

Controladores programables Twido

Guía de referencia de software

TWD USE 10AS Versión 1.0

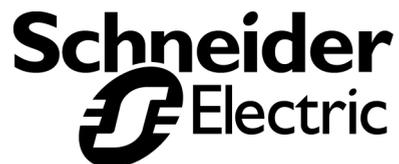


Tabla de materias



	Información de seguridad	9
	Acerca de este libro	13
Parte I	Descripción de software de Twido	15
	Presentación	15
Capítulo 1	Introducción al software Twido	17
	Presentación	17
	Introducción a TwidoSoft	18
	Introducción a los lenguajes de Twido	19
Capítulo 2	Objetos de lenguaje Twido	23
	Presentación	23
	Validación de objetos de lenguaje	24
	Objetos de bit	25
	Objetos de palabra	28
	Direccionamiento de objetos de bit	31
	Direccionamiento de objetos de palabra	32
	Direccionamiento de entradas/salidas	33
	Direccionamiento de red	35
	Objetos de bloques de función	36
	Objetos estructurados	37
	Palabras indexadas	40
	Simbolización de objetos	42
Capítulo 3	Memoria de usuario	43
	Estructura de la memoria de usuario	43
Capítulo 4	Modos de funcionamiento del controlador	47
	Presentación	47
	Exploración cíclica	48
	Exploración periódica	51
	Comprobación del tiempo de ciclo	54
	Modos de funcionamiento	56

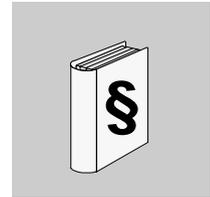
	Comportamiento ante cortes de corriente y recuperación de alimentación	58
	Uso de un reinicio en caliente	61
	Comportamiento ante un inicio en frío	64
	Inicialización del controlador	67
Parte II	Funciones especiales	69
	Presentación	69
Capítulo 5	Comunicaciones	71
	Presentación	71
	Vista general de las comunicaciones	72
	TwidoSoft a comunicaciones del controlador	74
	Comunicaciones de conexión remota	76
	Comunicaciones ASCII	89
	Comunicaciones Modbus	101
	Solicitudes Modbus estándar	119
Capítulo 6	Funciones analógicas incorporadas	125
	Presentación	125
	Potenciómetros	126
	Canal analógico	128
Capítulo 7	Módulos analógicos de gestión	129
	Presentación	129
	Vista general del módulo analógico	130
	Direccionamiento de entradas y salidas analógicas	131
	Configuración de E/S analógicas	133
	Ejemplo de uso de módulos analógicos	135
Capítulo 8	Funcionamiento del monitor de operación	137
	Presentación	137
	Monitor de operación	138
	Identificación del controlador e información de estado	141
	Objetos y variables del sistema	144
	Ajustes del puerto serie	151
	Reloj de fecha/hora	152
	Factor de corrección de tiempo real	153
Parte III	Descripción de lenguajes de Twido	155
	Presentación	155
Capítulo 9	Lenguaje Ladder Logic	157
	Presentación	157
	Introducción a los diagramas Ladder Logic	158
	Principios de programación para diagramas Ladder Logic	160
	Bloque de diagramas Ladder Logic	162

	Elementos gráficos del lenguaje Ladder Logic	165
	Instrucciones Ladder Logic especiales OPEN y SHORT	168
	Consejos sobre programación	169
	Reversibilidad de Ladder Logic/Lista	174
	Directrices para la reversibilidad Ladder Logic/Lista.	176
	Documentación del programa	178
Capítulo 10	Lenguaje de lista de instrucciones	181
	Presentación	181
	Vista general de programas de lista	182
	Operación de las instrucciones de lista	184
	Instrucciones del lenguaje de lista	185
	Utilización de paréntesis	189
	Instrucciones de stack (MPS, MRD, MPP)	191
Capítulo 11	Grafcet	193
	Presentación	193
	Descripción de las instrucciones Grafcet	194
	Descripción de la estructura del programa Grafcet.	198
	Acciones asociadas a pasos Grafcet	202
Parte IV	Descripción de instrucciones y funciones	205
	Presentación	205
Capítulo 12	Instrucciones básicas	207
	Presentación	207
12.1	Procesamiento booleario.	208
	Introducción al procesamiento booleario.	208
	Instrucciones booleanas	209
	Comprensión del formato para describir instrucciones booleanas	212
	Instrucciones de carga (LD, LDN, LDR, LDF)	214
	Instrucciones de almacenamiento (ST, STN, R, S)	216
	Instrucciones AND lógicas (AND, ANDN, ANDR, ANDF)	218
	Instrucciones OR lógicas (OR, ORN, ORR, ORF)	220
	Instrucciones de OR exclusivo (XOR, XORN, XORR, XORF)	222
	Instrucción NOT (N)	224
12.2	Bloques de función básicos.	225
	Presentación	225
	Bloques de función básicos.	226
	Principios para programar bloques de función	228
	Bloque de función del temporizador (%Tmi)	230
	Tipo de temporizador TOF	232
	Tipo de temporizador TON	233
	Tipo de temporizador TP.	234
	Programación y configuración de temporizadores	235
	Bloque de función del contador progresivo/regresivo (%Ci)	238

	Programación y configuración de contadores	242
	Bloque de función del registro de bits de desplazamiento (%SBRi)	243
	Bloques de función del contador de pasos (%SCi)	246
12.3	Procesamiento numérico	250
	Introducción al procesamiento numérico	250
	Introducción a las instrucciones numéricas	251
	Instrucciones de asignación	252
	Instrucciones de comparación	256
	Instrucciones aritméticas	258
	Instrucciones de lógica	262
	Instrucciones de desplazamiento	264
	Instrucciones de conversión	266
12.4	Instrucciones del programa	268
	Introducción a instrucciones del programa	268
	Instrucciones END	269
	Instrucción NOP	271
	Instrucciones de salto	272
	Instrucciones de subrutina	273
Capítulo 13	Instrucciones avanzadas	275
	Presentación	275
13.1	Bloques de función avanzados	276
	Presentación	276
	Objetos de palabra y de bit asociados a bloques de función avanzados	277
	Principios de programación para bloques de función avanzados	279
	Bloque de función de registro LIFO/FIFO (%Ri)	282
	Operación LIFO	284
	Operación FIFO	285
	Programación y configuración de registros	286
	Bloque de función de modulación de ancho de pulsos (%PWM)	289
	Bloque de función de la salida del generador de pulsos (%PLS)	293
	Bloque de función del controlador del conmutador de tambor (%DR)	296
	Operación de bloque de función del controlador del conmutador de tambor	298
	Programación y configuración de los controladores del conmutador de tambor	300
	Bloque de función de contador rápido (%FC)	302
	Bloque de función de contador muy rápido (%VFC)	306
	Transmisión/recepción de mensajes - La instrucción de intercambio (EXCH)	318
	Bloque de función de control de intercambio (%MSG)	319
13.2	Funciones de reloj	323
	Presentación	323
	Funciones de reloj	324
	Fechadores	325
	Fijación de la fecha y la hora	328
	Establecimiento de la fecha y la hora	330

Capítulo 14	Bits de sistema y palabras de sistema	335
	Presentación	335
	Bits del sistema (%S)	336
	Palabras de sistema (%SW)	342
Glosario	353
Índice	365

Información de seguridad



Información importante

AVISO

Lea atentamente estas instrucciones y observe el equipo para familiarizarse con el dispositivo antes de instalarlo, utilizarlo o realizar su mantenimiento. Los mensajes especiales que se ofrecen a continuación pueden aparecer a lo largo de la documentación o en el equipo para advertir de peligros potenciales o para ofrecer información que aclara o simplifica los distintos procedimientos.



La inclusión de este icono en una etiqueta de peligro o advertencia indica un riesgo de descarga eléctrica, que puede provocar daños personales si no se siguen las instrucciones.



Este es el icono de alerta de seguridad. Se utiliza para advertir de posibles riesgos de daños personales. Observe todos los mensajes que siguen a este icono para evitar posibles daños personales o incluso la muerte.



PELIGRO

PELIGRO indica una situación inminente de peligro que, si no se evita, puede **provocar** daños en el equipo, lesiones graves o incluso la muerte.



ADVERTENCIA

ADVERTENCIA indica una posible situación de peligro que, si no se evita, puede **provocar** daños en el equipo, lesiones graves o incluso la muerte.



AVISO

AVISO indica una posible situación de peligro que, si no se evita, puede **provocar** lesiones o daños en el equipo.

TENGA EN CUENTA

El mantenimiento de equipos eléctricos deberá ser realizado sólo por personal cualificado. Schneider Electric no asume las responsabilidades que pudieran surgir como consecuencia de la utilización de este material. Este documento no es un manual de instrucciones para personas sin formación. Las instrucciones de montaje e instalación figuran en el Manual de referencia de hardware Twido, TWD USE 10AS.

© 2002 Schneider Electric. Reservados todos los derechos.

Información de seguridad adicional

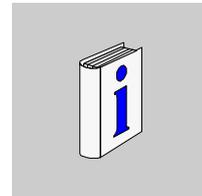
Las personas responsables de la aplicación, la implementación y el uso de este producto deben asegurarse de que se hayan tenido en cuenta todas las consideraciones de diseño necesarias y se hayan respetado totalmente las leyes, los requisitos de seguridad y de rendimiento, las regulaciones, los códigos y las normas aplicables.

Advertencias y avisos generales

	ADVERTENCIA
	<p>PELIGRO DE EXPLOSIÓN</p> <ul style="list-style-type: none"> ● La sustitución de los componentes puede anular la conformidad de Clase I, Div. 2. ● No desconecte el equipo a menos que haya desconectado la alimentación eléctrica o se cerciore de que la zona no es peligrosa. <p>Si no se respetan estas precauciones pueden producirse graves daños corporales y/o materiales.</p>

	ADVERTENCIA
	<p>OPERACIÓN INVOLUNTARIA DEL EQUIPO</p> <ul style="list-style-type: none"> ● Desconecte la alimentación antes de instalar, desmontar, cablear o realizar labores de mantenimiento. ● Este producto no está diseñado para su uso en condiciones peligrosas para la seguridad. En caso de que existan riesgos para el personal o los equipos, utilice los dispositivos de bloqueo de seguridad adecuados. ● No desmonte, repare o modifique los módulos. ● Este controlador está diseñado para su uso dentro de una caja. ● Instale los módulos en las condiciones de funcionamiento descritas. ● Utilice la alimentación de sensores sólo para alimentar los sensores conectados al módulo. ● Utilice un fusible aprobado por IEC60127 en la línea de alimentación y en el circuito de salida para responder a las necesidades de tensión y corriente. Fusible recomendado: Littelfuse 5 x 20 mm de acción retardada, serie 218000/Tipo T. <p>Si no se respetan estas precauciones pueden producirse graves daños corporales y/o materiales.</p>

Acerca de este libro



Presentación

Objeto

Este es el manual de referencia de software de los controladores programables Twido. Está compuesto por las siguientes partes principales:

- Descripción del software de programación Twido e introducción a los fundamentos necesarios para programar controladores Twido.
- Descripción de comunicaciones, gestión de E/S analógicas y otras funciones especiales.
- Descripción de los lenguajes de software utilizados para crear programas Twido.
- Descripción de las instrucciones y funciones de los controladores Twido.

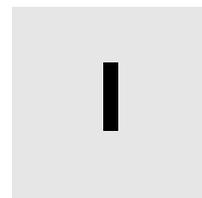
Campo de aplicación

La información contenida en este manual **sólo** es aplicable a los controladores programables Twido.

Advertencia

Schneider Electric no se hace responsable de cualquier error que pudiera aparecer en este documento. Está prohibida la reproducción de cualquier parte de este documento, en cualquier forma o medio, incluido el electrónico, sin autorización previa y por escrito de Schneider Electric.

Descripción de software de Twido



Presentación

Vista general

Esta parte proporciona una introducción a los lenguajes de software y a la información básica requerida para crear programas de control para controladores programables Twido.

Contenido

Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
1	Introducción al software Twido	17
2	Objetos de lenguaje Twido	23
3	Memoria de usuario	43
4	Modos de funcionamiento del controlador	47

Introducción al software Twido



Presentación

Vista general

Este capítulo ofrece una breve introducción a TwidoSoft, que es el software de configuración y programación para los controladores Twido, y a Lista, Ladder Logic y Grafcet, que son lenguajes de programación utilizados para crear programas de control.

Contenido:

Este capítulo contiene los siguiente apartados:

Apartado	Página
Introducción a TwidoSoft	18
Introducción a los lenguajes de Twido	19

Introducción a TwidoSoft

Introducción

TwidoSoft es un entorno de desarrollo gráfico para crear, configurar y mantener aplicaciones para controladores programables Twido. TwidoSoft permite introducir programas de control utilizando los editores de programa de lista o Ladder Logic de TwidoSoft y, a continuación, transferir el programa para ejecutarlo en un controlador.

TwidoSoft

TwidoSoft es un programa basado en Windows de 32 bits para un ordenador personal (PC) que se ejecute bajo los sistemas operativos Microsoft Windows 98 segunda edición o Microsoft Windows 2000 Professional.

Las principales funciones del software TwidoSoft son:

- Interfase de usuario estándar de Windows
- Programar y configurar controladores Twido
- Control y comunicaciones del controlador

Si desea más información, consulte el Manual de funcionamiento de Twido.

Introducción a los lenguajes de Twido

Introducción Un controlador programable lee entradas, escribe salidas y resuelve lógica basada en un programa de control. Crear un programa de control para un controlador Twido consiste en escribir una serie de instrucciones en uno de los lenguajes de programación de Twido.

Lenguajes de programación de Twido Para crear programas de control Twido se pueden utilizar los siguientes lenguajes de programación:

- Lenguaje de lista de instrucciones
Un programa de lista de instrucciones se compone de una serie de expresiones lógicas escritas como una secuencia de instrucciones booleanas.
- Diagramas Ladder Logic
Un diagrama Ladder Logic es una forma gráfica de mostrar una expresión lógica.
- Grafcet
Twido admite las instrucciones de lista Grafcet, pero no Grafcet gráfico.

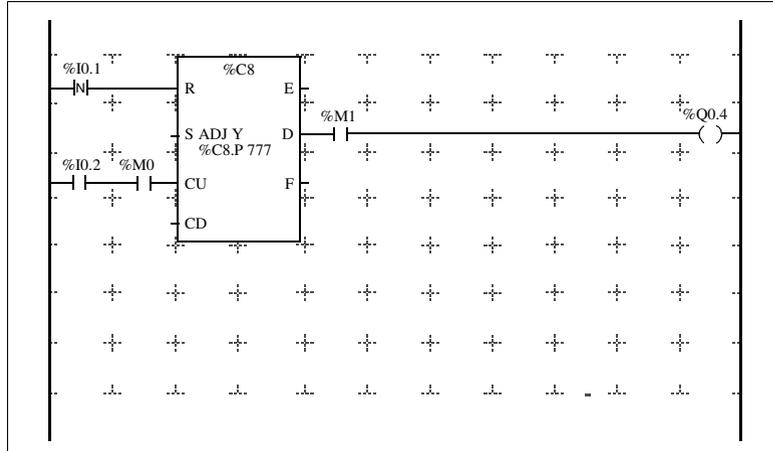
Puede utilizar un ordenador personal (PC) para crear y editar programas de control Twido mediante estos lenguajes de programación. La función de reversibilidad de Lista/Ladder Logic permite pasar un programa de Lista a Ladder Logic y viceversa, según convenga.

Lenguaje de lista de instrucciones Un programa escrito en lenguaje de lista de instrucciones ejecutado de manera secuencial por el controlador. A continuación aparece un ejemplo de programa de Lista.

```
0  BLK  %C8
1  LDF  %I0.1
2  R
3  LD   %I0.2
4  AND  %M0
5  CU
6  OUT_BLK
7  LD   D
8  AND  %M1
9  ST   %Q0.4
10 END_BLK
```

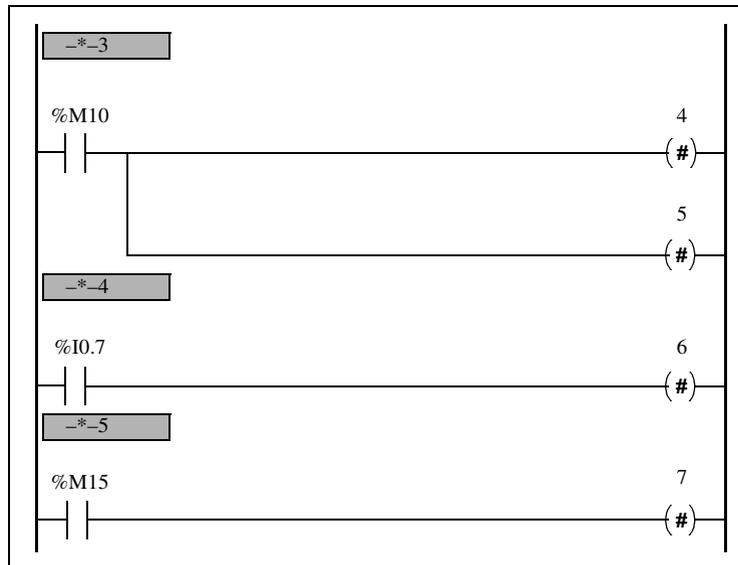
**Diagramas
Ladder Logic**

Los diagramas Ladder Logic son similares a los diagramas de lógica de relé que se utilizan para representar los circuitos de control de relé. Los elementos gráficos, como bobinas, contactos y bloques, representan las instrucciones. A continuación aparece un ejemplo de diagrama Ladder Logic.



Lenguaje Grafcet Grafcet es un método analítico que divide cualquier sistema de control secuencial en una serie de pasos con acciones, transiciones y condiciones asociadas. La ilustración que aparece a continuación muestra ejemplos de instrucciones Grafcet en programas Lista y Ladder Logic respectivamente.

0	-*-	3
1	LD	%M10
2	#	4
3	#	5
4	-*-	4
5	LD	%I0.7
6	#	6
7	-*-	5
8	LD	%M15
9	#	7
10	...	



Objetos de lenguaje Twido

2

Presentación

Vista general

Este capítulo contiene información detallada acerca de los objetos de lenguaje utilizados para programar controladores Twido.

Contenido:

Este capítulo contiene los siguiente apartados:

Apartado	Página
Validación de objetos de lenguaje	24
Objetos de bit	25
Objetos de palabra	28
Direccionamiento de objetos de bit	31
Direccionamiento de objetos de palabra	32
Direccionamiento de entradas/salidas	33
Direccionamiento de red	35
Objetos de bloques de función	36
Objetos estructurados	37
Palabras indexadas	40
Simbolización de objetos	42

Validación de objetos de lenguaje

Introducción

Los objetos de bit y de palabra son válidos si se les ha asignado espacio de memoria en el controlador. Para ello, se deben utilizar en la aplicación antes de descargarlos en el controlador.

Ejemplo

El rango de objetos válidos oscila entre cero y la referencia máxima para ese tipo de objeto. Por ejemplo, si la referencia máxima de su aplicación para palabras de memoria es %MW9, entonces está asignado el espacio de %MW0 a %MW9. En este ejemplo, %MW10 no es válido y no se puede acceder a él ni interna ni externamente.

Objetos de bit

Introducción

Los objetos de bit son bits de variables de software, es decir, bits simples de datos que se pueden utilizar como operandos y verificar mediante instrucciones booleanas. A continuación se ofrece una lista de objetos de bit.

- Bits de E/S
 - Bits internos (bits de memoria)
 - Bits de sistema
 - Bits de pasos
 - Bits extraídos de palabras
-

Lista de bits de operandos

En la tabla siguiente se enumeran y describen todos los objetos de bit principales que se utilizan como operandos en instrucciones booleanas.

Tipo	Descripción	Dirección o valor	Número máximo	Acceso de escritura ¹
Valores inmediatos	0 ó 1 (False o True)	0 ó 1	-	-
Entradas Salidas	Estos bits son las "imágenes lógicas" de los estados eléctricos de las E/S. Se almacenan en la memoria de datos y se actualizan durante cada ciclo de la lógica del programa.	%Ix.y.z ² %Qx.y.z ²	Nota ⁴	No Sí
Interna (memoria)	Los bits internos son áreas de memoria internas utilizadas para almacenar valores intermedios durante la ejecución de un programa. Nota: Los bits de E/S no utilizados no se pueden emplear como bits internos.	%Mi	128 TWDLCAA10 DRF, TWDLCAA16 DRF 256 Todos los controladores restantes	Sí
Sistema	Los bits de sistema %S0 a %S127 supervisan el funcionamiento correcto del controlador y la correcta ejecución del programa de aplicación.	%Si	128	Según i
Bloques de función	Los bits de bloques de función corresponden a las salidas de los bloques de función. Estas salidas pueden estar conectadas directamente o ser utilizadas como un objeto.	%T Mi.Q, %Ci.P, etc.	Nota ⁴	No ³
Bloques de función reversibles	Bloques de función programados mediante las instrucciones de programación reversibles BLK, OUT_BLK y END_BLK.	E, D, F, Q, TH0, TH1	Nota ⁴	No
Extractos de palabras	Uno de los 16 bits de algunas palabras se puede extraer como bit de operando.	Varía	Varía	Varía

Tipo	Descripción	Dirección o valor	Número máximo	Acceso de escritura ¹
Pasos Grafcet	Los bits %X1 a %Xi están asociados a pasos Grafcet. El bit de pasos Xi se pone a 1 cuando el paso correspondiente está activo. Se pone a 0 cuando el paso se desactiva.	%X21	62 TWDLCAA10 DRF, TWDLCAA16 DRF 94 TWDLCAA24 DRF, Controladores modulares	Sí

Notas:

1. Escrito por el programa o mediante el editor de tablas de animación.
2. Consulte el direccionamiento de E/S.
3. Excepto en el caso de %SB Ri,j y %SC i,j, estos bits se pueden leer y escribir.
4. El número está determinado por el modelo del controlador.

Objetos de palabra

Introducción

Objetos de palabra enviados en forma de palabras de 16 bits almacenados en la memoria de datos y que pueden contener un valor entero entre -32768 y 32767 (excepto para el bloque de función de contador rápido, que está entre 0 y 65535). Ejemplos de objetos de palabras:

- Valores inmediatos
- Palabras internas (%MWi) (palabras de memoria)
- Palabras constantes (%KWi)
- Palabras de intercambio de E/S (%IWi, %QWi)
- Palabras de sistema (%SWi)
- Bloques de función (datos de ejecución o configuración)

Formatos de palabra

El contenido de las palabras o los valores se almacena en la memoria de usuario en código binario de 16 bits (complemento de dos) mediante la convención que aparece a continuación.

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Posición de bit
0	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1	Estado de bit
+	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	Valor de bit

En la notación binaria con signo, el bit 15 se asigna por convención al signo del valor codificado:

- Si el bit 15 es 0: el contenido de la palabra es un valor positivo.
- Si el bit 15 es 1: el contenido de la palabra es un valor negativo (los valores negativos se expresan en lógica de complemento de dos).

Las palabras y los valores inmediatos se pueden introducir o recuperar en el siguiente formato:

- Decimal
Mín.: -32768. Máx: 32767 (por ejemplo, 1579)
- Hexadecimal
Mín.: 16#0000. Máx: 16#FFFF (por ejemplo, 16#A536)
Sintaxis alternativa: #A536

Descripción de los objetos de palabra

En la siguiente tabla se describen los objetos de palabra.

Palabras	Descripción	Dirección o valor	Número máximo	Acceso de escritura ¹
Valores inmediatos	Se trata de valores enteros expresados en el mismo formato que las palabras de 16 bits, lo que permite que los valores se puedan asignar a estas palabras.		-	No
	Base 10	de -32768 a 32767		
	Base 16	de 16#0000 a 16#FFFF		
Interna (memoria)	Utilizadas como palabras "de trabajo" para almacenar valores durante la operación en la memoria de datos. Las palabras %MWO a %MW255 se leen o escriben directamente mediante el programa.	%MWi	1500	Sí
Constantes	Almacenan mensajes alfanuméricos o constantes. Su contenido sólo se puede escribir o modificar utilizando TwidoSoft durante la configuración. Las palabras constantes %KW0 a %KW63 tienen acceso de sólo lectura para el programa.	%KWi	64	Sí (sólo utilizando TwidoSoft)
Sistema	Estas palabras de 16 bits tienen diversas funciones: <ul style="list-style-type: none"> ● Proporcionar acceso a los datos que proceden directamente del controlador leyendo palabras %SWi (por ejemplo, potenciómetros). ● Realizar operaciones en la aplicación (por ejemplo, ajustar fechadores). 	%SWi	128	Según i
Bloques de función	Estas palabras corresponden a valores o parámetros actuales de bloques de función.	%TM2.P, %Ci.P, etc.		Sí

Palabras	Descripción	Dirección o valor	Número máximo	Acceso de escritura ¹
Palabras de intercambio de E/S	Asignadas a controladores conectados como conexiones remotas. Estas palabras se utilizan para la comunicación entre controladores.			
	Entradas	%IWi.j	Nota ²	No
	Salidas	%QWi.j	Nota ²	Sí
Bits extraídos	Es posible extraer uno de los 16 bits de las siguientes palabras.			
	Interno	%MWi:Xk	1500	Sí
	Sistema	%SWi:Xk	128	Depende de i
	Constantes	%KWi:Xk	64	No
	Entrada	%IWi.j:Xk	Nota ²	No
	Salida	%QWi.j:Xk	Nota ²	Sí

Nota:

1. Escrito por el programa o mediante el editor de tablas de animación.
2. El número está determinado por el modelo del controlador.

Direccionamiento de objetos de bit

Formato

Utilice el siguiente formato para direccionar objetos de bit de pasos, de sistema e internos.

%	M, S, o X	i
Símbolo	Tipo de objeto	Número

Descripción

En la tabla siguiente se describen los elementos del formato de direccionamiento.

Grupo	Elemento	Descripción
Símbolo	%	El símbolo de porcentaje siempre precede a una variable de software.
Tipo de objeto	M	Los bits internos almacenan valores intermedios mientras se está ejecutando un programa.
	S	Los bits de sistema proporcionan información de control y de estado del controlador.
	X	Los bits de pasos proporcionan información de estado de las actividades de pasos.
Número	i	El valor numérico máximo depende del número de objetos configurados.

Ejemplos de direcciones de objetos de bit:

- %M25 = bit interno número 25
- %S20 = bit de sistema número 20
- %X6 = bit de pasos número 6

Objetos de bit extraídos de palabras

TwidoSoft se utiliza para extraer uno de los 16 bits de las palabras. La dirección de la palabra se completa mediante la fila de bit extraída de acuerdo con la sintaxis siguiente.

WORD	: X	k
Dirección de palabra	Posición k = rango de bits 0 - 15 en la dirección de palabra.	

Ejemplos:

- %MW5:X6 = bit número 6 de la palabra interna %MW5
- %QW5.1:X10 = bit número 10 de la palabra de salida %QW5.1

Direccionamiento de objetos de palabra

Introducción

El direccionamiento de objetos de palabra, excepto para el direccionamiento de entrada/salida (consulte *Direccionamiento de entradas/salidas*, p. 33) y bloques de función (consulte *Objetos de bloques de función*, p. 36), sigue el formato que aparece a continuación.

Formato

Utilice el siguiente formato para direccionar palabras del sistema, internas y constantes.

%	M, K o S	W	i
Símbolo	Tipo de objeto	Formato	Número

Descripción

La siguiente tabla describe los elementos en formato de direccionamiento.

Grupo	Elemento	Descripción
Símbolo	%	El símbolo de porcentaje siempre precede a una dirección interna.
Tipo de objeto	M	Las palabras internas almacenan valores intermedios mientras se está ejecutando un programa.
	K	Las palabras constantes almacenan valores constantes o mensajes alfanuméricos. Su contenido sólo puede sobrescribirse o modificarse utilizando TwidoSoft.
	S	Las palabras de sistema proporcionan información de control y de estado del controlador.
Formato	W	Palabra de 16 bits.
Número	i	El valor numérico máximo depende del número de objetos configurados.

Ejemplos de direccionamiento de objetos de palabra:

- %MW15 = número de palabra interna 15
- %KW26 = número de palabra constante 26
- %SW30 = número de palabra de sistema 30

Direccionamiento de entradas/salidas

Introducción

Cada punto de entrada/salida (E/S) de una configuración Twido tiene una única dirección: por ejemplo, una entrada específica de un controlador recibe la asignación de la dirección de "%I0.0.4".

Las direcciones de E/S pueden asignarse para el siguiente hardware:

- Controlador configurado como master de conexión remota
- Controlador configurado como E/S remota
- Módulos de ampliación de E/S

Varias referencias a una salida o bobina

En un programa, puede tener varias referencias a una sola salida o bobina. Sólo se actualiza el resultado de la última resuelta en las salidas de hardware. Por ejemplo, puede utilizarse %Q0.0.0 más de una vez en un programa y no aparecerá un mensaje de advertencia para varias apariciones. De este modo, es importante confirmar cuál va a ser la salida en la operación deseada.

	AVISO
	<p>Operación involuntaria</p> <p>No se proporciona ninguna advertencia o comprobación de salida duplicada. Compruebe el uso de las salidas o bobinas antes de realizar cambios en las mismas dentro de la aplicación.</p> <p>Si no se respetan estas precauciones pueden producirse daños corporales y/o materiales</p>

Formato

Utilice el siguiente formato para direccionar las entradas/salidas.

%	I, Q	x	y	z
Símbolo	Tipo de objeto	Posición del controlador	Tipo de E/S	Número de canal

Descripción

La tabla que aparece a continuación describe el formato de direccionamiento de E/S.

Grupo	Elemento	Valor	Descripción
Símbolo	%	-	El símbolo de porcentaje siempre precede a una dirección interna.
Tipo de objeto	I	-	Entrada. La "imagen lógica" del estado eléctrico de un controlador o entrada del módulo de E/S de ampliación.
	Q	-	Salida. La "imagen lógica" del estado eléctrico de un controlador o salida del módulo de E/S de ampliación.
Posición del controlador	x	0 1 - 7	Controlador master (master de conexión remota). Controlador remoto (slave de conexión remota).
Tipo de E/S	y	0 1 - 7	Unidad de E/S (E/S local del controlador). Módulos de ampliación de E/S.
Número de canal	z		Número de canal de E/S en el módulo de E/S de ampliación o el controlador. El número de puntos de E/S disponibles depende del modelo de controlador o del tipo de módulo de E/S de ampliación.

Ejemplos

La tabla que aparece a continuación muestra algunos ejemplos de direccionamiento de E/S.

Objeto de E/S	Descripción
%I0.0.5	Punto de entrada número 5 en el controlador base (E/S local).
%Q0.3.4	Punto de salida número 4 en el módulo de E/S de ampliación en la dirección de ampliación número 3 para el controlador base (E/S de ampliación).
%I0.0.3	Punto de entrada número 3 en el controlador base.
%I3.0.1	Punto de entrada número 1 en el controlador de E/S remotas en la dirección de conexión remota número 3.
%I0.3.2	Punto de entrada número 2 en el módulo de E/S de ampliación en la dirección número 3 para el controlador base.

Direccionamiento de red

Introducción En una red de conexión remota Twido, los datos de aplicación se intercambian mediante las palabras de red %INW y %QNW entre controladores peer y el controlador master. Para obtener más información al respecto, consulte "*Comunicaciones, p. 71*".

Formato Utilice el siguiente formato para el direccionamiento de red.

%	IN, QN	W	x	j
Símbolo	Tipo de objeto	Formato	Posición del controlador	Palabra

Descripción de formato En la tabla que aparece a continuación se describe el formato de direccionamiento de red.

Grupo	Elemento	Valor	Descripción
Símbolo	%	-	El símbolo de porcentaje siempre precede a una dirección interna.
Tipo de objeto	IN	-	Palabra de entrada de red. Transferencia de datos de master a peer.
	QN	-	Palabra de salida de red. Transferencia de datos de peer a master.
Formato	W	-	Palabra de 16 bits.
Posición del controlador	x	0 1 - 7	Controlador master (master de conexión remota). Controlador remoto (slave de conexión remota).
Palabra	j	0 - 3	Cada controlador peer utiliza entre una y cuatro palabras para intercambiar datos con el controlador master.

Ejemplos La tabla que aparece a continuación contiene algunos ejemplos de direccionamiento de red.

Objeto de red	Descripción
%INW3.1	Palabra de red número 1 del controlador remoto número 3.
%QNW0.3	Palabra de red número 3 del controlador base.

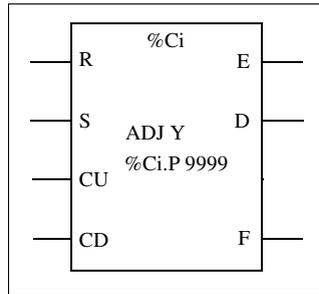
Objetos de bloques de función

Introducción

Los bloques de función proporcionan objetos de bit y palabras específicas a las que puede acceder el programa.

Ejemplo de un bloque de función

En la siguiente ilustración representa un bloque de función de contador.



Bloque del contador
progresivo/regresivo

Objetos de bit

Los objetos de bit corresponden a las salidas de bloque. A estos bits pueden acceder las instrucciones de verificación booleanas a través de uno de los métodos siguientes:

- Directamente (por ejemplo, LD E) si están cableados al bloque en programación reversible (consulte "*Principios para programar bloques de función, p. 228*").
- Especificando el tipo de bloque (por ejemplo, LD %Ci.E).

Puede accederse a las entradas en forma de instrucciones.

Objetos de palabra

Los objetos de palabra corresponden a parámetros especificados y a valores del siguiente modo:

- Parámetros de configuración de bloques: se puede acceder a algunos parámetros a través del programa (por ejemplo, parámetros de preselección) y a otros no (por ejemplo, base de tiempo).
 - Valores actuales: por ejemplo, %Ci.V, el valor de conteo actual.
-

Objetos accesibles a través del programa

Consulte las secciones siguientes para obtener una lista de los objetos accesibles a través del programa.

- Para los bloques de función básicos, consulte "*Bloques de función básicos, p. 226*".
 - Para los bloques de función avanzados, consulte *Objetos de palabra y de bit asociados a bloques de función avanzados, p. 277*.
-

Objetos estructurados

Introducción Los objetos estructurados son combinaciones de objetos simples. Twido admite los siguientes tipos de objetos estructurados:

- Cadenas de bits
 - Tablas de palabras
-

Cadenas de bits Las cadenas de bits son series de bits de objetos adyacentes del mismo tipo y con una longitud definida (L).

Ejemplo: Cadena de bits %M8:6

%M8 %M9 %M10 %M11 %M12 %M13

--	--	--	--	--	--

<p>Nota: %M8:6 es aceptable (8 es múltiplo de 8), pero %M10:16 no es aceptable (10 no es múltiplo de 8).</p>

Las cadenas de bits se pueden utilizar con la instrucción de asignación (consulte "*Instrucciones de asignación, p. 252*").

Tipos de bits disponibles

Tipos de bits disponibles para cadenas de bits.

Tipo	Dirección	Tamaño máximo	Acceso de escritura
Bits de entrada binaria	%I0.0:L o %I1.0:L1	0<L<17	No
Bits de salida binaria	%Q0.0:L o %Q1.0:L1	0<L<17	Sí
Bits de sistema	%Si:L con i múltiplo de 8	0<L<17 e i+L-128	En función de i
Bits de pasos Grafcet	%Xi:L con i múltiplo de 8	0<L<17 e i+L-95	Sí (por el programa)
Bits internos	%Mi:L con i múltiplo de 8	0<L<17 e i+L-256	Sí

Nota: (1) Sólo se pueden enviar los bits 0...L-1. No todas las E/S se pueden enviar en cadenas de bits.

Tablas de palabras

Las tablas de palabras son series de palabras adyacentes del mismo tipo y con una longitud definida (L).

Ejemplo: Tabla de palabras %KW10:7

%KW10	16 bits
%KW16	

Las tablas de palabras se pueden utilizar con la instrucción de asignación (consulte "*Instrucciones de asignación, p. 252*").

Tipos de palabras disponibles

Tipos de palabras disponibles para tablas de palabras.

Tipo	Dirección	Tamaño máximo	Acceso de escritura
Palabras internas	%MWi:L	$0 < L < 256$ e $i + L < o = 1500$	Sí
Palabras constantes	%KW:L	$0 < L$ e $i + L - 64$	No
Palabras de sistema	%SWi:L	$0 < L$ e $i + L - 128$	En función de i

Palabras indexadas

Introducción

Una palabra indexada es una palabra interna o constante con una dirección de objeto indexada. Existen dos tipos de direccionamiento de objetos:

- Direccionamiento directo
 - Direccionamiento indexado
-

Direccionamiento directo

Una dirección directa de un objeto se ajusta y se define cuando se escribe un programa.

Ejemplo: %M26 es un bit interno con la dirección directa 26.

Direccionamiento indexado

Una dirección indexada de un objeto proporciona un método para modificar la dirección de un objeto añadiendo un índice a la dirección directa de un objeto. El contenido del índice se añade a la dirección directa del objeto. El índice se define mediante una palabra interna %MWi. El número de "palabras indexadas" es ilimitado.

Ejemplo: %MW108[%MW2] es una palabra con una dirección compuesta por la dirección directa 108 más el contenido de la palabra %MW2.

Si la palabra %MW2 tiene un valor de 12, escribir en %MW108[%MW2] es igual que escribir en %MW120 (108 más 12).

Palabras disponibles para el direccionamiento indexado

A continuación se enumeran los tipos de palabras disponibles para el direccionamiento indexado.

Tipo	Dirección	Tamaño máximo	Acceso de escritura
Palabras internas	%MWi[%MWj]	0-i < o = %MWj < 1500	Sí
Palabras constantes	%KWi[%MWj]	0-i < o = %MWj < 64	No

Las palabras indexadas se pueden utilizar con la instrucción de asignación (consulte "*Instrucciones de asignación*, p. 252") y en instrucciones de comparación (consulte "*Instrucciones de comparación*, p. 256"). Este tipo de direccionamiento habilita series de objetos del mismo tipo (como palabras internas y constantes) para que se exploren de forma sucesiva modificando el contenido de la palabra de índice a través del programa.

Bit de sistema de desborde de índice %S20

El desborde de índice se produce cuando la dirección de un objeto indexado excede los límites de la zona de memoria que contiene el mismo tipo de objeto. En resumen:

- La dirección de objeto más el contenido del índice es menor que 0.
- La dirección de objeto más el contenido del índice es mayor que la palabra de mayor tamaño a la que se hace referencia directamente en la aplicación. El número máximo es 1499 (para las palabras %MWi) o 63 (para las palabras %KW_i).

En caso de desborde de índice, el sistema pone el bit %S20 a 1 y se asigna un valor de índice 0 al objeto.

Nota: El usuario es responsable de controlar cualquier desborde. El programa de usuario debe leer el bit %S20 para un posible procesamiento. El usuario debe confirmar que se restablece a 0.

