

Controladores Lógicos Programables (PLCs)

Instrumentación y Automatismos Industriales

U.Na.M. – Fac. de Ingeniería – Ingeniería en Computación

Año 2024



Modicon M221: Controladores Lógicos Programables para arquitecturas cableadas

Formato Estándar



16 I/O channels



24 I/O channels



40 I/O channels

Formato Libro



16 I/O channels



32 I/O channels

Características

Conexión:

- Voltaje de alimentación: 24 VCC ó 100 a 240 VCA / 50 ó 60 Hz.
- Entradas analógicas (0 a 10 V).
- Entradas digitales (lógica positiva o lógica negativa).
- Salidas tipo transistor o tipo relé.
- Cartuchos de expansión: Para E/S (entradas o salidas analógicas; medición de temperatura); Para comunicación.

Capacidad de procesamiento:

- Velocidad de ejecución: 0,2 μ s por instrucción booleana (asm)
- Memoria RAM: 256 K p/var. internas; 256 K p/var. de usuario.
- Memoria FLASH: 2 MB.

Funciones especiales:

- Control PID.
- Control de posición de ejes (PTO).
- Modulación de ancho de pulso (PWM).
- Generador de pulsos (PLS).
- Generador de frecuencia (FREQGEN).
- Contadores de alta frecuencia (HSC), hasta 100 kHz.

Sistemas de comunicación:

- Ethernet.
- Enlace serie RS 232/RS 485.
- Puerto de programación USB mini-B.

Programación:

- Software EcoStruxure Machine Expert – Basic

Expansión:

- A través de cartuchos y módulos.

TM221CE24T: Características

3

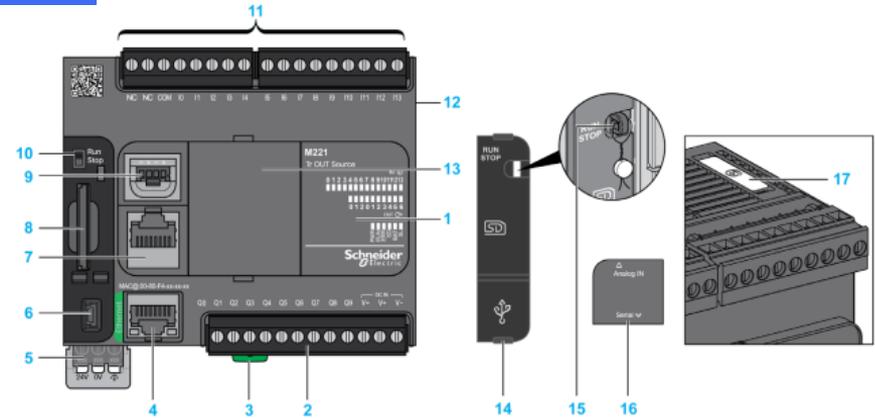


Cartucho de expansión

TM221CE24T

- ❑ Voltaje de alimentación: 24 VCC (de 20,4 V a 28,8 V).
- ❑ 14 Entradas digitales (4 del tipo rápidas).
- ❑ 2 Entradas analógicas (de 0 a 10 V), con ADC de 10 bits.
- ❑ 10 Salidas digitales tipo transistor PNP (2 del tipo rápidas).
- ❑ Funciones avanzadas: Controlador PID; Control de posición de eje (PTO); Generador de pulso (PLS); Generador PWM; Generador de frecuencia; Contador rápido (HSC).
- ❑ Protocolos de comunicación: USB para programar; Serie (RS232/RS485) ; Ethernet (Modbus RTU/ASCII).

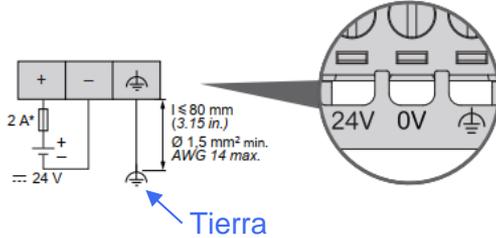
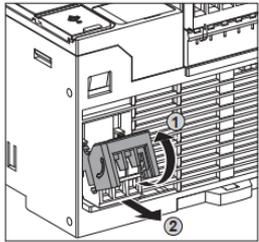
Partes



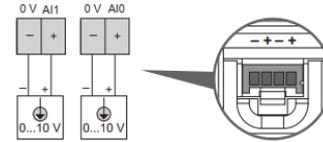
- 1) LEDs indicadores de estado.
- 2) Terminales de salidas (extraíble).
- 3) Seguro para montaje en riel DIN.
- 4) Puerto Ethernet (conector RJ 45).
- 5) Bornera p/fuente de alimentación.
- 6) Puerto USB mini-B p/programación.
- 7) Puerto serie (conector RJ 45).
- 8) Slot para tarjeta SD.
- 9) Entradas analógicas.
- 10) Interruptor RUN/STOP.
- 11) Terminales de entradas (extraíble).
- 12) Bus lateral p/módulos de expansión de E/S.
- 13) Slot p/cartucho de expansión 1.
- 14) Cubierta protectora.
- 15) Gancho de sujeción.
- 16) Cubierta extraíble de entradas analógicas.
- 17) Soporte batería de litio.

TM221CE24T: Conexiones

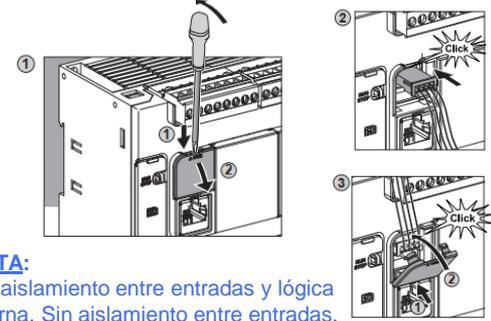
Fuente de Alimentación



Entradas Analógicas



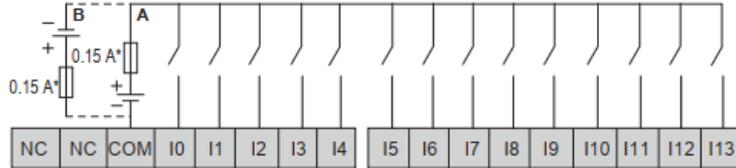
$$0\text{ V} \leq V_{in} \leq 10\text{ V}$$



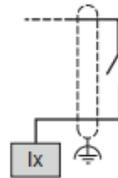
NOTA:
Sin aislamiento entre entradas y lógica interna. Sin aislamiento entre entradas.

Entradas Digitales

TM221C24T / TM221CE24T



Entradas rápidas
(I0, I1, I6, I7)



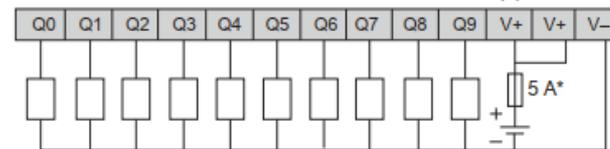
(A) Lógica positiva (*). (B) Lógica negativa.

(*) { **1 Lógico:** $V_{in} \geq 15\text{ V}$
0 Lógico: $V_{in} \leq 5\text{ V}$

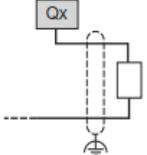
NOTA:
Aislamiento entre entradas y lógica interna: 500 VCA.
Sin aislamiento entre entradas.

Salidas Digitales

TM221C24T / TM221CE24T



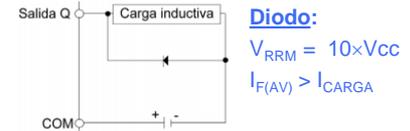
Salidas rápidas
(Q0, Q1)



(1) $V_{CC} \leq 24\text{ V}$. $P_{CARGA} < 12\text{ W}$.

**Protección de salidas
p/cargas inductivas**

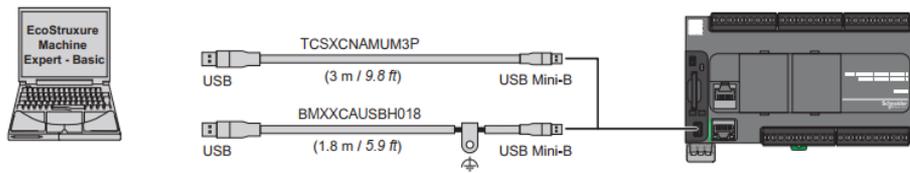
NOTA:
Aislamiento entre salidas y lógica interna:
500 VCA. Sin aislamiento entre salidas.



Diodo:
 $V_{RRM} = 10 \times V_{CC}$
 $I_{F(AV)} > I_{CARGA}$

Puerto USB

Para programación del PLC.

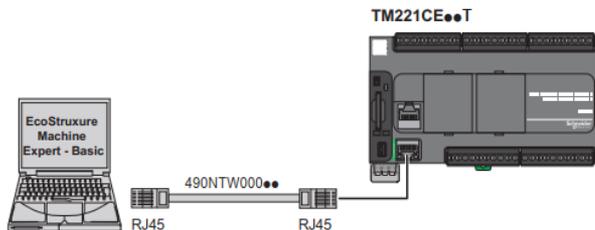
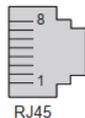


Puerto Ethernet

Para comunicación con PC/Notebook.

