5 (ver ejemplo 3)




```
LD      %I0.0
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
)
OR(     %I0.4
AND     %I0.3
)
ST      %Q0.1
```

Ejemplo 6 (ver ejemplo 4)




```
LD      %I0.0
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
AND     %I0.5
OR(     %I0.2
AND     %I0.4
)
)
ST      %Q0.1
```

7.3 Reactivación de salidas estáticas protegidas en TSX 07 •• ••12

Cuando una falla ha provocado la disyunción de las salidas de un autómata (autómata de base o extensión de E/S), es necesario reactivarlas. La reactivación puede realizarse:

- solicitándola mediante un comando de operador. Se aconseja este tipo de reactivación (véase el apartado A 1.7-2),
- automáticamente. Al usar este tipo de reactivación, es necesario conocer previamente sus consecuencias en el proceso.

La selección se efectúa con el bit de sistema %S49.

Disyunción de las salidas

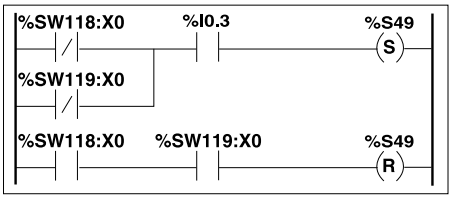
La aparición de una sobrecarga o un cortocircuito en una salida provoca:

- la limitación de corriente en la salida en cuestión,
- la disyunción de todas las salidas del bloque (autómata de base o extensión de E/S),
- la activación en modo fijo del (de los) indicador(es) I/O del autómata de base y de la extensión de E/S (disyunción de las vías de la extensión de E/S),
- la puesta a 0 del bit de sistema Falla E/S %S10,
- la puesta a 1 del bit de sistema %S118 (disyunción de las salidas del autómata de base) o del bit de sistema %S119 (disyunción de las salidas de la extensión de E/S),
- la puesta a 0 del bit de la palabra de sistema SW%118:X0 (disyunción de las salidas del autómata de base) o del bit de la palabra de sistema %SW119:X0 (disyunción de las salidas de la extensión de E/S).

Reactivación manual de las salidas (por comando de operador)

Una acción exterior pone el bit de sistema %S49 a 1. El bit %S49 debe ponerse a 0 después de la reactivación efectiva de las salidas. El programa correspondiente es:

```
LDN    %SW118:X0
ORN    %SW119:X0
AND    %I0.3 (entrada I0.3 por ejemplo)
S      %S49
LD     %SW118:X0
AND    %SW119:X0
R      %S49
```



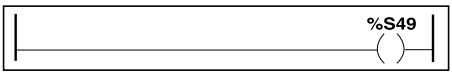
El tiempo mínimo entre dos reactivaciones, garantizado por el sistema, es de 10 segundos. Si la falla que causó la disyunción ha desaparecido:

- las salidas se activan de nuevo según el estado definido por el programa,
- los indicadores I/O están apagados,
- los bits de sistema y bits de palabra de sistema asumen sus valores por defecto: %S10, %SW118:X0, %SW119:X0 en el estado 1, %S118 y %S119 en el estado 0 (véase el apartado 6.2 de la sección B).

Reactivación automática de las salidas

El programa siguiente pone el bit de sistema %S49 a 1 de modo continuo:

```
LD     1
ST     %S49
```

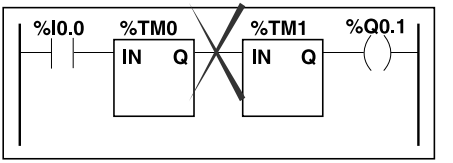


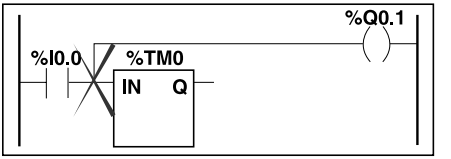
La reactivación es solicitada automáticamente cada 10 segundos. La base de tiempo de 10 segundos es síncrono respecto a la aparición del falla. Al reactivar, la reacción de las salidas, de los indicadores y de los bits y palabras de sistema es idéntica a la de la reactivación manual.

7.4 Condiciones de reversibilidad

Las condiciones siguientes deben verificarse para que un programa pueda ser totalmente reversible (1):

- las siguientes instrucciones no deben utilizarse: XOR, XORN, XORF, XORR, JMPCN, ENDCN o N
- los bloques de función se programarán de forma reversible (ver apart. 2.2-2),
- los bloques de función no deben programarse en cascada,
- las instrucciones de asignación quedan prohibidas entre las instrucciones BLK y OUT_BLK o BLK y END_BLK (si OUT_BLK no está programada).





Programación prohibida

```
BLK     %TM0
LD      %I0.0
ST      %Q0.1
IN
END_BLK
```

Programación aislada

```
BLK     %TM0
LD      %I0.0
IN
END_BLK
LD      %I0.0
ST      %Q0.1
```

(1) **Atención:** cuando una secuencia de instrucciones no es reversible, esta secuencia queda en lenguaje de lista de instrucciones, mientras que el resto del programa reversible se traduce en lenguaje de contactos.

7.5 Normas de reversibilidad

- Un escalón canónico completo no puede tener más de 7 celdas de alto y 11 celdas de ancho (cuadrícula 7x11).
- Una sentencia con el inicio LD debe terminar en una instrucción de acción condicional.
- Las instrucciones JMPCN, ENDCN, NOP y N no son reversibles.
- Las instrucciones de acción entre paréntesis no son reversibles.
- Las instrucciones de pila entre paréntesis no son reversibles.
- Una instrucción OR después de una instrucción de acción no es reversible.
- Las instrucciones RET, JMP y END son incondicionales. No pueden encontrarse otras instrucciones en el escalón completo.
- Sólo es posible acceder a las entradas y salidas de bloques de función mediante instrucciones estándares y reversibles de bloques de función. Las instrucciones de acceso directo a los bloques de función no son reversibles.

- Las instrucciones después de una END_BLK en una sentencia hacen que la sentencia se vuelva irreversible.
- Las salidas de bloques de función utilizadas con instrucciones AND y OR no son reversibles.
- Una instrucción OR dentro de un escalón de salida de un bloque de función debe encontrarse entre paréntesis.
- Una instrucción de acción incondicional entre una BLK y una END_BLK no es reversible.
- Una OUT_BLK debe venir seguida de una LD de una salida de un bloque de función válido o de una END_BLK.
- Una instrucción OR no anidada entre una MRD y MPP no es reversible.
- Una instrucción OR después de una MPS, MRD o MPP no es reversible.
- Una instrucción MCS no puede utilizarse en el mismo escalón con otras instrucciones de acción.
- Una instrucción de llamada de subprograma o JMPC debe ser la última instrucción de acción en un escalón.
- Cuando se introducen un título y comentarios antes de una instrucción en List, sólo prodrá haber un máximo de una línea para el título y 4 para los comentarios. Si se coloca una línea vacía entre el título y la instrucción en List, es posible que aparezca sólo parcialmente el encabezado del escalón correspondiente.
- Si se introducen más de 4 líneas de comentarios antes de la instrucción en List, la quinta línea de comentario se interpretará como una línea de título y las líneas de comentario precedentes no aparecerán en el encabezado del escalón de Ladder correspondiente.

Cuando una secuencia de instrucciones no es reversible, ésta permanece en lenguaje Lista, mientras que el resto del programa reversible se convierte en diagramas Ladder. Véase la ilustración siguiente.