%S20 (estado inicial = 0):

- En caso de desborde de índice: puesto a 1 por el sistema.
 - Acuse de recibo de desborde: puesto a 0 por el usuario después de modificar el índice.
-

Simbolización de objetos

Introducción	Se utilizan símbolos para direccionar objetos de lenguaje de software Twido por nombre o mnemotecnica personalizada. El uso de símbolos permite un rápido examen y análisis de la lógica del programa y simplifica enormemente el desarrollo y la comprobación de una aplicación.
Ejemplo	Por ejemplo, WASH_END es un símbolo que puede utilizarse para identificar un bloque con la función temporizador que represente el final de un ciclo de lavado. Recordar el propósito de este nombre debe ser más sencillo que intentar recordar la función de una dirección de programa como %TM3.
Directrices para definir símbolos	A continuación aparecen directrices para definir símbolos: <ul style="list-style-type: none">● Un máximo de 32 caracteres.● Letras (A-Z), números (0 -9) o guiones bajos (_).● El primer carácter debe ser alfabético o acentuado. No puede utilizar el signo de porcentaje (%).● No utilice espacios ni caracteres especiales.● Sin distinguir mayúsculas y minúsculas. Por ejemplo, Pump1 y PUMP1 son el mismo símbolo y sólo puede utilizarse una vez en una aplicación.
Edición de símbolos	Los símbolos se definen y asocian con los objetos de lenguaje en el editor de símbolos. Los símbolos y sus comentarios se guardan con la aplicación en el disco duro del PC, pero no en el controlador. Por lo tanto, no pueden transferirse con la aplicación al controlador. Para obtener más detalles sobre el uso de símbolos consulte el manual de funcionamiento de TwidoSoft.

Memoria de usuario



Estructura de la memoria de usuario

Introducción	<p>La memoria del controlador accesible a través de una aplicación de usuario está dividida en dos partes diferentes:</p> <ul style="list-style-type: none">● Valores de bit● Valores de palabra (valores con signo de 16 bits)
Memoria de bits	<p>La memoria de bits se almacena en la memoria RAM interna que está integrada en el controlador. Contiene el mapa de 1280 objetos de bit.</p>
Función de la memoria de palabras	<p>La memoria de palabras (16 bits) admite:</p> <ul style="list-style-type: none">● Datos: datos de sistema y datos de aplicación dinámicos.● Programa: descriptores y código ejecutable para tareas.● Constantes: palabras constantes, valores iniciales y configuración de entrada/salida.

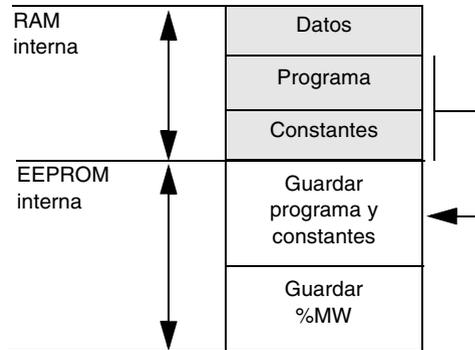
Tipos de memoria

A continuación se enumeran los distintos tipos de memoria para los controladores Twido.

- RAM interna (integrada)
Esta es la memoria RAM integrada del controlador. Los 10 primeros KB de la memoria RAM interna constituyen la RAM rápida. Los 32 KB siguientes constituyen la RAM estándar. La RAM interna contiene el programa, constantes y datos.
- EEPROM interna
EEPROM integrada de 32 KB que proporciona una copia de seguridad interna en el controlador de una aplicación. Protege la aplicación contra los daños provocados por fallos de batería o cortes de corriente superiores a 30 días. Contiene el programa y constantes.
- Cartucho de copia de seguridad de memoria externa
Cartucho de EEPROM externa opcional para realizar copias de seguridad de una aplicación o para dar cabida a una aplicación más grande. Se puede utilizar para actualizar la aplicación en la RAM del controlador. Contiene el programa y constantes, pero ningún dato.

Estructura sin cartucho de memoria externa

En el diagrama que aparece a continuación se describe la estructura de memoria sin cartucho de memoria externa.



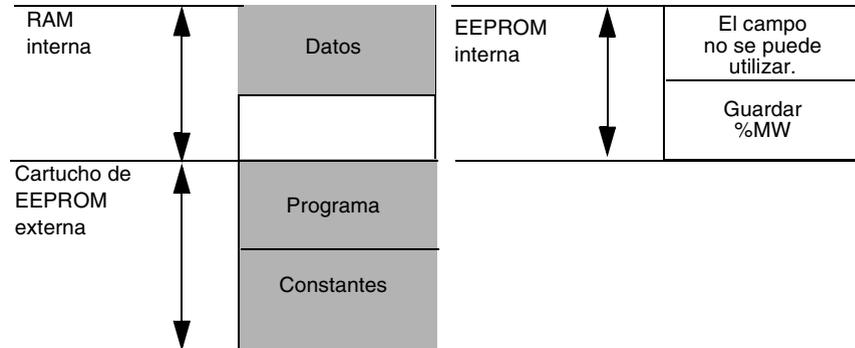
La EEPROM interna está integrada en el controlador y proporciona 32 KB de memoria para lo siguiente:

- El programa de aplicación (32 KB)
- 512 palabras internas (%MWi)

Estructura con cartucho de memoria externa

El cartucho de memoria externa opcional proporciona una copia de seguridad de los programas y constantes, al mismo tiempo que ofrece memoria ampliada para aplicaciones de mayor tamaño.

En el diagrama siguiente se describe la estructura de memoria con cartucho de memoria externa.



La EEPROM interna de 32 KB puede almacenar 512 palabras internas (%MWi).

Almacenamiento de la memoria

La memoria RAM interna del controlador se puede almacenar mediante:

- Batería interna (hasta 30 días)
- EEPROM interna (32 KB como máximo)
- Cartucho de memoria externa opcional (64 KB como máximo)

La transferencia de la aplicación desde la memoria EEPROM interna hasta la memoria RAM se realiza automáticamente cuando la aplicación se pierde en la RAM (si no se ha guardado o si no hay batería).

También se puede realizar una transferencia manual mediante TwidoSoft.

Configuraciones de la memoria

En la tabla que aparece a continuación se describe la configuración de los tipos de memoria posibles en los controladores Twido.

Tipo de memoria	Controladores compactos			Controladores modulares		
	10DRF	16DRF	24DRF	20DUK 20DTK	20DRT	40DUK 40DTK
RAM interna	10 KB	32 KB	32 KB	32 KB	32 KB	32 KB
Memoria ampliada disponible*					64 KB	64 KB
Tamaño máximo de aplicación	10 KB	32 KB	32 KB	32 KB	32 KB o 64 KB*	32 KB o 64 KB*
Copia de seguridad externa máxima	32 KB	32 KB	32 KB	64 KB	32 KB o 64 KB	32 KB o 64 KB

Nota: *La memoria se puede ampliar a 64 KB para los controladores TWDLMDA20DRT, TWDLMDA40DUK y TWDLMDA40DTK instalando el cartucho de memoria externa de 64 KB opcional. El cartucho debe permanecer instalado para ejecutar la aplicación y realizar copias de seguridad.

Modos de funcionamiento del controlador

4

Presentación

Vista general En este capítulo se describen los modos de funcionamiento del controlador y la ejecución cíclica y periódica del programa. Se incluye información detallada sobre cortes de corriente y recuperación de alimentación.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Exploración cíclica	48
Exploración periódica	51
Comprobación del tiempo de ciclo	54
Modos de funcionamiento	56
Comportamiento ante cortes de corriente y recuperación de alimentación	58
Uso de un reinicio en caliente	61
Comportamiento ante un inicio en frío	64
Inicialización del controlador	67

Exploración cíclica

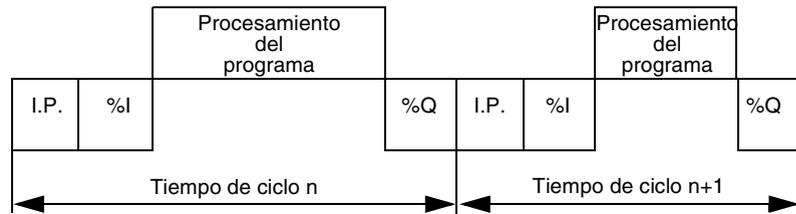
Introducción

La exploración cíclica une los ciclos de tarea de master uno detrás de otro sin esperar por nada excepto el procesamiento inevitable del sistema. Después de efectuar la actualización de las salidas (tercera fase del ciclo de tarea), el sistema ejecuta cierto número de sus propias tareas e inmediatamente dispara otro ciclo de tarea.

Nota: El temporizador watchdog del controlador supervisa el tiempo de ciclo del programa del usuario. Éste no debe exceder los 150 ms, ya que de lo contrario se producirá un fallo que provoque la detención inmediata del controlador en modo de parada. Las salidas en este modo se fuerzan a su estado de retorno predeterminado.

Operación

El siguiente dibujo muestra las fases de ejecución del tiempo de ciclo cíclico.



Descripción de las fases de operación

En la tabla siguiente se describen las fases de operación.

Dirección	Fase	Descripción
I.P.	Procesamiento interno	El sistema supervisa el controlador de forma implícita (gestionando las palabras y los bits de sistema, actualizando los valores de temporizador actuales, actualizando las luces de estado, detectando los cambios entre ejecución/detención, etc.) y procesa las solicitudes de TwidoSoft (modificaciones y animación).
%I	Adquisición de entradas	Se escribe en la memoria el estado de la información relativa a las entradas binarias y del módulo específico de la aplicación asociados a la tarea.
-	Procesamiento o del programa	Ejecución del programa de aplicación escrito por el usuario.
%Q	Actualización de salidas	Se escriben los bits de salida o las palabras asociadas a los módulos discretos y específicos de la aplicación asociados a la tarea según el estado definido por el programa de aplicación.

Modo de funcionamiento

Controlador en ejecución, el procesador realiza las siguientes operaciones:

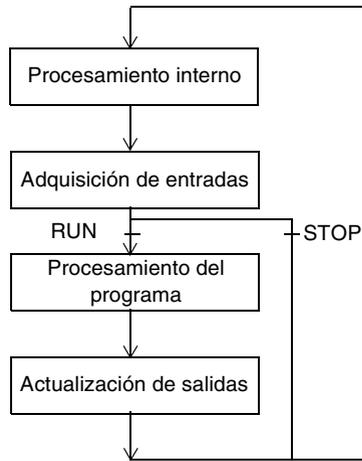
- Procesamiento interno
- Adquisición de entradas
- Procesamiento del programa de aplicación
- Actualización de salidas

Controlador en detención, el procesador realiza las siguientes operaciones:

- Procesamiento interno
- Adquisición de entradas

Ilustración

La siguiente ilustración muestra los ciclos de funcionamiento.



Ciclo de comprobación

El watchdog realiza el ciclo de comprobación.

Exploración periódica

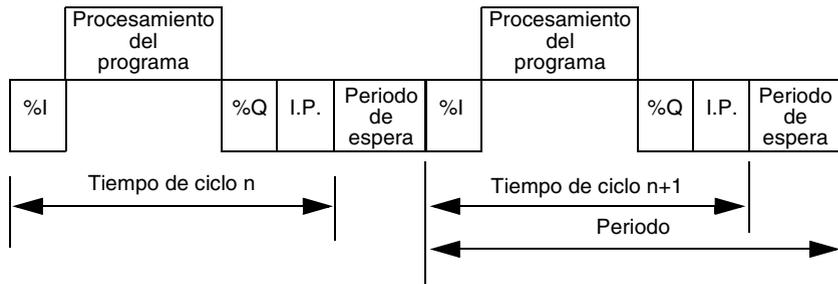
Introducción

En este modo de funcionamiento, la adquisición de entradas, el procesamiento del programa de aplicación y la actualización de salidas se realiza periódicamente con arreglo al tiempo definido en la configuración (de 2 a 150 ms).

Al comienzo del ciclo del controlador, un temporizador, cuyo valor se inicializa durante el periodo definido en la configuración, comienza con el conteo regresivo. El ciclo del controlador debe finalizar antes de que el temporizador haya finalizado y vuelve a ejecutar un nuevo ciclo.

Operación

El siguiente dibujo muestra las fases de ejecución del tiempo de ciclo periódico.



Descripción de las fases de operación

En la tabla siguiente se describen las fases de operación.

Dirección	Fase	Descripción
I.P.	Procesamiento interno	El sistema supervisa el controlador de forma implícita (gestionando las palabras y los bits de sistema, actualizando los valores de temporizador actuales, actualizando las luces de estado, detectando los cambios entre ejecución/detención, etc.) y procesa las solicitudes de TwidoSoft (modificaciones y animación).
%I	Adquisición de entradas	Se escribe en la memoria el estado de la información relativa a las entradas binarias y del módulo específico de la aplicación asociados a la tarea.
-	Procesamiento del programa	Ejecución del programa de aplicación escrito por el usuario.
%Q	Actualización de salidas	Se escriben los bits de salida o las palabras asociadas a los módulos discretos y específicos de la aplicación, asociados a la tarea según el estado definido por el programa de aplicación.

Modo de funcionamiento

Controlador en ejecución, el procesador realiza las siguientes operaciones:

- Orden de procesamiento interno
- Adquisición de entradas
- Procesamiento del programa de aplicación
- Actualización de salidas

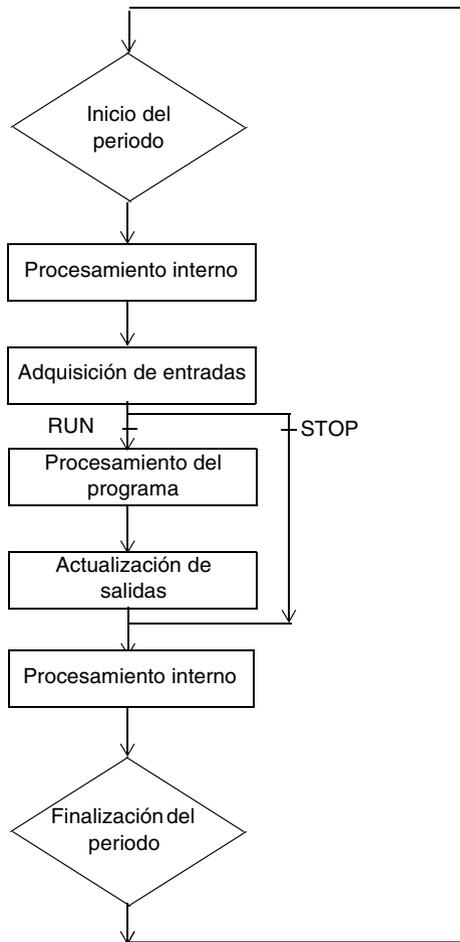
Si el periodo no ha finalizado, el procesador completa su ciclo de funcionamiento hasta el final del periodo de procesamiento interno. Si el tiempo de funcionamiento es superior al asignado al periodo, el controlador indica que se ha superado el periodo, estableciendo el bit de sistema de la tarea %S19 a 1. El proceso continúa y se ejecuta completamente (sin embargo, no debe superar el límite de tiempo del watchdog). El siguiente ciclo está conectado, después de escribir las salidas del ciclo en curso de forma implícita.

Controlador en detención, el procesador realiza las siguientes operaciones:

- Procesamiento interno
- Adquisición de entradas

Ilustración

La siguiente ilustración muestra los ciclos de funcionamiento.

**Ciclo de comprobación**

Se realizan dos comprobaciones:

- Desborde de periodo
- Watchdog

Comprobación del tiempo de ciclo

Generalidades

El ciclo de tarea master se controla mediante un temporizador watchdog llamado Tmax (duración máxima del ciclo de tarea master). Permite mostrar errores de aplicación (bucles infinitos, etc.) y garantiza una duración máxima para actualizar las salidas.

WatchDog del software (operación periódica o cíclica)

En una operación periódica o cíclica, la activación del watchdog provoca un error del software. La aplicación pasa a estado de pausa y establece el bit %S11 a 1. La nueva ejecución de la tarea necesita una conexión a Twido Soft con el fin de analizar la causa del error, la modificación de la aplicación para corregir el error y la nueva ejecución de las solicitudes de inicio y ejecución.

Nota: El estado de pausa se produce cuando la aplicación se detiene inmediatamente debido a un error del software de la aplicación, como un desborde de ciclo. Los datos conservan los valores actuales que permiten un análisis de la causa del error. Todas las tareas se detienen en la instrucción actual. Está disponible la comunicación con el controlador.

Comprobación de la operación periódica

En una operación periódica, se utiliza una comprobación adicional para detectar el periodo que se está excediendo.

- **%S19** indica que se ha superado el periodo. Se establece a:
 - 1 por el sistema cuando el tiempo de ciclo es mayor que el periodo de la tarea.
 - 0 por el usuario.
 - **%SW0** contiene el valor del periodo (0-150 ms). Es decir:
 - Se inicializa cuando se inicia a partir de un inicio en frío mediante el valor establecido en la configuración.
 - El usuario puede modificarlo.
-

Uso del tiempo de ejecución de la tarea master

Las siguientes palabras del sistema se utilizan para ofrecer información sobre el tiempo de ciclo de exploración del controlador:

- **%SW11** Se inicializa con el tiempo de vigilancia máximo de watchdog (10 a 500 ms).
- **%SW30** contiene el tiempo de ejecución para el último ciclo de exploración del controlador.
- **%SW31** contiene el tiempo de ejecución para el ciclo de exploración del controlador más largo.
- **%SW32** contiene el tiempo de ejecución para el ciclo de exploración del controlador más corto.

Nota: También puede accederse a esta información diferente desde el editor de configuración.

Modos de funcionamiento

Introducción

TwidoSoft se utiliza para tener en cuenta los tres grupos de modo de funcionamiento principales:

- Comprobación
- Ejecución o producción
- Detención

Nota: Estos modos de funcionamiento se definen en la guía "Design Guide for Operating and Stopping Modes" elaborada por la Applied Industrial Automation Development Agency.

Inicio mediante Grafcet

Estos diversos modos de funcionamiento se pueden obtener iniciando o utilizando los siguientes métodos Grafcet:

- Inicialización de Grafcet
- Preajuste de pasos
- Mantenimiento de una situación
- Congelación de diagramas

El procesamiento preliminar y el uso de bits del sistema garantiza una gestión eficaz de los modos de funcionamiento sin dificultar ni sobrecargar el programa de usuario.

**Bits del sistema
Grafcet**

El uso de los bits %S21, %S22 y %S23 se reserva sólo para el procesamiento preliminar. El sistema restablece automáticamente estos bits. Sólo se pueden escribir mediante la instrucción Establecer **S**.

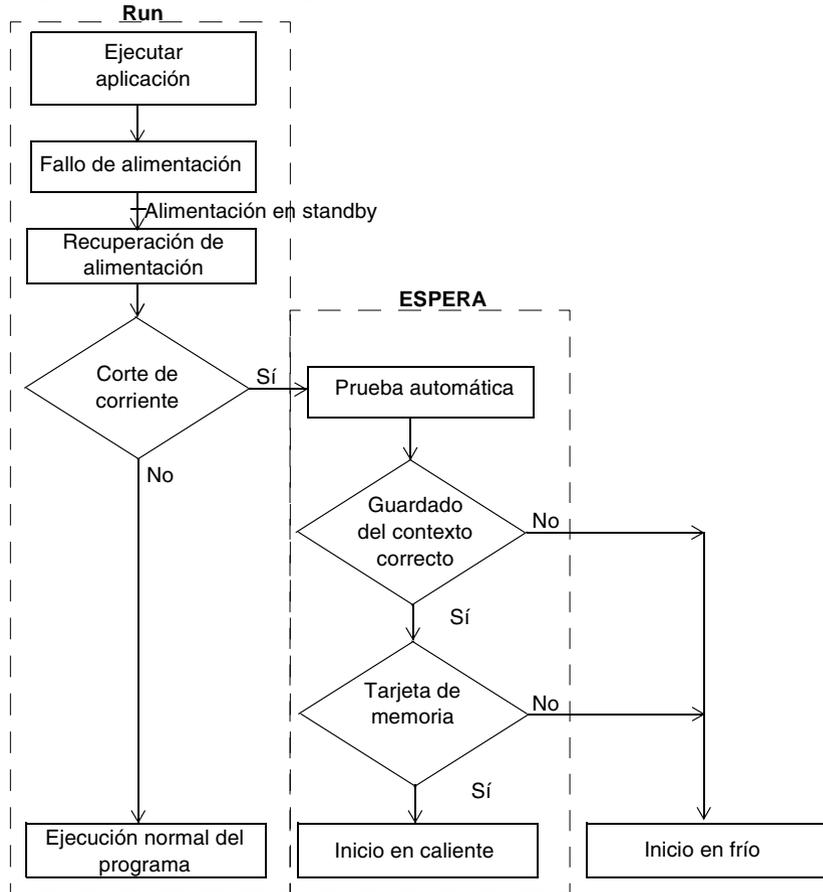
La siguiente tabla contiene bits del sistema relacionados con Grafcet:

Bit	Función	Descripción
%S21	Inicialización GRAFCET	Normalmente ajustado a 0. Se pone a 1 por: <ul style="list-style-type: none"> ● Un reinicio en frío, %S0=1. ● El usuario, sólo en la parte de pre-procesamiento del programa, mediante una instrucción Establecer S %S21 o una bobina de establecimiento -(S)- %S21. Consecuencias: <ul style="list-style-type: none"> ● Desactivación de todos los pasos activos. ● Activación de todos los pasos iniciales.
%S22	RESTABLECIMIENTO GRAFCET	Normalmente ajustado a 0. Sólo se puede poner a 1 por el programa durante el pre-procesamiento. Consecuencias: <ul style="list-style-type: none"> ● Desactivación de todos los pasos activos. ● Detención de la exploración del procesamiento secuencial.
%S23	Preajustar y congelar GRAFCET	Normalmente ajustado a 0. Sólo se puede poner a 1 por el programa durante el pre-procesamiento. <ul style="list-style-type: none"> ● Restablece Grafcet poniendo %S22 a 1. ● Precoloca los pasos que se deben activar mediante una serie de instrucciones S Xi. ● Habilita la precolocación poniendo %S23 a 1. Congelación de una situación: <ul style="list-style-type: none"> ● En situación inicial: manteniendo %S21 a 1 mediante el programa. ● En situación "vacía": manteniendo %S22 a 1 mediante el programa. ● En situación determinada: manteniendo %S23 a 1.

Comportamiento ante cortes de corriente y recuperación de alimentación

Ilustración

La ilustración que aparece a continuación muestra los distintos tipos de reinicio de alimentación detectados por el sistema. Si la duración del corte de corriente es inferior al tiempo de filtrado de suministro de alimentación (unos 10 ms para el suministro de corriente alterna o 1 ms para el suministro de corriente continua), el programa no lo advierte y sigue funcionando con normalidad.



Nota: El contexto se guarda en una memoria RAM mantenida con batería. Durante el arranque, el sistema comprueba el estado de la batería y el contexto guardado, y decide si puede producirse un arranque en caliente.

**Bit Ejecutar/
detener frente a
Ejecución
automática**

El bit de entrada Ejecutar/detener tiene prioridad sobre el Inicio automático en Ejecución, disponible en el cuadro de diálogo Modo de exploración (consulte el Manual de funcionamiento de TwidoSoft). Si se establece el bit Ejecutar/detener, el controlador se reiniciará en modo de ejecución cuando se restaure la alimentación. El modo del controlador se determina de la siguiente manera.

Bit de entrada Ejecutar/detener	Inicio automático en Ejecución	Estado resultante
Cero	Cero	Detenido
Cero	Uno	Detenido
Flanco ascendente	Desconocido	En ejecución
Uno	Desconocido	En ejecución
Sin configurar en el software	Cero	Detenido
Sin configurar en el software	Uno	En ejecución

Nota: En el caso de los controladores de tipo compacto, si, cuando se corta la alimentación, el controlador se encuentra en modo de ejecución y el indicador "Inicio automático en Ejecución " no se ha marcado en el cuadro de diálogo Modo de exploración, el controlador se reiniciará en modo de detención cuando se restaure la alimentación.

Nota: En el caso de los controladores de tipo modular, si la batería del controlador funciona con normalidad en el momento en que se corta la alimentación, el controlador arrancará en el modo en el que estaba. El indicador "Inicio automático en Ejecución", seleccionado en el cuadro de diálogo Modo de exploración, no tendrá ningún efecto sobre el modo cuando se restaure la alimentación.

Operación

La tabla siguiente describe las fases de procedimiento para cortes de corriente.

Fase	Descripción
1	En caso de corte de corriente, el sistema guarda el contexto de la aplicación y la hora del corte.
2	Ajusta todas las salidas a estado de retorno como una función de los parámetros de seguridad (%S9).
3	Cuando se recupera la alimentación, el contexto guardado se compara con el contexto en curso, lo que define el tipo de inicio que se va a realizar: <ul style="list-style-type: none">● Si el contexto de la aplicación ha cambiado (pérdida de contexto de sistema o aplicación nueva), el controlador inicializa la aplicación: arranque en frío.● Si el contexto de aplicación es el mismo, el controlador reinicia sin inicializar datos: reinicio en caliente.

Uso de un reinicio en caliente

Causa de un reinicio en caliente

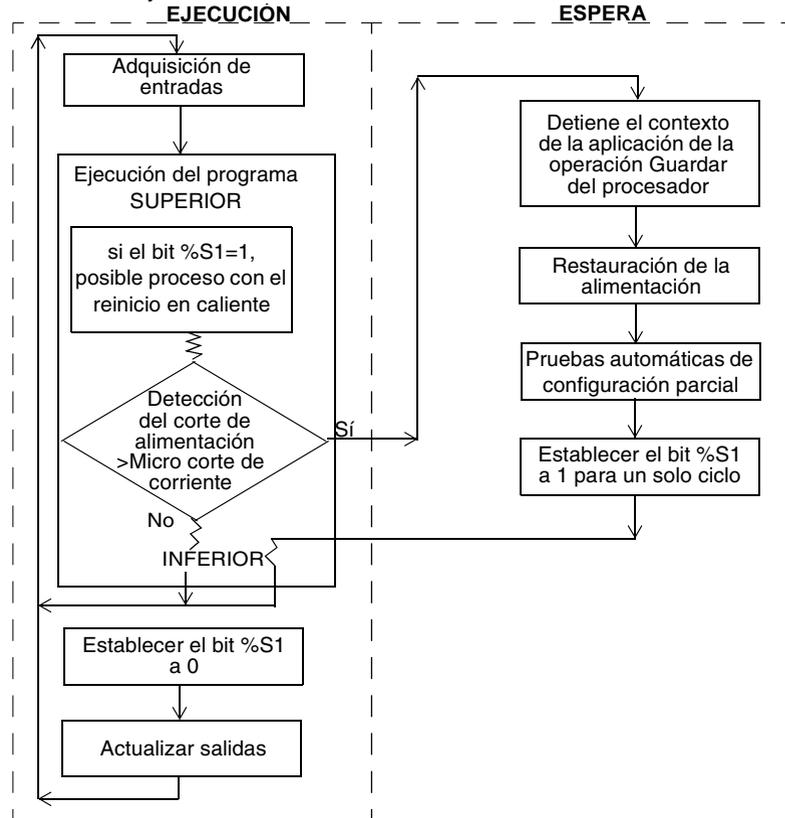
Un inicio en caliente puede producirse:

- Cuando se restaura la alimentación sin pérdida de contexto de las aplicaciones,
- Cuando el programa establece el bit **%S1** a estado 1,
- Desde la visualización del operador, cuando el controlador está en modo de detención.

Nota: Los controladores compactos siempre arrancan en frío. Los controladores modulares siempre se reinician en caliente.

Ilustración

El dibujo que aparece a continuación describe una operación de reinicio en caliente, en modo de ejecución.



Reinicio de la ejecución del programa

En la tabla siguiente se describen las fases de reinicio para ejecutar un programa después de un reinicio en caliente.

Fase	Descripción
1	La ejecución del programa se reanuda a partir del mismo elemento donde estaba antes del corte de alimentación, sin actualizar las salidas. Nota: Sólo se reinicia el mismo elemento del código de usuario. El código del sistema (por ejemplo, la actualización de salidas) no se reinicia.
2	Al final del ciclo de reinicio, el sistema: <ul style="list-style-type: none"> ● Elimina la reserva de la aplicación si se reservó (y hace que la aplicación se detenga en caso de depuración) ● Reinicializa los mensajes
3	El sistema realiza un ciclo de reinicio en el que: <ul style="list-style-type: none"> ● Ejecuta de nuevo la tarea con los bits %S1 (indicador de reinicio en frío) y %S13 (primer ciclo en ejecución) ajustados a 1. ● Restablece los bits %S1 y %S13 a 0 al final de este primer ciclo de tarea.

Procesamiento de un inicio en caliente

En caso de un inicio en caliente, si es necesario un proceso de aplicación determinado, el bit **%S1** debe comprobarse al comienzo del ciclo de tarea y debe llamarse al programa correspondiente.

Salidas después de un fallo de alimentación

Tan pronto como se detecta un fallo de alimentación, las salidas se ponen a un estado de recaída (predeterminado) de 0. Cuando se recupera la alimentación, las salidas permanecen con el último estado hasta que la tarea las actualice de nuevo.

Comportamiento ante un inicio en frío

Causas de un inicio en frío

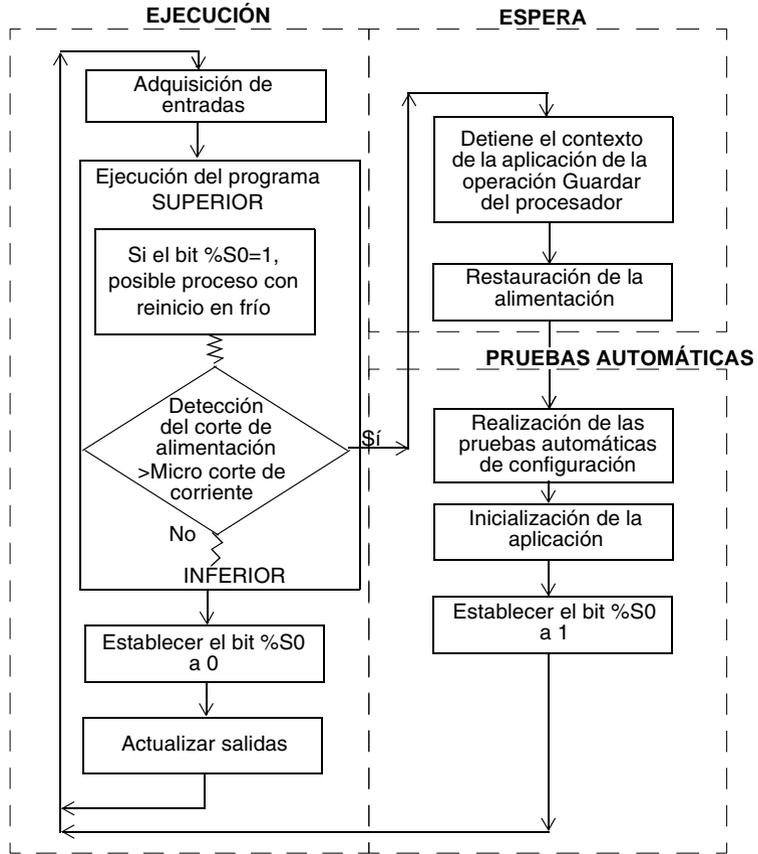
Un inicio en frío puede producirse:

- Al cargar una aplicación nueva en la RAM
- Cuando se restaura la alimentación con pérdida de contexto de las aplicaciones
- Cuando el programa ajusta el bit **%S0** a estado 1
- Desde el monitor de operación, cuando el controlador está en modo de detención

Nota: Los controladores compactos siempre arrancan en frío. Los controladores modulares siempre se reinician en caliente.

Ilustración

El dibujo de abajo describe una operación de reinicio en frío en modo de ejecución.



Operación

En la tabla siguiente se describen las fases de reinicio para ejecutar un programa después de un reinicio en frío.

Fase	Descripción
1	<p>Durante el arranque, el controlador está en modo de ejecución.</p> <p>Durante un reinicio en frío tras una detención debida a un ERROR, el sistema fuerza un reinicio en frío.</p> <p>La ejecución del programa se reinicia al comienzo del ciclo.</p>
2	<p>El sistema:</p> <ul style="list-style-type: none"> ● Restablece las palabras y los bits internos y las imágenes de E/S a 0 ● Inicializa las palabras y los bits de sistema. ● Inicializa los bloques de función de los datos de configuración.
3	<p>Durante este primer ciclo de reinicio, el sistema:</p> <ul style="list-style-type: none"> ● Ejecuta de nuevo la tarea con los bits %S0 (indicador de reinicio en frío) y %S13 (primer ciclo en ejecución) ajustados a 1. ● Restablece los bits %S0 y %S13 a 0 al final de este primer ciclo de tarea.

Procesamiento de un inicio en frío

En caso de inicio en frío, si se requiere un proceso de aplicación particular, se debe verificar el bit %S0 (que permanece a 1) durante el primer ciclo de la tarea.

Salidas después de un fallo de alimentación

Tan pronto como se detecta un fallo de alimentación, las salidas se ponen a un estado de recaída (predeterminado) de 0.

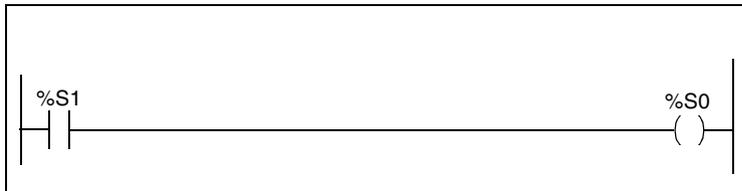
Cuando se recupera la alimentación, las salidas permanecen a 0 hasta que la tarea las actualice de nuevo.

Inicialización del controlador

Introducción Los controladores se pueden inicializar mediante TwidoSoft ajustando los bits de sistema **%S0** (reinicio en frío) y **%S1** (reinicio en caliente).

Inicialización de inicio en frío Para realizar una inicialización de inicio en fro, el bit de sistema **!%S0** se debe ajustar a 1.

Inicialización de inicio en caliente mediante %S0 y %S1 Para realizar una inicialización de inicio en caliente, los bits de sistema **%S1** y **%S0** se deben ajustar a 1. En el siguiente ejemplo se explica cómo programar una inicialización de reinicio en caliente mediante los bits de sistema.

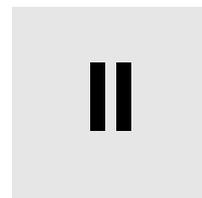


LD %S1 Si %S1 = 1 (reinicio en caliente), ajuste %S0 a 1 para inicializar el controlador.
ST %S0 El sistema restablece a 0 estos dos bits al final del ciclo siguiente.

Nota: No ajuste %S0 a 1 durante más de un ciclo del controlador.

Inicialización de inicio en caliente mediante el comando INIT También se puede solicitar una inicialización de inicio en caliente mediante un comando INIT. El comando INIT envía al controlador al estado IDLE, y la reinicialización de los datos de aplicación y el estado de la tarea al estado STOPPED.

Funciones especiales



Presentación

Vista general

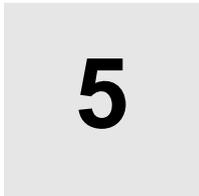
Esta parte describe comunicaciones, funciones analógicas incorporadas y gestión de módulos de E/S analógicas para controladores Twido.

Contenido

Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
5	Comunicaciones	71
6	Funciones analógicas incorporadas	125
7	Módulos analógicos de gestión	129
8	Funcionamiento del monitor de operación	137

Comunicaciones



Presentación

Vista general Este capítulo proporciona una vista general acerca de la configuración, la programación y la gestión disponibles de las comunicaciones con los controladores Twido.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Vista general de las comunicaciones	72
TwidoSoft a comunicaciones del controlador	74
Comunicaciones de conexión remota	76
Comunicaciones ASCII	89
Comunicaciones Modbus	101
Solicitudes Modbus estándar	119

Vista general de las comunicaciones

- Vista general** Twido ofrece uno o dos puertos de comunicaciones serie utilizados para la comunicación con controladores remotos, controladores peer o dispositivos externos generales. Cualquier puerto, si hay más de uno, se puede utilizar para cualquiera de los servicios, con excepción de la comunicación con Twido Soft, que sólo se puede establecer mediante el primer puerto. Los controladores Twido admiten tres protocolos básicos distintos: Conexión remota, ASCII o Modbus (master de Modbus o slave de Modbus).
-
- Conexión remota** El protocolo de conexión remota es un bus master/slave de alta velocidad diseñado para transferir una pequeña cantidad de datos entre el controlador master y hasta siete controladores remotos (slave). Se transfieren datos de E/S o de aplicación dependiendo de la configuración de los controladores remotos. Es posible realizar una mezcla de varios tipos de controladores remotos, donde unos pueden ser E/S remotas y otros, controladores peer.
-
- ASCII** El protocolo ASCII es un protocolo simple de modo de caracteres semi-dúplex que se utiliza para transmitir y/o recibir una cadena de caracteres hacia/desde un dispositivo simple (impresora o terminal). Este protocolo sólo se admite a través de la instrucción "EXCH".
-

Modbus

El protocolo Modbus es un protocolo master/slave que permite a un master, y sólo a uno, pedir respuestas de los slaves o realizar acciones dependiendo de las peticiones. El master puede dirigirse a slaves individuales o iniciar una difusión de mensajes para todos los slaves. Los slaves devuelven un mensaje (respuesta) a las peticiones que se les envían individualmente. No se devuelven respuestas a las peticiones de difusión desde el master.

Master de Modbus - El modo master de Modbus permite al controlador Twido iniciar una transmisión de peticiones Modbus, esperando una respuesta desde un slave de Modbus. El modo master de Modbus sólo se admite a través de la instrucción "EXCH". El modo master de Modbus admite los formatos ASCII Modbus y RTU Modbus.

Slave de Modbus - El modo slave de Modbus permite al controlador Twido responder a peticiones Modbus realizadas desde un master de Modbus y es, además, el modo de comunicaciones predeterminado si no hay ninguna comunicación configurada. El controlador Twido admite los datos Modbus estándar, las funciones de control y las ampliaciones de servicio para el acceso a objetos. El modo slave de Modbus admite los formatos ASCII Modbus y RTU Modbus.

<p>Nota: Pueden existir hasta 32 participantes en una red RS-485 (1 master y hasta 31 slaves) cuyas direcciones pueden estar en el intervalo 1-247.</p>
--

TwidoSoft a comunicaciones del controlador

Vista general

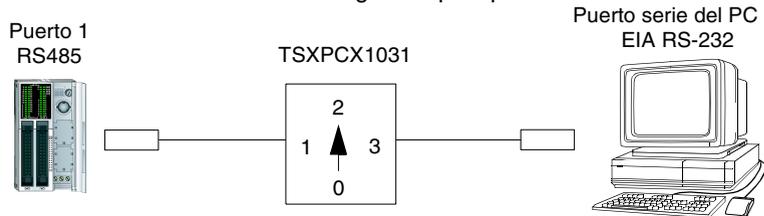
Cada controlador Twido tiene en su puerto 1 un puerto de terminales EIA RS-485 incorporado con fuente de alimentación interna. Debe utilizar el puerto 1 para comunicarse con el paquete de programación de TwidoSoft. No puede utilizarse ningún cartucho opcional o módulo de comunicaciones para esta conexión.