Ethernet

N°	Ethernet
1	TD+
2	TD-
3	RD+
4	—
5	—
6	RD-
7	—
8	—



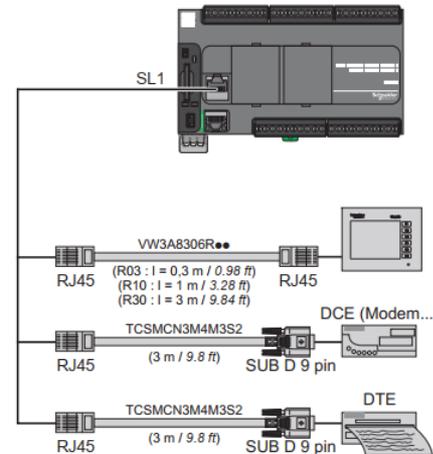
Puerto Serie

Para comunicación con dispositivos de campo.

SL1

N°	RS 232	RS 485
1	RxD	N.C.
2	TxD	N.C.
3	N.C.	N.C.
4	N.C.	D1
5	N.C.	D0
6	N.C.	N.C.
7	N.C. *	5 Vdc
8	Common	Common

* 5 Vdc delivered by the controller, do not connect.



Norma RS 232:

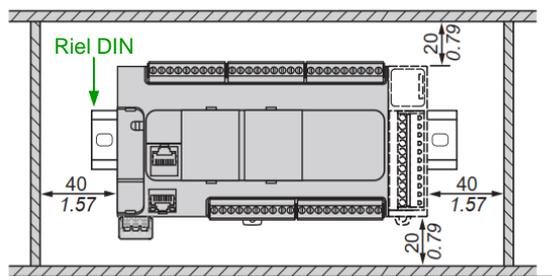
- Serie (sin conductor para CLK).
- Comunicación punto a punto (sólo entre dos dispositivos).
- Datos digitales en modo común.

Norma RS 485:

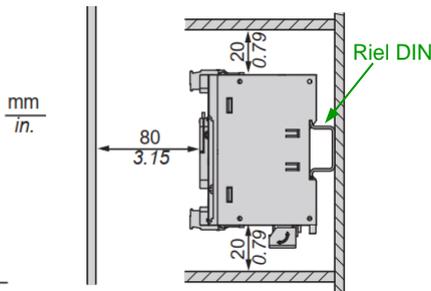
- Serie (sin conductor para CLK).
- Comunicación multipunto (entre varios dispositivos).
- Datos digitales en modo diferencial.

TM221CE24T: Montaje

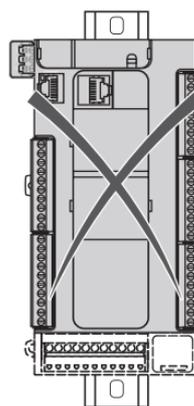
Distancias mínimas y posición



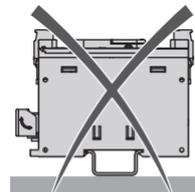
Distancias laterales mínimas.



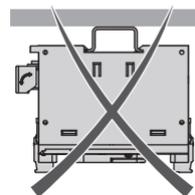
Distancia frontal mínima.



NO vertical.



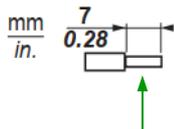
NO frente hacia arriba.



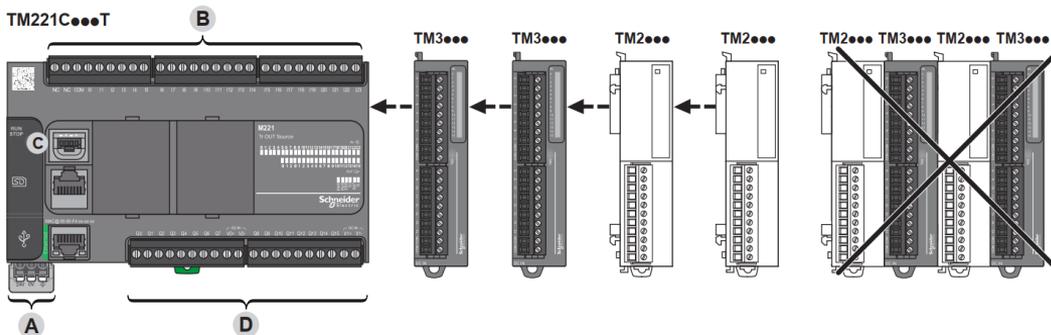
NO frente hacia abajo.

Módulos de Expansión

Conductores para conexión:



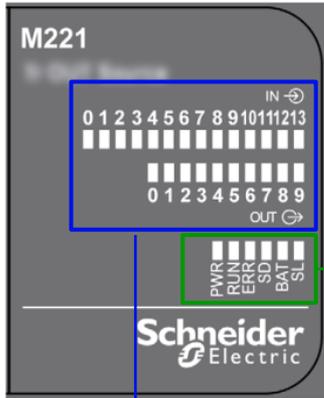
φ 0,2 a 2,5 mm²



TM2...
Diversos módulos para expansión de E/S digitales.

TM3...
Diversos módulos para expansión de E/S digitales, Analógicas y Puertos de Comunicación.

LEDs Indicadores



Indican el estado de las entradas y salidas del PLC. Se encienden cada vez que una de ellas se activa.

❑ PWR (Alimentación)

Verde ACTIVADO: Indica que recibe alimentación.
Verde DESACTIVADO: Indica que NO recibe alimentación.

❑ RUN (Estado del equipo)

Verde ACTIVADO: Indica que el PLC ejecuta una aplicación válida (modo RUN).
Verde INTERMITENTE: Indica que el PLC tiene una aplicación válida detenida (modo STOP).
Verde DESACTIVADO: Indica que el PLC no está programado.

❑ ERR (Error)

Rojo ACTIVADO: Durante el proceso de arranque o con una excepción. El programa no se ejecuta.
Rojo INTERMITENTE (con RUN apagado): Error interno. La aplicación no se ejecuta.
Rojo INTERMITENTE (intermitencia lenta): Error menor detectado. La aplicación se ejecuta.
Rojo INTERMITENTE (intermitencia normal): No hay aplicación

❑ SD (Acceso a la tarjeta SD)

Verde ACTIVADO: Se está accediendo a la tarjeta SD.
Verde INTERMITENTE: Error durante el acceso a la tarjeta SD.
Verde DESACTIVADO: No hay acceso o no hay tarjeta SD.

❑ BAT (Batería)

Rojo ACTIVADO: La batería debe sustituirse.
Rojo INTERMITENTE: La batería tiene baja carga.
Rojo DESACTIVADO: La batería está en buen estado.

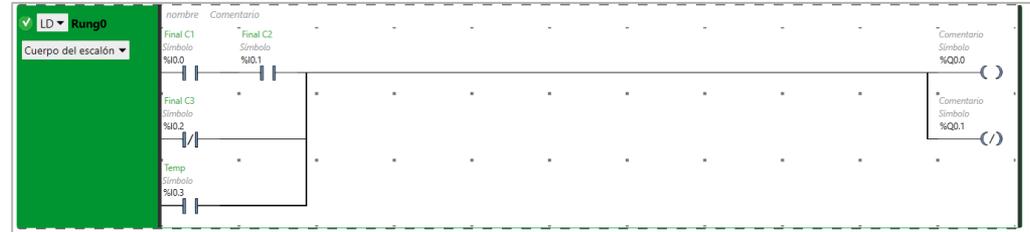
❑ SL (Línea Serie)

Verde ACTIVADO: Indica el estado de la línea serie.
Verde INTERMITENTE: Línea serie con actividad.
Verde DESACTIVADO: No existe comunicación serie.

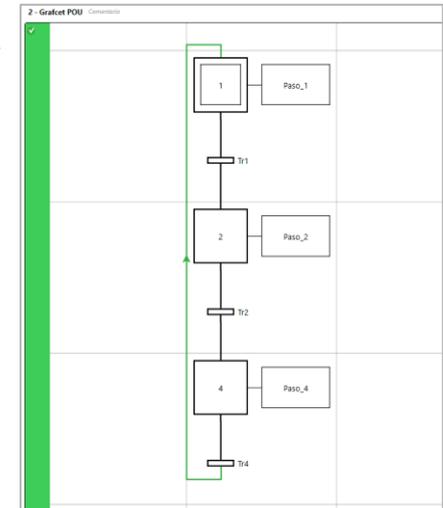
Software y Lenguajes de Programación



Diagrama de Contactos (Ladder)