	AVISO
	<p>DAÑOS MATERIALES INESPERADOS</p> <p>Es posible que TwidoSoft no perciba la desconexión si se retira físicamente el cable de comunicaciones TSXPCX1031 de un primer controlador y se inserta rápidamente en un segundo controlador. Para evitar esto, utilice TwidoSoft para desconectar antes de retirar el cable.</p> <p>Si no se respetan estas precauciones pueden producirse daños corporales y/o materiales</p>

Conexión de los cables

El puerto EIA RS-232C de su PC está conectado al puerto 1 del controlador utilizando el cable de comunicaciones con varias funciones TSXPCX1031. Este cable convierte las señales comprendidas entre EIA RS-232 y EIA RS-485. Este cable está equipado con un conmutador giratorio de 4 posiciones para seleccionar diferentes modos de funcionamiento. El conmutador designa las cuatro posiciones como "0-3" y el ajuste apropiado de TwidoSoft para el controlador Twido es la ubicación 2.

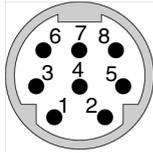
Esta conexión se ilustra en el diagrama que aparece a continuación.



Nota: La señal DPT no está puesta a tierra. La señal se detiene internamente indicando al firmware Executive que ésta es una conexión TwidoSoft.

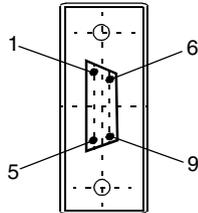
Clavijas de conectores macho y hembra

La siguiente figura muestra las clavijas de un conector miniDIN macho de 8 pins.



Clavijas	RS-485
1	A (+)
2	B (-)
3	NC
4	/DE
5	DPT
6	NC
7	0 V
8	5 V

La siguiente figura muestra las clavijas de un conector subD hembra de 9 pins.



Clavijas	RS-232
1	DCD
2	RX
3	TX
4	DTR
5	SG
6	NC
7	RTS
8	CTS
9	NC

Comunicaciones de conexión remota

Introducción

El protocolo de conexión remota es un bus master/slave de alta velocidad diseñado para transferir una pequeña cantidad de datos entre el controlador master y hasta siete controladores remotos (slave). Se transfieren datos de E/S o de aplicación dependiendo de la configuración de los controladores remotos. Es posible realizar una mezcla de varios tipos de controladores remotos, donde unos pueden ser E/S remotas y otros, controladores peer.

Nota: El controlador master contiene información acerca de la dirección de una E/S remota. Desconoce qué controlador específico está en la dirección. Por lo tanto, el master no puede validar que realmente existen todas las entradas y salidas remotas utilizadas en el aplicación de usuario. Compruebe que éstas existan.

Nota: El protocolo y el bus de E/S utilizados están patentados y no se permite utilizar dispositivos de otros fabricantes en la red.

	AVISO
	<p>FUNCIONAMIENTO INESPERADO DEL EQUIPO</p> <ul style="list-style-type: none"> ● Asegúrese de que sólo exista un controlador master en una conexión remota y que cada slave tenga una dirección exclusiva. Si no se respeta esta precaución, puede producir daños en los datos o resultados inesperados o ambiguos. ● Asegúrese de que todos los slaves tengan direcciones exclusivas. Dos slaves no deben tener la misma dirección. Si no se respeta esta precaución, puede producir daños en los datos o resultados inesperados o ambiguos. <p>Si no se respetan estas precauciones pueden producirse daños corporales y/o materiales</p>

Nota: La conexión remota requiere una conexión EIA RS-485 y sólo puede ejecutarse en un puerto de comunicaciones cada vez.

Configuración de hardware

La conexión remota debe utilizar un puerto mínimo EIA RS-485 de 3 conductores. Esto significa que puede configurarse para utilizar el primero o un segundo puerto opcional si está presente.

Nota: Sólo puede configurarse un puerto de comunicaciones como conexión remota.

La tabla que aparece a continuación enumera los dispositivos que se pueden utilizar.

Dispositivo	Puerto	Características
TWDCAA10/16/24DRF, TWDLMDA20/40DUK, TWDLMDA20/40DTK, TWDLMDA20DRT	1	Controlador base que admite EIA RS-485 de 3 conductores utilizando un conector miniDin.
TWDNOZ232D	2	Módulo de comunicaciones que admite EIA RS-232 de 3 conductores utilizando un conector miniDin. Nota: Este módulo sólo está disponible para los controladores modulares. Cuando está conectado el módulo, el controlador no puede tener un módulo de ampliación del monitor de operación.
TWDNOZ485D	2	Módulo de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector miniDin. Nota: Este módulo sólo está disponible para los controladores modulares. Cuando está conectado el módulo, el controlador no puede tener un módulo de ampliación del monitor de operación.
TWDNOZ485T	2	Módulo de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector de terminales. Nota: Este módulo sólo está disponible para los controladores modulares. Cuando el módulo está conectado, el controlador no puede tener un módulo de ampliación del monitor de operación.
TWDNAC232D	2	Adaptador de comunicaciones que admite EIA RS-232 de 3 conductores utilizando un conector miniDin. Nota: Este adaptador sólo está disponible para los controladores compactos de 16 y 24 E/S, y el módulo de ampliación del monitor de operación.
TWDNAC485D	2	Adaptador de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector miniDin. Nota: Este adaptador sólo está disponible para los controladores compactos de 16 y 24 E/S, y el módulo de ampliación del monitor de operación.

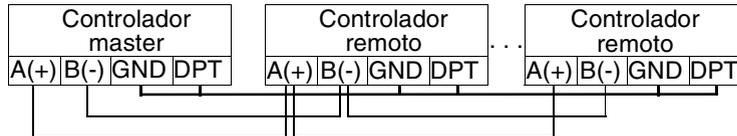
Dispositivo	Puerto	Características
TWDNAC485T	2	Adaptador de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector de terminales. Nota: Este adaptador sólo está disponible para los controladores compactos de 16 y 24 E/S, y el módulo de ampliación del monitor de operación.
TWDXCPODM	2	El módulo de ampliación del monitor de operación que admite un EIA RS-232 de 3 conductores utilizando un conector miniDIN, EIA RS-485 utilizando un conector miniDIN o EIA RS-485 utilizando un conector de terminales. Nota: Este módulo sólo está disponible para los controladores modulares. Cuando el módulo está conectado, el controlador no puede tener un módulo de ampliación de comunicaciones.

Nota: La configuración del puerto 2 (disponibilidad y tipo) sólo se comprueba durante el encendido o reinicio.

Conexión del cable a cada dispositivo

Nota: La señal DPT en el pin 5 debe estar conectada a tierra en el pin 7 para destacar el uso de comunicaciones de conexión remota. Cuando esta señal no está conectada a tierra, el controlador Twido (como master o slave) pasará de forma predeterminada a un modo de intentar establecer comunicaciones con TwidoSoft.

Las conexiones de cable a cada dispositivo se muestran a continuación.



Nota: La conexión DPT a GND sólo es necesaria si está conectado a un controlador base en el puerto 1.

Configuración de software

Sólo debe haber un controlador master definido en la conexión remota. Además, cada controlador remoto debe mantener una dirección slave exclusiva. Es posible que varios master o slave que utilicen direcciones idénticas dañen las transmisiones o creen ambigüedad.

	AVISO
	<p>Funcionamiento inesperado del equipo</p> <p>Asegúrese de que sólo exista un controlador master en una conexión remota y que cada slave tenga una dirección exclusiva. Si no se respeta esta precaución, puede producir daños en los datos o resultados inesperados o ambiguos.</p> <p>Si no se respetan estas precauciones pueden producirse daños corporales y/o materiales</p>

Configuración del controlador master

El controlador master se configura utilizando TwidoSoft para gestionar una red de conexión remota de hasta siete controladores remotos. El master admite una mezcla heterogénea de ambos controladores remotos (como E/S remotas o controladores Peer) en la conexión remota. La dirección del master se configura utilizando TwidoSoft para que quede en la dirección 0.

Configuración del controlador remoto

Es posible utilizar cada uno de los controladores remotos como E/S remotas o en un controlador peer. Se configuran utilizando TwidoSoft para que se asignen las direcciones 1 a 7 (observe que 0 está reservado para el master de conexión remota).

La tabla que aparece a continuación resume las diferencias y restricciones de cada uno de estos tipos de configuraciones del controlador remoto.

Tipo	Programa de aplicación	Acceso de datos
E/S remotas	No Ni siquiera una sentencia "END" simple	%I y %Q Sólo se puede acceder a E/S locales en el controlador remoto. (No su E/S de ampliación)
Controlador peer	Sí El modo de ejecución no está acoplado al de master	%INW y %QNW Puede transmitirse un máximo de 4 palabras de entrada y 4 palabras de salida procedentes y dirigidas a cada Peer

Sincronización del ciclo del controlador remoto

El ciclo de actualización de la conexión remota no está sincronizado con el ciclo del controlador master. Las comunicaciones con los controladores remotos están controladas por los interrupts y se produce como una tarea de fondo de forma paralela con la ejecución del ciclo del controlador master. Al final del ciclo de exploración, los valores más actualizados se leen en los datos de aplicación que se van a utilizar para la siguiente resolución. Este procesamiento es el mismo para E/S remotas y controladores peer.

Cualquier controlador puede comprobar actividad de conexión general utilizando el bit del sistema %S111. Sin embargo, para lograr la sincronización, un master o peer deberán utilizar un bit del sistema %S110. Éste se establece a 1 cuando se ha producido un ciclo de actualización completo. El programa de aplicación es responsable de restablecerlo a 0.

El master puede habilitar o inhabilitar la conexión remota utilizando el bit del sistema %S112. Los controladores pueden comprobar la correcta configuración y estado de la conexión remota utilizando %S113. La señal DPT en el puerto 1 (utilizada para determinar si está conectado TwidoSoft) se detecta y notifica en %S100.

Todas estas se resumen en la siguiente tabla.

Bit de sistema	Estado	Indicación
%S100	0	master/slave: DPT no activa (cable TwidoSoft NO conectado)
	1	master/slave: DPT activa (cable TwidoSoft conectado)
%S110	0	master/slave: restablecido por aplicación
	1	master: todos los intercambios de conexión remota completados (sólo E/S remotas) slave: intercambio con master completado
%S111	0	master: intercambio único de conexión remota completado slave: intercambio único de conexión remota detectado
	1	master: intercambio único de conexión remota activo slave: intercambio único de conexión remota detectado
%S112	0	master: conexión remota inhabilitada
	1	master: conexión remota habilitada
%S113	0	master/slave: configuración/funcionamiento de conexión remota correctos
	1	master: configuración/funcionamiento de conexión remota erróneos slave: funcionamiento de conexión remota erróneo

Reinicio del controlador master

Si se reinicia un controlador master, se producirá uno de los siguientes eventos:

- Un inicio en frío (%S0 = 1) fuerza una reinicialización de las comunicaciones.
- Un inicio en caliente (%S1 = 1) fuerza una reinicialización de las comunicaciones.
- En modo de detención, el master continúa comunicándose con los slave, con el bit ejecutar/detener establecido para indicar detención.

Reinicio del controlador slave

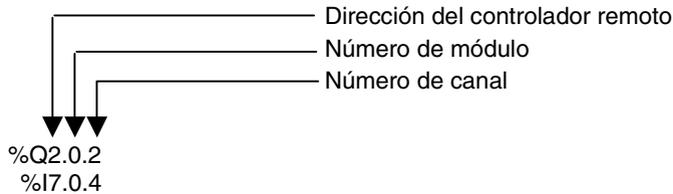
Si se reinicia un controlador slave, se producirá uno de los siguientes eventos:

- Un inicio en frío (%S0 = 1) fuerza una reinicialización de las comunicaciones.
- Un inicio en caliente (%S1 = 1) fuerza una reinicialización de las comunicaciones.
- En modo de detención, el slave continúa comunicándose con el master. Si el master indica que es necesaria una detención:
 - La E/S remota afecta a una detención
 - Un controlador peer continúa en su estado actual.

Acceso de datos de E/S remotas

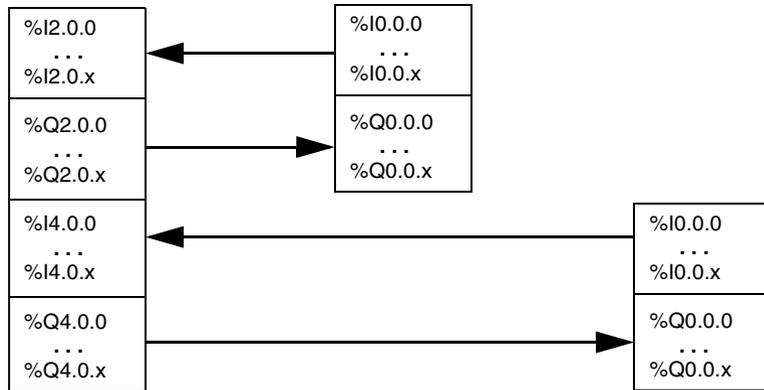
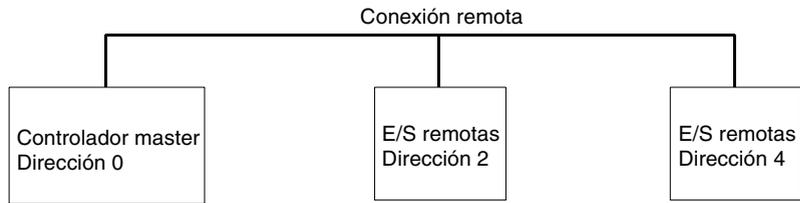
El controlador remoto configurado para ser una E/S remota no tiene ningún programa de aplicación o ejecuta el suyo propio. Las entradas digitales base del controlador remoto son una sola ampliación del controlador master. La aplicación sólo debe utilizar el mecanismo de direccionamiento de tres dígitos completo suministrado.

Nota: El número de módulo es siempre cero para las E/S remotas.



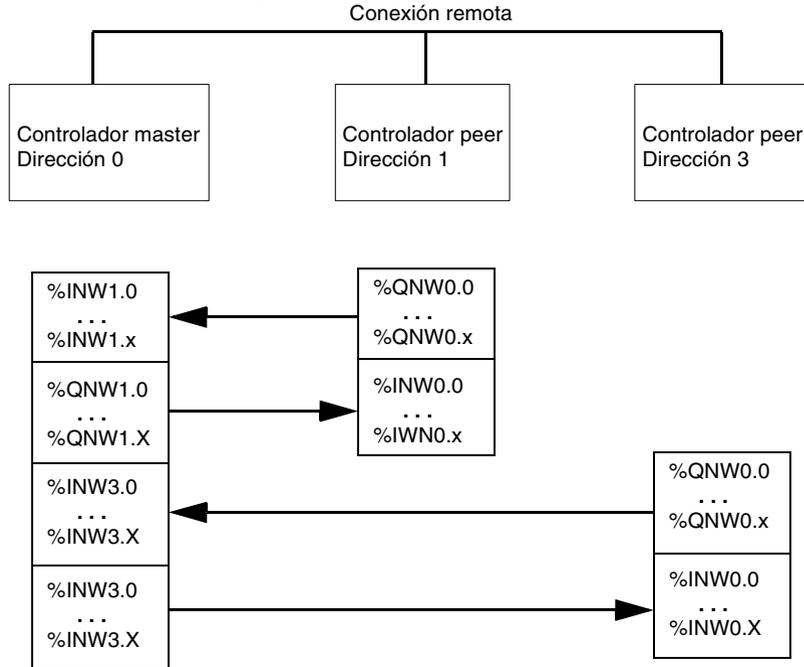
Para comunicarse con la E/S remota, el controlador master utiliza la anotación de entrada y salida estándar de %I y %Q. Para acceder al tercer bit de salida de la E/S remota configurada en la dirección 2, el master se establecería en %Q2.0.2. De forma similar, para leer el quinto bit de entrada de la E/S remota configurada en la ubicación 7, el master cargaría %I7.0.4.

Nota: El master tiene restricciones para acceder únicamente a la E/S digital que forma parte de la E/S local remota. No puede transferirse ninguna E/S analógica ni de ampliación, a menos que utilice comunicaciones peer.



Acceso de datos del controlador peer

Para comunicarse con los controladores peer, el master utiliza palabras de red %INW y %QNW para intercambiar datos. Debe accederse a cada peer de la red mediante su dirección remota "j" utilizando las palabras %INWj.k y %QNWj.k. Cada controlador peer de la red utiliza %INW0.0 a %INW0.3 y %QNW0.0 a %QNW0.3 para acceder a los datos del master. Las palabras de red se actualizan de forma automática cuando el controlador está en modo de ejecución o detenido. El ejemplo que aparece a continuación ilustra el intercambio de un master con dos controladores peer configurados.



No existe mensajería peer-to-peer dentro de la conexión remota. Los programas de aplicación pueden utilizarse junto con las palabras de red para transferir información entre los controladores remotos, en ejecución utilizando el master como puerto.

Información de estado

Además de los bits del sistema explicados anteriormente, el master conserva su estado con el aspecto y la configuración de los controladores remotos. Esto se realiza en las palabras de sistema %SW111 y %SW113. Tanto el controlador remoto como el master pueden adquirir el valor del último error producido durante la comunicación en la conexión remota de la palabra de sistema %SW112. Cada uno de ellos se describe en la siguiente tabla.

Palabras de sistema	Uso	
%SW111	Estado de conexión remota: dos bits para cada controlador remoto (sólo master)	
	x0-5 0 - Controlador remoto 1-6 ausente 1 - Controlador remoto 1-6 presente	
	x6 0 - Controlador remoto 7 ausente 1 - Controlador remoto 7 presente	
	x8-13 0 - E/S remota detectada en el controlador remoto 1-6 1 - Controlador peer detectado en el controlador remoto 1-6	
	x14 0 - E/S remota detectada en el controlador remoto 7 1 - Controlador peer detectado en el controlador remoto 7	
	%SW112	Código de error de configuración/funcionamiento de conexión remota
		0 - operaciones correctas
1 - timeout detectado (slave)		
2 - error de suma de control detectado (slave) 3 - discrepancia de configuración (slave)		
%SW113	Configuración de conexión remota: dos bits para cada controlador remoto (sólo master)	
	x0-5 0 - Controlador remoto 1-6 no configurado 1 - Controlador remoto 1-6 configurado	
	x6 0 - Controlador remoto 7 no configurado 1 - Controlador remoto 7 configurado	
	x8-13 0 - E/S remota configurada como controlador remoto 1-6 1 - Controlador peer configurado como controlador remoto 1-6	
	x14 0 - E/S remota configurada como controlador remoto 7 1 - Controlador peer configurado como controlador remoto 7	

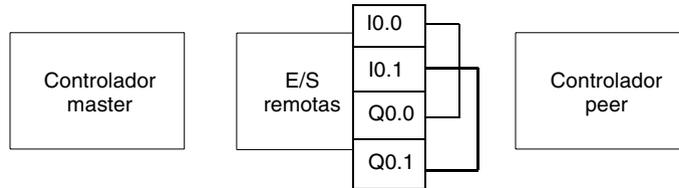
Ejemplo de conexión remota

Para configurar una conexión remota debe seguir estos pasos.

1. Configurar el hardware.
2. Conectar el cableado del controlador.
3. Conectar el cable de comunicaciones entre el PC y los controladores.
4. Configurar el software.
5. Escribir una aplicación.

Los diagramas que aparecen a continuación ilustran el uso de la conexión remota con la E/S remota y un controlador peer.

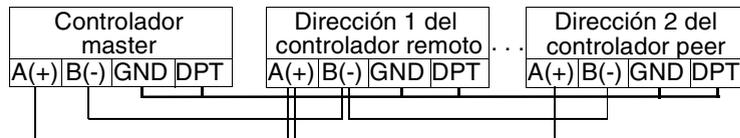
Paso 1: Configurar el hardware:



La configuración del hardware se compone de tres controladores base de cualquier tipo. El puerto 1 se utiliza en modo dual. Un modo se utiliza para configurar y transferir el programa de aplicación con TwidoSoft. El segundo, para la red de conexión remota. Si está disponible, puede utilizarse el puerto 2 opcional de los controladores, pero un controlador sólo admite una conexión remota.

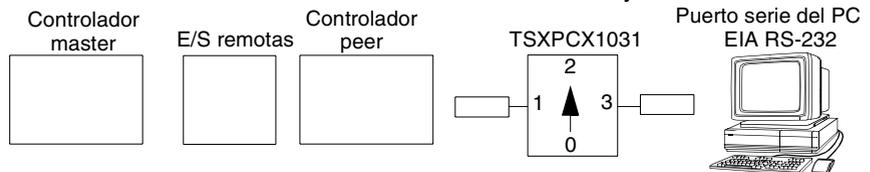
Nota: En este ejemplo, las dos primeras entradas de la E/S remota están cableadas a sus salidas.

Paso 2: Conectar el cableado del controlador:



Conectar los cables de señal D(+) y D(-) juntos. En cada controlador, la señal DPT está puesta a tierra. Aunque no es necesario, es recomendable poner a tierra la señal para utilizarla con una conexión remota en el puerto 2 (el cartucho opcional o el módulo de comunicación).

Paso 3: Conectar el cable de comunicaciones entre el PC y los controladores:



El cable de programación de varias funciones TSXPCX1031 se utiliza para comunicarse con cada uno de los tres controladores base. Asegúrese de que el cable esté en la posición 2 del controlador. Para programar cada uno de los controladores, será necesario establecer una comunicación punto a punto con cada controlador. Para establecer esta comunicación: conecte el puerto 1 del primer controlador, transfiera los datos de aplicación y configuración, y establezca el controlador al estado de ejecución. Repita este procedimiento para cada controlador.

Nota: El cable necesita desplazarse después de la configuración de cada controlador y la transferencia de la aplicación.

Una vez programados los tres controladores, conecte los controladores de la red de conexión remota como se describe en el paso 2.

Paso 4: Configurar el software:

Configuración de com. del controlador
 Tipo: Conexión remota
 Dirección: 0 (Master)

Configuración de com. del controlador
 Tipo: Conexión remota
 Dirección: 1

Configuración de com. del controlador
 Tipo: Conexión remota
 Dirección: 2

Agregar controladores remotos
 Utilización del controlador: E/S remotas
 Dirección remota: 1

Utilización del controlador: Peer
 Dirección remota: 2

Cada uno de los tres controladores utiliza TwidoSoft para crear una configuración y, si fuera pertinente, el programa de aplicación. Para el controlador master, edite la configuración de comunicaciones del controlador para establecer el protocolo como "conexión remota" y la dirección como "0 (Master)".

Nota: Sólo puede haber un controlador configurado como master en una conexión remota.

En TwidoSoft, agregue una "E/S remota" en la dirección "1" y un "controlador peer" en la dirección "2".

Para el controlador configurado como una E/S remota, compruebe que la configuración de comunicaciones del controlador está establecida como "conexión remota" y la dirección como "1". Para el controlador configurado como peer, compruebe que la configuración de comunicaciones del controlador está establecida como "conexión remota" y la dirección como "2".

Paso 5: Escribir una aplicación:

```
LD 1
[%MW0 := %MW0 +1]
[%QNW2.0 := %MW0]
[%MW1 := %INW2.0]

LD %I0.0
ST %Q1.0.0
LD %I1.0.0
ST %Q0.0

LD %I0.1
ST %Q1.0.1
LD %I1.0.1
ST %Q0.1
```

```
LD 1
[%QNW0.0 := %INW0.0]
```

En este ejemplo, la aplicación master incrementa una palabra de memoria interna y la comunica al controlador peer utilizando una sola palabra de red. El controlador peer toma la palabra recibida del master y la devuelve. En el master, una palabra de memoria diferente recibe y almacena esta transmisión.

Para establecer comunicación con el controlador de E/S remota, el master envía sus entradas locales a las salidas de E/S remotas. Con el cableado de E/S externa de la E/S remota, las señales se devuelven y recuperan mediante el master.

Nota: Esta comunicación tiene lugar bajo la aplicación master. No existe ninguna aplicación en el controlador de E/S.

Comunicaciones ASCII

Introducción

El protocolo ASCII proporciona a los controladores Twido un protocolo simple de modo de caracteres semi-dúplex para transmitir o recibir datos hacia/desde un dispositivo simple. Este protocolo se admite si se utiliza la instrucción EXCHx y se controla mediante el bloque de función %MSGx.

Hay tres tipos de comunicaciones posibles utilizando el protocolo ASCII:

- Sólo transmisión
- Transmisión/Recepción
- Sólo recepción

El tamaño máximo de las tramas transmitidas o recibidas mediante la instrucción EXCHx es de 128 bytes.

Configuración de hardware

Una conexión ASCII se puede establecer en el puerto EIA RS-232 o EIA RS-485 y se puede ejecutar hasta en dos puertos de comunicaciones al mismo tiempo. La tabla que aparece a continuación enumera los dispositivos que se pueden utilizar.

Dispositivo	Puerto	Características
TWDCAA10/16/24DRF, TWDLMDA20/40DUK, TWDLMDA20/40DTK, TWDLMDA20DRT	1	Controlador base que admite EIA RS-485 de 3 conductores utilizando un conector miniDin.
TWDNOZ232D	2	Módulo de comunicaciones que admite EIA RS-232 de 3 conductores utilizando un conector miniDin. Nota: Este módulo sólo está disponible para los controladores modulares. Cuando el módulo está conectado, el controlador no puede tener un módulo de ampliación del monitor de operación.
TWDNOZ485D	2	Módulo de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector miniDin. Nota: Este módulo sólo está disponible para los controladores modulares. Cuando el módulo está conectado, el controlador no puede tener un módulo de ampliación del monitor de operación.
TWDNOZ485T	2	Módulo de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector de terminales. Nota: Este módulo sólo está disponible para los controladores modulares. Cuando el módulo está conectado, el controlador no puede tener un módulo de ampliación del monitor de operación.
TWDNAC232D	2	Adaptador de comunicaciones que admite EIA RS-232 de 3 conductores utilizando un conector miniDin. Nota: Este adaptador sólo está disponible para los controladores compactos de 16 y 24 E/S, y el módulo de ampliación del monitor de operación.
TWDNAC485D	2	Adaptador de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector miniDin. Nota: Este adaptador sólo está disponible para los controladores compactos de 16 y 24 E/S, y el módulo de ampliación del monitor de operación.
TWDNAC485T	2	Adaptador de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector de terminales. Nota: Este adaptador sólo está disponible para los controladores compactos de 16 y 24 E/S, y el módulo de ampliación del monitor de operación.

Dispositivo	Puerto	Características
TWDXCPODM	2	<p>El módulo de ampliación del monitor de operación que admite un EIA RS-232 de 3 conductores utilizando un conector miniDIN, EIA RS-485 utilizando un conector miniDIN o EIA RS-485 utilizando un conector de terminales.</p> <p>Nota: Este módulo sólo está disponible para los controladores modulares. Cuando el módulo está conectado, el controlador no puede tener un módulo de ampliación de comunicaciones.</p>

Nota: La configuración del puerto 2 (disponibilidad y tipo) sólo se comprueba durante el encendido o reinicio mediante el firmware Executive.

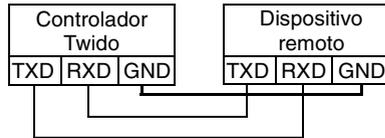
Cableado nominal

A continuación, se ilustran las conexiones de cableado nominal para los tipos EIA RS-232 y EIA RS-485.

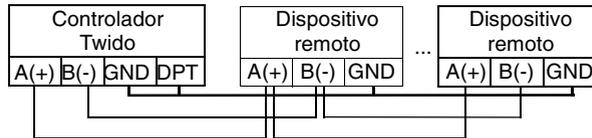
Nota: Si se utiliza el puerto 1 en el controlador Twido, la señal DPT deberá estar conectada a tierra. Esto indica al controlador Twido que la comunicación a través del puerto 1 es ASCII y no el protocolo utilizado para comunicarse con el software TwidoSoft. El dispositivo remoto específico puede requerir la utilización de señales adicionales (DTR, DSR, etc.).

Las conexiones de cable a cada dispositivo se muestran a continuación.

Cable EIA RS-232



Cable EIA RS-485



Nota: La conexión DPT a GND sólo es necesaria si está conectado a un controlador base en el puerto 1.

Configuración de software

Para configurar el controlador con el fin de utilizar una conexión serie para enviar y recibir caracteres utilizando el protocolo ASCII, deberá seguir estos pasos.

Etapa	Descripción
1	Configurar el puerto serie para ASCII utilizando TwidoSoft.
2	Crear en su aplicación un búfer de transmisión/recepción para ASCII con el fin de utilizar la instrucción EXCHx.

Configuración del puerto

Un controlador Twido puede utilizar su puerto 1 primario o un puerto 2 configurado de forma opcional para utilizar el protocolo ASCII. Para configurar un puerto serie para ASCII, siga estos pasos.

Paso	Acción
1	Definir físicamente cualquier módulo o cartucho opcionales configurados en la base.
2	Hacer clic con el botón derecho en el puerto y hacer clic en Configuración de com. del controlador... y cambiar el tipo de puerto serie a "ASCII".
3	Establecer los parámetros de comunicaciones asociados.

Configuración del búfer de transmisión/recepción para ASCII

El tamaño máximo de las tramas transmitidas o recibidas es 128 bytes, y la tabla de palabras asociada a la instrucción EXCHx está compuesta por tablas de transmisión y de recepción.

	Byte de mayor valor	Byte de menor valor
Palabras de control	Comando	Longitud (Tx/Rx)
	Reservado (0)	Reservado (0)
Tabla de transmisión	Byte 1 transmitido	Byte 2 transmitido

	...	Byte n transmitido
	Byte n+1 transmitido	
Tabla de recepción	Byte 1 recibido	Byte 2 recibido

	...	Byte p recibido
	Byte p+1 recibido	

Parámetros de control

El byte de **longitud** contiene la longitud que se va a transmitir, sobrescrita por el número de caracteres recibidos al final de la recepción, en caso de que ésta se solicite.

El byte de **comando** debe contener uno de los valores siguientes:

- 0: Sólo transmisión
- 1: Transmisión/Recepción
- 2: Sólo recepción

Tablas de transmisión/recepción

Cuando está activo el modo Sólo transmisión, las tablas de transmisión y control se completan antes de ejecutar la instrucción EXCHx, y pueden ser del tipo %KW o %MW. No se requiere ningún espacio para la recepción de caracteres en el modo Sólo transmisión. Una vez transmitidos todos los bytes, el estado de %MSGx.D se pone a 1 y se puede ejecutar una instrucción EXCHx nueva.

Cuando está activo el modo Transmisión/Recepción, las tablas de transmisión y control se completan antes de ejecutar la instrucción EXCHx, y deben ser del tipo %MW. Se requiere espacio para hasta 128 bytes de recepción al final de la tabla de transmisión. Una vez transmitidos todos los bytes, el controlador Twido cambia a modo de recepción y espera a recibir los bytes.

Cuando está activo el modo Sólo recepción, la tabla de control se completa antes de ejecutar la instrucción EXCHx y debe ser del tipo %MW. Se requiere espacio para hasta 128 bytes de recepción al final de la tabla de control. El controlador Twido cambia inmediatamente a modo de recepción y espera a recibir los bytes.

La recepción concluye cuando se recibe el byte de final de trama o cuando la tabla de recepción está llena. Si se configura un timeout que no sea cero, la recepción concluye cuando el timeout se completa. Si se selecciona un valor de timeout cero, no hay timeout de recepción. Por lo tanto, para detener la recepción hay que activar la entrada %MSGx.R.

No hay direccionamiento inherente asociado con el protocolo ASCII a menos que el dispositivo simple lo tenga incorporado en el protocolo. No obstante, el controlador Twido no lo admite.

Intercambio de mensajes

El controlador Twido puede configurarse para enviar o recibir mensajes en modo carácter. El lenguaje ofrece dos servicios para esto:

- **Instrucción EXCHx:** para transmitir/recibir mensajes
- **Bloque de función %MSGx:** para controlar los intercambios de mensajes.

Cuando se procesa una instrucción EXCHx, el controlador Twido utiliza el protocolo configurado para dicho puerto.

Nota: Cada puerto de comunicaciones puede configurarse para protocolos diferentes o para el mismo. El modo de acceder a la instrucción EXCHx o al bloque de función %MSGx para cada puerto de comunicaciones es agregando el número de puerto (1 ó 2).

**Instrucción
EXCHx**

La instrucción EXCHx permite al controlador Twido enviar o recibir información dirigida a, o procedente de, dispositivos ASCII. El usuario define una tabla de palabras (%MWi:L o %KW:L) que contiene información de control y los datos que se van a enviar o recibir (hasta 64 palabras en la transmisión o recepción). El formato de la tabla se describe en secciones anteriores.

Un intercambio de mensajes se realiza utilizando la instrucción EXCHx.

Sintaxis: [EXCHx %MWi:L] o [EXCHx %KW:L]

donde: x = número de puerto (1 ó 2).

L = número de palabras en la tabla de palabras.

El controlador Twido debe finalizar el intercambio desde la primera instrucción EXCHx antes de que se ejecute una segunda. El bloque de función %MSGx debe utilizarse cuando se envíen varios mensajes.

El procesamiento de la instrucción de lista EXCHx se produce inmediatamente, con cualquier transmisión iniciada bajo el control de interrupción (la recepción de datos también se encuentra bajo el control de interrupción), que se considera procesamiento de fondo.

Bloque de función %MSGx

El uso del bloque de función %MSGx es opcional; puede utilizarse para gestionar los intercambios de datos. El bloque de función %MSGx tiene tres propósitos.

- **Comprobación de errores de comunicación**

La comprobación de errores verifica que la longitud del bloque (tabla de palabras) programada con la instrucción EXCHx es lo suficientemente grande para contener la longitud del mensaje que se va a enviar. Esto se compara con la longitud programada en el byte de menor valor de la primera palabra de la tabla de palabras.

- **Coordinación de varios mensajes**

Para asegurar la coordinación cuando se envíen varios mensajes, el bloque de función %MSGx proporciona la información requerida para determinar cuándo está completo un mensaje anterior.

- **Transmisión de mensajes prioritarios**

El bloque de función %MSGx permite la detención de la transmisión del mensaje actual para permitir el envío inmediato de un mensaje urgente.

El bloque de función %MSGx tiene una entrada y dos salidas asociadas.

Entrada/salida	Definición	Descripción
R	Restablecer entrada	Poner a 1: reinicializa la comunicación o restablece el bloque (%MSGx.E = 0 y %MSGx.D = 1).
%MSGx.D	Comunicación completa	0: solicitud en curso. 1: comunicación realizada si se produce el final de la transmisión, se recibe el carácter final, se produce un error o se restablece el bloque.
%MSGx.E	Error	0: longitud del mensaje y enlace correctos. 1: si hay un comando inválido, la tabla se configura de forma incorrecta, se recibe un carácter incorrecto (velocidad, paridad, etc.) o la tabla de recepción está llena.

Limitaciones

Es importante observar las siguientes limitaciones:

- La disponibilidad y el tipo de puerto 2 sólo se comprueban durante el encendido o reinicio.
- El procesamiento de un mensaje en el puerto 1 se cancela cuando se conecta TwidoSoft.
- EXCHx o %MSG no se puede procesar en un puerto configurado como conexión remota.
- EXCHx interrumpe el procesamiento de slave de Modbus activo (excepto en el caso del procesamiento de TwidoSoft).
- El procesamiento de las instrucciones EXCHx no se vuelve a intentar en caso de error.
- R %MSGx se puede utilizar para interrumpir el procesamiento de recepción de instrucción EXCHx.
- Las instrucciones EXCHx se pueden configurar con un timeout para interrumpir la recepción.
- Se controlan mensajes múltiples a través de %MSGx.D.

Condiciones de modo de funcionamiento y error

Si se produce un error durante el uso de una instrucción EXCHx, los bits %MSGx.D y %MSGx.E se ponen a 1 y la palabra de sistema %SW63 contiene el código de error para el puerto 1, y %SW64 contiene el código de error para el puerto 2.

Palabras de sistema	Uso
%SW63	Código de error EXCH1: 0 - operación correcta 1 - búfer de transmisión demasiado largo (> 128) 2 - búfer de transmisión demasiado pequeño 3 - tabla de palabras demasiado pequeña 4 - tabla de recepción desbordada 5 - timeout transcurrido 6 - error de transmisión (error recibido en la respuesta) 7 - comando incorrecto en la tabla 8 - puerto seleccionado no configurado/disponible 9 - error de recepción 10 - no se puede utilizar %KW si se está recibiendo 11 - offset de transmisión mayor que tabla de transmisión 12 - offset de recepción mayor que tabla de recepción 13 - procesamiento EXCH detenido por el controlador
%SW64	Código de error EXCH2: consulte %SW63.

Reinicio del controlador master/slave

Si se reinicia un controlador master/slave, se producirá uno de los siguientes eventos:

- Un inicio en frío (%S0 = 1) fuerza una reinicialización de las comunicaciones.
- Un inicio en caliente (%S1 = 1) fuerza una reinicialización de las comunicaciones.
- En modo de detención, el controlador detiene todas las comunicaciones ASCII.

Ejemplo de conexión ASCII

Para configurar una conexión ASCII debe seguir estos pasos.

1. Configurar el hardware.
2. Conectar el cable de comunicaciones ASCII.
3. Configurar el puerto.
4. Escribir una aplicación.
5. Inicializar el editor de tablas de animación.

El diagrama que aparece a continuación ilustra el uso de las comunicaciones ASCII con un emulador terminal de un PC.

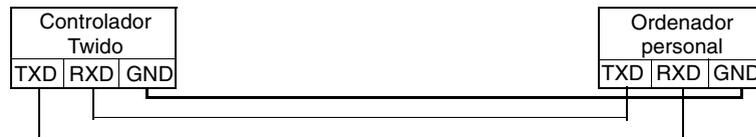
Paso 1: Configurar el hardware



La configuración de hardware está formada por dos conexiones serie del PC al controlador Twido con un puerto 2 opcional EIA RS-232. En un controlador modular, el puerto 2 opcional es un TWDNOZ232D. En el controlador compacto, el puerto 2 opcional es un TWDNAC232D.

Para configurar el controlador, conecte el cable TSXPCX1031 (no se muestra en la figura) al puerto 1 del controlador Twido. A continuación, conecte el cable al puerto COM 1 del PC. Asegúrese de que el conmutador se encuentre en la posición 2. Por último, conecte el puerto COM 2 del PC al puerto 2 opcional EIA RS-232 del controlador Twido. Las conexiones de pin y el cableado se describen en el paso siguiente.

Paso 2: Conectar el cable de comunicaciones ASCII (EIA RS-232)



El requisito mínimo para el cableado del cable de comunicaciones ASCII es una conexión básica de 3 conductores. Cruce las señales de transmisión y recepción.

Nota: En el extremo del cable que se va a conectar al PC pueden ser necesarias conexiones adicionales (como Terminal de datos preparada y Paquete de datos preparado) para satisfacer los requisitos del establecimiento de enlace. No es necesaria ninguna conexión adicional para satisfacer los requisitos del controlador Twido.