GRAFSET

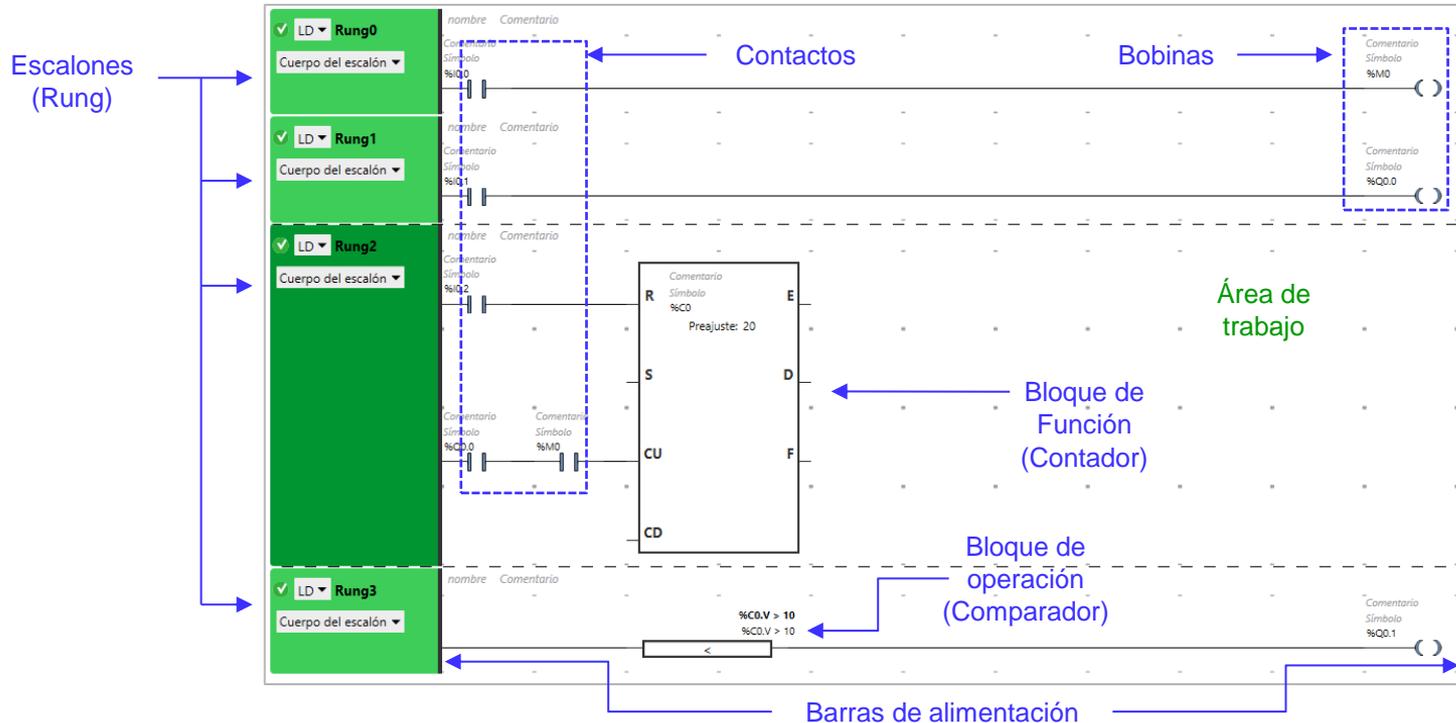


Lista de Instrucciones

nombre	Comentario		
0000	LD	%I0.0	Comentario
0001	AND	%I0.1	Comentario
0002	ORN	%I0.2	Comentario
0003	OR	%I0.3	Comentario
0004	ST	%Q0.0	Comentario
0005	STN	%Q0.1	Comentario

Programación en LADDER

El programa en este lenguaje consiste en construir una escalera donde cada escalón posee un circuito que relaciona contactos (elemento de lectura), bobinas (elemento de escritura), bloques de funciones y bloques de operaciones.



TM221CE24T: Programación. Objetos

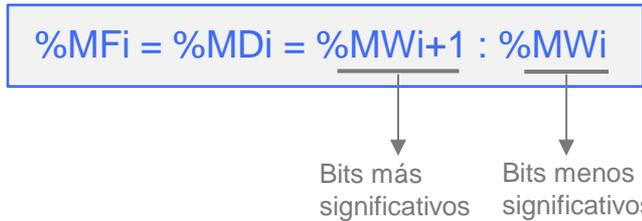
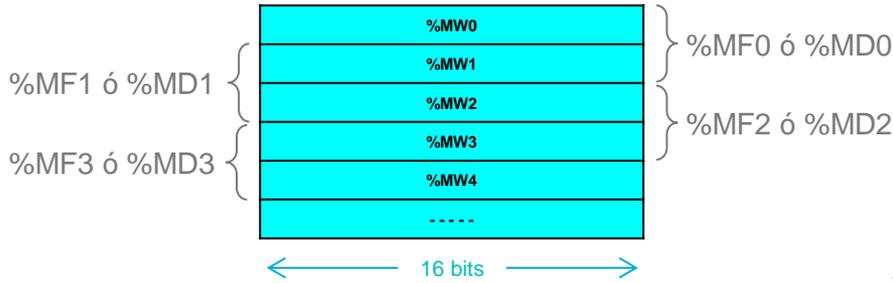
Los **objetos** son áreas de memoria del PLC que están reservadas para el uso de una aplicación. Los objetos disponibles en estos PLCs pueden representar:

- Variables del programa.
- Entradas y salidas (digitales y analógicas)
- Variables internas del sistema.
- Variables de bloques de funciones y bloques de operaciones

Tipo de objeto	Direccionamiento (etiqueta)	Tamaño en memoria	Rango de objetos	Descripción
Bits de sistema	%Si	1 bit	$0 \leq i \leq 123$	Estos bits proporcionan información de control y estado del PLC. Ejemplo: %S75, ESTADO DE LA BATERÍA (0: batería normal; 1: sin batería o batería baja).
Palabras de sistema	%SWi	16 bits (binario)	$0 \leq i \leq 212$	Estas palabras proporcionan información de control y estado del PLC. Ejemplo: %SW6, ESTADO DEL PLC (0: empty, 2: stopped, 3: running, 4: halted, 5: powerless).
Bits de memoria	%Mi	1 bit	$0 \leq i \leq 1023$	Permiten almacenar estados lógicos auxiliares que el usuario puede utilizar en la aplicación. Constituyen variables de usuario del tipo bit. Ejemplo: %M1, BIT DE MEMORIA N° 1.
Palabras de memoria	%MWi	16 bits (entero)	$0 \leq i \leq 7999$	Permiten almacenar datos que pueden utilizarse en la aplicación. Constituyen variables de usuario del tipo enteros. Nota: Las primeras 50 palabras de memoria (de %MW0 a MW49) son guardadas automáticamente en la memoria no volátil del PLC. Ejemplo: %MW10, PALABRA DE MEMORIA N° 10.
Palabras de memoria doble	%MDi	32 bits (doble)	$0 \leq i \leq 7998$	Ídem anterior pero con variables de usuario del tipo "double". Ejemplo: %MD110, PALABRA DE MEMORIA DOBLE N° 110.
Palabras de memoria flotante	%MFi	32 bits (flotante)	$0 \leq i \leq 7998$	Ídem anterior pero con variables de usuario del tipo "float". Ejemplo: %MF210, PALABRA DE MEMORIA FLOTANTE N° 210.

Tipo de objeto	Direccionamiento (etiqueta)	Tamaño en memoria	Rango de objetos	Descripción
Palabras constantes	%KWi	16 bits (entero)	$0 \leq i \leq 511$	Estos espacios de memoria almacenan valores constantes o mensajes alfanuméricos de 16 bits escritos con el software de programación. No pueden modificarse con la aplicación. Ejemplo: %KW21, PALABRA CONSTANTE N° 21.
Palabras constantes dobles	%KDi	32 bits (doble)	$0 \leq i \leq 510$	Ídem anterior pero con constantes o mensajes alfanuméricos de 32 bits. Ejemplo: %KD12, PALABRA CONSTANTE DOBLE N° 12.
Palabras constantes flotante	%KFi	32 bits (flotante)	$0 \leq i \leq 510$	Ídem anterior pero con constantes o mensajes alfanuméricos de 32 bits. Ejemplo: %KF12, PALABRA CONSTANTE FLOTANTE N° 12.
Bits de entradas digitales	%I0.i	1 bit	$0 \leq i \leq (n-1)$	Representan los estados lógicos presentes en las entradas digitales del PLC. Ejemplo: %I0.2, ENTRADA DIGITAL N°2.
Bits de salidas digitales	%Q0.i	1 bit	$0 \leq i \leq (m-1)$	Representan los estados lógicos presentes en las salidas digitales del PLC. Ejemplo: %Q0.10, SALIDA DIGITAL N°10.
Palabras de entradas analógicas	%IW0.i	16 bits (de 0 a 1023)	$0 \leq i \leq 1$	Representan los valores digitales de una señal analógica conectada a la entrada analógica correspondiente del PLC. Este dispositivo posee dos entradas analógicas de 0 a 10 V. El convertidor interno posee una resolución de 10 bits. Ejemplo: %IW0.0, VALOR DIGITAL DEL CANAL ANALÓGICO N°0.
Palabras de salidas analógicas	%QW0.z0y	16 bits (de 0 a 4095)	Depende del cartucho	Representan los valores digitales para obtener una señal analógica de salida en el canal deseado del cartucho conectado al PLC. Dependiendo el cartucho, la salida puede ser una señal de voltaje o de corriente.
Bits y palabras de los bloques funcionales	%BLKi.x	Depende del bloque	i = N° de bloque x = Variable del bloque	Representan valores o parámetros de los bloques funcionales, como así también palabras que poseen los mismos. Ejemplos: %C1.V, VALOR DE CONTEO DEL CONTADOR N°1; %C1.F, BIT DE SALIDA DE DESBORDAMIENTO CONTADOR N°1.