Paso 3: Configurar el puerto

Hardware -> Agregar opción TWDNOZ232D
Hardware => Configuración de com. del controlador Puerto: 2 Tipo: ASCII Velocidad de transmisión: 19200 Datos: 8 bits Paridad: Ninguno Detener: 1 bit Fin de trama: 65 Timeout de respuesta: 100 x 100 ms

Emulador terminal en un PC	
Puerto:	COM2
Velocidad de transmisión:	19200
Datos:	8 bits
Paridad:	Ninguna
Detener:	1 bit
Control de flujo:	Ninguno

Utilice una aplicación de emulador de terminal simple en el PC para configurar una configuración de puerto básica y para asegurarse de que no exista control de flujo. Utilice TwidoSoft para configurar el puerto del controlador. Primero se configura la opción de hardware. En este ejemplo, se añade el TWDNOZ232D al controlador base modular.

En segundo lugar, se inicializa la instalación de comunicaciones del controlador con los mismos ajustes de parámetros que en el emulador terminal del PC. En este ejemplo, se elige la letra mayúscula "A" para el carácter "fin de trama" para concluir el búfer de recepción de entradas. Se selecciona un timeout de 10 segundos para el parámetro "Timeout de respuesta". Sólo se ejecutará uno de estos dos parámetros dependiendo de cuál se produzca primero.

Paso 4: Escribir una aplicación

```
LD 1
[%MW10 := 16#0104]
[%MW11 := 16#0000]
[%MW12 := 16#4F4B]
[%MW13 := 16#0A0D]
LD 1
AND %MSG2.D
[EXCH2 %MW10:8]
LD %MSG2.E
ST %Q0.0
END
```

Utilice TwidoSoft para crear un programa de aplicación con tres partes principales. Primero, inicialice el búfer de transferencia y control para utilizarlo con la instrucción EXCH. En este ejemplo, se ajusta un comando tanto para enviar como para recibir datos. La cantidad de datos que se va a enviar se pone a 4 bytes y se inicializa con los caracteres: "O", "K", CR y LF.

A continuación, compruebe el bit de finalización asociado a %MSG2 y ejecute la instrucción EXCH2 sólo si el puerto está listo. Para la instrucción EXCH2 se especifica un valor de 8 caracteres. Hay dos palabras de control (%MW10 y %MW11), dos palabras utilizadas para transmitir información (%MW12 y %MW13) y cuatro palabras para recibir datos (de %MW14 a %MW17).

Por último, se detecta el estado de error de %MSG2 y se guarda en el primer bit de salida de las E/S locales del controlador base. También se podría añadir una comprobación de errores adicional mediante %SW64 para aumentar la seguridad.

Paso 5: Inicializar el editor de tablas de animación

	Dirección	Actual	Guardado	Formato
1	%MW10	0104	0000	Hexadecimal
2	%MW11	0000	0000	Hexadecimal
3	%MW12	4F4B	0000	Hexadecimal
4	%MW13	0A0D	0000	Hexadecimal
5	%MW14	TW	0000	ASCII
6	%MW15	ID	0000	ASCII
7	%MW16	O	0000	ASCII
8	%MW17	A	0000	ASCII

El último paso es descargar el controlador de aplicación y ejecutarlo. Inicialice un editor de tablas de animación para animar y visualizar las palabras %MW10 a %MW17. En el emulador terminal se visualizan los caracteres "O"- "K"-CR-LF. Puede haber muchos de estos caracteres dependiendo del número de veces que se produce un timeout del bloque EXCH y se ejecuta uno nuevo. En el emulador terminal, escriba "T"- "W"- "I"- "D"- "O"- " "- "A". Estos datos se intercambian con el controlador Twido y se muestran en el editor de tablas de animación.

Comunicaciones Modbus

Introducción

El protocolo Modbus es un protocolo master/slave que permite a un master, y sólo a uno, pedir respuestas de los slaves o realizar acciones dependiendo de las peticiones. El master puede dirigirse a slaves individuales o iniciar una difusión de mensajes para todos los slaves. Los slaves devuelven un mensaje (respuesta) a las peticiones que se les envían individualmente. No se devuelven respuestas a las peticiones de difusión desde el master.

Configuración de hardware

Una conexión Modbus puede establecerse en el puerto EIA RS-232 o EIA RS-485 y puede ejecutarse hasta en dos puertos de comunicaciones al mismo tiempo. La tabla que aparece a continuación enumera los dispositivos que se pueden utilizar.

Dispositivo	Puerto	Características
TWDCAA10/16/24DRF, TWDLMDA20/40DUK, TWDLMDA20/40DTK, TWDLMDA20DRT	1	Controlador base que admite EIA RS-485 de 3 conductores utilizando un conector miniDin.
TWDNOZ232D	2	Módulo de comunicaciones que admite EIA RS-232 de 3 conductores utilizando un conector miniDin. Nota: Este módulo sólo está disponible para los controladores modulares. Cuando el módulo está conectado, el controlador no puede tener un módulo de ampliación del monitor de operación.
TWDNOZ485D	2	Módulo de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector miniDin. Nota: Este módulo sólo está disponible para los controladores modulares. Cuando el módulo está conectado, el controlador no puede tener un módulo de ampliación del monitor de operación.
TWDNOZ485T	2	Módulo de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector de terminales. Nota: Este módulo sólo está disponible para los controladores modulares. Cuando el módulo está conectado, el controlador no puede tener un módulo de ampliación del monitor de operación.
TWDNAC232D	2	Adaptador de comunicaciones que admite EIA RS-232 de 3 conductores utilizando un conector miniDin. Nota: Este adaptador sólo está disponible para los controladores compactos de 16 y 24 E/S, y el módulo de ampliación del monitor de operación.
TWDNAC485D	2	Adaptador de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector miniDin. Nota: Este adaptador sólo está disponible para los controladores compactos de 16 y 24 E/S, y el módulo de ampliación del monitor de operación.
TWDNAC485T	2	Adaptador de comunicaciones que admite EIA RS-485 de 3 conductores utilizando un conector de terminales. Nota: Este adaptador sólo está disponible para los controladores compactos de 16 y 24 E/S, y el módulo de ampliación del monitor de operación.

Dispositivo	Puerto	Características
TWDXCPODM	2	<p>El módulo de ampliación del monitor de operación que admite un EIA RS-232 de 3 conductores utilizando un conector miniDIN, EIA RS-485 utilizando un conector miniDIN o EIA RS-485 utilizando un conector de terminales.</p> <p>Nota: Este módulo sólo está disponible para los controladores modulares. Cuando el módulo está conectado, el controlador no puede tener un módulo de ampliación de comunicaciones.</p>

Nota: La configuración del puerto 2 (disponibilidad y tipo) sólo se comprueba durante el encendido o reinicio mediante el firmware Executive.

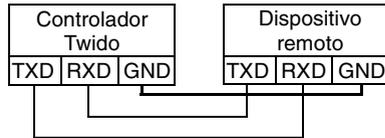
Cableado nominal

A continuación, se ilustran las conexiones de cableado nominal para los tipos EIA RS-232 y EIA RS-485.

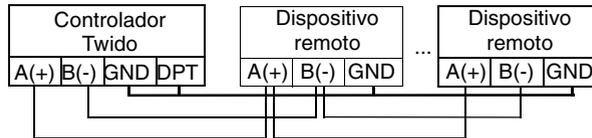
Nota: Si se utiliza el puerto 1 en el controlador Twido, la señal DPT deberá estar conectada a tierra. Esto indica al controlador Twido que la comunicación a través del puerto 1 es Modbus y no el protocolo utilizado para comunicarse con el software TwidoSoft. El dispositivo remoto específico puede requerir la utilización de señales adicionales (DTR, DSR, etc.).

Las conexiones de cable a cada dispositivo se muestran a continuación.

Cable EIA RS-232



Cable EIA RS-485



Nota: La conexión DPT a GND sólo es necesaria si está conectado a un controlador base en el puerto 1.

Configuración de software

Para configurar el controlador con el fin de utilizar una conexión serie para enviar y recibir caracteres utilizando el protocolo Modbus, deberá:

Etapa	Descripción
1	Configurar el puerto serie para Modbus utilizando TwidoSoft.
2	Crear en su aplicación un búfer de transmisión/recepción para Modbus con el fin de utilizar la instrucción EXCHx.

Configuración del puerto

Un controlador Twido puede utilizar su puerto 1 primario o un puerto 2 configurado de forma opcional para utilizar el protocolo Modbus. Para configurar un puerto serie para Modbus, siga estos pasos.

Paso	Acción
1	Definir físicamente cualquier módulo o cartucho opcionales configurados en la base.
2	Hacer clic con el botón derecho en el puerto y hacer clic en Configuración de com. del controlador... y cambiar el tipo de puerto serie a "Modbus".
3	Establecer los parámetros de comunicaciones asociados.

Master Modbus

El modo master de Modbus permite al controlador iniciar una transmisión de peticiones Modbus, esperando una respuesta desde un slave Modbus. El modo master de Modbus sólo se admite a través de la instrucción EXCHx. El modo master de Modbus admite los formatos ASCII Modbus y RTU Modbus. El tamaño máximo de las tramas transmitidas o recibidas es 128 bytes y la tabla de palabras asociada a la instrucción EXCHx está compuesta por tablas de transmisión y recepción.

	Byte de mayor valor	Byte de menor valor
Palabras de control	Comando	Longitud (Tx/Rx)
	Rx Offset	Tx Offset
Tabla de transmisión	Byte 1 transmitido	Byte 2 transmitido

	...	Byte n transmitido
	Byte n+1 transmitido	
Tabla de recepción	Byte 1 recibido	Byte 2 recibido

	...	Byte p recibido
	Byte p+1 recibido	

Parámetros de control

El byte de **longitud** contiene la longitud que se va a transmitir, sobrescrita por el número de caracteres recibidos al final de la recepción, en caso de que ésta se solicite.

Este parámetro es la longitud en bytes de la tabla de transmisión. Si el parámetro Tx Offset es igual a 0, este parámetro será igual que la propia longitud de trama menos 2 bytes CRC. Si el parámetro Tx Offset no es igual a 0, no se transmitirá un byte del búfer (indicado por el valor de offset) y este parámetro será igual a la propia longitud de trama más 1.

El byte de **comando**, en caso de que se produzca una solicitud RTU Modbus (excepto para la difusión), debe ser siempre igual a 1 (Tx y Rx).

El byte **Tx Offset** contiene el offset (1 para el primer byte, 2 para el segundo byte, etc.) dentro de la tabla de transmisión que se ignorará cuando se transmita el paquete. Esto se utiliza para gestionar los problemas asociados a los valores de bytes/palabras del protocolo Modbus. Por ejemplo, si este byte contiene 3, el tercer byte se ignorará, haciendo que el cuarto byte de la tabla sea el tercero en transmitirse.

El byte **Rx Offset** contiene el offset (1 para el primer byte, 2 para el segundo byte, etc.) dentro de la tabla de recepción que se agregará cuando se transmita el paquete. Esto se utiliza para gestionar los problemas asociados a los valores de bytes/palabras del protocolo Modbus. Por ejemplo, si este byte contiene 3, el tercer byte de la tabla se completará con un cero y el tercer byte recibido se introducirá en la cuarta ubicación de la tabla.

Tablas de transmisión/recepción

Cuando se utiliza cualquier modo (ASCII Modbus o RTU Modbus), la tabla de transmisión se completará con la solicitud previa a la ejecución de la instrucción EXCHx. En el momento de la ejecución, el controlador determina qué es la capa de enlace de datos y realiza todas las conversiones necesarias para procesar la transmisión y la respuesta. Los caracteres de inicio, fin y comprobación no se almacenan en las tablas de transmisión/recepción.

Una vez transmitidos todos los bytes, el controlador cambia a modo de recepción y espera a recibir los bytes. La recepción se completa de una de estas formas: el carácter de final de trama se recibe en modo ASCII; se detecta el timeout de un carácter o trama; la tabla de recepción está llena.

Las entradas de **byte X transmitido** contienen los datos del protocolo Modbus (codificación RTU) que se va a transmitir. Si el puerto de comunicaciones está configurado para ASCII Modbus, los caracteres de trama correctos se agregan a la transmisión. El primer byte contiene la dirección de dispositivo (específica o difusión), el segundo byte contiene el código de función y el resto contienen información asociada al código de función.

Nota: Ésta es una aplicación típica, pero no define todas las posibilidades. No se realizará ninguna validación de los datos que se están transmitiendo.

Las entradas de **byte X recibido** contienen los datos del protocolo Modbus (codificación RTU) que se va a recibir. Si el puerto de comunicaciones está configurado para ASCII Modbus, los caracteres de trama correctos se eliminan de la respuesta. El primer byte contiene la dirección de dispositivo, el segundo byte contiene el código de función (o código de respuesta) y el resto contienen información asociada al código de función.

Nota: Ésta es una aplicación típica, pero no define todas las posibilidades. No se realizará ninguna validación de los datos que se están recibiendo, excepto para la verificación de la suma de control.

Slave Modbus

El modo slave Modbus permite al controlador responder a las solicitudes de Modbus desde un master Modbus. El controlador admite los datos Modbus estándar y las funciones de control, así como las ampliaciones UMAS para el acceso a objetos y la configuración.

Cuando el cable TSXPCX1031 se conecta al controlador, las comunicaciones en modo slave Modbus se inician en el puerto, inhabilitando temporalmente el modo de comunicaciones que estaba en ejecución antes de que se conectara el cable. El protocolo Modbus admite dos formatos de capa de enlace de datos: ASCII y RTU. Cada uno está definido por la implementación de la capa física: ASCII utiliza 7 bits de datos y RTU utiliza 8 bits de datos.

Cuando se utiliza el modo ASCII Modbus, cada byte del mensaje se envía como dos caracteres ASCII. La trama ASCII Modbus comienza con un carácter inicial (':') y finaliza con dos caracteres finales (CR y LF). El carácter de final de trama se establece de forma predeterminada como 0x0A (avance de línea) y el usuario puede modificar el valor de este byte durante la configuración. El valor de comprobación para la trama ASCII Modbus es un complemento de dos de la trama, excluyendo los caracteres inicial y final.

El modo RTU Modbus no vuelve a formatear el mensaje antes de transmitirlo; sin embargo, utiliza un modo de cálculo de suma de verificación diferente, especificado como CRC.

La capa de enlace de datos de Modbus tiene las siguientes limitaciones:

- Dirección 1-247
- Bits: 128 bits al realizar la solicitud utilizando solicitudes abiertas de Modbus
- Palabras: 64 palabras de 16 bits al realizar la solicitud utilizando solicitudes abiertas de Modbus

Intercambio de mensajes

El controlador Twido puede configurarse para enviar o recibir mensajes en modo carácter. El lenguaje ofrece dos servicios para esto:

- **Instrucción EXCHx**: para transmitir/recibir mensajes
- **Bloque de función %MSGx**: para controlar los intercambios de mensajes.

Cuando se procesa una instrucción EXCHx, el controlador Twido utiliza el protocolo configurado para dicho puerto.

Nota: Cada puerto de comunicaciones puede configurarse para protocolos diferentes o para el mismo. El modo de acceder a la instrucción EXCHx o al bloque de función %MSGx para cada puerto de comunicaciones es agregando el número de puerto (1 ó 2).

**Instrucción
EXCHx**

La instrucción EXCHx permite al controlador Twido enviar o recibir información dirigida o procedente de dispositivos Modbus. El usuario define una tabla de palabras (%MWi:L o %KW:L) que contiene información de control y los datos que se van a enviar o recibir (hasta 64 palabras en la transmisión o recepción). El formato de la tabla se describe en secciones anteriores.

Un intercambio de mensajes se realiza utilizando la instrucción EXCHx.

Sintaxis: [EXCHx %MWi:L] o [EXCHx %KW:L]

donde: x = número de puerto (1 ó 2).

L = número de palabras en la tabla de palabras.

El controlador Twido debe finalizar el intercambio desde la primera instrucción EXCHx antes de que se ejecute una segunda. El bloque de función %MSGx debe utilizarse cuando se envíen varios mensajes.

El procesamiento de la instrucción de lista EXCHx se produce inmediatamente, con cualquier transmisión iniciada bajo el control de interrupción (la recepción de datos también se encuentra bajo el control de interrupción), que se considera procesamiento de fondo.

Bloque de función %MSGx

El uso del bloque de función %MSGx es opcional; puede utilizarse para gestionar los intercambios de datos. El bloque de función %MSGx tiene tres propósitos:

- **Comprobación de errores de comunicación**

La comprobación de errores verifica que la longitud del bloque (tabla de palabras) programada con la instrucción EXCHx es lo suficientemente grande para contener la longitud del mensaje que se va a enviar. Esto se compara con la longitud programada en el byte de menor valor de la primera palabra de la tabla de palabras.

- **Coordinación de varios mensajes**

Para asegurar la coordinación cuando se envíen varios mensajes, el bloque de función %MSGx proporciona la información requerida para determinar cuándo está completo un mensaje anterior.

- **Transmisión de mensajes prioritarios**

El bloque de función %MSGx permite la detención de la transmisión del mensaje actual para permitir el envío inmediato de un mensaje urgente.

El bloque de función %MSGx tiene una entrada y dos salidas asociadas.

Entrada/salida	Definición	Descripción
R	Restablecer entrada	Poner a 1: reinicializa la comunicación o restablece el bloque (%MSGx.E = 0 y %MSGx.D = 1).
%MSGx.D	Comunicación completa	0: solicitud en curso. 1: comunicación realizada si se produce el final de la transmisión, se recibe el carácter final, se produce un error o se restablece el bloque.
%MSGx.E	Error	0: longitud del mensaje y enlace correctos. 1: si hay un comando inválido, la tabla se configura de forma incorrecta, se recibe un carácter incorrecto (velocidad, paridad, etc.) o la tabla de recepción está llena.

Limitaciones

Es importante observar las siguientes limitaciones:

- La disponibilidad y el tipo de puerto 2 sólo se comprueban durante el encendido o reinicio.
- El procesamiento de un mensaje en el puerto 1 se cancela cuando se conecta TwidoSoft.
- EXCHx o %MSG no se puede procesar en un puerto configurado como conexión remota.
- EXCHx interrumpe el procesamiento de slave de Modbus activo (excepto en el caso del procesamiento de TwidoSoft).
- El procesamiento de las instrucciones EXCHx no se vuelve a intentar en caso de error.
- R %MSGx se puede utilizar para interrumpir el procesamiento de recepción de instrucción EXCHx.
- Las instrucciones EXCHx se pueden configurar con un timeout para interrumpir la recepción.
- Se controlan mensajes múltiples a través de %MSGx.D.

Condiciones de modo de funcionamiento y error

Si se produce un error durante el uso de una instrucción EXCHx, los bits %MSGx.D y %MSGx.E se ponen a 1 y la palabra de sistema %SW63 contiene el código de error para el puerto 1, y %SW64 contiene el código de error para el puerto 2.

Palabras de sistema	Uso
%SW63	Código de error EXCH1: 0 - operación correcta 1 - búfer de transmisión demasiado largo (> 128) 2 - búfer de transmisión demasiado pequeño 3 - tabla de palabras demasiado pequeña 4 - tabla de recepción desbordada 5 - timeout transcurrido 6 - error de transmisión (error recibido en la respuesta) 7 - comando incorrecto en la tabla 8 - puerto seleccionado no configurado/disponible 9 - error de recepción 10 - no se puede utilizar %KW si se está recibiendo 11 - offset de transmisión mayor que tabla de transmisión 12 - offset de recepción mayor que tabla de recepción 13 - procesamiento EXCH detenido por el controlador
%SW64	Código de error EXCH2: consulte %SW63.

Reinicio del controlador master

Si se reinicia un controlador master/slave, se producirá uno de los siguientes eventos:

- Un inicio en frío (%S0 = 1) fuerza una reinicialización de las comunicaciones.
- Un inicio en caliente (%S1 = 1) fuerza una reinicialización de las comunicaciones.
- En modo de detención, el controlador detiene todas las comunicaciones Modbus.

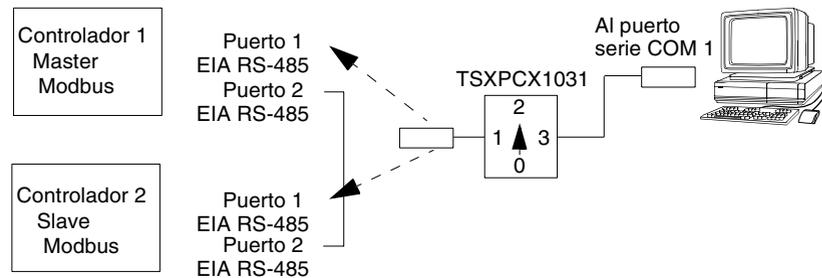
Ejemplo 1 de conexión Modbus

Para configurar una conexión Modbus debe seguir estos pasos.

1. Configurar el hardware.
2. Conectar el cable de comunicaciones Modbus.
3. Configurar el puerto.
4. Escribir una aplicación.
5. Inicializar el editor de tablas de animación.

Los siguientes diagramas ilustran el uso del código de función Modbus 3 para leer las palabras de salida de un slave. En este ejemplo se utilizan dos controladores Twido.

Paso 1: Configurar el hardware



La configuración del hardware está integrada por dos controladores Twido. Uno se configura como master de Modbus y el otro, como slave de Modbus.

Nota: En este ejemplo, cada controlador se ha configurado para utilizar EIA RS-485 en el puerto 1 y un puerto 2 opcional EIA RS-485. En un controlador modular, el puerto 2 opcional puede ser un TWDNOZ485D o un TWDNOZ485T. En un controlador compacto, el puerto 2 opcional puede ser un TWDNAC485D o un TWDNAC485T.

Para configurar cada controlador, conecte el cable TSXPCX1031 al puerto 1 del primer controlador.

Nota: El cable TSXPCX1031 no se puede conectar a más de un controlador al mismo tiempo y sólo en el puerto 1 EIA RS-485.

A continuación, conecte el cable al puerto COM 1 del PC. Asegúrese de que el conmutador está en posición 2. Descargue y compruebe cada aplicación. Repita el procedimiento con el segundo controlador.

Paso 2: Conectar el cable de comunicaciones Modbus



El cableado de este ejemplo muestra una conexión punto a punto sencilla. Las tres señales A(+), B(-) y GND están cableadas de acuerdo con el diagrama.

Si se utiliza el puerto 1 del controlador Twido, la señal DPT se debe poner a tierra. Este condicionamiento de DPT determina si TwidoSoft está conectado. Si está puesto a tierra, el controlador utilizará la configuración de puerto ajustada en la aplicación para determinar el tipo de comunicación.

Paso 3: Configurar el puerto

Hardware -> Agregar opción TWDNOZ485-	
Hardware => Configuración de com. del controlador	
Puerto:	2
Tipo:	Modbus
Dirección:	1
Velocidad de transmisión:	19200
Datos:	8 bits
Paridad:	Ninguna
Detener:	1 bit
Fin de trama:	65
Timeout de respuesta:	10 x 100 ms
Timeout de trama:	10 ms

Hardware -> Agregar opción TWDNOZ485-	
Hardware => Configuración de com. del controlador	
Puerto:	2
Tipo:	Modbus
Dirección:	2
Velocidad de transmisión:	19200
Datos:	8 bits
Paridad:	Ninguna
Detener:	1 bit
Fin de trama:	65
Timeout de respuesta:	100 x 100 ms
Timeout de trama:	10 ms

En las aplicaciones master y slave, están configurados los puertos opcionales EIA RS-485. Asegúrese de cambiar las comunicaciones del controlador para inicializar las direcciones Modbus o el puerto 2 a dos direcciones diferentes. En este ejemplo, el master se pone a una dirección de 1 y el slave a una dirección de 2. El número de bits se pone a 8, lo que indica que se utilizará el modo RTU Modbus. Si se ajustara a 7, se utilizaría el modo ASCII Modbus. El otro cambio realizado es incrementar el valor predeterminado de timeout de respuesta a 1 segundo.

Nota: Puesto que se ha seleccionado el modo RTU Modbus, no se tiene en cuenta el parámetro "Fin de trama".

Paso 4: Escribir una aplicación

```
LD 1
[%MW0 := 16#0106 ]
[%MW1 := 16#0300 ]
[%MW2 := 16#0203 ]
[%MW3 := 16#0000 ]
[%MW4 := 16#0004 ]
LD 1
AND %MSG2.D
[EXCH2 %MW0:11]
LD %MSG2.E
ST %Q0.0
END
```

```
LD 1
[%MW0 := 16#6566 ]
[%MW1 := 16#6768 ]
[%MW2 := 16#6970 ]
[%MW3 := 16#7172 ]
END
```

Mediante TwidoSoft, se escribe un programa de aplicación tanto para el master como para el slave. Para el slave, simplemente se inicializan algunas palabras de memoria para un conjunto de valores conocidos. En el master, el bloque de intercambio se inicializa para leer 4 palabras del slave en la dirección Modbus 2 comenzando por la ubicación %MW0.

Nota: Preste atención al uso del offset RX ajustado en %MW1 del master de Modbus. El offset de tres añadirá un byte (valor = 0) en la tercera posición del área de recepción de la tabla. De este modo, las palabras se alinean en el master, de forma que se mantienen dentro de los límites de palabras. Sin este offset, cada palabra de datos se dividiría entre dos palabras en el bloque de intercambio. Este offset se utiliza por comodidad.

Antes de ejecutar la instrucción EXCH2, la aplicación comprueba el bit de finalización asociado a %MSG2. Por último, se detecta el estado de error de %MSG2 y se guarda en el primer bit de salida de las E/S locales del controlador base. También se podría añadir una comprobación de errores adicional mediante %SW64 para aumentar la seguridad.

Paso 5: Inicializar el editor de tablas de animación

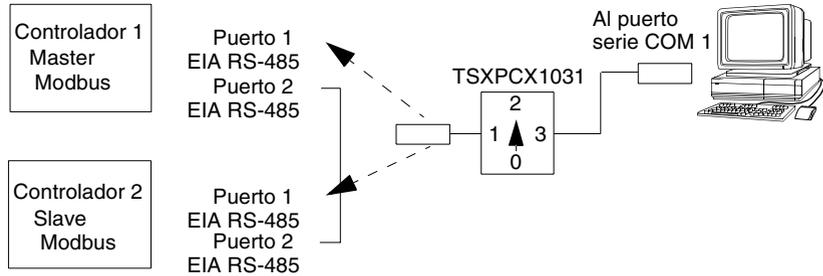
Dirección	Actual	Guardado	Formato
1	%MW5	0203	0000 Hexadecimal
2	%MW6	0008	0000 Hexadecimal
3	%MW7	6566	0000 Hexadecimal
4	%MW8	6868	0000 Hexadecimal
5	%MW9	6970	0000 Hexadecimal
6	%MW10	7172	0000 Hexadecimal

Después de descargar y ajustar cada controlador para que se ejecute, abra una tabla de animación en el master. Examine la sección de respuesta de la tabla para comprobar que el código de respuesta sea 3 y asegurarse de que se haya leído el número de bytes correcto. En este ejemplo también se aprecia que las palabras leídas del slave (comenzando por %MW7) están correctamente alineadas con los límites de palabras del master.

Ejemplo 2 de conexión Modbus

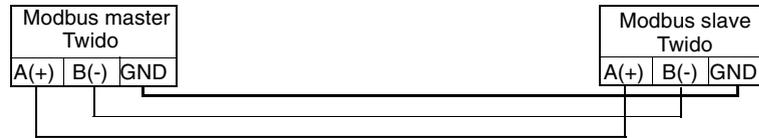
El diagrama que aparece a continuación ilustra el uso del código de función Modbus 16 para escribir las palabras de salida en un slave. En este ejemplo se utilizan dos controladores Twido.

Paso 1: Configurar el hardware



La configuración de hardware es idéntica a la del ejemplo anterior.

Paso 2: Conectar el cable de comunicaciones Modbus



El cableado de las comunicaciones Modbus es idéntico al del ejemplo anterior.

Paso 3: Configurar el puerto

Hardware -> Agregar opción TWDNOZ485-	
Hardware => Configuración de com. del controlador	
Puerto:	2
Tipo:	Modbus
Dirección:	1
Velocidad de transmisión:	19200
Datos:	8 bits
Paridad:	Ninguna
Detener:	1 bit
Fin de trama:	65
Timeout de respuesta:	10 x 100 ms
Timeout de trama:	10 ms

Hardware -> Agregar opción TWDNOZ485-	
Hardware => Configuración de com. del controlador	
Puerto:	2
Tipo:	Modbus
Dirección:	2
Velocidad de transmisión:	19200
Datos:	8 bits
Paridad:	Ninguna
Detener:	1 bit
Fin de trama:	65
Timeout de respuesta:	100 x 100 ms
Timeout de trama:	10 ms

La configuración del puerto es idéntica a la del ejemplo anterior.

Paso 4: Escribir una aplicación

```
LD 1
[%MW0 := 16#010C ]
[%MW1 := 16#0007 ]
[%MW2 := 16#0210 ]
[%MW3 := 16#0010 ]
[%MW4 := 16#0002 ]
[%MW5 := 16#0004 ]
[%MW6 := 16#6566 ]
[%MW7 := 16#6768 ]
```

```
LD 1
AND %MSG2.D
[EXCH2 %MW0:11]
LD %MSG2.E
ST %Q0.0
END
```

```
LD 1
[%MW18 := 16#FFFF ]
END
```

Mediante TwidoSoft, se crea un programa de aplicación tanto para el master como para el slave. Para el slave, inicialice una sola palabra de memoria %MW18. De este modo, se asignará espacio en el slave para las direcciones de memoria de %MW0 a %MW18. Si no se asigna este espacio, el bloque de intercambio intentará escribir en ubicaciones que no existen en el slave.

En el master, el bloque de intercambio se inicializa para escribir 12 palabras (0C hexadecimal) en el slave en la dirección Modbus 2 comenzando por la ubicación %MW16 (10 hexadecimal).

Nota: Preste atención al uso del offset TX ajustado en %MW1 de la aplicación del master de Modbus. El offset de siete suprimirá el byte superior de la sexta palabra (el valor hexadecimal 00 en %MW5). De esta forma se alinean los valores de datos en la tabla de transmisión del bloque de intercambio de tal modo que se mantienen dentro de los límites de palabras.

Antes de ejecutar la instrucción EXCH2, la aplicación comprueba el bit de finalización asociado a %MSG2. Por último, se detecta el estado de error de %MSG2 y se guarda en el primer bit de salida de las E/S locales del controlador base. También se podría añadir una comprobación de errores adicional mediante %SW64 para aumentar la seguridad.

Paso 5: Inicializar el editor de tablas de animación

Dirección	Actual	Guardado	Formato
1 %MW0	010C	0000	Hexadecimal
2 %MW1	0007	0000	Hexadecimal
3 %MW2	0210	0000	Hexadecimal
4 %MW3	0010	0000	Hexadecimal
5 %MW4	0002	0000	Hexadecimal
6 %MW5	0004	0000	Hexadecimal
7 %MW6	6566	0000	Hexadecimal
8 %MW7	6768	0000	Hexadecimal
9 %MW8	0210	0000	Hexadecimal
10 %MW9	0010	0000	Hexadecimal
11 %MW10	0004	0000	Hexadecimal

Dirección	Actual	Guardado	Formato
1 %MW16	6566	0000	Hexadecimal
2 %MW17	6768	0000	Hexadecimal

Después de descargar y ajustar cada controlador para que se ejecute, abra una tabla de animación. Los dos valores de %MW16 y %MW17 se escriben en el slave. En el master, la tabla de animación se puede utilizar para examinar la parte de la tabla de recepción de los datos de intercambio. Estos datos indican la dirección del slave, el código de respuesta, la primera palabra escrita y el número de palabras escritas comenzando por %MW8 en el ejemplo anterior.

Solicitudes Modbus estándar

Introducción

Puede utilizar estas solicitudes para intercambiar datos entre dispositivos con el fin de acceder a información de bits y palabras. Se utiliza el mismo formato de tabla para los modos RTU y ASCII.

Formato	Referencia
Bit	Registros %Mi, 0x ó 1x
Palabra	Registros %MWi, 3x ó 4x

Master Modbus: Leer bits de entrada y salida N

Esta tabla representa las solicitudes 01 y 02.

	Tabla Índice	Byte de mayor valor	Byte de menor valor
Control	0	01 (Tx/Rx)	06 (Longitud Tx)
	1	00 (Rx Offset)	00 (Tx Offset)
Tabla de transmisión	2	Slave@(1..247)	01 (Código de solicitud)
	3	Número del primer bit que se va a leer	
	4	N = Número de bits que se van a leer	
Tabla de recepción (después de respuesta)	5	Slave@(1..247)	01 (Código de respuesta)
	6	Número de bytes de datos transmitidos (1 byte por bit)	
	7	Primer byte leído (valor = 00 ó 01)	Segundo byte leído (si N>1)
	8	Tercer byte leído	
	...		
	(N/2)+6	Byte N leído (si N>1)	

Master Modbus: Esta tabla representa las solicitudes 03 y 04.

Leer palabras de entrada y salida N

	Tabla Índice	Byte de mayor valor	Byte de menor valor
Control	0	01 (Tx/Rx)	06 (Longitud Tx)
	1	03 (Rx Offset)	00 (Tx Offset)
Tabla de transmisión	2	Slave@(1..247)	03 (Código de solicitud)
	3	Número de la primera palabra que se va a leer	
	4	N = Número de palabras que se van a leer	
Tabla de recepción (después de respuesta)	5	Slave@(1..247)	03 (Código de respuesta)
	6	00 (byte agregado por acción Rx Offset)	2*N (número de bytes leídos)
	7	Primera palabra leída	
	8	Segunda palabra leída (si N>1)	
	...		
	N+6	palabra N leída (si N>2)	

Nota: Rx Offset=3 agregará un byte (valor=0) a la tercera posición en la tabla de recepción. Permitir una buena ubicación del número de bytes leídos y de los valores de palabras leídas en esta tabla.

Master Modbus: Esta tabla representa la solicitud 05.
Escribir bit de salida 1

	Tabla Índice	Byte de mayor valor	Byte de menor valor
Control	0	01 (Tx/Rx)	06 (Longitud Tx)
	1	00 (Rx Offset)	00 (Tx Offset)
Tabla de transmisión	2	Slave@(1..247)	05 (Código de solicitud)
	3	Número del bit que se va a escribir	
	4	Valor de bit que se va a escribir	
Tabla de recepción (después de respuesta)	5	Slave@(1..247)	05 (Código de respuesta)
	6	Número del bit escrito	
	7	Valor escrito	

Nota:

- Esta solicitud no necesita utilizar el offset.
- Aquí la trama de respuesta es la misma que la de solicitud (en un caso normal).
- Para que un bit escriba 1, la palabra asociada en la tabla de transmisión debe contener el valor FF00H. 0 para un valor de bit 0.

Master Modbus: Esta tabla representa la solicitud 06.
Escribir palabra de salida 1

	Tabla Índice	Byte de mayor valor	Byte de menor valor
Control	0	01 (Tx/Rx)	06 (Longitud Tx)
	1	00 (Rx Offset)	00 (Tx Offset)
Tabla de transmisión	2	Slave@(1..247)	06 (Código de solicitud)
	3	Número de la palabra que se va a escribir	
	4	Valor de la palabra que se va a escribir	
Tabla de recepción (después de respuesta)	5	Slave@(1..247)	06 (Código de respuesta)
	6	Número de la palabra escrita	
	7	Valor escrito	

Nota:

- Esta solicitud no necesita utilizar el offset.
- Aquí la trama de respuesta es la misma que la de solicitud (en un caso normal).

**Master Modbus:
Escribir bits de
salida N**

Esta tabla representa la solicitud 15.

	Tabla Índice	Byte de mayor valor	Byte de menor valor
Control	0	01 (Tx/Rx)	8 + número de bytes (Tx)
	1	00 (Rx Offset)	07 (Tx Offset)
Tabla de transmisión	2	Slave@(1..247)	15 (Código de solicitud)
	3	Número del primer bit que se va a escribir	
	4	N_1 = Número de bits que se van a escribir	
	5	00 (byte no enviado, efecto offset)	N_1 = Número de bytes de datos que se van a escribir
	6	Valor del primer byte	Valor del segundo byte
	7	Valor del tercer byte	
	...		
	$6+(N_2/2)$	Valor del byte N_2	
Tabla de recepción (después de respuesta)		Slave@(1..247)	15 (Código de respuesta)
		Número del primer bit escrito	
		Número de bits escritos (= N_1)	

Nota:

- Tx Offset=7 eliminará el séptimo byte en la trama enviada. Permitir una buena correspondencia de los valores de palabra en la tabla de transmisión.

Master Modbus: Esta tabla representa la solicitud 16.
Escribir palabras de salida N

	Tabla Índice	Byte de mayor valor	Byte de menor valor
Control	0	01 (Tx/Rx)	8 + (2*N) (Longitud Tx)
	1	00 (Rx Offset)	07 (Tx Offset)
Tabla de transmisión	2	Slave@(1..247)	16 (Código de solicitud)
	3	Número de la primera palabra que se va a escribir	
	4	N = Número de palabras que se van a escribir	
	5	00 (byte no enviado, efecto offset)	2*N=Número de bytes que se van a escribir
	6	Valor de la primera palabra que se va a escribir	
	7	Segundo valor que se va a escribir	
	...		
	N+5	Valor N que se va a escribir	
Tabla de recepción (después de respuesta)	N+6	Slave@(1..247)	16 (Código de respuesta)
	N+7	Número de la primera palabra escrita	
	N+8	Número de palabra escrita (= N)	

Nota: Tx Offset=7 eliminará el quinto byte MMSB en la trama enviada. Permitir una buena correspondencia de los valores de palabra en la tabla de transmisión.

Funciones analógicas incorporadas



Presentación

Vista general

Este capítulo describe el modo de gestionar los potenciómetros y el canal analógico incorporado.

Contenido:

Este capítulo contiene los siguiente apartados:

Apartado	Página
Potenciómetros	126
Canal analógico	128

Potenciómetros

Introducción

Los controladores Twido tienen:

- Un potenciómetro en los controladores TWDLCAA10DRF y TWDLCAA16DRF
 - Dos potenciómetros en el controlador TWDLCAA24DRF
-

Programación

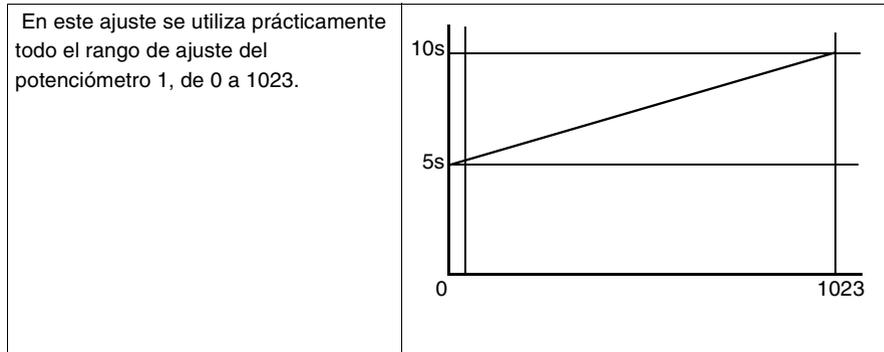
Los valores numéricos, de 0 a 1023 para el potenciómetro 1 y de 0 a 511 para el potenciómetro 2, correspondientes a los valores analógicos que indican estos potenciómetros, forman parte de las dos palabras de sistema siguientes:

- %IW0.0.0 para el potenciómetro 1 (más a la izquierda)
- %IW0.0.1 para el potenciómetro 2 (más a la derecha)

Estas palabras se pueden utilizar en operaciones aritméticas y para cualquier ajuste, por ejemplo, preestablecer un retardo o un contador, ajustar la frecuencia del generador de pulsos o el precalentamiento de una máquina.