Solapamiento de palabras



Nota: Se cumple la misma relación entre las palabras constantes %KFi, %KDi y %KWi.

Extracción de bit de palabras

- De palabra de sistema → %SWi:Xk
- De palabra constante → %KWi:Xk
- De palabra de memoria → %MWi:Xk

Nota: “k” representa la posición del bit en la palabra.

Ejemplo:

%MW1:X0 → Bit 0 de la palabra de memoria MW1.

Formación de cadenas de bits

- Cadena de bits de sistema → %Si:L
- Cadena de bits de memoria → %Mi:L
- Cadena de bits de entradas → %IO.0:L
- Cadena de bits de salidas → %Q0.0:L

Nota: “i” Inicio de la cadena (0; 8 o múltiplo de 8).
“L” representa la longitud de la cadena de bits (para entradas y salidas digitales Lmax = 17).

Ejemplo:

%M8:8 → Cadena conformada por:

%M8	%M9	%M10	%M11	%M12	%M13	%M14	%M15
-----	-----	------	------	------	------	------	------

Formación de tablas

- Tabla de palabras de sistema → %SWi:L
- Tabla de palabras constantes → %KWi:L
- Tabla de palabras de memoria → %MWi:L

Nota: “i” Inicio de la tabla (p/palabras de sistema, depende de la palabra). “L” representa la longitud de la tabla.

Ejemplo:

%KW0:4 →

%KW0
%KW1
%KW2
%KW3

P1-ContactosBobinas.smbp

LD Rung	nombre	Comentario	Comentario	Comentario
LD Rung0	Cuerpo del escalón	%I0.0	← Contacto NA	Bobina NA → %Q0.0
LD Rung1	Cuerpo del escalón	%I0.0	← Contacto NC	Bobina NC → %Q0.1
LD Rung2	Cuerpo del escalón	%I0.1		Bobina NC → %Q0.2
LD Rung3	Cuerpo del escalón	%I0.2	← Contacto Flanco Ascendente	Bobina Set → %Q0.3
LD Rung4	Cuerpo del escalón	%I0.3	← Contacto Flanco Descendente	Bobina Reset → %Q0.3

CONTACTOS



Operandos: 0/1; %I0.i; %Q0.i; %Mi; %BLKi.x; %Si; %IW:Xk; %QW:Xk; %MW:Xk; %KW:Xk; %SW:Xk



Operandos: %I0.i; %Mi

BOBINAS

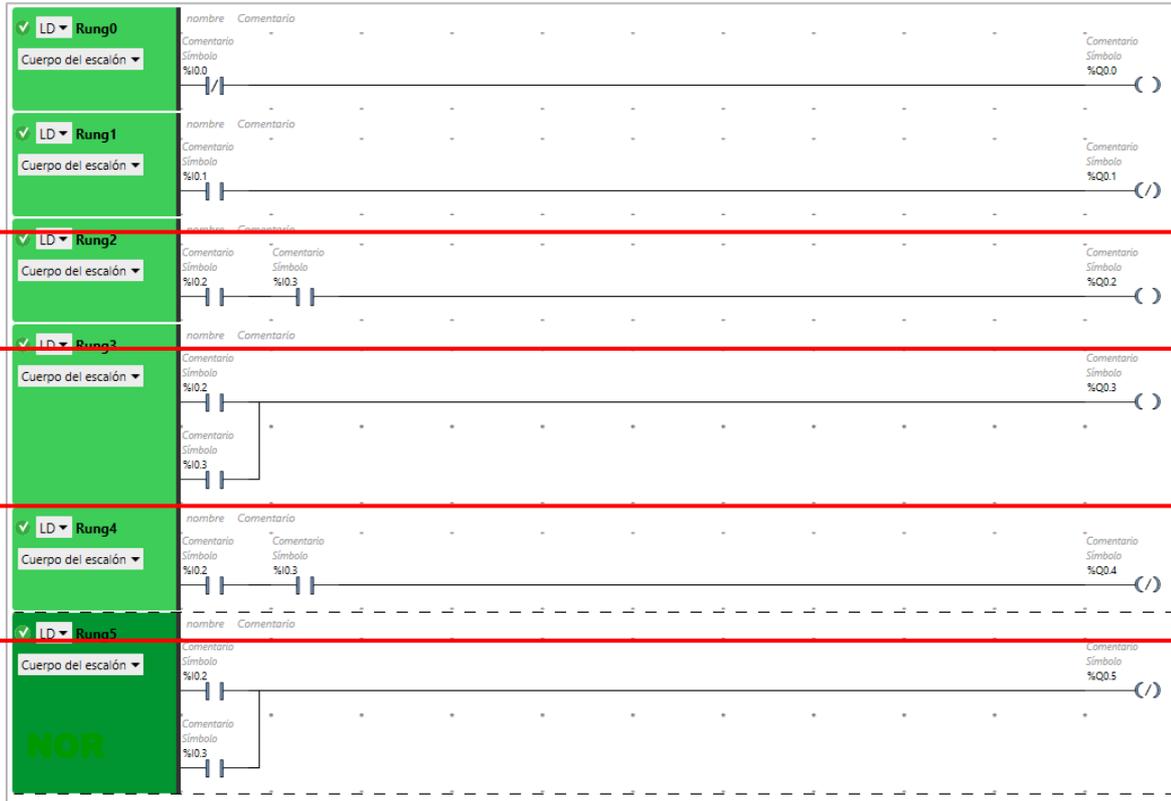


Operandos: %Q0.i; %Mi; %BLKi.x; %Si (*); %QW:Xk; %MW:Xk; %SW:Xk (*)

(*) Siempre que no sean de sólo lectura.

P2-OperacionesLógicas.smbp

**NOT
(2 formas)**



← $\%Q0.0 = \overline{\%I0.1}$

← $\%Q0.1 = \%I0.2$

AND

← $\%Q0.2 = \%I0.2 \times \%I0.3$

OR

← $\%Q0.3 = \%I0.2 + \%I0.3$

NAND

← $\%Q0.4 = \overline{\%I0.2 \times \%I0.3}$

NOR

← $\%Q0.5 = \overline{\%I0.2 + \%I0.3}$

NOR

Bloque de Operaciones



← Símbolo en software EcoStruxure Machine Expert – Basic



Operación	Operador	Sintaxis	Op1	Op2	Op3
ASIGNACIÓN	:=	Op1 := Op2	%MWi; %MDi; %MFi; %SWi; %Mi:L; %Si:L; %Q0.0:L; %QW0.z0y; %BLKi.x	Valor inmediato (ej.: 16#FF00; 45537; etc.); %MWi; %MDi; %MFi; %KW; %KDi; %KFi; %SWi; %IW0.i; %QW0.z0y; %Mi:L; %Q0.0:L; %Si:L; %I0.0:L; %BLKi.x	
ARITMÉTICA	+ ; - ; * ; /	Op1 := Op2 operador Op3	%MWi; %MDi; %QW0.z0y; %SWi; %BLKi.x	%MWi; %KW; %SWi; %IW0.i; %QW0.z0y; %MDi; %KDi; %BLKi.x	%MWi; %KW; %SWi; %IW0.i; %QW0.z0y; %MDi; %KDi; %BLKi.x
	DEC; INC	operador Op1	%MWi; %MDi; %QW0.z0y; %SWi; %BLKi.x		
LÓGICA	AND; OR; XOR	Op1 := Op2 operador Op3	%MWi; %MDi; %QW0.z0y; %SWi; %BLKi.x	Valor inmediato; %MWi; %KW; %IW0.i; %QW0.z0y, %SWi, %BLKi.x	Valor inmediato; %MWi; %KW; %IW0.i; %QW0.z0y, %SWi, %BLKi.x
	NOT	Op1:= operador (Op2)	%MWi; %MDi; %QW0.z0y; %SWi; %BLKi.x	%MWi; %KW; %IW0.i; QW0.z0y, %SWi, %BLKi.x	
DESPLAZAM.	SHL; SHR	Op1 = operador (Op2,n) Palabra de 16 bits: $1 \leq n \leq 16$ Palabra de 32 bits: $1 \leq n \leq 32$	%MWi; %QW0.z0y; %SWi; %MDi; %BLKi.x	%MWi; %KW; %IW0.i; %QW0.z0y; %SWi; %MDi; %KDi; %BLKi.x	
	ROL; ROR		%MWi; %QW0.z0y; %SWi; %MDi; %BLKi.x	%MWi; %KW; %IW0.i; %QW0.z0y; %SWi; %MDi; %KDi; %BLKi.x	

SHL y SHR: Desplazamientos Lógicos (hacia izquierda y hacia derecha). **ROL y ROR:** Desplazamientos Circular (hacia izquierda y hacia derecha).