Ejemplo

Ajuste de un retardo de 5 a 10 segundos utilizando el potenciómetro 1:

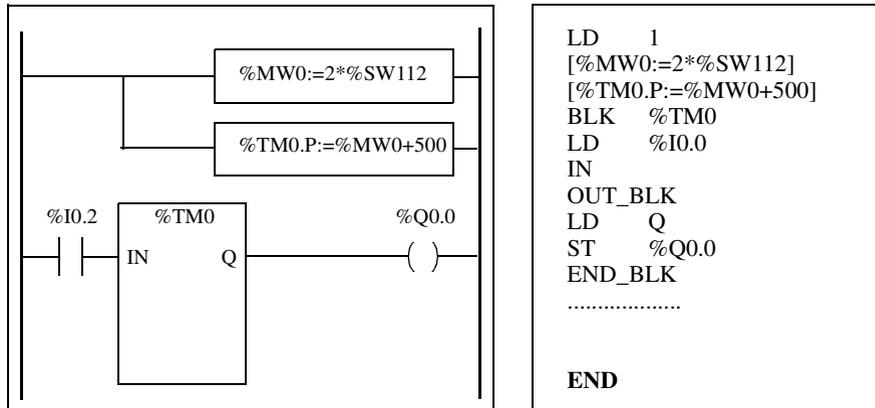


Los siguientes parámetros están seleccionados en la configuración del bloque de retardo %TM0:

- Tipo TON
- Base de tiempo TB: 10 ms

El valor predeterminado del retardo se calcula a partir del valor de ajuste del potenciómetro utilizando la siguiente ecuación $\%TM0.P := 2*\%SW112+500$.

Código del ejemplo anterior:



Canal analógico

Introducción

Todos los controladores modulares (TWDLMDA20DTK, TWDLMDA20DUK, TWDLMDA20DRT, TWDLMD40DTK y TWDLMD40DUK) disponen de un canal analógico incorporado. La entrada de tensión varía entre 0 y 10 V y la señal digitalizada entre 0 y 511. El canal analógico aprovecha un esquema de promedio simple que se aplica a ocho muestras..

Principio

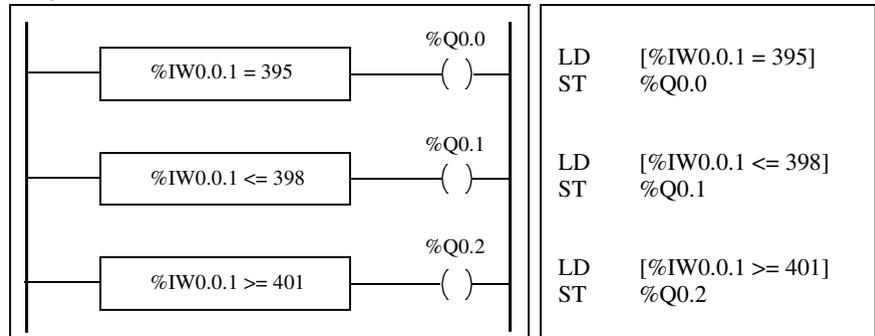
Un convertidor de digital a analógico muestrea una tensión de entrada de 0 a 10 V con un valor digital de 0 a 511. Este valor se almacena en la palabra de sistema %IW0.0.1. El valor es lineal en todo el rango, de modo que cada conteo es aproximadamente de 20 mV (10 V/512). Una lectura de 511 se utiliza para detectar si se ha superado el valor máximo de la señal de entrada.

Ejemplo de programación

Control de la temperatura de un horno: La temperatura del horno se fija en 350 °C. Una variación de +/- 2,5 °C supone la interrupción de las salidas %Q0.1 y %Q0.2. En este ejemplo se utilizan prácticamente todos los rangos de configuración posibles del canal analógico de 0 a 511. La configuración analógica de los valores teóricos de temperatura es la siguiente.

Temperatura (°C)	Tensión	Palabra de sistema %IW0.0.1
0	0	0
347,5	7,72	395
350	7,77	398
352,5	7,83	401
450	10	511

Código del ejemplo anterior:



Módulos analógicos de gestión



Presentación

Vista general Este capítulo ofrece una vista general de los módulos analógicos de gestión para controladores Twido.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Vista general del módulo analógico	130
Direccionamiento de entradas y salidas analógicas	131
Configuración de E/S analógicas	133
Ejemplo de uso de módulos analógicos	135

Vista general del módulo analógico

Introducción

Además del potenciómetro integrado de 10 bits y el canal analógico de 9 bits, todos los controladores Twido que apoyan ampliaciones de E/S pueden comunicar y configurar módulos de E/S analógicas.

Los módulos son los siguientes:

Nombre	Canales	Rango de señal	Codificado
TWDAMI2HT	2 entrada	0 a 10 V o 4 a 20 mA	12 Bit
TWDAM01HT	1 salida	0 a 10 V o 4 a 20 mA	12 Bit
TWDAMM3HT	2 entrada, 1 salida	0 a 10 V o 4 a 20 mA	12 Bit
TWDALM3LT	2 entrada, 1 salida	0 a 10 V, entradas Th o RTD, salidas 4 a 20 mA	12 Bit

Funcionamiento de módulos analógicos

Las palabras de entrada y de salida (%IW y %QW) se utilizan para intercambiar datos entre la aplicación del usuario y cualquier canal analógico. La actualización de estas palabras se lleva a cabo de manera sincronizada con la ejecución del controlador con el modo de ejecución.

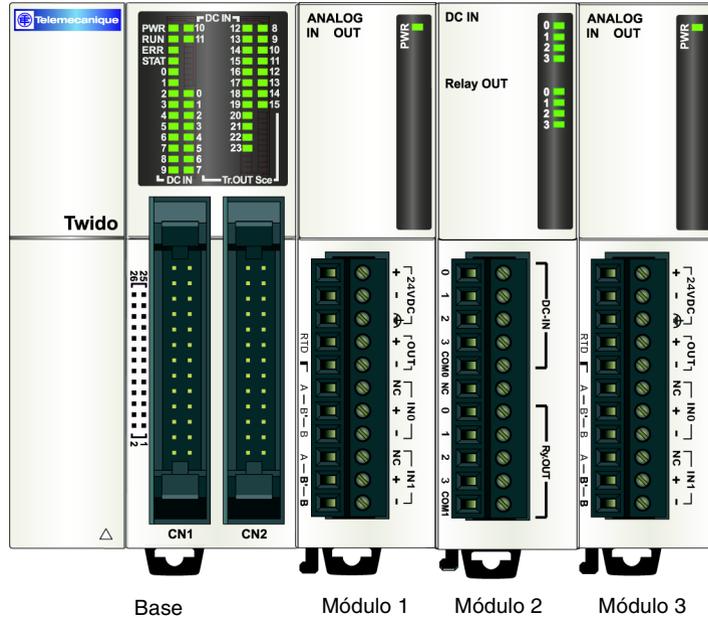
	AVISO
	<p>Funcionamiento inesperado del equipo</p> <p>Cuando el control se establece en Detener, la salida analógica se establece en su posición anterior. Como en el caso de la salida digital, esta posición es cero.</p> <p>Si no se respetan estas precauciones pueden producirse daños corporales y/o materiales</p>

Direccionamiento de entradas y salidas analógicas

Introducción Se asignan direcciones a los canales analógicos según su ubicación en el bus de ampliación.

Ejemplo de direccionamiento de E/S analógicas

En este ejemplo, el controlador TWDLMDA40DUK tiene el potenciómetro integrado de 10 bit y un canal analógico integrado de 9 bits. En el bus de ampliación, se configuran un modelo analógico TWDAMM3HT, un modelo de relé digital de entrada/salida TWDDMM8DRT y un segundo módulo analógico TWDAMM3HT.



La tabla que aparece a continuación proporciona información acerca del direccionamiento de cada salida.

Descripción	Base	Módulo 1	Módulo 2	Módulo 3
Potenciómetro 1	%IW0.0.0			
Canal analógico integrado o potenciómetro 2	%IW0.0.1			
Canal 1 de entrada analógica		%IW0.1.0		%IW0.3.0
Canal 2 de entrada analógica		%IW0.1.1		%IW0.3.1
Canal 1 de salida analógica		%QW0.1.0		%QW0.3.0
Canales de entrada digital			%I0.2.0 - %I0.2.3	
Canales de salida digital			%Q0.2.0 - %Q0.2.3	

Configuración de E/S analógicas

Introducción Esta sección proporciona información acerca de la configuración de las entradas y salidas del módulo analógico.

Configuración de E/S analógicas El cuadro de diálogo Configurar módulo se utiliza para administrar los parámetros de los módulos analógicos.

Nota: Los parámetros sólo se pueden modificar en estado offline, cuando no esté conectado al controlador.

Las direcciones se asignan a los canales analógicos según su ubicación en el bus de ampliación. Para facilitar la programación, también puede asignar símbolos previamente definidos para gestionar los datos en la aplicación.

Es posible configurar los tipos de canal para que el único canal de salida de los modelos TWDAM01HT, TWDAMM3HT y TWDALM3LT sea:

- No utilizado
- 0 - 10 V
- 4 - 20 mA

Es posible configurar los tipos de canal para que los dos canales de entrada de los modelos TWDAMI2HT y TWDAMM3HT sean:

- No utilizado
- 0 - 10 V
- 4 - 20 mA

	<p>AVISO</p>
	<p>Daños inesperados en el equipo</p> <p>Si ha cableado su entrada para una medición de tensión y configura TwidoSoft para el tipo de configuración actual, puede dañar el módulo analógico de forma permanente. Asegúrese de que el cableado se realiza con arreglo a la configuración de TwidoSoft.</p> <p>Si no se respetan estas precauciones pueden producirse daños corporales y/o materiales</p>

Los dos canales de entrada del modelo TWDALM3LT se pueden configurar como:

- No utilizado
- Termopar K
- Termopar J
- Termopar T
- PT 100

Cuando se configura un canal, puede elegir entre asignar unidades y asignar el rango de entradas según la tabla que aparece a continuación.

Rango	Unidades	Descripción
Normal	Ninguna	Rango establecido desde un mínimo de 0 hasta 4.095.
Personalizado	Ninguna	Definido por el usuario, con un mínimo no inferior a -32.768 y un máximo no superior a 32.767.
Centígrados	0,1 °C	Escala termométrica internacional. Sólo disponible para los canales de entrada del modelo TWDALM3LT.
Fahrenheit	0,1 °F	Escala termométrica en la que el punto de ebullición del agua es 212 °F (100 °C) y el de congelación es 32 °F (0 °C). Sólo disponible para los canales de entrada del modelo TWDALM3LT.

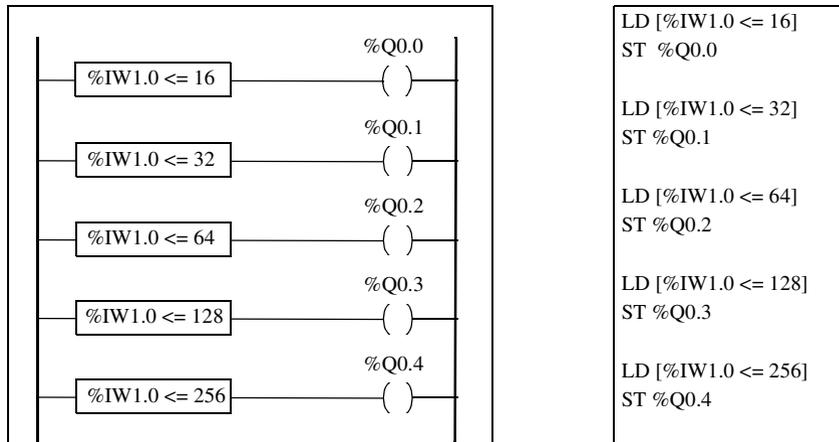
Ejemplo de uso de módulos analógicos

Introducción

En esta sección se ofrece un ejemplo de uso de módulos analógico disponible para Twido.

Ejemplo

En este ejemplo, la señal de entrada analógica se compara con cinco valores de umbral independientes. Se realiza una comparación de la entrada analógica y se ajusta un bit en el controlador base si la entrada es menor que el umbral.



Funcionamiento del monitor de operación



Presentación

Vista general

En este capítulo se ofrece información detallada acerca del uso del monitor de operación Twido opcional.

Contenido:

Este capítulo contiene los siguiente apartados:

Apartado	Página
Monitor de operación	138
Identificación del controlador e información de estado	141
Objetos y variables del sistema	144
Ajustes del puerto serie	151
Reloj de fecha/hora	152
Factor de corrección de tiempo real	153

Monitor de operación

Introducción

El monitor de operación es una opción de Twido que proporciona una interfaz para visualizar y controlar datos de la aplicación y algunas funciones del controlador, como el estado de funcionamiento y el reloj de tiempo real (RTC). Esta opción está disponible como un cartucho (TWDXCPODC) para los controladores compactos o como un módulo de ampliación (TWDXCPODM) para los controladores modulares. El monitor de operación tiene dos modos de funcionamiento:

- Modo de visualización: sólo muestra datos.
- Modo de edición: permite modificar datos.

Nota: El monitor de operación se actualiza en un intervalo específico del ciclo de exploración del controlador. Esto puede provocar confusión al interpretar la pantalla de salidas especializadas para pulsos %PLS o %PWM. En el momento en que se muestran estas salidas, sus valores serán siempre cero y se mostrará este valor. Asegúrese de que la salida especializada real resulte modificada por la configuración del bloque de función.

Visualizaciones y funciones

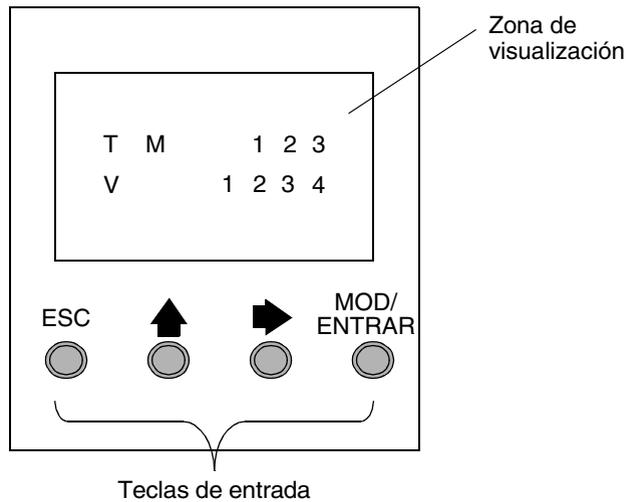
El monitor de operación proporciona las siguientes visualizaciones independientes con las funciones asociadas que puede realizar para cada visualización.

- Identificación del controlador e información de estado
Muestra una revisión del firmware y el estado del controlador. Cambie el estado del controlador con los comandos Ejecutar, Inicial y Detener. Muestra códigos de error en estado Detenido.
- Objetos y variables del sistema
Selecciona los datos de aplicación por la dirección: %I, %Q y el resto de los objetos de software en el controlador base. Controla y modifica el valor de un objeto de datos de software seleccionado.
- Ajustes del puerto serie
Muestra y configura ajustes del puerto de comunicación.
- Reloj de fecha/hora
Muestra y configura la fecha y la hora actuales (si está instalado el RTC).
- Factor de corrección de tiempo real
Muestra y modifica el valor de corrección RTC para el RTC opcional.

Nota: El reloj de fecha/hora y la corrección de tiempo real sólo están disponibles si está instalado el cartucho opcional (TWDXCPRTC) del reloj de tiempo real (RTC).

Ilustración

A continuación se muestra un diagrama simplificado del monitor de operación que se compone de una zona de visualización y cuatro teclas de entrada de botones de comando.

**Zona de visualización**

El monitor de operación proporciona una pantalla LCD capaz de visualizar dos líneas de caracteres.

- La primera línea de la pantalla tiene tres caracteres de 12 segmentos y cuatro caracteres de 7 segmentos.
- La segunda línea tiene un carácter de 13 segmentos, un carácter de 3 segmentos (para un signo más/menos) y cinco caracteres de 7 segmentos.

Teclas de entrada

Las funciones de los cuatro botones de comando de entrada dependen del modo del monitor de operación:

Tecla	En modo de visualización	En modo de edición
ESC		Rechaza los cambios y regresa a la visualización anterior.
▲		Cambia el elemento de edición actual con el valor sucesor.
▶	Avanza hasta la siguiente visualización	Avanza hasta el siguiente elemento de edición.
MOD/ ENTRAR	Va al modo de edición.	Acepta los cambios y regresa a la visualización anterior.

Selección y navegación por las distintas visualizaciones

La visualización o pantalla inicial del monitor de operación muestra la identificación del controlador y la información de estado. Pulse el botón de comando ▶ para avanzar por cada una de las visualizaciones. Las pantallas del reloj de fecha/hora o del factor de corrección del reloj de tiempo real (RTC) no se muestran si no se detecta el cartucho de RTC opcional (TWDXCPRTC) en el controlador. Como método abreviado, pulse la tecla ESC para regresar a la pantalla de visualización inicial. Para la mayoría de las pantallas, al pulsar ESC regresará a la pantalla de identificación del controlador e información de estado. Sólo cuando se editen objetos y variables del sistema que no sean la entrada inicial (%I0.0.0), al pulsar ESC le llevará a la entrada inicial o primera del objeto del sistema. Para modificar un valor del objeto, en lugar de pulsar el botón ▶ de comando para ir al primer dígito de valor, vuelva a pulsar MOD/ENTRAR.

Identificación del controlador e información de estado

Introducción La visualización o pantalla inicial del monitor de operación opcional Twido muestra la identificación del controlador y la información de estado.

Ejemplo La revisión del firmware aparece en la esquina superior derecha de la zona de visualización y el estado del controlador se muestra en la esquina superior izquierda de la zona de visualización, como se muestra a continuación:



Estados del controlador

Los estados del controlador incluyen lo siguiente:

- **NCF: Sin configurar**

El controlador está en estado NCF hasta que se carga una aplicación. No se permite ningún otro estado hasta que se carga un programa de aplicación. Puede comprobar la E/S modificando el bit S8 del sistema (consulte *Bits del sistema (%S)*, p. 336).

- **STP: Detenido**

Cuando hay una aplicación presente en el controlador, el estado cambia a STP o detenido. En este estado, la aplicación no se está ejecutando. Las entradas se actualizan y los datos internos se mantienen con sus últimos valores. Las salidas no se actualizan en este estado.

- **INI: Inicial**

Puede modificar el controlador a estado INI o inicial sólo desde el estado STP. La aplicación no se está ejecutando. Las salidas del controlador se actualizan y los valores de datos se establecen con su estado inicial. Las salidas no se actualizan en este estado.

- **RUN: En ejecución**

Cuando está en estado RUN o en ejecución, la aplicación se está ejecutando. Las salidas del controlador se actualizan y los valores de datos se establecen con arreglo a la aplicación. Éste es el único estado donde se actualizan las salidas.

- **HLT: Detenido (Error de la aplicación de usuario)**

Si el controlador ha introducido un estado ERR o de error, la aplicación se detendrá. Las entradas se actualizan y los valores de datos se detienen con su último valor. Desde este estado, las salidas no se actualizan. En este modo, el código de error se muestra en la zona inferior derecha del monitor de operación como un valor decimal sin signo.

- **NEX: No ejecutable**

Se ha realizado un cambio en línea en la lógica de aplicación que provocó que la aplicación ya no se vuelva a ejecutar. La aplicación del PLC no volverá al estado ejecutable hasta que se hayan resuelto todas las causas para la no ejecución.

Visualización y cambio de estados del controlador

Con el monitor de operación, puede cambiar a estado INI desde el estado STP o desde STP a RUN, o bien desde RUN a STP. Realice lo siguiente para cambiar el estado del controlador:

Paso	Acción
1	Pulse la tecla  hasta que aparezca el monitor de operaciones (o pulse ESC). Los estados del controlador actual aparecen en la esquina superior izquierda de la zona de visualización.
2	Pulse la tecla MOD/ENTRAR para entrar en el modo de edición.
3	Pulse la tecla  para seleccionar un estado del controlador.
4	Pulse la tecla MOD/ENTRAR para aceptar el valor modificado. O bien, pulse la tecla ESC para eliminar las modificaciones realizadas en el modo de edición.

Objetos y variables del sistema

Introducción

El monitor de operación opcional proporciona estas funciones para controlar y ajustar los datos de la aplicación:

- Seleccionar los datos de la aplicación por dirección (como %I o %Q).
 - Controlar el valor de un objeto o una variable del software seleccionado.
 - Cambiar el valor del objeto de datos visualizado actualmente (incluido el forzado de entradas y salidas).
-

Objetos y variables del sistema

La siguiente tabla enumera los objetos y variables del sistema en el mismo orden en que se accede a los mismos, que pueden visualizarse y modificarse mediante el monitor de operación.

Objeto	Variable/Atributo	Descripción	Acceso
Entrada	%I.x.y.z	Valor	Lectura/Forzado
Salida	%Q.x.y.z	Valor	Lectura/Escritura/ Forzado
Temporizador	%TMX.V %TMX.P %TMX.Q	Valor actual Valor preestablecido Hecho	Lectura/Escritura Lectura/Escritura Lectura
Contador	%Cx.V %Cx.P %Cx.D %Cx.E %Cx.F	Valor actual Valor preestablecido Hecho Vacío Completo	Lectura/Escritura Lectura/Escritura Lectura Lectura Lectura
Bit de memoria	%Mx	Valor	Lectura/Escritura
Palabra de memoria	%MWx	Valor	Lectura/Escritura
Palabra constante	%KWx	Valor	Lectura
Bit de sistema	%Sx	Valor	Lectura/Escritura
Palabra de sistema	%SWx	Valor	Lectura/Escritura
Entrada analógica	%IW.x.y.z	Valor	Lectura
Salida analógica	%QW.x.y.z	Valor	Lectura/Escritura
Contador rápido	%FCx.V %FCx.P %FCx.D	Valor actual Valor preestablecido Hecho	Lectura/Escritura Lectura/Escritura Lectura
Contador muy rápido	%VFCx.V %VFCx.P %VFCx.U %VFCx.C %VFCx.S0 %VFCx.S1 %VFCx.F %VFCx.M %VFC.T %VFC.R %VFC.S	Valor actual Valor preestablecido Dirección de conteo Valor rápido Valor de umbral 0 Valor de umbral 1 Desborde Frecuencia realizada Base de tiempo Salida refleja habilitada Entrada refleja habilitada	Lectura/Escritura Lectura/Escritura Lectura Lectura Lectura/Escritura Lectura/Escritura Lectura Lectura/Escritura Lectura/Escritura Lectura/Escritura Lectura/Escritura
Palabra de red de entrada	%INWx.z	Valor	Lectura/Escritura

Objeto	Variable/Atributo	Descripción	Acceso
Palabra de red de salida	%QNWx.z	Valor	Lectura/Escritura
Grafcet	%Xx	Bit de pasos	Lectura
Generador de pulsos	%PLS.N	Número de pulsos	Lectura/Escritura
	%PLS.P	Valor preestablecido	Lectura/Escritura
	%PLS.D	Hecho	Lectura
	%PLS.Q	Salida de corriente	Lectura
Modulador de ancho de pulso	%PMW.R	Ratio	Lectura/Escritura
	%PMW.P	Valor preestablecido	Lectura/Escritura
Controlador del conmutador de tambor	%DRx.S	Número de paso actual completo	Lectura
	%DRx.F		Lectura
Contador de pasos	%SCx.n	Bit del contador de pasos	Lectura/Escritura
Registro	%Rx.I	Entrada	Lectura/Escritura
	%Rx.O	Salida	Lectura
	%Rx.E	Vacío	Lectura
	%Rx.F	Completo	Lectura
Registro de bits de desplazamiento	%SBR.x.yy	Bit de registro	Lectura/Escritura
Mensaje	%MSGx.D	Hecho	Lectura
	%MSGx.E	Error	Lectura

Notas:

1. Las variables no se visualizarán si no se utilizan en una aplicación, ya que Twido utiliza una asignación de memoria dinámica.
2. Si el valor de %MW es superior a +32767 o inferior a -32787, el monitor de operación continuará parpadeando.
3. Si el valor de %SW es superior a 65535, el monitor de operación continúa parpadeando, excepto para %SW0 y %SW11. Si el valor introducido es superior al límite, volverá al valor configurado.
4. Si el valor introducido para %PLS.P es superior al límite, el valor se establecerá como saturación.

Visualización y modificación de objetos y variables

Puede accederse a cada tipo de objeto del sistema comenzando con el objeto de entrada (%I), continuando a través del objeto del mensaje (%MSG) y, finalmente, realizando un bucle de prueba al objeto de entrada (%I).

Para visualizar un objeto del sistema:

Paso	Acción
1	Presionar la tecla  hasta que aparezca la pantalla de visualización de datos. El objeto de entrada ("I") aparecerá en la esquina superior izquierda de la zona de visualización. El carácter "I" (o nombre de objeto anterior) no parpadea.
2	Pulsar la tecla MOD/ENTRAR para entrar en el modo de edición. El carácter "I" del objeto de entrada (o nombre de objeto anterior) comienza a parpadear.
3	Pulsar la tecla  para avanzar de forma secuencial a través de la lista de objetos.
4	Pulsar la tecla  para avanzar de forma secuencial a través del campo de un tipo de objeto y pulsar la tecla  para aumentar el valor de dicho campo. Se pueden utilizar las teclas  y  para navegar y modificar todos los campos del objeto visualizado.
5	Repetir los pasos 3 y 4 hasta que se complete la edición.
6	Pulse la tecla MOD/ENTRAR para aceptar los valores modificados. Nota: Antes de aceptar cualquier modificación, deben validarse el nombre y la dirección del objeto. Es decir, deben existir en la configuración del controlador antes de utilizar el monitor de operación. Pulse ESC para rechazar los cambios realizados en el modo de edición.

Valores de datos y formatos de visualización

En general, el valor de datos de un objeto o variable se muestra como entero con o sin signo en la esquina inferior derecha de la zona de visualización. Además, todos los campos suprimen los ceros no significativos de los valores visualizados. La dirección de cada objeto aparece en el monitor de operación en uno de estos seis formatos:

- Formato de E/S
- Formato del bloque de función
- Formato simple
- Formato de E/S de red
- Formato de contador de pasos
- Formato de registros de bits de desplazamiento

Formato de entrada/salida

Los objetos de entrada/salida (%I, %Q, %IW, and %QW) tienen direcciones de tres partes, como %IX.Y.Z y se visualizan del siguiente modo:

- Tipo de objeto y dirección del controlador en la esquina superior izquierda
- Dirección de ampliación en la parte central superior
- Canal de E/S en la esquina superior derecha

En el caso de una sola entrada (%I) y salida (%Q), el área inferior izquierda de la visualización contendrá un carácter que puede ser "U" para bit no forzado o "F" para forzado. El valor forzado se muestra en la esquina inferior derecha de la pantalla. El objeto de salida %Q0.3.11 aparece en la zona de visualización del siguiente modo:

Q	0	3	1	1
F				1

Formato del bloque de función

Los bloques de función (%TM, %C, %FC, %VFC, %PLS, %PWM, %DR, %R y %MSGj) tienen direcciones de dos partes que contienen un número de objeto y una variable o nombre de atributo y se visualizan del siguiente modo:

- Nombre del bloque de función en la esquina superior izquierda
- Número de bloque de función (o instancia) en la esquina superior derecha
- La variable o el atributo de la esquina inferior izquierda
- Valor del atributo de la esquina inferior derecha

En el siguiente ejemplo, el valor actual del número de temporizador 123 se establece a 1.234.

T	M	1	2	3
V			1	234

Formato simple

Se utiliza un solo formato para los objetos %M, %MW, %KW, %S, %SW y %X del siguiente modo:

- Número de objeto en la esquina superior derecha
- Valor con signo para los objetos en la zona inferior

En el siguiente ejemplo, el número de palabras de memoria 67 contiene el valor +123.

M	W		6	7	
		+	1	2	3

Formato de entrada/salida de red

Los objetos de entrada/salida de red (%INW y %QNW) aparecen en la zona de visualización del siguiente modo:

- Nombre del objeto de la esquina superior izquierda
- Dirección del controlador en la parte central superior
- Número de objeto en la esquina superior derecha
- Valor con signo para el objeto en la zona inferior

En el siguiente ejemplo, la primera entrada o palabra de red del controlador remoto configurado en la dirección remota nº 2 se establece con el valor -4.

M	N	W	2	1
		-		4

Formato de contador de pasos

El formato de contador de pasos (%SC) muestra el número de objeto y el bit del contador de pasos del siguiente modo:

- Nombre y número del objeto en la esquina superior izquierda
- El bit del contador de pasos en la esquina superior derecha
- El valor del objeto en la parte inferior de la visualización

En el siguiente ejemplo, el número 129 de bit del número 3 del contador de pasos se establece a -1.

S	C	3	1	2	9
		-			1

Formato del registro de bits de desplazamiento

El formato del registro de bits de desplazamiento (%SBR) muestra un número de objeto y el bit de registro del siguiente modo:

- Nombre y número del objeto en la esquina superior izquierda
- El bit de registro en la esquina superior derecha

El siguiente ejemplo muestra la visualización del registro de bits de desplazamiento número 4.

S	B	R	4	9
				1

Ajustes del puerto serie

Introducción

Puede visualizar y modificar ajustes de protocolo utilizando el monitor de operación. El número máximo de puertos serie es dos. En el ejemplo que aparece a continuación, el primer puerto se configura como protocolo Modbus con una dirección 123. El segundo puerto serie se configura como conexión remota con una dirección de 5.

M	1 2 3
R	4

Visualización y modificación de ajustes del puerto serie

Los controladores Twido soportan hasta dos puertos serie. Para visualizar los ajustes del puerto serie utilizando el monitor de operación:

Paso	Acción
1	Presionar la tecla  hasta que aparezca la pantalla de comunicación. La única letra del ajuste del protocolo del primer puerto serie ("M", "R" o "A") se visualizará en la esquina superior izquierda del monitor de operación.
2	Pulsar la tecla MOD/ENTRAR para entrar en el modo de edición.
3	Presionar la tecla  hasta que se encuentre en el campo que desea modificar.
4	Al presionar la tecla  se aumenta el valor de dicho campo.
5	Continúe con los pasos 3 y 4 hasta que se completen los ajustes del puerto serie.
6	Presionar la tecla MOD/ENTRAR para aceptar los valores modificados o ESC para rechazar las modificaciones realizadas durante el modo de edición.

Reloj de fecha/hora

Introducción

Puede modificar la fecha y la hora utilizando el monitor de operación si está instalado el cartucho opcional RTC (TWDXCPRTC) en su controlador Twido. El mes se visualiza en el extremo superior izquierdo de la pantalla HMI. Hasta que se haya introducido una hora válida, el campo de mes contendrá el valor "RTC". El día del mes se muestra en la esquina superior derecha de la pantalla. La hora del día aparece en formato militar. Las horas y los minutos se muestran en la esquina inferior derecha de la visualización, separados por la letra "h". El ejemplo que aparece a continuación muestra que el RTC se ha establecido para el 28 de marzo a las 2:22 de la tarde.

M	A	R	2	8
			1	4 h 2 2

Visualización y modificación del reloj de fecha/hora

Para visualizar y modificar el reloj de fecha/hora:

Paso	Acción
1	Presionar la tecla  hasta que aparezca la pantalla de visualización de fecha/hora. El valor del mes ("JAN", "FEB") aparecerá en la esquina superior izquierda de la zona de visualización. Si no se ha inicializado ningún mes, el valor "RTC" se visualizará en la esquina superior izquierda.
2	Pulsar la tecla MOD/ENTRAR para entrar en el modo de edición.
3	Presionar la tecla  hasta que se encuentre en el campo que desea modificar.
4	Al presionar la tecla  se aumenta el valor de dicho campo.
5	Continuar con los pasos 3 y 4 hasta que se haya completado el valor de hora del día.
6	Presionar la tecla MOD/ENTRAR para aceptar los valores modificados o ESC para rechazar las modificaciones realizadas durante el modo de edición.

Factor de corrección de tiempo real

Introducción

Puede visualizar y modificar el factor de corrección del reloj de tiempo real (RTC) utilizando el monitor de operación. Cada módulo opcional de reloj de tiempo real (RTC) tiene un valor de factor de corrección de RTC que se utiliza para corregir imprecisiones en el cristal del módulo RTC. El factor de corrección es un entero de 3 dígitos sin signo de 0 a 127. Este valor se muestra en la esquina inferior derecha de la pantalla.

El ejemplo que aparece a continuación muestra un factor de corrección de 127.

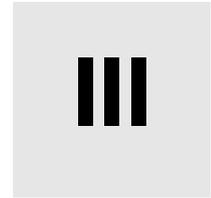
R T C	C o r r
	1 2 7

Visualización y modificación de la corrección RTC

Para visualizar y modificar el factor de corrección de tiempo real, siga estos pasos.

Paso	Acción
1	Presionar la tecla ▶ hasta que aparezca la pantalla del factor RTC. Aparecerá "RTC Corr" en la línea superior del monitor de operación.
2	Pulsar la tecla MOD/ENTRAR para entrar en el modo de edición.
3	Presionar la tecla ▶ hasta que se encuentre en el campo que desea modificar.
4	Al presionar la tecla ▲ se aumenta el valor de dicho campo.
5	Continuar con los pasos 3 y 4 hasta que se haya completado el valor de corrección RTC.
6	Presionar la tecla MOD/ENTRAR para aceptar los valores modificados o ESC para rechazar las modificaciones realizadas durante el modo de edición.

Descripción de lenguajes de Twido



Presentación

Vista general

Esta parte proporciona instrucciones para utilizar los lenguajes de programación Ladder Logic, de lista y Grafcet para crear programas de control para controladores programables Twido.

Contenido

Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
9	Lenguaje Ladder Logic	157
10	Lenguaje de lista de instrucciones	181
11	Grafcet	193

Lenguaje Ladder Logic

9

Presentación

Vista general Este capítulo describe la programación con el lenguaje Ladder Logic.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Introducción a los diagramas Ladder Logic	158
Principios de programación para diagramas Ladder Logic	160
Bloque de diagramas Ladder Logic	162
Elementos gráficos del lenguaje Ladder Logic	165
Instrucciones Ladder Logic especiales OPEN y SHORT	168
Consejos sobre programación	169
Reversibilidad de Ladder Logic/Lista	174
Directrices para la reversibilidad Ladder Logic/Lista	176
Documentación del programa	178

Introducción a los diagramas Ladder Logic

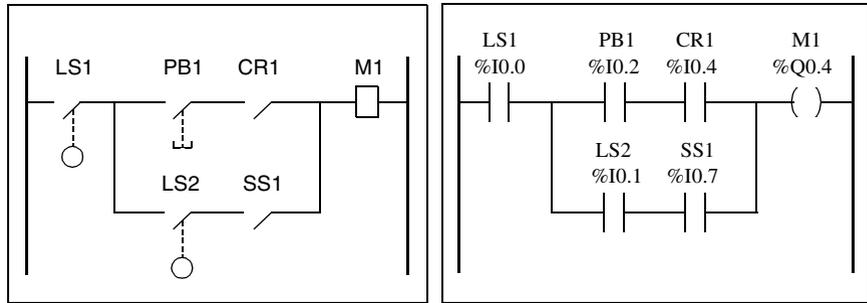
Introducción

Los diagramas Ladder Logic son similares a los diagramas de lógica de relé que representan circuitos de control de relé. Las principales diferencias entre los dos son las siguientes funciones de la programación de Ladder Logic que no aparecen en los diagramas de lógica de relé:

- Todas las entradas están representadas por símbolos de contactos (—|—).
- Todas las salidas están representadas por símbolos de bobinas (—|—).
- Las operaciones numéricas están incluidas en el conjunto de instrucciones de Ladder Logic gráfico.

Equivalentes Ladder Logic a los circuitos de relé

La siguiente ilustración muestra un diagrama simplificado del cableado de un circuito de lógica de relé y el diagrama Ladder Logic equivalente.



Circuito de lógica de relé

Diagrama Ladder Logic

Observe en la ilustración anterior que todas las entradas asociadas al dispositivo de conmutación en el diagrama de lógica de relé aparecen como contactos en el diagrama Ladder Logic. La bobina de salida M1 del diagrama de lógica de relé se representa con un símbolo de bobina de salida en el diagrama Ladder Logic. Los números de dirección que aparecen sobre cada uno de los símbolos de contactos o bobinas en el diagrama Ladder Logic hacen referencia a la posición que ocupan las conexiones de entrada/salida con el controlador.

Escalones Ladder Logic

Un programa escrito en lenguaje Ladder Logic está compuesto por escalones, que son conjuntos de instrucciones gráficas dibujadas entre dos barras verticales de potencia. El controlador ejecuta los escalones secuencialmente.

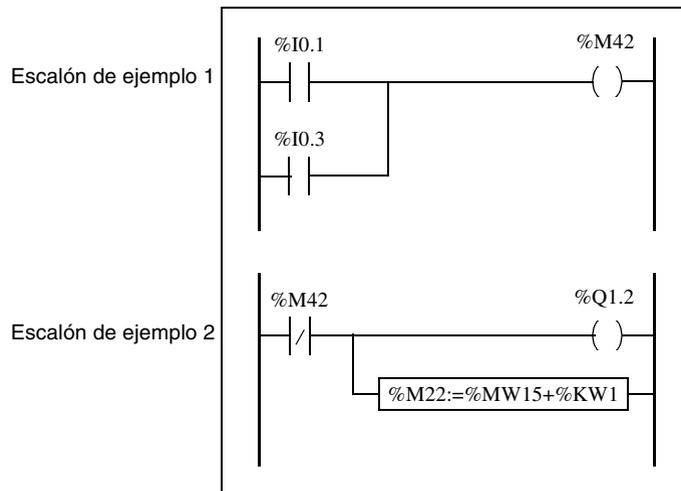
El conjunto de instrucciones gráficas representan las siguientes funciones:

- Entradas/salidas del controlador (*, sensores, relés, luces de pilotos...)
- Funciones del controlador (temporizadores, contadores...)
- Operaciones lógicas y matemáticas (adición, división, AND, XOR...)
- Operadores de comparación y otras operaciones numéricas ($A < B$, $A = B$, desplazamiento, rotación...)
- Variables internas del controlador (bits, palabras...)

Estas instrucciones gráficas se organizan con conexiones horizontales y verticales que eventualmente llevan a una o varias salidas o acciones. Una red no puede admitir más de un grupo de instrucciones vinculadas.

Ejemplo de escalones Ladder Logic

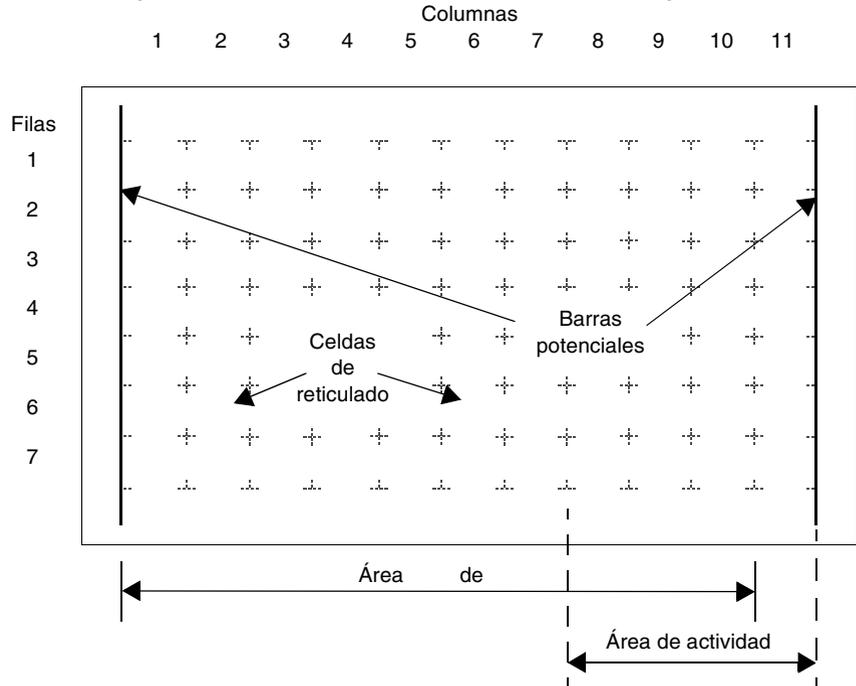
El siguiente diagrama es un ejemplo de un programa Ladder Logic compuesto por dos escalones.



Principios de programación para diagramas Ladder Logic

Reticulado de programación

Cada escalón de Ladder Logic se compone de un reticulado de siete filas por once columnas organizadas en dos áreas como se muestra en la siguiente ilustración.



Áreas del reticulado

El reticulado de programación del diagrama Ladder Logic está dividido en dos áreas:

- **Área de prueba**
Contiene las condiciones que se han de probar a fin de realizar acciones. Está formada por las columnas 1 a 10 y contiene contactos, bloques de función y bloques de comparación.
- **Área de actividad**
Contiene la salida u operación que será realizada según sean los resultados de las pruebas llevadas a cabo en el área de prueba. Está formada por las columnas 8 a 11 y contiene bobinas y bloques de operación.

Introducción de instrucciones en el reticulado

Un escalón de Ladder Logic proporciona un reticulado de programación de siete por once que comienza en la primera celda de la esquina superior izquierda del reticulado. La programación consiste en introducir instrucciones en las celdas del reticulado. Las funciones, comparaciones e instrucciones de prueba se introducen en celdas en el área de prueba y se alinean a la izquierda. La lógica de prueba proporciona continuidad al área de actividad donde se introducen bobinas, operaciones numéricas e instrucciones de flujo del programa y se justifican a la derecha.

El escalón se soluciona o ejecuta (pruebas realizadas y salidas asignadas) dentro del reticulado de arriba a abajo y de izquierda a derecha.

Cabeceras de escalón

Además del escalón, una cabecera de escalón aparece directamente por encima del escalón. Utilice la cabecera de escalón para documentar el propósito lógico del escalón. La cabecera de escalón puede contener la siguiente información:

- Número de escalón
- Etiquetas (%Li)
- Declaraciones de subrutina (SRi:)
- Título de escalón
- Comentarios de escalón

Para obtener más información acerca de la utilización de la cabecera de escalón para documentar los programas, consulte *Documentación del programa*, p. 178.

Bloque de diagramas Ladder Logic

Introducción

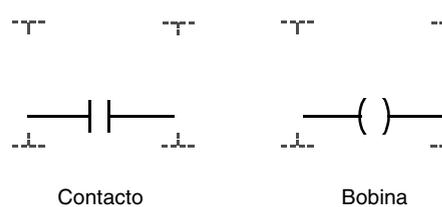
Los diagramas Ladder Logic están compuestos por bloques que representan el flujo de programas y las funciones, por ejemplo:

- Contactos
- Bobinas
- Instrucciones de flujo de programas
- Bloques de función
- Bloques de comparación
- Bloques de operación

Contactos, bobinas y flujo de programas

Las instrucciones contactos, bobinas y flujo de programas (saltar y llamadas) ocupan una única celda en el reticulado de programación de Ladder Logic. Los bloques de función, comparación y operación ocupan varias.

A continuación se muestran ejemplos de contactos y bobinas.

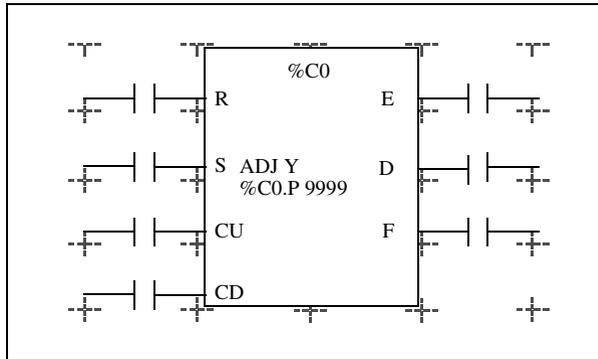


Bloques de función

Los bloques de función están ubicados en el área de prueba del reticulado de programación. El bloque debe aparecer en la primera fila; no deberían aparecer instrucciones Ladder Logic ni líneas de continuidad por encima ni por debajo del bloque de función. Las instrucciones de prueba Ladder Logic llevan al lateral de entrada del bloque y las instrucciones de prueba o acción llevan al lateral de salida del bloque.

Los bloques de función están colocados en vertical y ocupan dos columnas en cuatro filas del reticulado de programación.

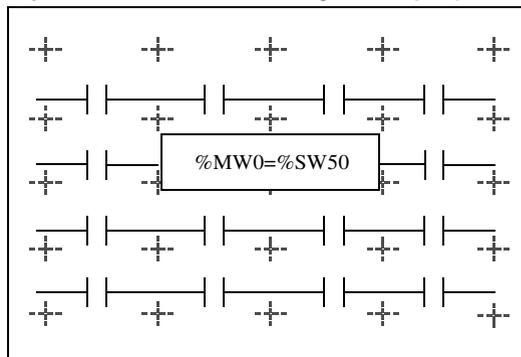
A continuación se muestra un ejemplo de un bloque de función del contador.



Bloques de comparación

Los bloques de comparación están ubicados en el área de prueba del reticulado de programación. El bloque puede aparecer en cualquier fila o columna del área de prueba siempre que la longitud completa de la instrucción esté en esta área.

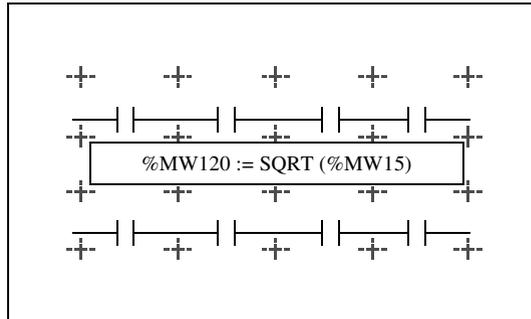
Los bloques de comparación están colocados en horizontal y ocupan dos columnas en una fila del reticulado de programación, tal y como se muestra en el siguiente ejemplo de bloque de comparación.



Bloques de operación

Los bloques de operación están ubicados en el área de acción del reticulado de programación. El bloque puede aparecer en cualquier fila del área de acción. La instrucción está justificada a la derecha . Aparece en la derecha y termina en la última columna.

Los bloques de operación están colocados en horizontal y ocupan cuatro columnas en una fila del reticulado de programación, a continuación se muestra un ejemplo de un bloque de operación.



Elementos gráficos del lenguaje Ladder Logic

Introducción

Las instrucciones de los diagramas Ladder Logic se componen de elementos gráficos. Esta sección enumera y describe los elementos gráficos utilizados en las instrucciones Ladder de Twido. Si desea más información acerca de cómo utilizar estos elementos gráficos en programas Ladder de Twido, consulte el Manual de funcionamiento de TwidoSoft.

Contactos

Los elementos gráficos de los contactos se programan en el área de prueba y ocupan una celda (el alto de una fila por el ancho de una columna).

Nombre	Elemento gráfico	Instrucción	Función
Contacto normal abierto		LD	Establece contacto cuando el objeto de bit de control está en estado 1.
Contacto normal cerrado		LDN	Establece contacto cuando el objeto de bit de control está en estado 0.
Contacto para detectar un flanco ascendente		LDR	Flanco ascendente: detecta el cambio de 0 a 1 del objeto de bit de control.
Contacto para detectar un flanco descendente		LDF	Flanco descendente: detecta el cambio de 1 a 0 del objeto de bit de control.

Elementos de conexión

Los elementos gráficos de conexión se utilizan para conectar los elementos gráficos de acción y de prueba.

Nombre	Elemento gráfico	Funciones
Conector horizontal		Conecta en serie los elementos gráficos de prueba y acción entre dos barras potenciales.
Conector inferior		Conecta los elementos de prueba y acción de forma paralela (conexión vertical).

Bobinas

Los elementos gráficos de bobina se programan en el área de acción y ocupan una celda (el alto de una fila por el ancho de una columna).

Nombre	Elemento gráfico	Instrucción	Funciones
Bobina directa		ST	El objeto de bit asociado toma el valor del resultado del área de prueba.
Bobina negada		STN	El objeto de bit asociado toma el valor negado del resultado del área de prueba.
Establecer bobina		S	El objeto de bit asociado se establece en 1 cuando el resultado del área de prueba es 1.
Restablecer bobina		R	El objeto de bit asociado se establece en 0 cuando el resultado del área de prueba es 1.
Llamada de salto o subrutina	->>%Li ->>%SRi	JMP SR	Se conecta a una instrucción etiquetada ubicada delante o detrás.
Bobina de condición de transición			Proporcionado en lenguaje Grafcet, se utiliza cuando la programación de las condiciones de transición asociadas a las transiciones provoca una inversión de corriente en el siguiente paso.
Retorno desde una subrutina	<RET>	RET	Situado al final de las subrutinas para regresar al programa principal.
Detener programa	<END>	END	Define el final del programa.

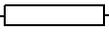
Bloques de función

Los elementos gráficos de los bloques de función se programan en la misma área de prueba y requieren cuatro filas y dos columnas de celdas (excepto para contadores muy rápidos que requieren cinco filas y dos columnas).

Nombre	Elemento gráfico	Funciones
Temporizadores, contadores, registros, etc.		Cada bloque de función utiliza entradas y salidas que permiten conexiones a otros elementos gráficos.. Nota: Las salidas de los bloques de función no pueden conectarse entre sí (conexiones verticales).

Bloques de operación y comparación

Los bloques de comparación se programan en el área de prueba, mientras que los de operación lo hacen en el área de acción.

Nombre	Elemento gráfico	Funciones
Bloque de comparación		Compara dos operandos y la salida cambia a 1 cuando se comprueba el resultado. Tamaño: una fila por dos columnas
Bloque de operación		Realiza operaciones aritméticas y lógicas. Tamaño: una fila por cuatro columnas

Instrucciones Ladder Logic especiales OPEN y SHORT

Introducción

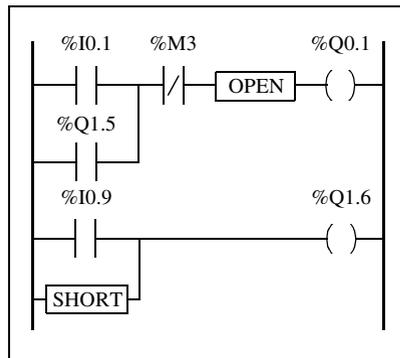
Las instrucciones OPEN y SHORT proporcionan un método apropiado para depurar y solucionar posibles problemas en los programas Ladder. Estas instrucciones especiales alteran la lógica de un escalón, ya sea acortando o abriendo la continuidad de un escalón tal y como se explica en la siguiente tabla.

Instrucción	Descripción	Instrucción de lista
OPEN	Establece una interrupción en la continuidad de un escalón de Ladder Logic sin tener en cuenta los resultados de la última operación lógica.	AND 0
SHORT	Permite la continuidad a través del escalón sin tener en cuenta los resultados de la última operación lógica.	OR 1

En la programación de Lista, las instrucciones OR y AND se utilizan para crear las instrucciones OPEN y SHORT utilizando valores inmediatos de 0 y 1 respectivamente.

Ejemplos

A continuación se muestran ejemplos de uso de las instrucciones OPEN y SHORT.



```

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

Consejos sobre programación

Tratamiento de los saltos de programa

Utilice los saltos de programa con precaución para evitar bucles largos que prolonguen el tiempo de ciclo. Evite los saltos en las instrucciones ubicadas "detrás" (una instrucción ubicada detrás aparece justo antes de un salto en un programa, mientras que una instrucción ubicada delante aparece después del salto).

Programación de salidas

Un bit de salida o un bit interno sólo se pueden controlar una vez en el programa. En el caso de los bits de salida, sólo se tiene en cuenta el último valor examinado cuando se actualizan las salidas.

Utilización de los sensores de parada de emergencia de cableado directo

Los sensores utilizados directamente para paradas de emergencia no deben ser procesados por el controlador. Se deben conectar directamente a las salidas correspondientes.

Tratamiento de recuperación de la alimentación

Haga que la recuperación de la alimentación dependa de una operación manual, ya que un reinicio automático de la instalación podría provocar un funcionamiento inesperado del equipo (utilice los bits del sistema %S0, %S1 y %S9).

Gestión de los fechadores y del bloque de tiempo

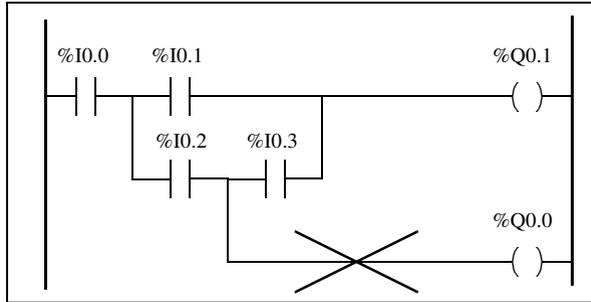
El bit de estado del sistema %S1, que indica cualquier fallo en el fechador, debe ser comprobado.

Comprobación de errores y sintaxis

Cuando se introduce un programa, TwidoSoft comprueba la sintaxis de instrucciones, operandos y sus asociaciones. Para obtener más detalles consulte el manual de funcionamiento de TwidoSoft.

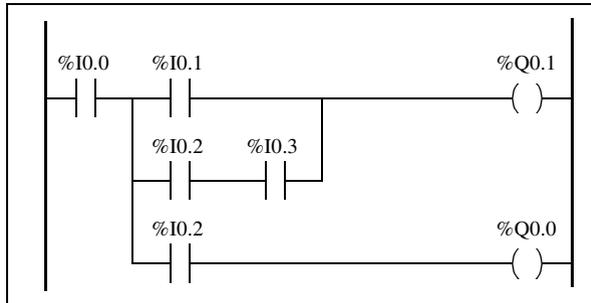
**Notas
adicionales
sobre cómo
utilizar los
paréntesis**

Las operaciones de asignación nunca deben aparecer entre paréntesis.



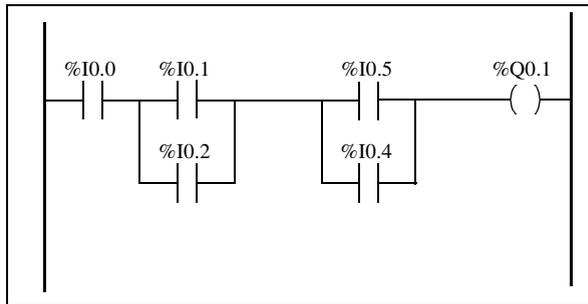
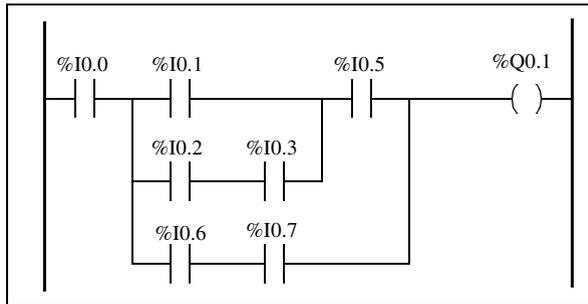
```
LD    %I0.0
AND   %I0.1
OR(   %I0.2
ST  %Q0.0
AND %I0.3
)
ST    %Q0.1
```

Para realizar la misma función, se deben programar las siguientes ecuaciones.

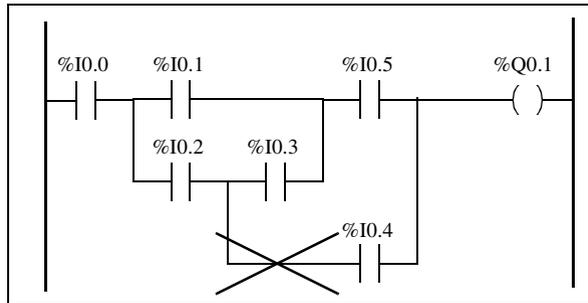
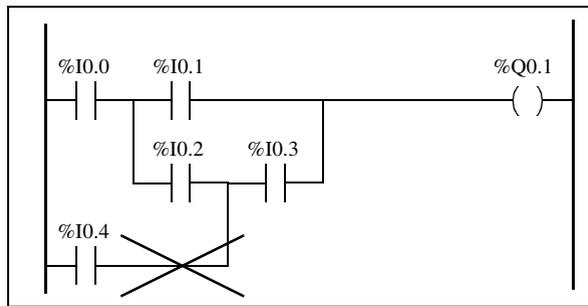


```
LD    %I0.0
MPS
AND(  %I0.1
OR(   %I0.2
AND   %I0.3
)
)
ST    %Q0.1
MPP
AND   %I0.2
ST    %Q0.0
```

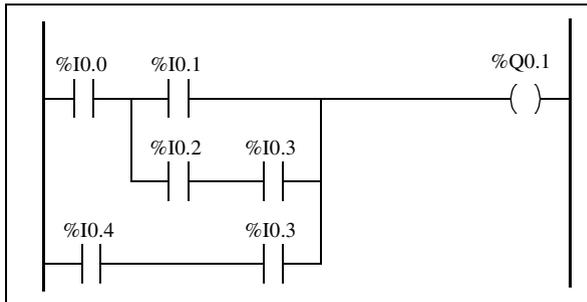
Si existen varios contactos colocados en paralelo, deben estar anidados uno dentro de otro o bien estar totalmente disociados.



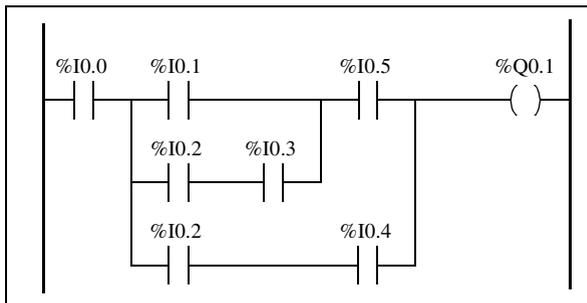
No se puede programar el siguiente esquema.



Para ejecutar esquemas equivalentes a estos, se deben modificar como se especifica a continuación.



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



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

Reversibilidad de Ladder Logic/Lista

Introducción

La reversibilidad de programa es una función del software de programación TwidoSoft que facilita la conversión de programas de aplicación de Ladder Logic a Lista y viceversa.

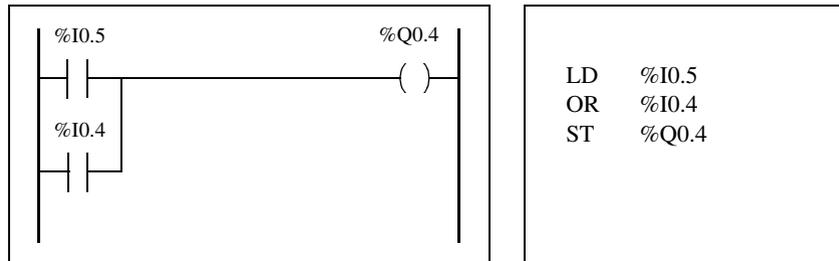
Utilice TwidoSoft para establecer la visualización predeterminada de programas tanto para formato Lista como para formato Ladder Logic (establecido según las preferencias del usuario) y para alternar las vistas de Lista y de Ladder Logic (si desea más información, consulte el Manual de funcionamiento de TwidoSoft).

Comprender la reversibilidad

Para llegar a comprender la función reversibilidad de programas, es necesario examinar la relación de un escalón de Ladder Logic con la secuencia de Lista de instrucción asociada.

- **Escalón de Ladder Logic:** conjunto de instrucciones Ladder Logic que constituyen una expresión lógica.
- **Secuencia de Lista:** conjunto de instrucciones de programación de Lista que corresponde a las instrucciones Ladder Logic y representa la misma expresión lógica.

La siguiente ilustración muestra un escalón de Ladder normal y el equivalente de la lógica de programación expresado como una secuencia de instrucciones de Lista.



Un programa de aplicación se guarda internamente como instrucciones de Lista, sin importar si el programa está escrito en lenguaje Ladder Logic o Lista. TwidoSoft aprovecha las similitudes de la estructura de programa entre ambos lenguajes y utiliza esta imagen de Lista interna del programa para mostrarla en los visualizadores y editores de Ladder Logic y de Lista como un programa de Lista (su forma básica) o gráficamente como diagrama Ladder Logic, según lo que desee el usuario.

**Asegurar la
reversibilidad**

Los programas creados en Ladder Logic siempre se pueden invertir a Lista, pero puede darse el caso que parte de la lógica de Lista no pueda invertirse a Ladder Logic. Para asegurar la reversibilidad de Lista a Ladder Logic, es importante seguir una serie de directrices de programación de Lista que aparecen en "*Directrices para la reversibilidad Ladder Logic/Lista, p. 176*".

Directrices para la reversibilidad Ladder Logic/Lista

Instrucciones básicas para la reversibilidad

Las instrucciones que aparecen a continuación son necesarias para la estructura de un bloque de función reversible en lenguaje de lista.

- **BLK** indica el comienzo del bloque y define el inicio del escalón y de la parte de entrada al bloque.
- **OUT_BLK** indica el comienzo de la parte de salida del bloque.
- **END_BLK** indica el final del bloque y del escalón.

El uso de las instrucciones del bloque de función reversible no es obligatorio cuando el programa de lista funciona correctamente. Es posible programar en lista algunas instrucciones, lo cual no es reversible. Si desea más información acerca de la programación de lista no reversible de bloques de función estándar, consulte "*Principios para programar bloques de función, p. 228*".

Instrucciones no equivalentes que se deben evitar

Evite utilizar determinadas instrucciones de lista o determinadas combinaciones de instrucciones y operandos, ya que no tienen ningún equivalente en diagramas Ladder Logic. Por ejemplo, la instrucción N (invierte el valor en el acumulador booleano) no tiene una instrucción Ladder Logic equivalente.

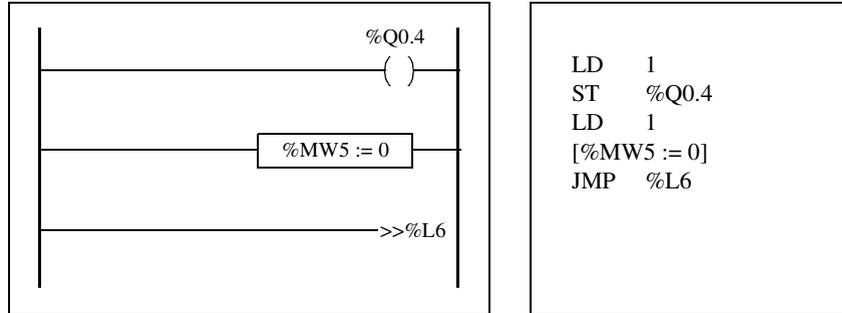
La tabla que aparece a continuación enumera todas las instrucciones de programación de lista que no se pueden invertir a Ladder Logic.

Instrucción de lista	Operando	Descripción
JMPCN	%Li	NOT condicional de salto
N	ninguno	Negación (NOT)
ENDCN	ninguno	NOT condicional de finalización

Escalones incondicionales

Programar escalones incondicionales también necesita las siguientes directrices de programación de lista para asegurar la reversibilidad de lista a Ladder Logic. Los escalones incondicionales no tienen pruebas ni condiciones, las instrucciones de acción o de salidas se cargan o se ejecutan.

El diagrama que aparece a continuación muestra ejemplos de escalones incondicionales y la secuencia de lista equivalente.



Tenga en cuenta que cada una de las secuencias de lista incondicionales mencionadas con anterioridad comienza con una instrucción de carga seguida de un 1, excepto la instrucción JMP. Esta combinación ajusta el valor del acumulador booleano a uno, de tal manera que pone la bobina a 1 (instrucción de almacenamiento) y %MW5 a cero cada vez que se ejecuta el programa. La excepción es la instrucción de lista incondicional JUMP (JMP %L6) que se ejecuta independientemente del valor del acumulador. No es necesario poner el acumulador a 1.

Escalón de lista Ladder Logic

Si un programa de lista que no es totalmente reversible se invierte, las partes reversibles se muestran en la vista Ladder Logic y las partes que no se pueden invertir aparecen en los escalones de lista Ladder Logic.

Un escalón de lista Ladder Logic funciona como un editor de lista pequeño, permitiendo al usuario visualizar y modificar las partes del programa Ladder Logic que no se pueden invertir.

Documentación del programa

Documentación del programa

Puede documentar el programa introduciendo comentarios a través de los editores de Lista o de Ladder Logic (si desea más información acerca del uso de estos editores de programas, consulte el Manual de funcionamiento de TwidoSoft).

- Utilice el editor de Lista para documentar el programa con los comentarios de la línea de lista. Estos comentarios pueden aparecer en la misma línea que las instrucciones de programación o en sus propias líneas.
- Utilice el editor de Ladder Logic para documentar el programa utilizando cabeceras de escalón encontradas directamente por encima del escalón.

El software de programación de TwidoSoft utiliza estos comentarios como reversibilidad. Cuando se invierte un programa de lista a Ladder, TwidoSoft utiliza algunos de los comentarios de lista para crear una cabecera de escalón y los comentarios insertados entre las secuencias de lista se utilizan para las cabeceras de escalón.

Ejemplo de comentarios de la línea de lista

A continuación se muestra un ejemplo de un programa de lista con los comentarios de la línea de lista.

```

---- (* ÉSTE ES EL TÍTULO DE CABECERA PARA EL ESCALÓN 0 *)
---- (* ÉSTE ES EL PRIMER COMENTARIO DE CABECERA PARA EL ESCALÓN 0 *)
---- (* ÉSTE ES EL SEGUNDO COMENTARIO DE CABECERA PARA EL ESCALÓN 0 *)
0 LD %I0.0 (* ESTO ES UN COMENTARIO DE LÍNEA *)
1 OR %I0.1 (* SE IGNORA UN COMENTARIO DE LÍNEA CUANDO SE INVIERTE A
LADDER LOGIC *)
2 ANDM %M10
3 ST M101
---- (* ÉSTA ES LA CABECERA PARA EL ESCALÓN 1 *)
---- (* ESTE ESCALÓN CONTIENE UNA ETIQUETA *)
---- (* ÉSTE ES EL SEGUNDO COMENTARIO DE CABECERA PARA EL ESCALÓN 1 *)
---- (* ÉSTE ES EL TERCER COMENTARIO DE CABECERA PARA EL ESCALÓN 1 *)
---- (* ÉSTE ES EL CUARTO COMENTARIO DE CABECERA PARA EL ESCALÓN 1 *)
4 %L5:
5 LD %M101
6 [ %MW20 := %KW2 * 16 ]
---- (* ESTE ESCALÓN SÓLO CONTIENE UN TÍTULO DE CABECERA *)
7 LD %Q0.5
8 OR %I0.3
9 ORR I0.13

```

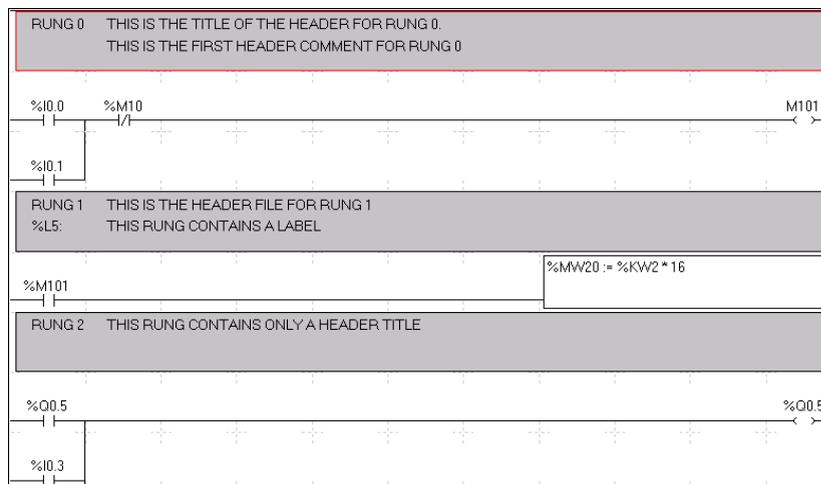
Inversión de comentarios de lista a Ladder Logic

Cuando se invierten las instrucciones de lista en un diagrama de Ladder Logic, los comentarios de la línea de lista aparecen en el editor de Ladder Logic con arreglo a las siguientes normas:

- El primer comentario que aparece en una línea por sí solo se asigna a la cabecera de escalón.
- Cualquier comentario encontrado después del primero se convierte en el cuerpo del escalón.
- Una vez ocupadas las líneas del cuerpo de la cabecera, el resto de los comentarios de línea entre las secuencias de lista se ignoran, igual que cualquier otro comentario que aparezca en líneas de la lista que también contengan instrucciones de lista.

Ejemplo de comentarios de cabecera de escalón

A continuación se muestra un ejemplo de un programa de Ladder Logic con los comentarios de cabecera de escalón.



Inversión de comentarios de Ladder Logic a lista

Cuando se invierte un diagrama de Ladder Logic a instrucciones de lista, los comentarios de la cabecera de escalón aparecen en el editor de Ladder Logic con arreglo a las siguientes normas:

- Cualquier comentario de la cabecera de escalón se inserta entre las secuencias de lista asociadas.
 - Cualquier etiqueta (%Li:) o declaraciones de subrutina (SRi:) se encuentran en la siguiente línea siguiendo a la cabecera e inmediatamente antes del comienzo de la secuencia de lista.
 - Si la lista se invirtió a Ladder Logic, los comentarios ignorados volverán a aparecer en el editor de lista.
-

Lenguaje de lista de instrucciones

10

Presentación

Vista general Este capítulo describe la programación en el lenguaje de lista de instrucciones.

Contenido: Este capítulo contiene los siguiente apartados:

Apartado	Página
Vista general de programas de lista	182
Operación de las instrucciones de lista	184
Instrucciones del lenguaje de lista	185
Utilización de paréntesis	189
Instrucciones de stack (MPS, MRD, MPP)	191

Vista general de programas de lista

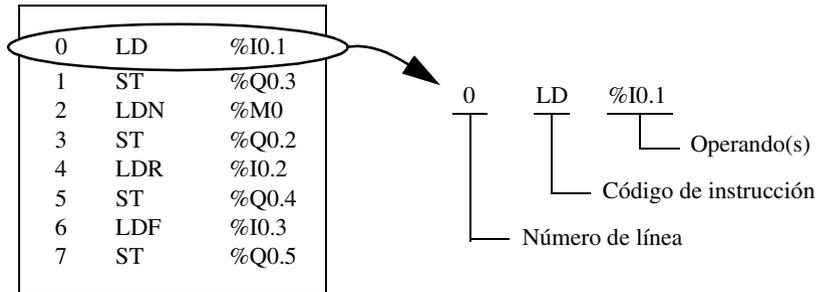
Introducción

Un programa escrito en lenguaje de lista está formado por una serie de instrucciones que el controlador ejecuta de forma secuencial. Cada instrucción de lista está representada por una línea de programa y tiene tres componentes:

- Número de línea
- Código de instrucción
- Operando(s)

Ejemplo de un programa de lista

A continuación se muestra un ejemplo de un programa de lista.



Número de línea

Los números de línea se generan automáticamente al introducir una instrucción. Las líneas vacías y las líneas de comentario no tienen números de línea.

Código de instrucción

El código de instrucción es un símbolo para un operador que identifica la operación que se va a realizar utilizando los operandos. Los operadores típicos especifican operaciones numéricas y booleanas.

Por ejemplo, en el programa de ejemplo anterior, LD es la abreviatura del código de instrucción para una instrucción LOAD. La instrucción LOAD coloca (carga) el valor del operando %I0.1 en un registro interno llamado el acumulador.

Hay dos tipos de instrucciones básicas:

- Instrucciones de prueba
Estas instrucciones configuran o comprueban las condiciones necesarias para realizar una acción. Por ejemplo, LOAD (LD) y AND.
- Instrucciones de acción
Estas instrucciones realizan acciones como resultado de las condiciones configuradas. Por ejemplo, instrucciones de asignación como STORE (ST) y RESET (R).

Operando

Un operando es un número, dirección o símbolo que representa un valor que puede manipular un programa en una instrucción. Por ejemplo, en el programa de ejemplo anterior, el operando %I0.1 es una dirección que tiene asignado el valor de una entrada del controlador. Una instrucción puede tener de cero a tres operandos dependiendo del tipo de código de instrucción.

Los operandos pueden representar los siguientes elementos:

- Entradas y salidas del controlador, como sensores, botones y relés.
 - Funciones de sistema predefinidas, como temporizadores y contadores.
 - Operaciones aritméticas, numéricas y de comparación.
 - Variables internas del controlador, como bits y palabras.
-

Operación de las instrucciones de lista

Introducción

Las instrucciones de lista tienen un único operando explícito, el otro está implícito. El operando implícito es el valor del acumulador booleario. Por ejemplo, en la instrucción LD %I0.1, %I0.1 es el operando explícito. El operando implícito se almacenará en el acumulador y se sobrescribirá con el valor de %I0.1.

Operación

Una instrucción de lista realiza una operación específica en el contenido del acumulador y el operando explícito, y sustituye el contenido del acumulador con el resultado. Por ejemplo, la operación AND %I1.2 realiza la instrucción AND lógica entre el contenido del acumulador y la salida 1.2, y sustituye el contenido del acumulador con este resultado.

Todas las instrucciones boolearias, excepto Carga, Almacenamiento y No, funcionan con dos operandos. El valor de los operandos puede ser True o False, y la ejecución de programa de las instrucciones produce un valor único, True o False. Las instrucciones de carga colocan el valor del operando en el acumulador mientras las de almacenamiento transfieren el valor del acumulador al operando. La instrucción No no tiene operandos explícitos, así que simplemente invierte el estado del acumulador.

Instrucciones de lista admitidas

La tabla que aparece a continuación es un resumen de los tipos de instrucciones de lista admitidas.

Tipo de instrucción	Ejemplo	Función
Instrucción sobre bit	LD %M10	Lee el bit %M10 interno
Instrucción sobre bloque	IN %TMO	Inicia el contador %TMO
Instrucción de palabra	[%MW10 := %MW50+100]	Operación de adición
Instrucciones del programa	SR5	Llama a subrutina nº 5
Instrucción Grafcet	-*8	Paso nº 8

Instrucciones del lenguaje de lista

Introducción

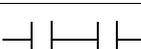
Un lenguaje de lista se compone de los siguientes tipos de instrucciones:

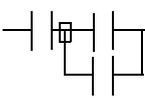
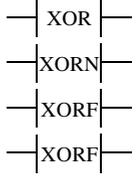
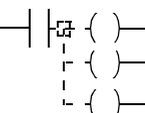
- Instrucciones de prueba
- Instrucciones de acción
- Instrucciones sobre bloques de función

Esta sección identifica y describe las instrucciones Twido para la programación de listas.

Instrucciones de prueba

La siguiente tabla describe las instrucciones de prueba en lenguaje de listas.

Nombre	Elemento gráfico equivalente	Función
LD		El resultado booleario es el mismo que el estado del operando.
LDN		El resultado booleario es el mismo que el estado inverso del operando.
LDR		El resultado booleario cambia a 1 durante la detección del operando (flanco ascendente) que cambia de 0 a 1.
LDF		El resultado booleario cambia a 1 durante la detección del operando (flanco descendente) que cambia de 1 a 0.
AND		El resultado booleario es igual a la instrucción lógica AND entre el resultado booleario de la instrucción anterior y el estado del operando.
ANDN		El resultado booleario es igual a la instrucción lógica AND entre el resultado booleario de la instrucción anterior y el estado inverso del operando.
ANDR		El resultado booleario es igual a la instrucción lógica AND entre el resultado booleario de la instrucción anterior y el flanco ascendente del operando (1 = flanco ascendente).
ANDF		El resultado booleario es igual a la instrucción lógica AND entre el resultado booleario de la instrucción anterior y la detección del flanco descendente del operando (1 = flanco descendente).
OR		El resultado booleario es igual a la instrucción lógica OR entre el resultado booleario de la instrucción anterior y el estado del operando.

Nombre	Elemento gráfico equivalente	Función
AND(	Instrucción lógica AND (8 niveles de paréntesis)
OR(	Instrucción lógica OR (8 niveles de paréntesis)
XOR, XORN, XORR, XORF		OR exclusivo
MPS MRD MPP		Conmutación a las bobinas.
N	-	Negación (NOT)

Instrucciones de acción

La siguiente tabla describe las instrucciones de acción en lenguaje de listas.

Nombre	Elemento gráfico equivalente	Funciones
ST	-()-	El operando asociado toma el valor del resultado del área de prueba.
STN	-(/)-	El operando asociado toma el valor inverso del resultado del área de prueba.
S	-(S)-	El operando asociado se establece en 1 cuando el resultado del área de prueba es 1.
R	-(R)-	El operando asociado se establece en 0 cuando el resultado del área de prueba es 1.
JMP	->>%Li	Se conecta de forma incondicional a una secuencia etiquetada ubicada delante o detrás.
SRn	->>%SRi	Conexión al comienzo de una subrutina.
RET	<RET>	Retorno desde una subrutina.
END	<END>	Fin del programa.
ENDC	<ENDC>	Fin del programa condicionado en un resultado booleario de 1.
ENDCN	<ENDCN>	Fin del programa condicionado en un resultado booleario de 0.

Instrucciones sobre bloques de función

La siguiente tabla describe los bloques de función en lenguaje de listas.

Nombre	Elemento gráfico equivalente	Funciones
Temporizadores, contadores, registros, etc.		<p>Para cada bloque de función existen instrucciones para controlar el bloque.</p> <p>Para cablear las entradas y salidas de bloques directamente se utiliza una forma estructurada.</p> <p>Nota: Las salidas de los bloques de función no pueden conectarse entre sí (conexiones verticales).</p>

Utilización de paréntesis

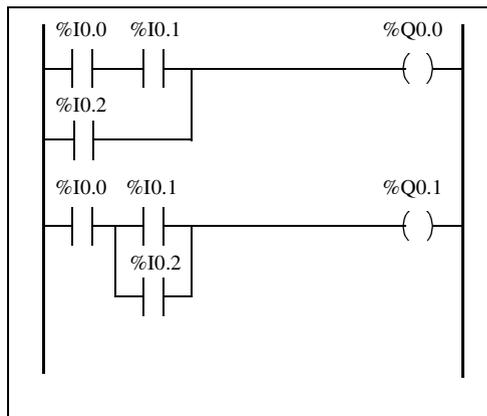
Introducción

Utilice paréntesis con las instrucciones lógicas AND y OR para indicar bifurcaciones paralelas en los diagramas Ladder Logic. Los paréntesis de apertura y cierre se asocian con instrucciones tal y como se indica a continuación:

- El paréntesis de apertura se asocia con la instrucción AND u OR.
- El paréntesis de cierre es una instrucción necesaria siempre que se haya abierto un paréntesis.

Ejemplo de uso de una instrucción AND

Los siguientes diagramas son ejemplos de utilización de paréntesis con una instrucción AND: AND(...).

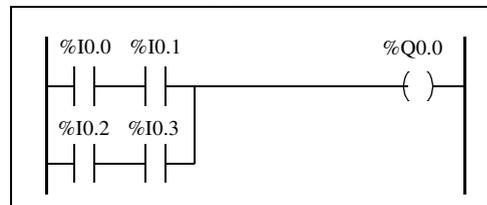