Bloque de Comparaciones



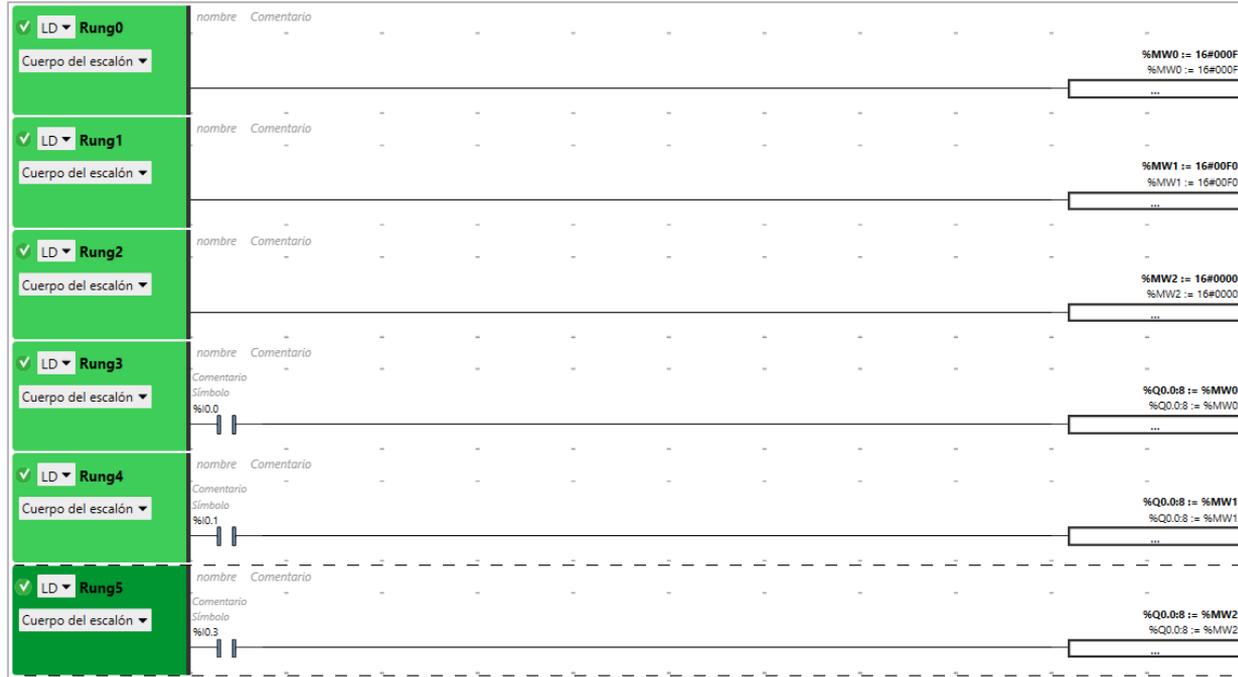
← Símbolo en software EcoStruxure Machine Expert – Basic

Símbolo
Expresión de comparación/operación

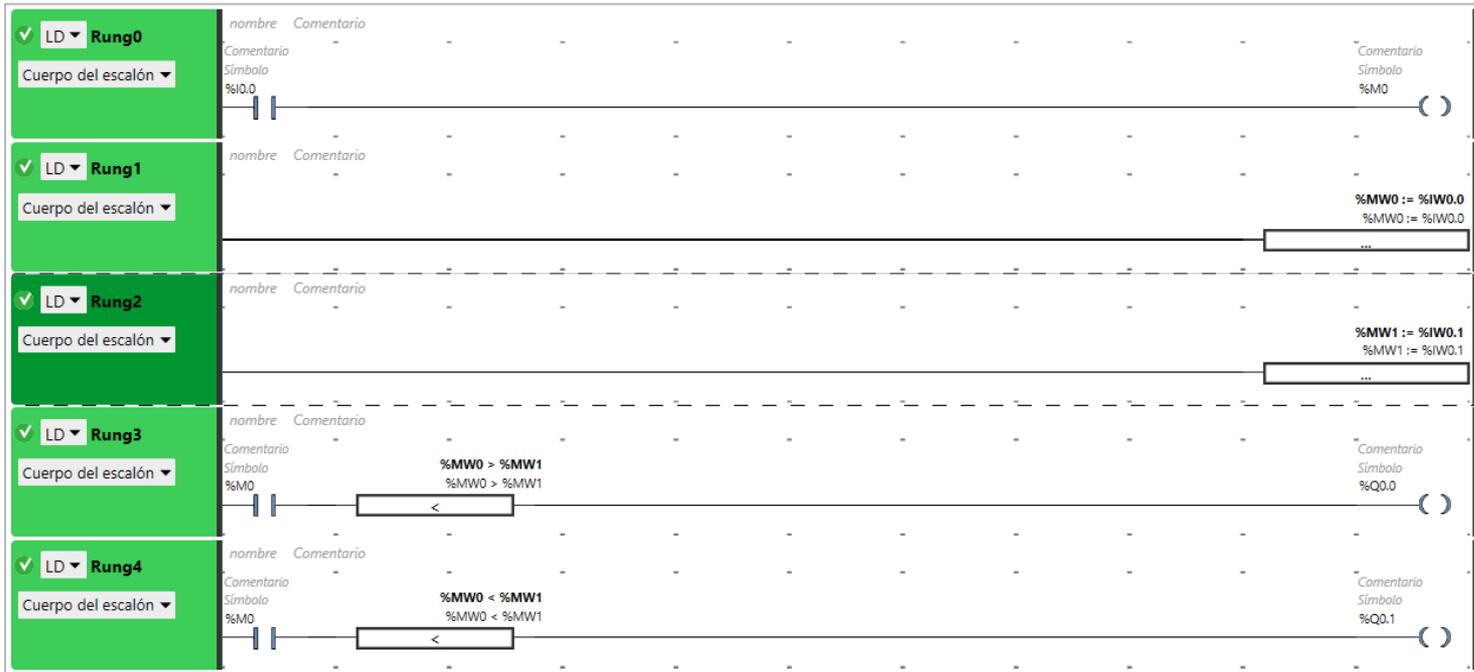


Operación	Operador	Sintaxis	Op1	Op2
Es mayor que	>	Op1 > Op2	%MWi; %KWi; %MDi; %KDi; %MFi; %KFi; %SWi; %IW0.i; %QW0.z0y; %BLKi.x	Valor inmediato; %MWi; %KWi; %MDi; %KDi; %MFi; %KFi; %SWi; %IW0.i; %QW0.z0y; %BLKi.x,
Es mayor o igual que	>=	Op1 >= Op2		
Es menor que	<	Op1 < Op2		
Es menor o igual que	<=	Op1 <= Op2		
Es menor igual que	=	Op1 = Op2		
Es distinto que	<>	Op1 <> Op2		

P3-BloquesOperaciones.smbp



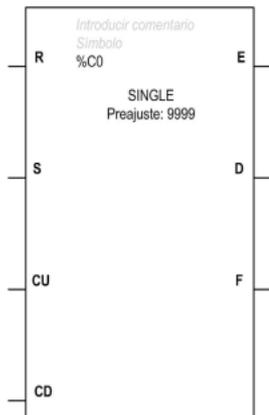
P4-BloquesComparaciones.smbp



Contador (%Ci)

123

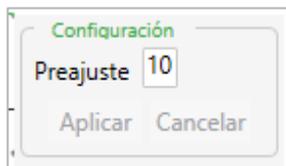
← Símbolo en software EcoStruxure Machine Expert – Basic



%Ci con $0 \leq i \leq 254$

Objeto	Nombre	Descripción
%Ci.V	Valor actual de conteo	Palabra de 16 bits que cambia cuando el contador recibe un flanco \uparrow en la entrada CU o CD (o los bits asociados).
%Ci.P	Valor de preajuste (de 0 a 9999)	Palabra de 16 bits que tiene un valor que será alcanzado por el conteo.
%Ci.E	Desbordamiento descendente	Bit que refleja el estado de la salida "E".
%Ci.D	Igualación del conteo con el preajuste	Bit que refleja el estado de la salida "D".
%Ci.F	Desbordamiento ascendente	Bit que refleja el estado de la salida "F".

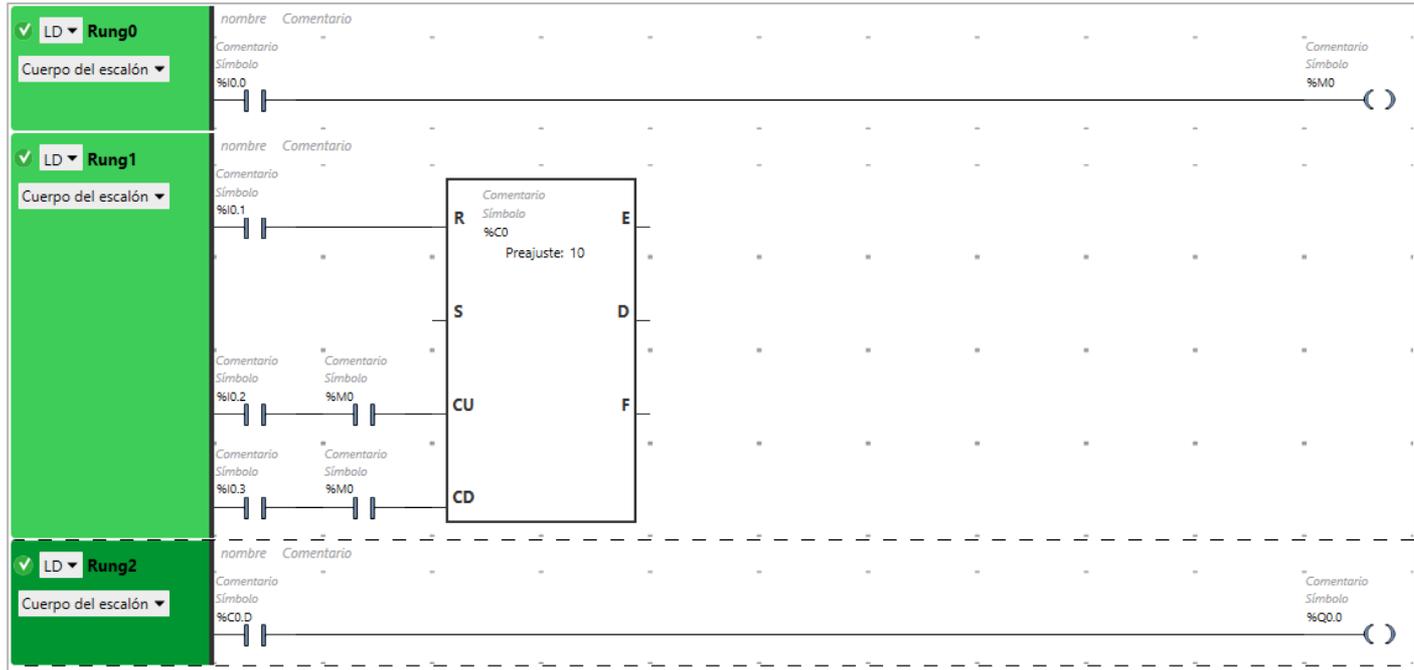
Etiqu.	Descripción	Operación
R	Entrada reseteo .	Cuando R = 1 se hace %Ci.V = 0.
S	Entrada seteo .	Cuando S = 1 se hace %Ci.V = %Ci.P.
CU	Entrada para conteo ascendente .	Cuando CU recibe un flanco \uparrow , %Ci.V = %Ci.V+1.
CD	Entrada para conteo descendente .	Cuando CU recibe un flanco \uparrow , %Ci.V = %Ci.V-1 .
E	Salida indicativa de desborde en conteo descendente .	Cuando %Ci.V pasa de 0 a 9999, se hace E = 1. Si sigue el conteo descendente, se hace E = 0.
D	Salida indicativa de que se alcanzó el preajuste	Cuando %Ci.V = %Ci.P, se hace D = 1.
F	Salida indicativa de desborde en conteo ascendente .	Cuando %Ci.V pasa de 9999 a 0, se hace F = 1. Si sigue el conteo ascendente, se hace F = 0.



Ventana de configuración

Contador (%Ci)

P5-Contador.smbp



Temporizador (%TMi)



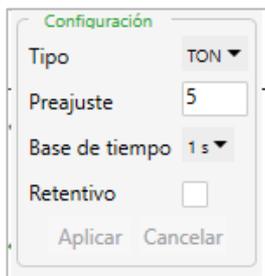
← Símbolo en software EcoStruxure Machine Expert – Basic



%TMi con $0 \leq i \leq 254$

Este bloque es utilizado para retrasar el encendido de un evento (TON), retrasar el apagado de un evento (TOFF) o bien generar un evento con una duración determinada (TP).

Objeto	Nombre	Descripción
%TMi.V	Valor actual del temporizador.	Palabra de 16 bits que cambia cuando la temporización está en curso.
%TMi.P	Valor de preajuste (de 0 a 9999).	Palabra de 16 bits usada como multiplicador de la base de tiempo para determinar la temporización.
%TMi.Q	Salida del temporizador.	Bit que refleja el estado de la salida "Q".



Ventana de configuración

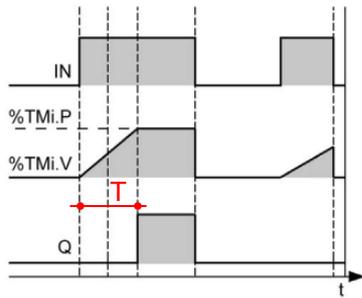
Etiqu.	Descripción	Operación
IN	Entrada de disparo .	TON y TP: Cuando esta entrada detecta un flanco \uparrow , inicia la temporización. TOFF: Cuando esta entrada detecta un flanco \downarrow , inicia la temporización.
Q	Salida.	TON: Esta salida se hace 1 cuando finaliza la temporización. TOFF: Esta salida se hace 0 cuando finaliza la temporización. TP: Esta salida permanece en 1 durante la temporización.

Temporización:
 $T = \%TMi.P \times TB$

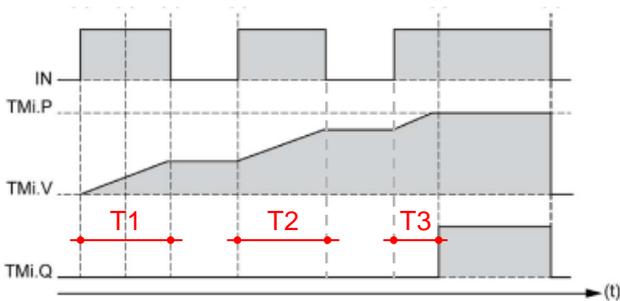
Base de Tiempo (TB) $\left\{ \begin{array}{l} 1 \text{ ms (sólo para \%TM0... \%TM5)} \\ 10 \text{ ms} \quad 100 \text{ ms} \\ 1 \text{ s} \quad 1 \text{ min} \end{array} \right.$

Temporizador (%TMI)

TON



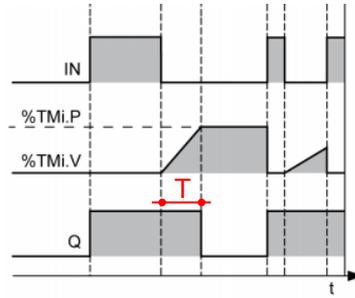
TON Normal ($T = \%TMI.P \times TB$)



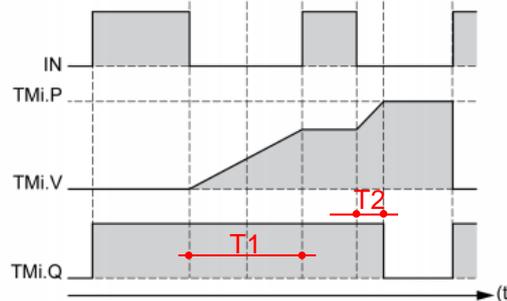
TON Retentivo

($T = T1 + T2 + T3 = \%TMI.P \times TB$)

TOFF



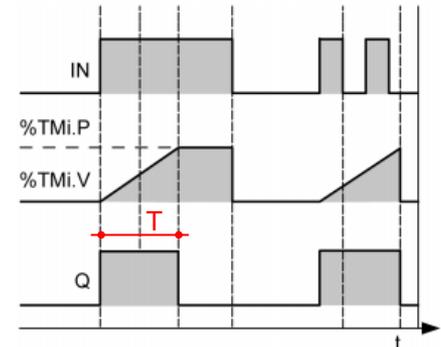
TOFF Normal ($T = \%TMI.P \times TB$)



TOFF Retentivo

($T = T1 + T2 = \%TMI.P \times TB$)

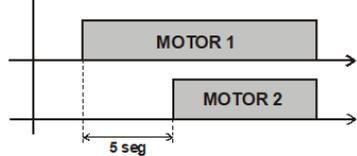
TP



TP ($T = \%TMI.P \times TB$)

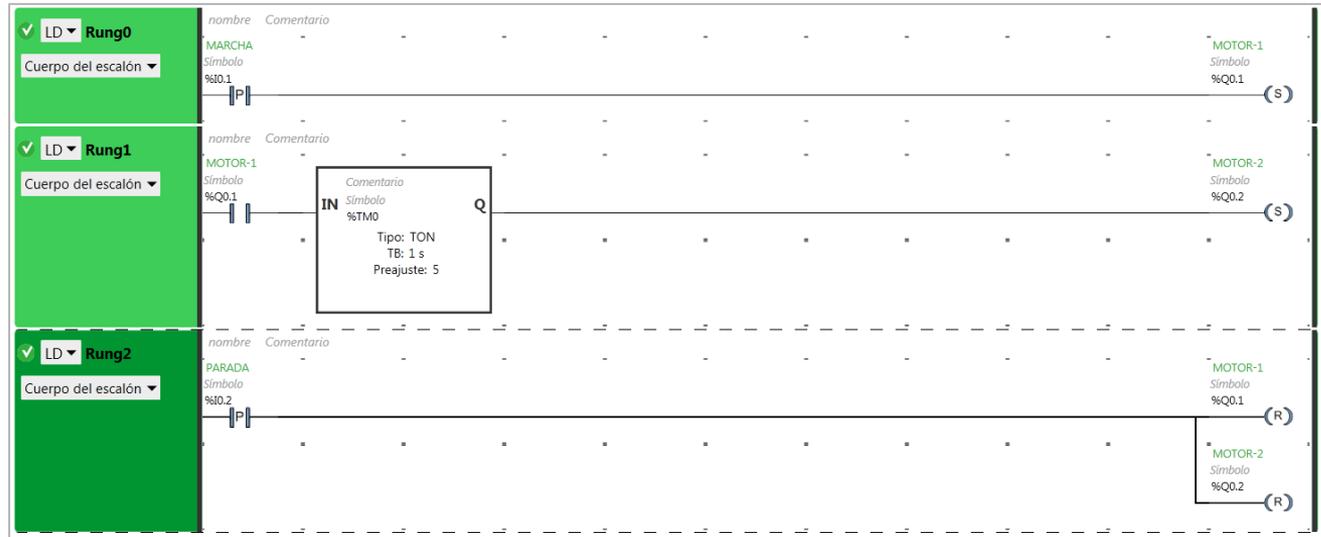
Temporizador (%Tmi)

Se debe comandar la MARCHA-PARADA de dos motores como se muestra en el siguiente diagrama temporal:



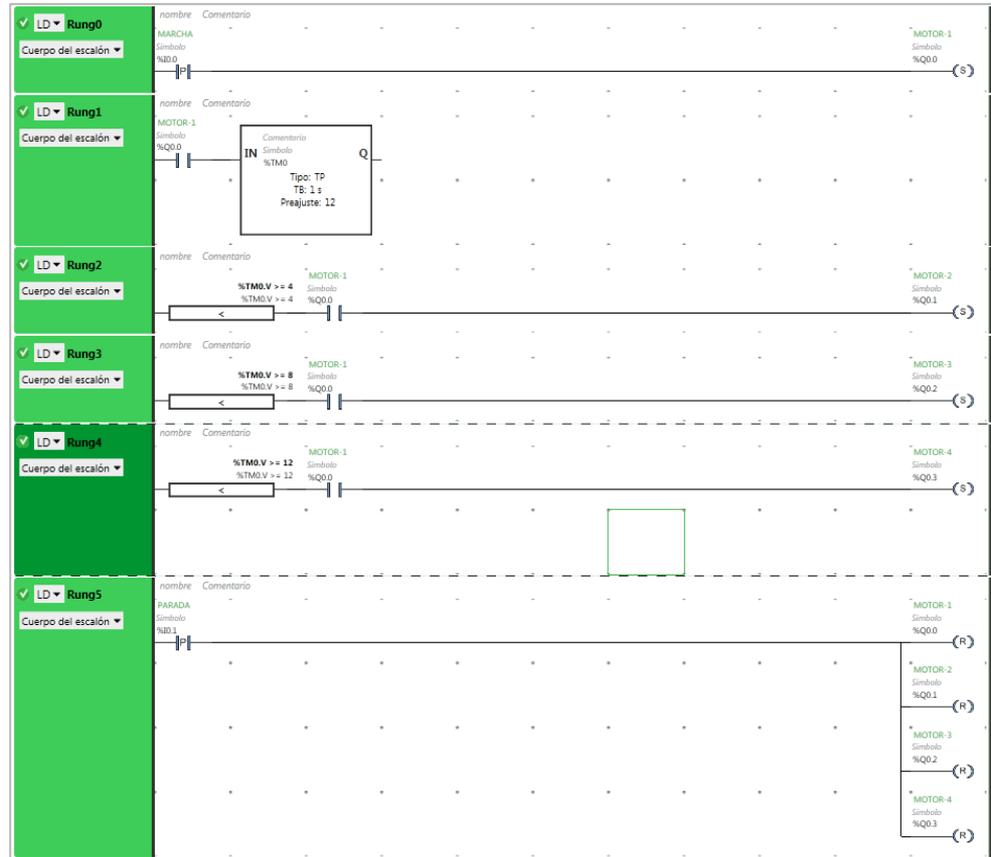
Considerar las siguientes especificaciones:
Pulsador (NA) MARCHA ==> %I0.1
Pulsador (NA) PARADA ==> %I0.2
Encendido/Apagado MOTOR1 ==> %Q0.1
Encendido/Apagado MOTOR2 ==> %Q0.2

P6-Temporizador1.smbp



Temporizador (%TMI)

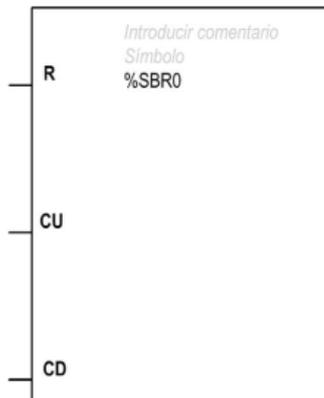
P7-Temporizador2.smbp



Registro de Desplazamiento (%SBRi)



← Símbolo en software EcoStruxure Machine Expert – Basic



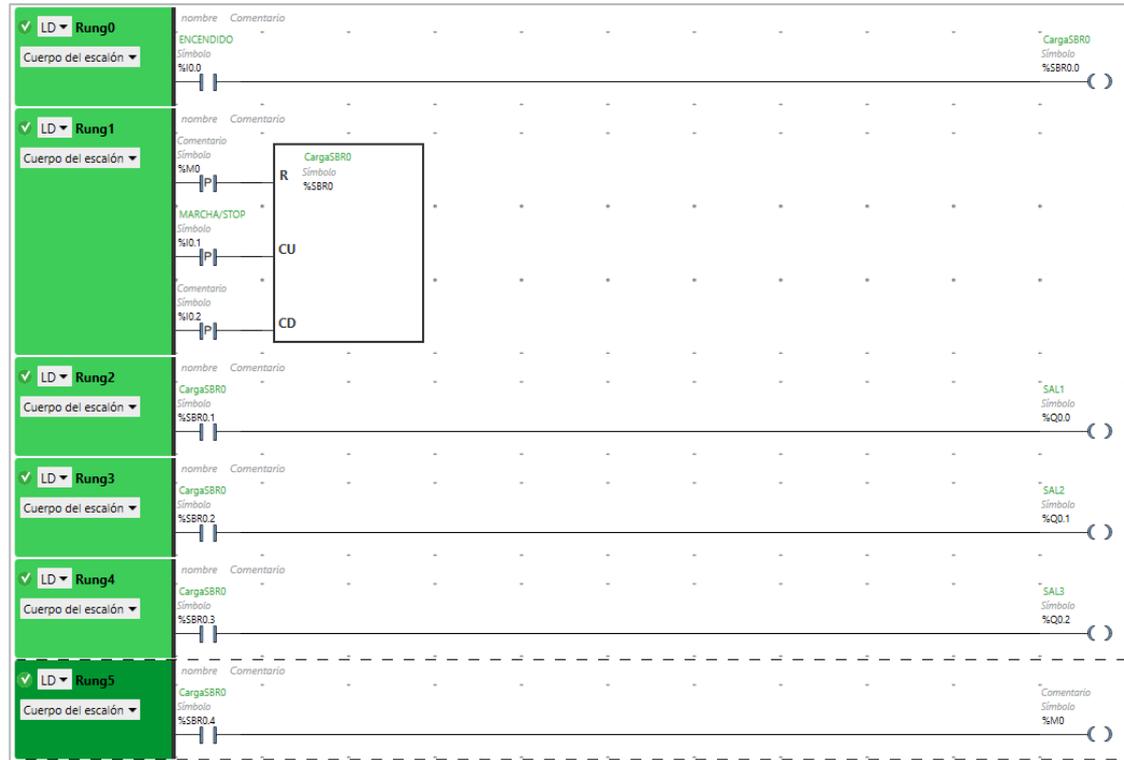
%SBRi con $0 \leq i \leq 7$

El bloque de funciones %SBRi proporciona un desplazamiento de un bit hacia la izquierda (con CU) o hacia la derecha (con CD) de un dato binario.

Objeto	Nombre	Descripción
%SBRi.j	Bit de registro.	Se puede usar para acceder a un bit del registro, ya sea para escribirlo o leerlo. Número de registro: $0 \leq i \leq 7$; Número de bit: $0 \leq j \leq 15$.

Etiqu.	Descripción	Operación
R	Entrada de reset .	Cuando $R = 1$ todos los bits del registro se borran.
CU	Entrada desplazar hacia la izquierda .	Cuando CU recibe un flanco \uparrow , el valor del registro se desplaza un bit hacia la izquierda.
CD	Entrada desplazar hacia la derecha .	Cuando CD recibe un flanco \uparrow , el valor del registro se desplaza un bit hacia la derecha.

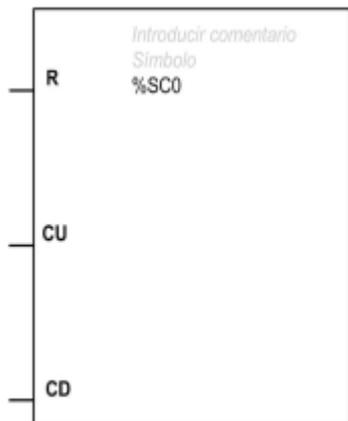
P8-RegistroDesplazamiento.smbp



Contador de Pasos (%SCi)



← Símbolo en software EcoStruxure Machine Expert – Basic



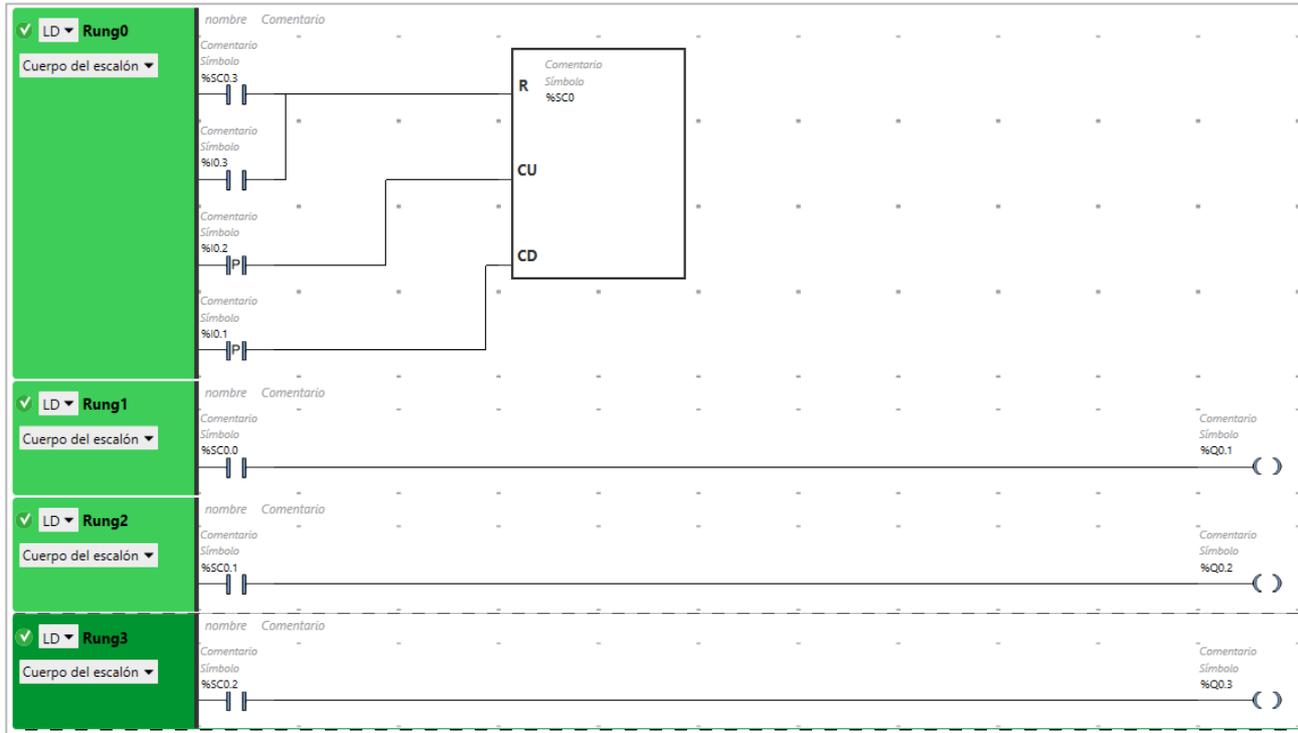
%SCi con $0 \leq i \leq 7$

Este bloque permite realizar una serie de pasos a los que pueden asignarse acciones (por ej. activar salidas). El desplazamiento de un paso a otro está comandado por las entradas CU y CD, las cuales pueden asociarse a eventos internos o externos. Cada vez que un paso esté activo, el bit asociado (%SCi.j) se pone en 1. Solo puede activarse un paso de %SCi cada vez.

Objeto	Nombre	Descripción
%SCi.j	Bit de paso.	Se puede usar para acceder a un bit de paso a través de la lectura del mismo. Número de registro: $0 \leq i \leq 7$; Número de bit: $0 \leq j \leq 255$.

Etiqu.	Descripción	Operación
R	Entrada de reset .	Cuando R = 1 todos los bits del registro se borran.
CU	Entrada para incrementar el paso .	Cuando CU recibe un flanco \uparrow , se avanza en un paso.
CD	Entrada para decrementar el paso .	Cuando CD recibe un flanco \uparrow , se retrocede en un paso.

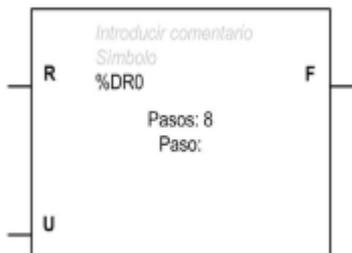
P9-ContadorPasos.smbp



Drums (%DRi)



← Símbolo en software EcoStruxure Machine Expert – Basic



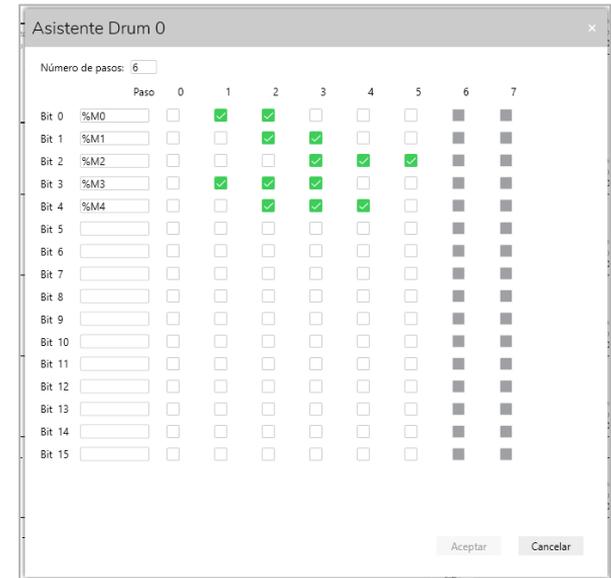
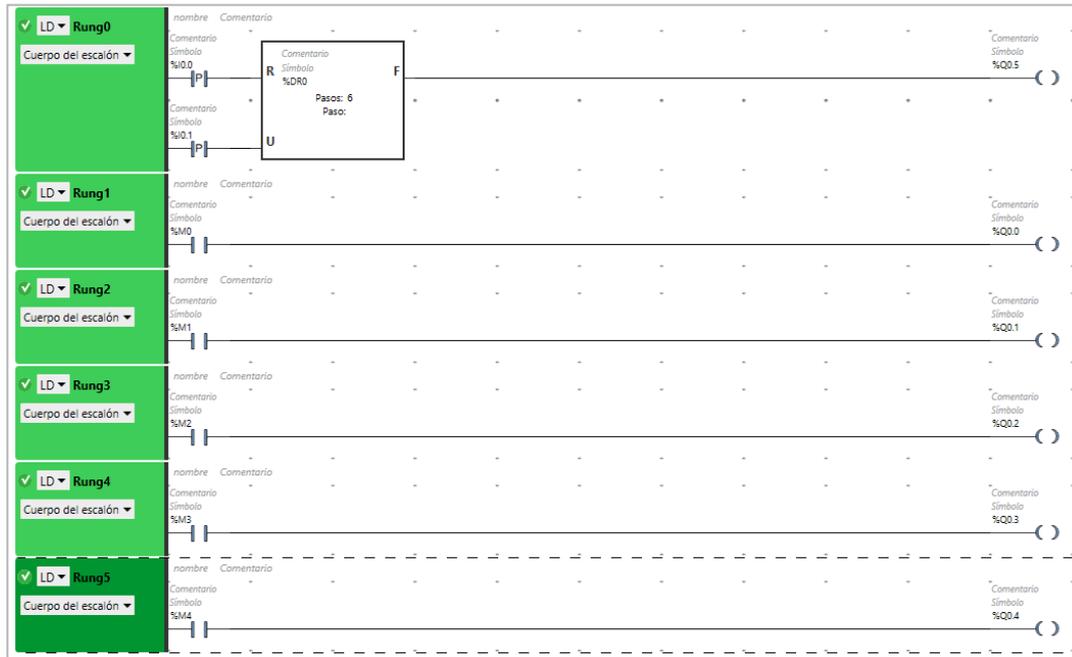
%DRi con $0 \leq i \leq 7$

El bloque de funciones Drum funciona en forma similar a un secuenciador del tipo tambor electromecánico, que cambia de un paso a otro según eventos externos (por ej. una entrada que se activa). Cada paso proporciona una combinación de 16 bits, con estados configurables. Cada bit puede asociarse a un objeto (tipo bit) determinado del PLC. La cantidad de pasos del bloque Drum también es configurable, siendo 8 el número máximo de pasos.

Objeto	Nombre	Descripción
%DRi.S	Palabra correspondiente al Nº de paso actual.	Corresponde a una palabra que contiene la combinación de los 16 bits del paso actual. Para escribir el paso actual debe utilizarse un valor inmediato decimal. El efecto de la escritura se produce en la siguiente ejecución del bloque de funciones.
%DRi.F	Salida.	Bit que refleja el estado de la salida "F".

Etq.	Descripción	Operación
R	Entrada de reset (para volver al paso 0).	Cuando R = 1, retorna el bloque Drum al paso 0.
U	Entrada de avance .	Cuando U recibe un flanco \uparrow , el bloque Drum avanza en un paso y actualiza la combinación de bits que forman el paso.
F	Salida indicadora de último paso.	Esta salida se hace F = 1 cuando el bloque Drum alcanza el último paso.

P10-Drums.smbp



Ventana de configuración