```
LD    %I0.0
AND   %I0.1
OR    %I0.2
ST    %Q0.0
```

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

Ejemplo de uso de una instrucción OR

Los siguientes diagramas son ejemplos de utilización de paréntesis con una instrucción OR: OR(...).



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

Modificadores

En la siguiente tabla se enumeran los modificadores que se pueden asignar a los paréntesis.

Modificador	Función	Ejemplo
N	Negación	AND(N u OR(N
F	Flanco descendente	AND(F u OR(F
R	Flanco ascendente	AND(R u OR(R
[Comparación	Consulte <i>Instrucciones de comparación, p. 256</i>

Intercalado de paréntesis

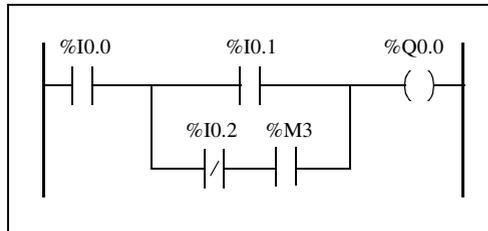
Es posible intercalar hasta ocho niveles de paréntesis.

Tenga en cuenta las siguientes reglas para intercalar paréntesis:

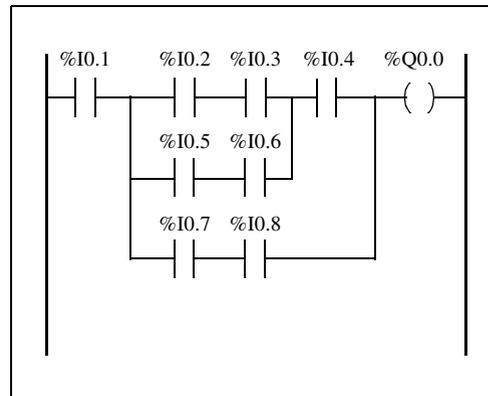
- Cada paréntesis de apertura debe tener su paréntesis de cierre correspondiente.
- No se deben colocar etiquetas (%Li:), subrutinas (SRi:), instrucciones de salto (JMP) ni instrucciones de bloque de función en instrucciones entre paréntesis.
- No se deben programar instrucciones de almacenamiento ST, STN, S o R entre paréntesis.
- No se pueden utilizar instrucciones de stack MPS, MRD o MPP entre paréntesis.

Ejemplos de intercalado de paréntesis

Los siguientes diagramas muestran ejemplos de intercalado de paréntesis.



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



```
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
```

Instrucciones de stack (MPS, MRD, MPP)

Introducción

Las instrucciones de stack procesan el direccionamiento a bobinas. Las instrucciones MPS, MRD y MPP utilizan un área de almacenamiento temporal llamada stack, que puede almacenar hasta ocho expresiones booleanas.

Nota: Estas instrucciones no se pueden utilizar en una expresión entre paréntesis.

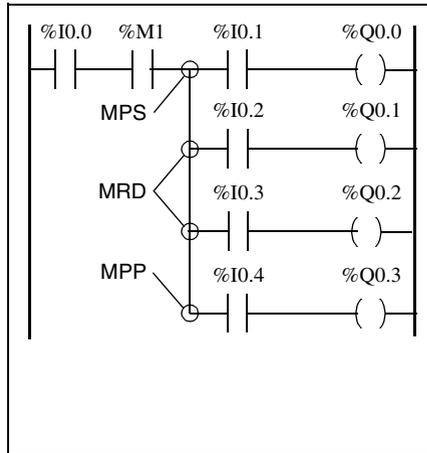
Funcionamiento de las instrucciones de stack

En la siguiente tabla se describe el funcionamiento de las tres instrucciones de stack.

Instrucción	Descripción	Función
MPS	Introducir memoria en stack	Almacena el resultado de la última instrucción lógica (contenidos del acumulador) en la parte superior del stack (introducir) y desplaza el resto de valores a la parte inferior del stack.
MRD	Leer memoria desde stack	Lee la parte superior del stack en el acumulador.
MPP	Sacar memoria de stack	Copia el valor de la parte superior del stack en el acumulador (sacar) y desplaza el resto de valores hacia la parte superior del stack.

Ejemplos de instrucciones de stack

Los siguientes diagramas son ejemplos de utilización de las instrucciones de stack.

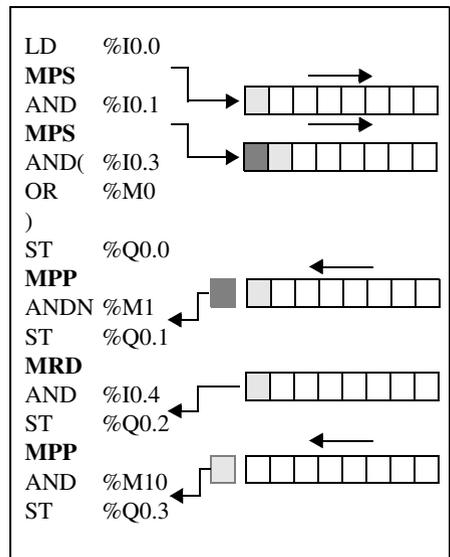
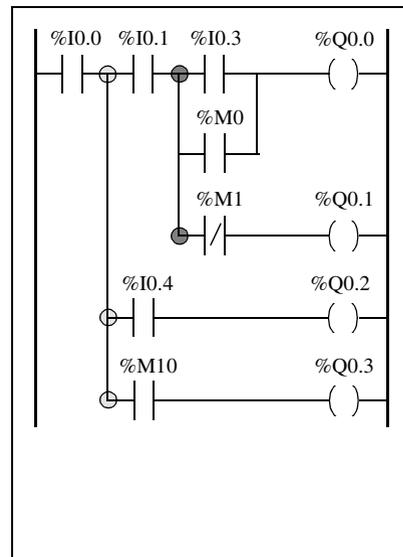


```

LD    %I0.0
AND   %M1
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
    
```

Ejemplos de funcionamiento de stack

Los siguientes diagramas muestran el funcionamiento de las instrucciones de stack.



Presentación

Vista general

Este capítulo describe la programación con el lenguaje Grafcet.

Contenido:

Este capítulo contiene los siguiente apartados:

Apartado	Página
Descripción de las instrucciones Grafcet	194
Descripción de la estructura del programa Grafcet	198
Acciones asociadas a pasos Grafcet	202

Descripción de las instrucciones Grafcet

Introducción

Las instrucciones Grafcet de TwidoSoft ofrecen un método sencillo de traducir una secuencia de control (diagrama Grafcet).

La cantidad máxima de pasos Grafcet depende del tipo de controlador Twido. La cantidad de pasos activos simultáneamente sólo está limitada por el número total de pasos.

Para TWDLCAA10DRF y TWDLCAA16DRF, están disponibles los pasos del 1 al 62. Para los otros controladores, están disponibles los pasos del 1 al 94.

Instrucciones Grafcet

La tabla que aparece a continuación enumera todas las instrucciones y objetos necesarios para programar un diagrama Grafcet.

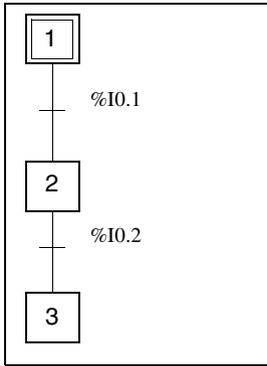
Representación gráfica (1)	Transcripción en lenguaje TwidoSoft	Función
 Paso inicial	=*= i	Comenzar paso inicial (2)
 Paso de transición	# i	Activar paso i tras desactivar el paso actual
	-*- i	Comenzar el paso i y validar la transición asociada (2)
	#	Desactivar el paso actual sin activar ningún otro paso
	#Di	Desactivar el paso i y el paso actual
	=*= POST	Iniciar procesamiento posterior y finalizar procesamiento secuencial
	%Xi	Se puede comprobar y escribir el bit asociado con el paso i (el número máximo de pasos depende del controlador).
	LD %Xi, LDN %Xi AND %Xi, ANDN %Xi, OR %Xi, ORN %Xi XOR %Xi, XORN %Xi	Comprobar actividad del paso i
	S %Xi	Activar paso i
	R %Xi	Desactivar paso i

(1) No apoya Grafcet gráfico.

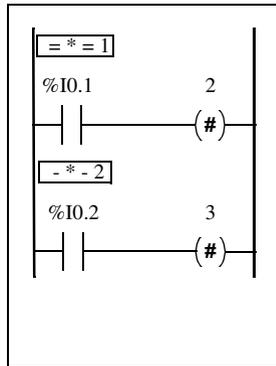
(2) El primer paso =*=i o -*-i escrito indica el inicio del procesamiento secuencial y, por lo tanto, el final del procesamiento previo.

Ejemplos de Grafcet

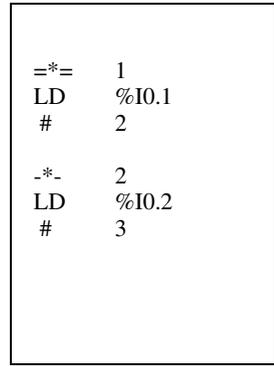
Secuencia linear:



No apoyado

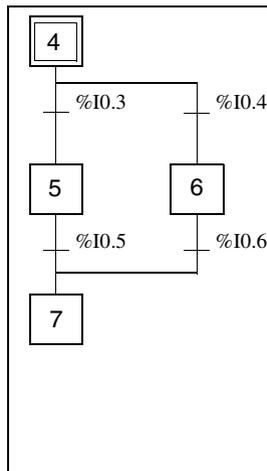


Programa Ladder de Twido

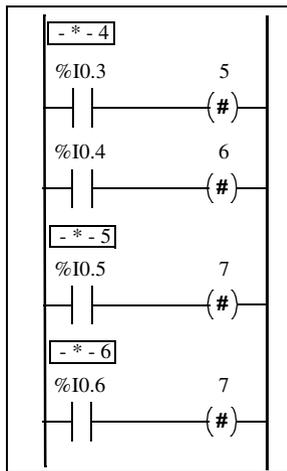


Programa de lista de Twido

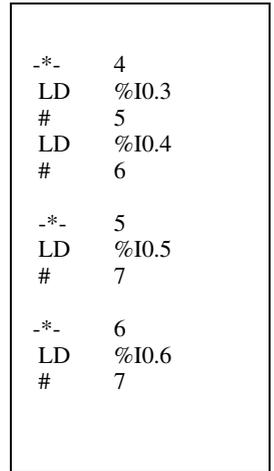
Secuencia alternativa:



No apoyado

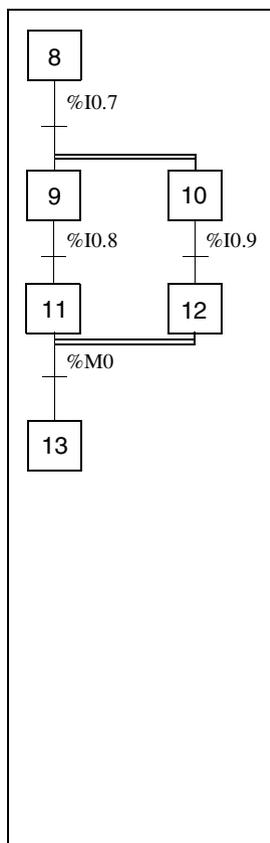


Programa Ladder de Twido

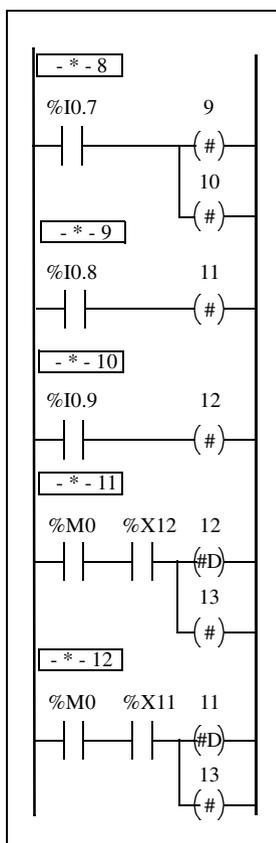
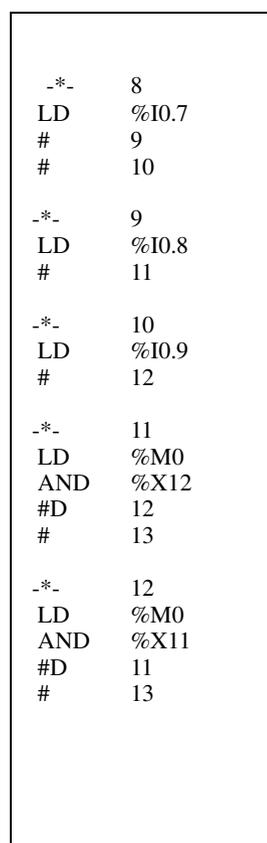


Programa de lista de Twido

Secuencias simultáneas:



No apoyado

Programa Ladder
de TwidoPrograma de lista
de Twido

Nota: Para que un diagrama Grafcet funcione, debe haber al menos un paso activo utilizando la instrucción *=i (paso inicial) o el diagrama debe ubicarse antes durante el procesamiento previo utilizando el bit de sistema %S23 y la instrucción S %Xi.

Descripción de la estructura del programa Grafcet

Introducción

Un programa Grafcet de TwidoSoft consta de tres partes:

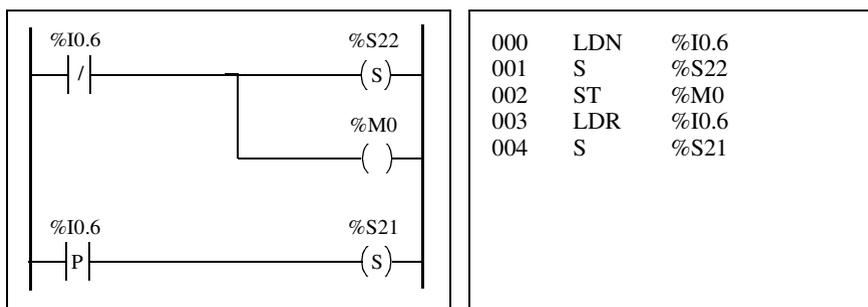
- Procesamiento anterior
 - Procesamiento secuencial
 - Procesamiento posterior
-

Procesamiento previo

El procesamiento previo consta de las siguientes partes.

- Recuperación de la alimentación
- Errores
- Cambios de modo de funcionamiento
- Pasos Grafcet de ubicación previa
- Entrada lógica

En el ejemplo de ubicación previa que aparece a continuación (área anterior al primer paso Grafcet), el estado 0 de la entrada %I0.6 solicita que el diagrama Grafcet se restaure estableciendo el bit de sistema %S22 en 1. Esto desactivará los pasos activos. El flanco ascendente de la entrada %I0.6 coloca el diagrama antes del paso X1. Finalmente, la utilización del bit de sistema %S21 fuerza la inicialización de Grafcet.



El procesamiento previo comienza con la primera línea del programa y finaliza con la primera aparición de una instrucción "= * =" o "- * -".

Existen tres bits de sistema designados al control de Grafcet: %S21, %S22 y %S23. La aplicación establece cada uno de estos bits de sistema en 1 (si fuera necesario), normalmente durante el procesamiento previo. El sistema lleva a cabo la función asociada cuando finaliza el procesamiento previo y, entonces, el sistema restaura bit de sistema a 0.

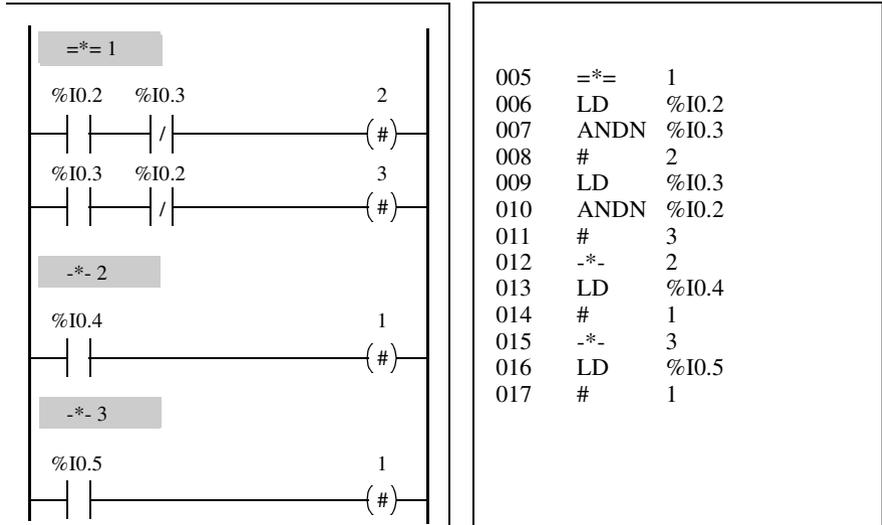
Bit de sistema	Nombre	Descripción
%S21	Inicialización de Grafcet	Todos los pasos activos se desactivan y los pasos iniciales se activan.
%S22	Restablecer Grafcet	Se desactivan todos los pasos.
%S23	Ubicación previa de Grafcet	Este bit se debe establecer en 1 si %Xi han sido escritos de manera explícita por la aplicación durante el procesamiento previo. Si el procesamiento previo mantiene el bit en 1 sin ningún cambio explícito de los objetos %Xi, Grafcet se congela (no se tienen en cuenta las actualizaciones).

Procesamiento secuencial

El procesamiento secuencial se realiza en el diagrama (instrucciones que representan el diagrama).

- Pasos
- Acciones asociadas a los pasos
- Transiciones
- Condiciones de transición

Ejemplo:



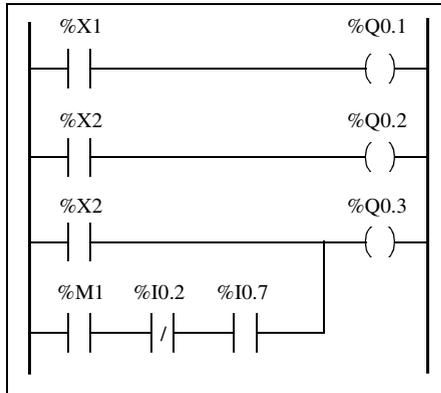
El procesamiento secuencial termina con la ejecución de la instrucción "= * = POST" o con la finalización del programa.

Procesamiento posterior

El procesamiento posterior consta de las siguientes partes.

- Comandos del procesamiento secuencial para controlar las salidas
- Dispositivos de bloqueo de seguridad específicos para las salidas

Ejemplo:



```

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
  
```

Acciones asociadas a pasos Grafcet

Introducción

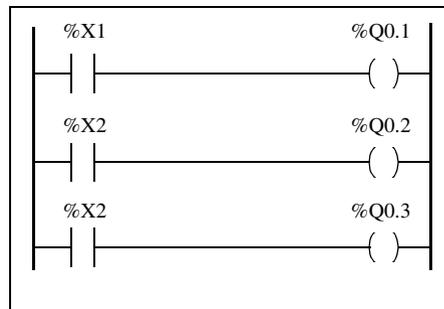
Un programa Grafcet de TwidSoft ofrece dos modos de programar acciones asociadas con los pasos:

- En la sección de procesamiento posterior
- En las instrucciones de lista o escalones de Ladder Logic de los propios pasos

Asociación de acciones en el procesamiento posterior

En caso de que existan limitaciones en el modo de seguridad o de ejecución, es preferible programar acciones en la sección de procesamiento posterior de una aplicación Grafcet. Puede utilizar las instrucciones de lista Establecer y Restablecer o conectar bobinas en el programa Ladder Logic para activar los pasos de Grafcet (%Xi).

Ejemplo:



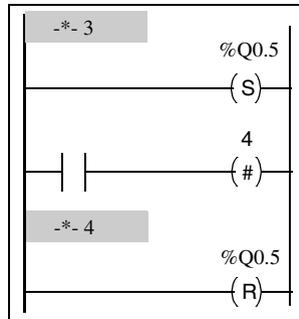
```

018  ==  POST
019  LD  %X1
020  ST  %Q0.1
021  LD  %X2
022  ST  %Q0.2
023  LD  %X3
024  ST  %Q0.3
  
```

Asociación de acciones desde una aplicación

Puede programar las acciones asociadas a los pasos dentro de las instrucciones de lista o escalones de Ladder Logic. En este caso, la instrucción de lista o el escalón de Ladder Logic no se examina a menos que esté activo el paso. Éste es el modo más eficaz, claro y sostenible de utilizar Grafcet.

Ejemplo:



```

020  -*_  3
021  LD   1
022  S    %Q0.5
023  LD   %M10
024  #    4
025  -*_  4
026  LD   1
027  R    %Q0.5
028  ...
029  ...

```

Descripción de instrucciones y funciones



Presentación

Vista general

Esta parte proporciona descripciones detalladas de instrucciones básicas y avanzadas, así como palabras y bits del sistema para lenguajes de Twido.

Contenido

Esta parte contiene los siguientes capítulos:

Capítulo	Nombre del capítulo	Página
12	Instrucciones básicas	207
13	Instrucciones avanzadas	275
14	Bits de sistema y palabras de sistema	335

Instrucciones básicas

12

Presentación

Vista general

Este capítulo proporciona detalles acerca de los bloques de función e instrucciones utilizados para crear programas de control básico para controladores Twido.

Contenido:

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
12.1	Procesamiento booleario	208
12.2	Bloques de función básicos	225
12.3	Procesamiento numérico	250
12.4	Instrucciones del programa	268

12.1 Procesamiento booleario

Introducción al procesamiento booleario

Vista general Esta sección proporciona una introducción al procesamiento booleario, incluidas las descripciones y directrices de programación para instrucciones boolearias.

Contenido Esta sección contiene los siguientes apartados:

Apartado	Página
Instrucciones boolearias	209
Comprensión del formato para describir instrucciones boolearias	212
Instrucciones de carga (LD, LDN, LDR, LDF)	214
Instrucciones de almacenamiento (ST, STN, R, S)	216
Instrucciones AND lógicas (AND, ANDN, ANDR, ANDF)	218
Instrucciones OR lógicas (OR, ORN, ORR, ORF)	220
Instrucciones de OR exclusivo (XOR, XORN, XORR, XORF)	222
Instrucción NOT (N)	224

Instrucciones booleanas

Introducción

Las instrucciones booleanas pueden compararse con elementos del lenguaje Ladder Logic como se resume en la siguiente tabla.

Elemento	Instrucción	Ejemplo	Descripción
Elementos de prueba	La instrucción Cargar (LD) equivale a un contacto abierto.	LD %I0.0	El contacto se cierra cuando el bit de control está en estado 1.
Elementos de acción	La instrucción Almacenar (ST) equivale a una bobina.	ST %Q0.0	El objeto de bit asociado toma un valor lógico del acumulador de bits (resultado de lógica anterior).

El resultado booleano de los elementos de prueba se aplica a los elementos de acción como muestran las siguientes instrucciones.

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

Comprobación de las entradas del controlador

Pueden utilizarse instrucciones de prueba booleana para detectar flancos ascendentes o descendentes en las entradas del controlador. Se ha detectado un flanco cuando el estado de una entrada ha cambiado entre "explorar n-1" y "explorar n" actual y permanece detectado durante la exploración actual.

Detección del flanco ascendente

La instrucción LDR (Cargar flanco ascendente) equivale a un contacto de detección del flanco ascendente. El flanco ascendente detecta un cambio de las entradas de control de 0 a 1.

Se utiliza un contacto de detección de transición positiva para detectar un flanco ascendente como se muestra en el siguiente diagrama.

LDR %I0.0  P: Contacto de detección de transición positiva

Detección del flanco descendente

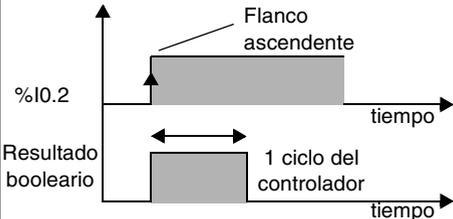
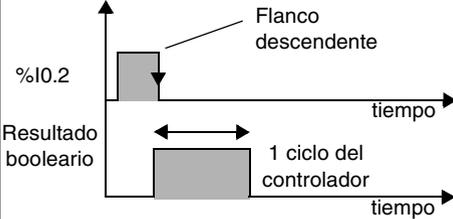
La instrucción LDF (Cargar flanco descendente) equivale a un contacto de detección del flanco descendente. El flanco descendente detecta un cambio de la entrada de control de 1 a 0.

Se utiliza un contacto de detección de transición negativa para detectar un flanco descendente como se muestra en el siguiente diagrama.



Temporización de detección de flanco

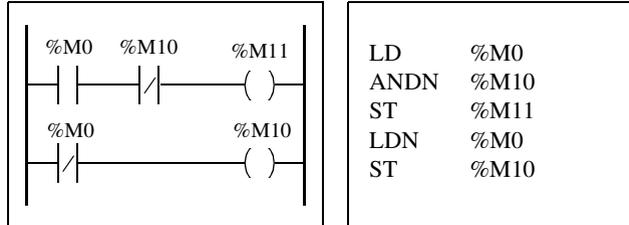
La siguiente tabla resume las instrucciones y la temporización de las instrucciones booleanas utilizadas para comprobar los flancos ascendentes y descendentes.

Flanco	Instrucción de prueba	Diagrama Ladder Logic	Temporización
Flanco ascendente	LDR %I0.0		
Flanco descendente	LDF %I0.0		

Uso de bits internos para la detección de flanco

Las instrucciones sobre un flanco ascendente o descendente hacen referencia a entradas %I, pero es posible para detectar flancos en cualquier otro bit (o resultado booleano) utilizando dos bits internos.

En el siguiente ejemplo, el bit %M11 registra el flanco ascendente del bit %M0.



Nota: En un reinicio en frío o en caliente, la aplicación detecta un flanco ascendente aunque la entrada permanezca en 1. Esta tarea se puede enmascarar iniciando el programa en las instrucciones LD %S1 y ENDC.

Nota: La detección directa de los flancos ascendente y descendente sólo puede realizarse utilizando bits de entrada (%I).

Comprensión del formato para describir instrucciones booleanas

Introducción

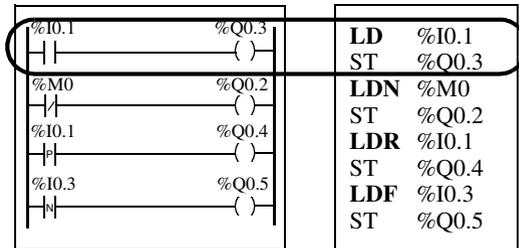
Todas las instrucciones booleanas de esta sección se describen utilizando la siguiente información:

- Breve descripción
- Ejemplo de la instrucción y del correspondiente diagrama de Ladder Logic
- Lista de operandos permitidos
- Cronograma

Las siguientes explicaciones proporcionan más detalles acerca del modo en que se describen las instrucciones booleanas en esta sección.

Ejemplos

La siguiente ilustración muestra el modo en que se proporcionan ejemplos para cada instrucción.



Equivalentes del diagrama de Ladder Logic

Instrucciones de lista

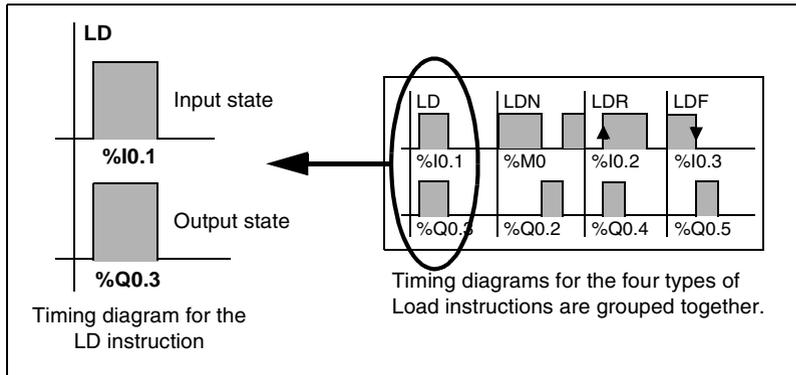
Operandos permitidos

La siguiente tabla define los tipos de operandos permitidos utilizados para las instrucciones booleanas.

Operando	Descripción
0/1	Valor inmediato de 0 ó 1
%I	Entrada del controlador %Ii.j
%Q	Salida del controlador %Qi.j
%M	Bit interno %Mi
%S	Bit de sistema %Si
%X	Bit de pasos %Xi
%BLK.x	Bit del bloque de función (por ejemplo, %TMi.Q)
%•:Xk	Bit de palabra (por ejemplo, %MWi:Xk)
[Expresión de comparación (por ejemplo, [%MWi<1000])

Cronogramas

La siguiente ilustración muestra el modo en que se muestran cronogramas para cada instrucción.



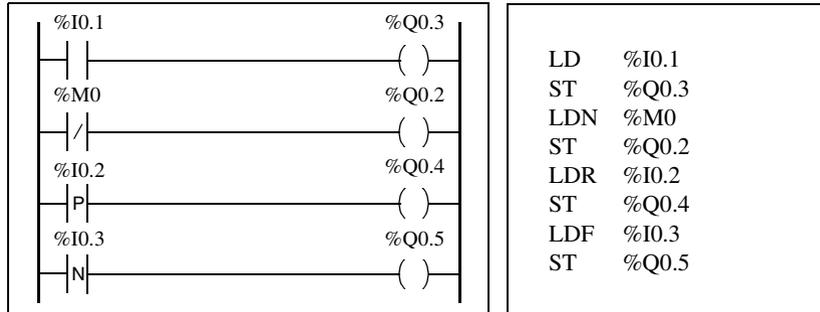
Instrucciones de carga (LD, LDN, LDR, LDF)

Introducción

Las instrucciones de carga LD, LDN, LDR y LDF corresponden respectivamente a los contactos abierto, cerrado, flanco ascendente y flanco descendente (LDR y LDF sólo se utilizan con entradas del controlador).

Ejemplos

Los siguientes diagramas son ejemplos de instrucciones de carga.



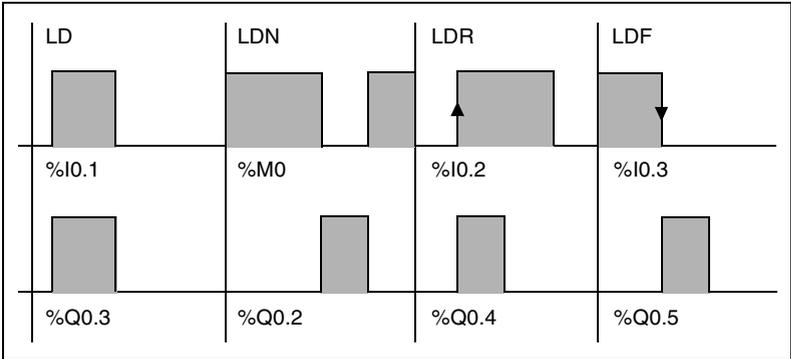
Operandos permitidos

La siguiente tabla enumera los tipos de instrucciones de carga con operandos equivalentes y permitidos de Ladder Logic.

Instrucción de lista	Equivalente Ladder Logic	Operandos permitidos
LD		0/1,%I,%Q,%M,%S,%X,%BLK.x,%*:Xk,[
LDN	/	%I,%Q,%M,%S,%X,%BLK.x,%*:Xk,[
LDR	P	%I
LDF	N	%I

Cronograma

El siguiente diagrama muestra la temporización de las instrucciones de carga.



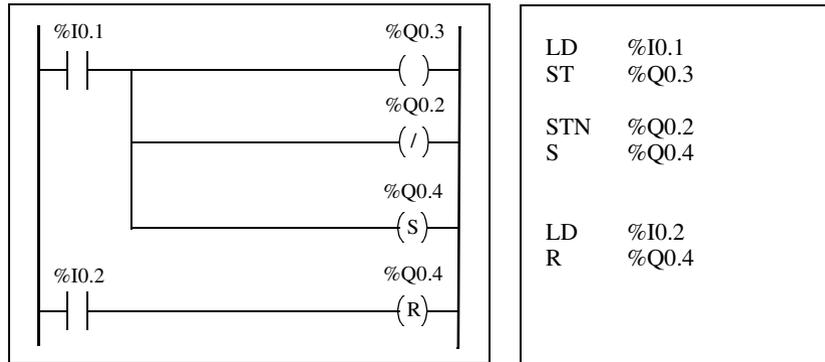
Instrucciones de almacenamiento (ST, STN, R, S)

Introducción

Las instrucciones de almacenamiento ST, STN, S y R corresponden respectivamente a las bobinas directa, inversa, establecida y restablecida.

Ejemplos

Los siguientes diagramas son ejemplos de instrucciones de almacenamiento.



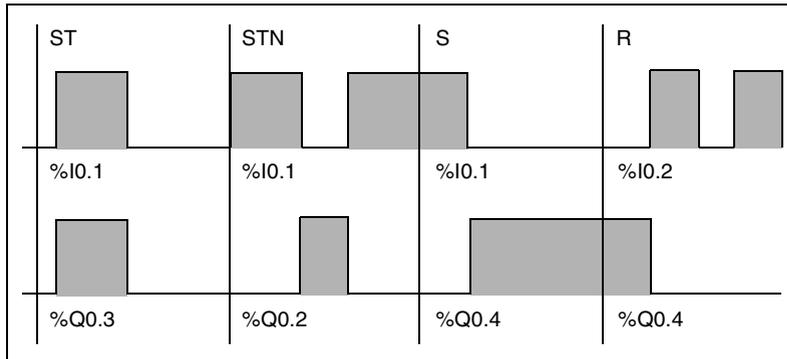
Operandos permitidos

La siguiente tabla enumera los tipos de instrucciones de almacenamiento con operandos equivalentes y permitidos de Ladder Logic.

Instrucción de lista	Equivalente Ladder Logic	Operandos permitidos
ST	()	%Q,%M,%S,%BLK.x,%•:Xk
STN	(/)	%Q,%M,%S,%BLK.x,%•:Xk
S	(s)	%Q,%M,%S,%X,%BLK.x,%•:Xk
R	(R)	%Q,%M,%S,%X,%BLK.x,%•:Xk

Cronograma

El siguiente diagrama muestra la temporización de las instrucciones de almacenamiento.



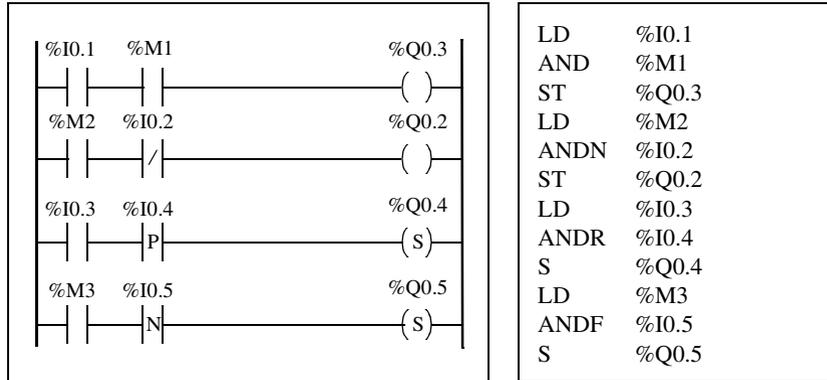
Instrucciones AND lógicas (AND, ANDN, ANDR, ANDF)

Introducción

Las instrucciones AND realizan una operación lógica AND entre el operando (o su inverso; o su flanco ascendente o descendente) y el resultado booleano de la instrucción precedente.

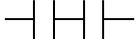
Ejemplos

Los siguientes diagramas son ejemplos de instrucciones AND lógicas.



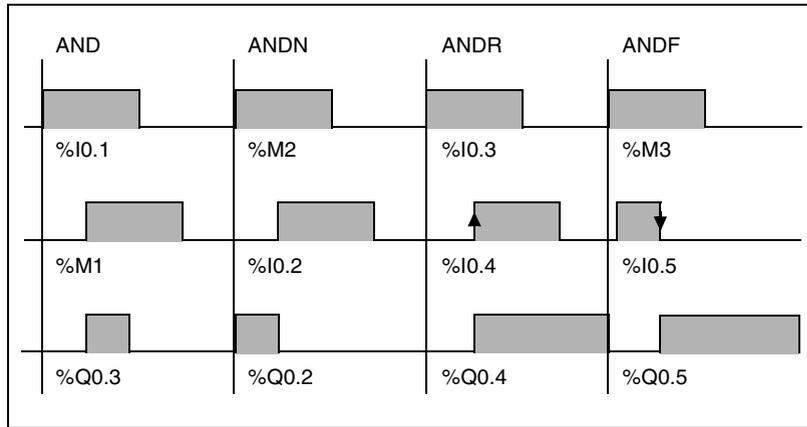
Operandos permitidos

La siguiente tabla enumera los tipos de instrucciones AND con operandos equivalentes y permitidos de Ladder Logic.

Instrucción de lista	Equivalente Ladder Logic	Operandos permitidos
AND		0/1,%I,%Q,%M,%S,%X,%BLK.x,%*:Xk, [
ANDN		%I,%Q,%M,%S,%X,%BLK.x,%*:Xk, [
ANDR		%I
ANDF		%I

Cronograma

El siguiente diagrama muestra la temporización de las instrucciones AND.



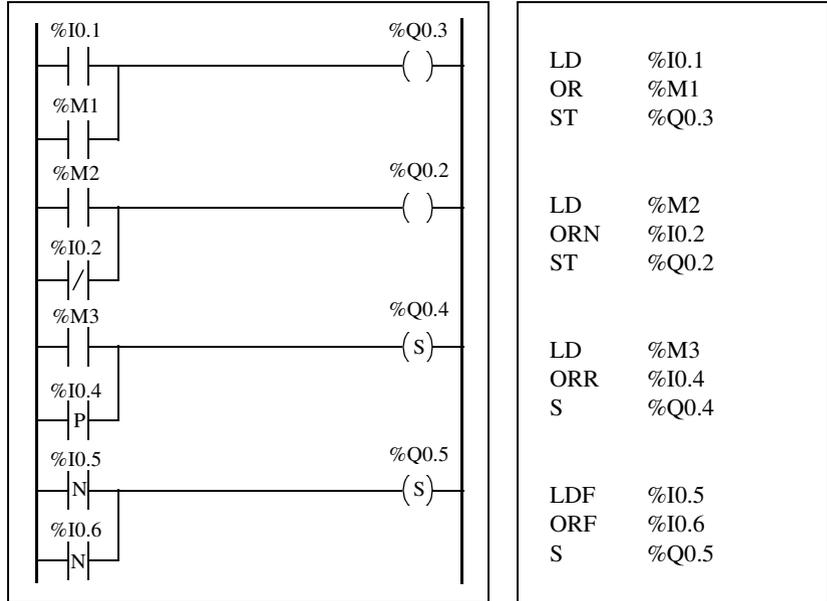
Instrucciones OR lógicas (OR, ORN, ORR, ORF)

Introducción

Las instrucciones OR realizan una operación lógica OR entre el operando (o su inverso; o su flanco ascendente o descendente) y el resultado booleano de la instrucción precedente.

Ejemplos

Los siguientes diagramas son ejemplos de instrucciones OR lógicas.



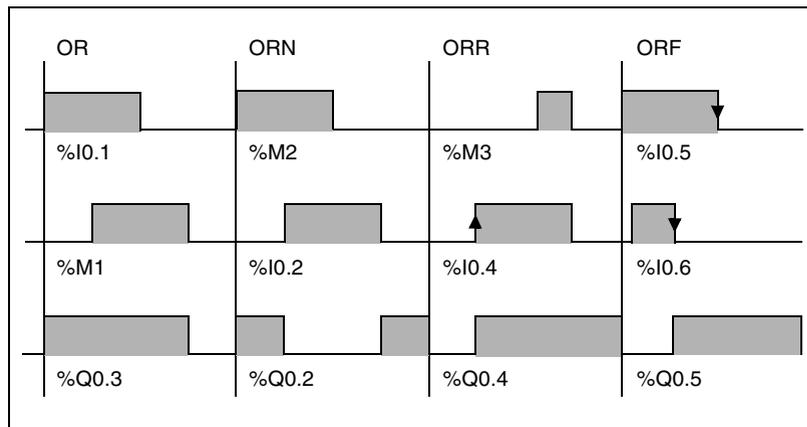
Operandos permitidos

La siguiente tabla enumera los tipos de instrucciones OR con operandos equivalentes y permitidos de Ladder Logic.

Instrucción de lista	Equivalente Ladder Logic	Operandos permitidos
OR		0/1,%I,%Q,%M,%S,%X,%BLK.x,%•:Xk
ORN		%I,%Q,%M,%S,%X,%BLK.x,%•:Xk
ORR		%I
ORF		%I

Cronograma

El siguiente diagrama muestra la temporización de las instrucciones OR.



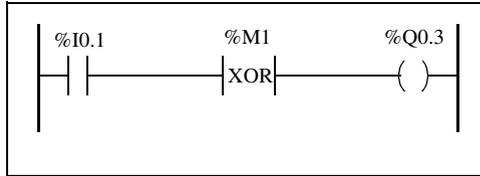
Instrucciones de OR exclusivo (XOR, XORN, XORR, XORF)

Introducción

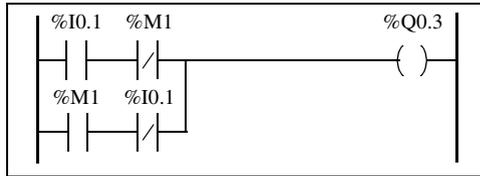
Las instrucciones XOR realizan una operación de OR exclusivo entre el operando (o su inverso; o su flanco ascendente o descendente) y el resultado booleano de la instrucción precedente.

Ejemplos

Las instrucciones XOR pueden utilizarse como se muestra en los siguiente ejemplos.



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



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

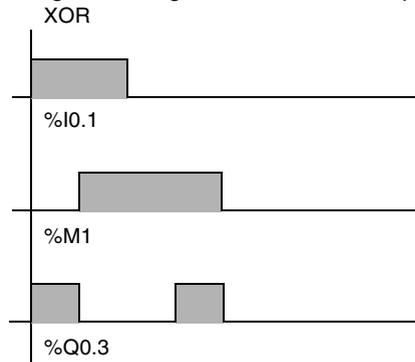
Operandos permitidos

La siguiente tabla enumera los tipos de instrucciones XOR y operandos permitidos.

Lista de instrucciones	Operandos permitidos
XOR	%I,%Q,%M,%S,%X,%BLK.x,%*:Xk
XORN	%I,%Q,%M,%S,%X,%BLK.x,%*:Xk
XORR	%I
XORF	%I

Cronograma

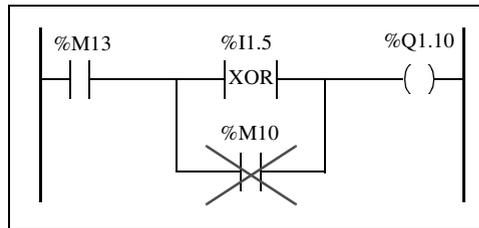
El siguiente diagrama muestra la temporización de las instrucciones XOR.

**Casos especiales**

A continuación aparecen precauciones especiales para utilizar instrucciones XOR en programas de Ladder Logic.

- No inserte contactos XOR en la primera posición de un escalón.
- No inserte contactos XOR de forma paralela con otros elementos de Ladder Logic (consulte el siguiente ejemplo).

Como se muestra en el siguiente ejemplo, la inserción de un elemento de forma paralela con el contacto XOR generará un error de validación.



Instrucción NOT (N)

Introducción La instrucción NOT (N) niega el resultado booleano de la instrucción anterior.

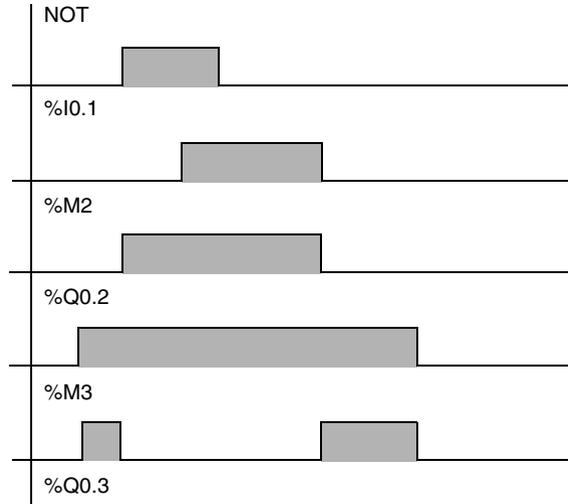
Ejemplo A continuación se muestra un ejemplo de uso de la instrucción NOT.

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

Nota: La instrucción NOT no es reversible.

Operandos permitidos No aplicable.

Cronograma El siguiente diagrama muestra la temporización de la instrucción NOT.



12.2 Bloques de función básicos

Presentación

Vista general Esta sección proporciona descripciones y directrices de programación para utilizar bloques de función básicos.

Contenido Esta sección contiene los siguientes apartados:

Apartado	Página
Bloques de función básicos	226
Principios para programar bloques de función	228
Bloque de función del temporizador (%TMi)	230
Tipo de temporizador TOF	232
Tipo de temporizador TON	233
Tipo de temporizador TP	234
Programación y configuración de temporizadores	235
Bloque de función del contador progresivo/regresivo (%Ci)	238
Programación y configuración de contadores	242
Bloque de función del registro de bits de desplazamiento (%SBRi)	243
Bloques de función del contador de pasos (%SCi)	246

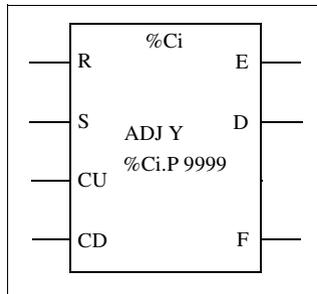
Bloques de función básicos

Introducción

Los bloques de función son los orígenes de los objetos de bit y palabras específicas utilizados por los programas. Los bloques de función básicos proporcionan funciones simples como temporizadores o conteo progresivo/regresivo.

Ejemplo de un bloque de función

A continuación se muestra una ilustración del bloque de función del contador progresivo/regresivo.



Bloque del contador
progresivo/regresivo

Objetos de bit

Los objetos de bit corresponden a las salidas de bloque. Puede accederse a estos bits mediante instrucciones de prueba booleanas utilizando cualquiera de los siguientes métodos:

- Directamente (por ejemplo, LD E) si están cableados al bloque en programación reversible (consulte *Principios para programar bloques de función*, p. 228).
- Especificando el tipo de bloque (por ejemplo, LD %Ci.E).

Puede accederse a las entradas en forma de instrucciones.

Objetos de palabra

Los objetos de palabra corresponden a parámetros especificados y a valores del siguiente modo:

- **Parámetros de configuración de bloques:** Se puede acceder a algunos parámetros a través del programa (por ejemplo, parámetros de preselección) y a otros no (por ejemplo, base de tiempo).
- **Valores actuales:** Por ejemplo, %Ci.V, el valor de conteo.

Objetos de palabra y bit accesibles

La siguiente tabla describe los objetos de palabra y bits de bloques de función básicos a los que puede acceder el programa.

Bloque de función básico	Símbolo	Rango (i)	Tipos de objetos	Descripción	Dirección	Acceso de escritura
Temporizador	%Tmi	0 - 127	Palabra	Valor actual	%Tmi.V	no
				Valor preestablecido	%Tmi.P	sí
			Bit	Salida de temporizador	%Tmi.Q	no
Contador progresivo/ regresivo	%Ci	0 - 31	Palabra	Valor actual	%Ci.V	no
				Valor preestablecido	%Ci.P	sí
			Bit	Salida de transgresión por debajo de rango (vacío)	%Ci.E	no
				Salida predeterminada alcanzada	%Ci.D	no
				Salida de desborde (lleno)	%Ci.F	no

Principios para programar bloques de función

Introducción

Utilice uno de los métodos para programar bloques de función básicos que aparecen a continuación.

- Instrucciones de bloque de función (por ejemplo, `BLK %TM2`): método reversible de programación en lenguaje Ladder Logic que permite que las operaciones que se van a realizar en el bloque se lleven a cabo en un único lugar del programa.
- Instrucciones específicas (por ejemplo, `CU %Ci`): método no reversible que permite que las operaciones que se van a realizar en las entradas del bloque se lleven a cabo en varias partes del programa (por ejemplo, `line 100 CU %C1`, `line 174 CD %C1`, `line 209 LD %C1.D`).

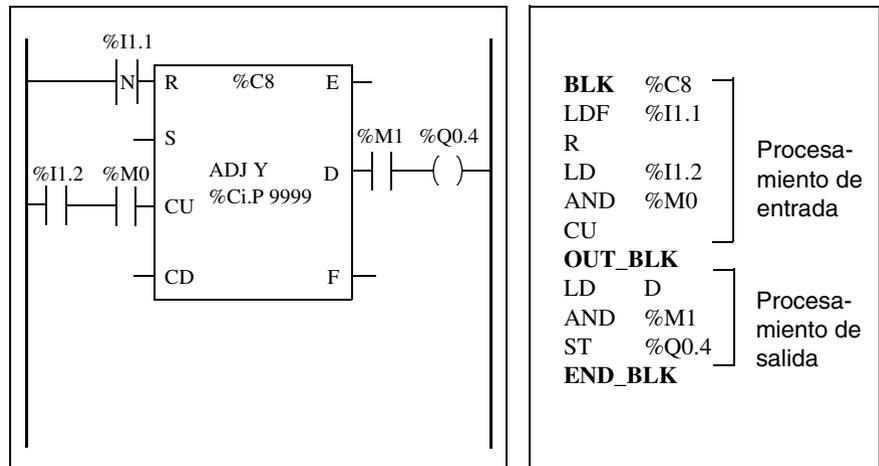
Programación reversible

Utilice las instrucciones `BLK`, `OUT_BLK` y `END_BLK` para programación reversible.

- **BLK**: Indica el principio de un bloque.
- **OUT_BLK**: Se utiliza para cablear directamente las salidas de bloque.
- **END_BLK**: Indica el final de un bloque.

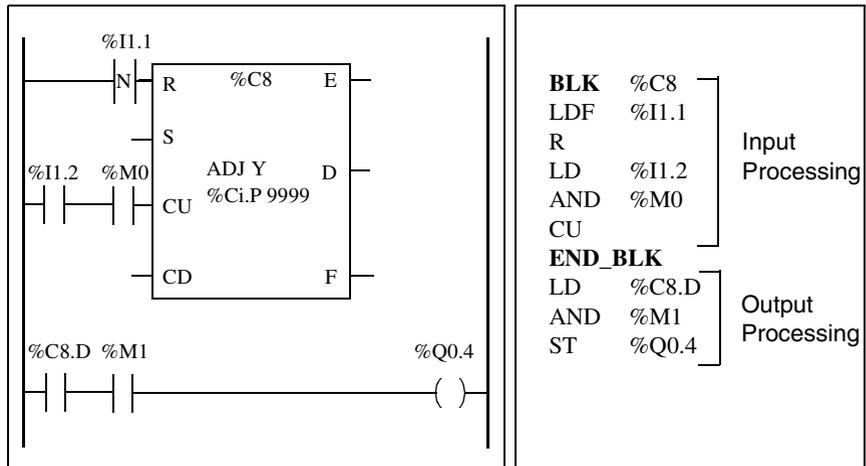
Ejemplo con salidas cableadas

El siguiente ejemplo muestra la programación reversible de un bloque de función del contador con salidas cableadas.



Ejemplo sin cableado de salida

Este ejemplo muestra la programación reversible de un bloque de función del contador sin cableado en las salidas.



Nota: Sólo se pueden colocar instrucciones de entrada y de prueba en el bloque relevante entre las instrucciones BLK y OUT_BLK (o entre BLK y END_BLK cuando no se programa OUT_BLK).

Bloque de función del temporizador (%TMi)

Introducción

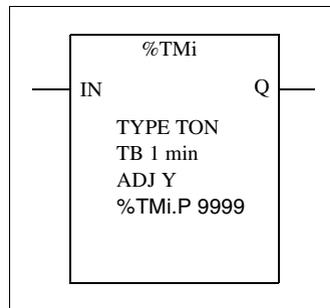
Existen tres tipos de bloques de función del temporizador:

- TON (temporizador de retardo a la conexión): utilice este tipo de temporizador para controlar las acciones de retardo a la conexión.
- TOF (temporizador de retardo a la desconexión): utilice este tipo de temporizador para controlar las acciones de retardo a la desconexión.
- TP (pulso de temporizador): utilice este tipo de temporizador para generar pulsos de duración determinada.

Los retardos o períodos de pulsos se pueden programar y modificar utilizando TwidoSoft.

Ilustración

A continuación se muestra una ilustración del bloque de función del contador.



Bloque de función del
temporizador

Parámetros

El bloque de función del temporizador presenta los siguientes parámetros:

Parámetro	Etiqueta	Valor
Número de temporizador	%TMi	Controlador compacto 0 a 63 Controladores modulares 0 a 127
Tipo	TON	• retardo a la conexión (predeterminado)
	TOF	• retardo a la desconexión
	TP	• pulso (monoestable)
Base de tiempo	TB	1 min (predeterminado), 1 s, 100 ms, 10 ms, 1 ms (para TM0 y TM1).
Valor actual	%TMi.V	Palabra que aumenta de 0 a %TMi.P cuando el temporizador está en funcionamiento. Se puede leer y comprobar, pero no se puede escribir desde el programa. %TMi.V se puede modificar utilizando el editor de datos.
Valor preestablecido	%TMi.P	0 - 9999. Palabra que se puede leer, comprobar y escribir desde el programa. El valor predeterminado es 9999. El período o retardo generado es igual a %TMi.P x TB.
Editor de datos	Y/N	Y: Sí, el valor preestablecido %TMi.P puede modificarse utilizando el editor de datos. N: No, el valor preestablecido %TMi.P no se puede modificar.
Establecimiento de entrada (o instrucción)	IN	Inicia el temporizador en flanco ascendente (tipos TON o TP) o en flanco descendente (tipo TOF).
Salida del temporizador	Q	El bit asociado %TMi.Q se establece en 1 dependiendo de la función realizada: TON, TOF o TP.1.

Nota: Cuanto mayor sea el valor preestablecido, mayor será la precisión del temporizador.

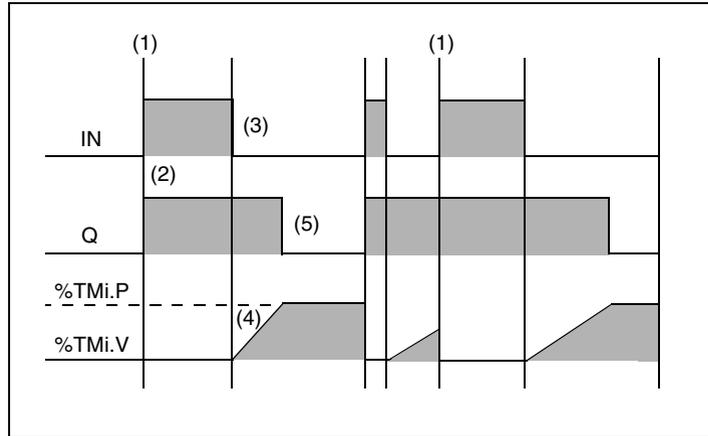
Tipo de temporizador TOF

Introducción

El tipo de temporizador TOF (temporizador de retardo a la desconexión) se utiliza para controlar las acciones de retardo a la desconexión. Este retardo se puede programar con TwidoSoft.

Cronograma

El siguiente cronograma ilustra el funcionamiento del temporizador de tipo TOF.



Operación

En la siguiente tabla se describe el funcionamiento del temporizador de tipo TOF.

Fase	Descripción
1	El valor actual %Tmi.V se establece en 0 en un flanco ascendente en la entrada IN, aun cuando el temporizador se encuentre en ejecución.
2	El bit de salida %Tmi.Q se establece en 1 cuando se detecte un flanco ascendente en la entrada N.
3	El temporizador inicia en el flanco descendente de la entrada IN.
4	El valor actual %Tmi.V aumenta a %Tmi.P en incrementos de una unidad por pulso de la base de tiempo TB.
5	El bit de salida %Tmi.Q se restablece a 0 cuando el valor actual llega a %Tmi.P.

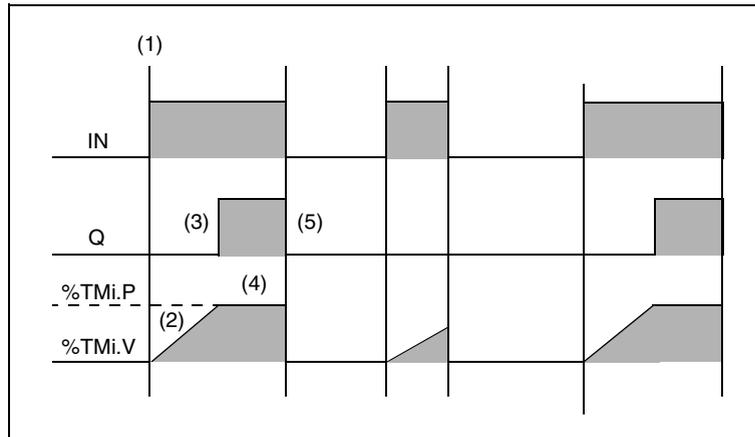
Tipo de temporizador TON

Introducción

El tipo de temporizador TON (temporizador de retardo a la conexión) se utiliza para controlar las acciones de retardo a la conexión. Este retardo se puede programar con TwidoSoft.

Cronograma

El siguiente cronograma ilustra el funcionamiento del temporizador de tipo TON.



Operación

En la siguiente tabla se describe el funcionamiento del temporizador de tipo TON.

Fase	Descripción
1	El temporizador inicia en el flanco ascendente de la entrada IN.
2	El valor actual %TMI.V aumenta de 0 a %TMI.P en incrementos de una unidad por pulso de la base de tiempo TB.
3	El bit de salida %TMI.Q se establece en 1 cuando el valor actual llega a %TMI.P.
4	El bit de salida %TMI.Q permanece en 1 mientras la entrada IN esté en 1.
5	Si se detecta un flanco descendente en la entrada IN, el temporizador se detiene, aun cuando el temporizador no haya alcanzado el valor %TMI.P, y %TMI.V se establece en 0.

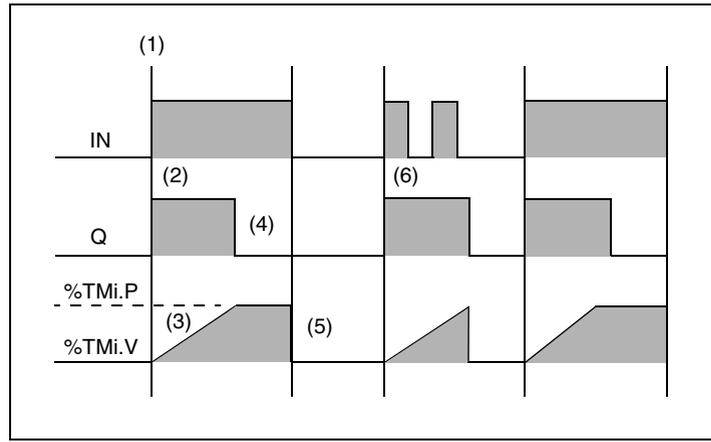
Tipo de temporizador TP

Introducción

El tipo de temporizador TP (pulso de temporizador) se utiliza para generar pulsos de duración determinada. Este retardo se puede programar con TwidoSoft.

Cronograma

El siguiente cronograma ilustra el funcionamiento del temporizador de tipo TP.



Operación

En la siguiente tabla se describe el funcionamiento del temporizador de tipo TP.

Fase	Descripción
1	El temporizador se inicia en el flanco ascendente de la entrada IN. El valor actual %Tmi.V se establece en 0 si el temporizador todavía no se ha iniciado.
2	El bit de salida %Tmi.Q se establece en 1 cuando se inicia el temporizador.
3	El valor actual %Tmi.V del temporizador aumenta de 0 a %Tmi.P en incrementos de una unidad por pulso de la base de tiempo TB.
4	El bit de salida %Tmi.Q se establece en 0 cuando el valor actual llega a %Tmi.P.
5	El valor actual %Tmi.V se establece en 0 cuando %Tmi.V es igual a %Tmi.P y la entrada IN vuelve a 0.
6	Este temporizador no se puede restablecer. Una vez %Tmi.V es igual a %Tmi.P y la entrada IN es 0, %Tmi.V se establecerá en 0.

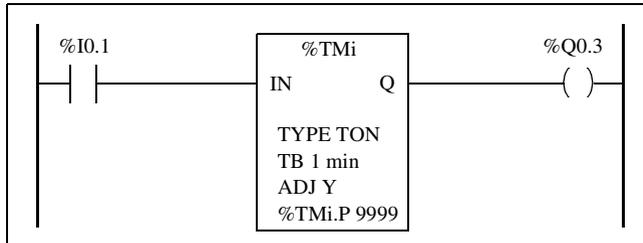
Programación y configuración de temporizadores

Introducción

Los bloques de función del temporizador (%T_{Mi}) se programan de la misma manera, independientemente del modo en que vayan a utilizarse. La función del temporizador (TON, TOF o TP) se selecciona durante la configuración.

Ejemplos

La siguiente ilustración es un bloque de función del temporizador con ejemplos de programación reversible y no reversible.



Programación reversible

```

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

```

Programación no reversible

```

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

```

Configuración

Durante la configuración, deben introducirse los siguientes parámetros:

- Tipo de temporizador: TON, TOF o TP
- Tiempo base (TB): 1 min, 1s, 100 ms, 10 ms o 1 ms
- Valor preestablecido (%T_{Mi}.P): 0 a 9999
- Ajuste: Sí o No (S o N)

Casos especiales

La siguiente tabla contiene una lista de casos especiales de programación y configuración de temporizadores.

Caso especial	Descripción
Efecto de un reinicio en frío (%S0=1)	Fuerza el valor actual a 0. Establece la salida %TMi.Q en 0. El valor preestablecido se restablece al valor definido durante la configuración.
Efecto de un reinicio en caliente (%S1=1)	No tiene ningún efecto en los valores actuales y presentes del temporizador. El valor actual no varía durante un corte de alimentación.
Efecto de una detención del controlador	No inmovilizará el valor actual.
Efecto de un salto del programa	Un salto sobre el bloque del temporizador no mantendrá el temporizador. El temporizador continúa aumentando hasta que alcanza el valor preestablecido (%TMi.P). En este punto, el bit de finalización (%TMi.Q) asignado a la salida Q del bloque del temporizador cambia de estado; sin embargo, la salida asociada cableada directamente a la salida del bloque no se activa y el controlador no la explora.
Comprobación por bit %TMi.Q (bit de finalización)	Es recomendable realizar una prueba del bit %TMi.Q una única vez en el programa.
Efecto de modificar el valor preestablecido %TMi.P	Modificar el valor presente mediante una instrucción o ajustando el valor sólo tiene efecto cuando se vuelve a activar el temporizador.

Temporizadores con un tiempo base de 1 ms

El tiempo base de 1 ms sólo está disponible en temporizadores %TM0 y %TM1. Las cuatro palabras del sistema %SW76, %SW77, %SW78 y SW79 se pueden utilizar como "relojes de arena". El sistema hace que estas cuatro palabras disminuyan individualmente cada milisegundo **si tienen un valor positivo**.

Se pueden conseguir varias temporizaciones, cargando de manera sucesiva una de estas palabras o realizando comprobaciones de los valores inmediatos. Si el valor de uno de estas cuatro palabras es menor que 0, no se modificará. Es posible inmovilizar un temporizador estableciendo el bit 15 correspondiente en 1 y cancelar la inmovilización restableciéndolo en 0.

Ejemplo de programación

A continuación se muestra un ejemplo de programación de un bloque de función del temporizador.

```

LDR %I0.1      (Ejecución del temporizador en el flanco ascendente de %I0.1)
[%SW76:=XXXX] (XXXX = valor necesario)
LD  %I0.2      (gestión opcional de inmovilización, la entrada I0.2
               queda inmovilizada)

ST  %SW76:X15
LD  [%SW76=0] (restablecimiento final del temporizador)
ST  %M0
.....
    
```



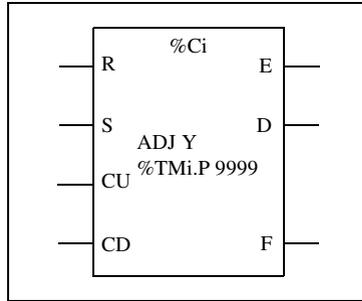
Bloque de función del contador progresivo/regresivo (%Ci)

Introducción

El bloque de función del contador (%Ci) proporciona un recuento de eventos progresivo o regresivo. Estas dos operaciones pueden realizarse de forma simultánea.

Ilustración

A continuación se muestra una ilustración del bloque de función del contador progresivo/regresivo.



Up/down counter function block

Parámetros

El bloque de función del contador tiene los siguientes parámetros:

Parámetro	Etiqueta	Valor
Número de contador	%Ci	0 a 31
Valor actual	%Ci.V	La palabra aumenta o disminuye con arreglo a las entradas (o instrucciones) CU y CD. El programa puede leerla y comprobarla, pero no escribirla. Utilice el editor de datos para modificar %Ci.V.
Valor preestablecido	%Ci.P	0 - %Ci.P-9999. La palabra puede leerse comprobarse y escribirse (valor preestablecido: 9999).
Editar utilizando el editor de datos	S/N	<ul style="list-style-type: none"> ● S: Sí, el valor preestablecido puede modificarse utilizando el editor de datos. ● N: No, el valor preestablecido no puede modificarse utilizando el editor de datos.
Restablecer entrada (o instrucción)	R	En estado 1: %Ci.V = 0.
Establecer entrada (o instrucción)	S	En estado 1: %Ci.V = %Ci.P.
Entrada de conteo progresivo (o instrucción)	CU	Incrementos %Ci.V en un flanco ascendente.
Entrada de conteo regresivo (o instrucción)	CD	Disminuciones %Ci.V en un flanco ascendente.
Salida de transgresión por debajo de rango	E (Vacío)	El bit asociado %Ci.E=1, cuando el contador regresivo %Ci.V cambia de 0 a 9999 (establecido a 1 cuando %Ci.V alcanza 9999 y se restablece a 0 si el contador continúa con el conteo regresivo).
Salida predeterminada alcanzada	D (Hecho)	El bit asociado %Ci.D=1, cuando %Ci.V=%Ci.P.
Salida de desborde	F (Llena)	El bit asociado %Ci.F=1, cuando %Ci.V cambia de 9999 a 0 (establecido a 1 cuando %Ci.V alcanza 0 y se restablece a 0 si el contador continúa con el conteo progresivo).

Operación

La siguiente tabla describe las fases principales de la operación del contador progresivo/regresivo.

Operación	Acción	Resultado
Conteo progresivo	Aparece un flanco ascendente en la CU de entrada de conteo progresivo (o se activa la CU de instrucción).	El valor actual de %Ci.V aumenta en una unidad.
	El valor actual de %Ci.V es igual al valor %Ci.P preestablecido.	El bit %Ci.D de salida "preestablecida alcanzada" asignado a la salida D cambia a estado 1.
	El valor actual %Ci.V cambia de 9999 a 0.	El bit de salida %Ci.F (desborde de conteo progresivo) cambia a estado 1.
	Si el contador continúa con el conteo progresivo.	El bit de salida %Ci.F (desborde de conteo progresivo) se restablece a 0.
Conteo regresivo	Aparece un flanco ascendente en la CD de entrada de conteo regresivo (o se activa la CD de instrucción).	El valor actual de %Ci.V disminuye en una unidad.
	El valor actual %Ci.V cambia de 0 a 9999.	El bit de salida %Ci.E (transgresión por debajo de rango) cambia a estado 1.
	Si el contador continúa con el conteo regresivo.	El bit de salida %Ci.E (transgresión por debajo de rango) se restablece como 0.
Conteo progresivo/ regresivo	Para utilizar simultáneamente las funciones de conteo progresivo y regresivo (o para activar las instrucciones CD y CU), deben controlarse las dos entradas correspondientes CU y CD. Estas dos entradas se examinan sucesivamente. Si ambas están en 1, el valor actual permanece intacto.	
Restablecer	La entrada R se establece a estado 1 (o la instrucción R se activa).	El valor actual %Ci.V se fuerza a 0. Las salidas %Ci.E, %Ci.D y %Ci.F están a 0. La entrada restablecida tiene prioridad.
Establecer	Si la entrada S está en estado 1 (o se activa la instrucción S) y la entrada restablecida está a 0 (o la instrucción R está inactiva).	El valor actual %Ci.V toma el valor %Ci.P y la salida %Ci.D se establece a 1.

Casos especiales

La siguiente tabla contiene una lista de casos especiales de programación y configuración de contadores.

Caso especial	Descripción
Efecto de un reinicio en frío (%S0=1)	<ul style="list-style-type: none">● El valor actual %Ci se establece a 0.● Los bits de salida %Ci.E, %Ci.D y %Ci.F se establecen a 0.● El valor preestablecido se inicializa con el valor definido durante la configuración.
Efecto de un reinicio en caliente (%S1=1) de una detención del controlador	No tiene ningún efecto sobre el valor actual del contador (%Ci.V).
Efecto de modificar el valor preestablecido %Ci.P	La modificación del valor preestablecido mediante una instrucción o ajustándolo entra en vigor cuando la aplicación procesa el bloque (activación de una de las entradas).

Programación y configuración de contadores

Introducción

El siguiente ejemplo es un contador que proporciona un conteo de elementos hasta 5000. Cada pulso de entrada %I1.2 (cuando el bit interno %M0 está en 1) incrementa el contador %C8 hasta su valor preestablecido final (bit %C8.D=1). El contador se restablece mediante la entrada %I1.1.

Ejemplo de programación

La siguiente ilustración es un bloque de función del contador con ejemplos de programación reversibles y no reversibles.

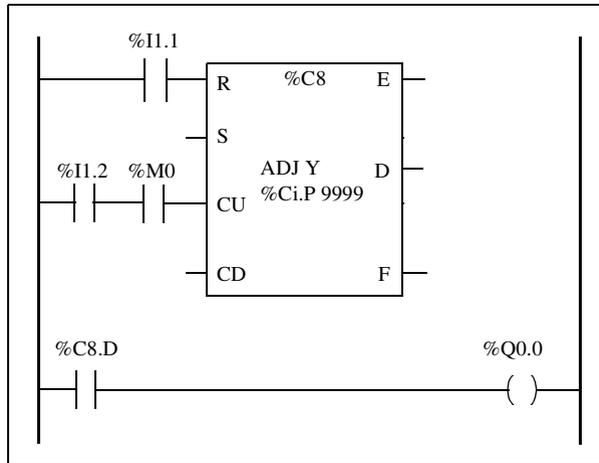


Diagrama Ladder Logic

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

Programación reversible

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

Programación no reversible

Configuración

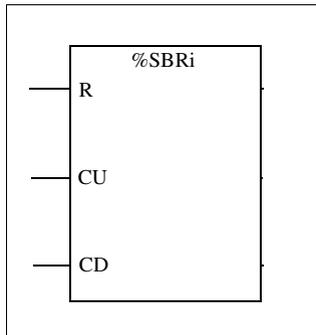
Deben introducirse los siguientes parámetros durante la configuración:

- Valor preestablecido (%Ci.P): establecido a 5000 en este ejemplo
- Ajuste: Sí

Bloque de función del registro de bits de desplazamiento (%SBRi)

Introducción El bloque de función del registro de bits de desplazamiento (%SBRi) proporciona un desplazamiento de bits de datos binarios a la izquierda o la derecha (0 ó 1).

Ilustración A continuación se muestra un ejemplo de un bloque de función del registro de desplazamiento.



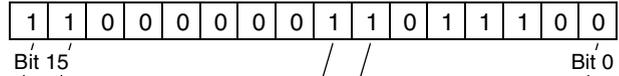
Parámetros El bloque de función del registro de bits de desplazamiento tiene los siguientes parámetros.

Parámetro	Etiqueta	Valor
Número de registro	%SBRi	0 a 7
Bit de registro	%SBRi.j	Los bits 0 a 15 (j = 0 a 15) del registro de desplazamiento pueden probarse mediante una instrucción de prueba y escribirse utilizando una instrucción de asignación.
Restablecer entrada (o instrucción)	R	En un flanco ascendente, establece los bits de registro 0 a 15 %SBRi.j a 0.
Desplazar hacia la entrada izquierda (o instrucción)	CU	En un flanco ascendente, desplaza un bit de registro a la izquierda.
Desplazar hacia la entrada derecha (o instrucción)	CD	En un flanco ascendente, desplaza un bit de registro a la derecha.

Operación

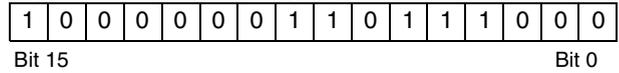
La siguiente ilustración muestra un modelo de bit antes y después de una operación de desplazamiento.

Estado inicial de la operación



CU %SBRi realiza un desplazamiento a la izquierda

El bit 15 se pierde

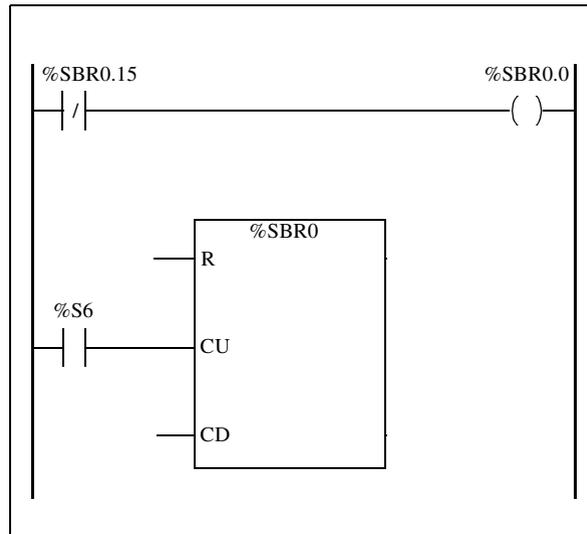


También es posible que una solicitud para desplazar un bit a la derecha (Bit 15 a Bit 0) utilizando la instrucción CD. El bit 0 se pierde.

Si un registro de 16 bits no es adecuado, es posible utilizar el programa para mostrar en cascada varios registros.

Programación

En el siguiente ejemplo, un bit se desplaza a la izquierda cada segundo mientras el bit 0 asume el estado opuesto al 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 especiales

La siguiente tabla contiene una lista de casos especiales para programar el bloque de función del registro de bits de desplazamiento.

Caso especial	Descripción
Efecto de un reinicio en frío (%S0=1)	Establece todos los bits de la palabra de registro a 0.
Efecto de un reinicio en caliente (%S1=1)	No tiene efecto sobre los bits de la palabra de registro.

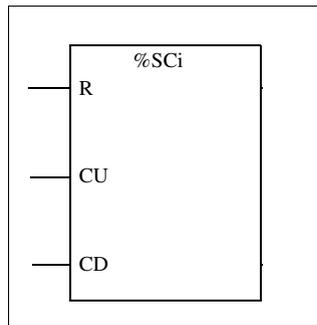
Bloques de función del contador de pasos (%SCi)

Introducción

Un bloque de función del contador de pasos (%SCi) proporciona una serie de pasos a los que se pueden asignar las acciones. El desplazamiento de un paso a otro depende de eventos internos o externos. Cada vez que un paso esté activo, el bit asociado se establecerá en 1. Sólo se puede activar un paso del contador de paso a la vez.

Ilustración

A continuación se muestra un ejemplo de un bloque de función del contador de pasos.



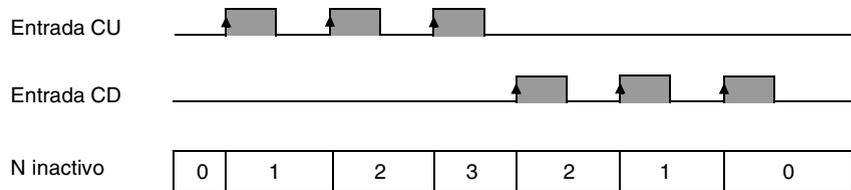
Parámetros

El bloque de función del contador de pasos tiene los siguientes parámetros:

Parámetro	Etiqueta	Valor
Número del contador de pasos	%SCi	0 a 7
Bit del contador de pasos	%SCi.j	Los bits de contador de pasos del 0 al 255 (j = 0 a 255) se pueden comprobar mediante una operación lógica Cargar y se pueden escribir con la instrucción Asignación.
Restablecer entrada (o instrucción)	R	En un flanco ascendente, pone el contador de pasos a cero.
Aumentar entrada (o instrucción)	CU	En un flanco ascendente, aumenta un paso el contador de pasos.
Disminuir entrada (o instrucción)	CD	En un flanco ascendente, disminuye un paso el contador de pasos.

Cronograma

El diagrama que aparece a continuación muestra el funcionamiento del bloque de función del contado de pasos.

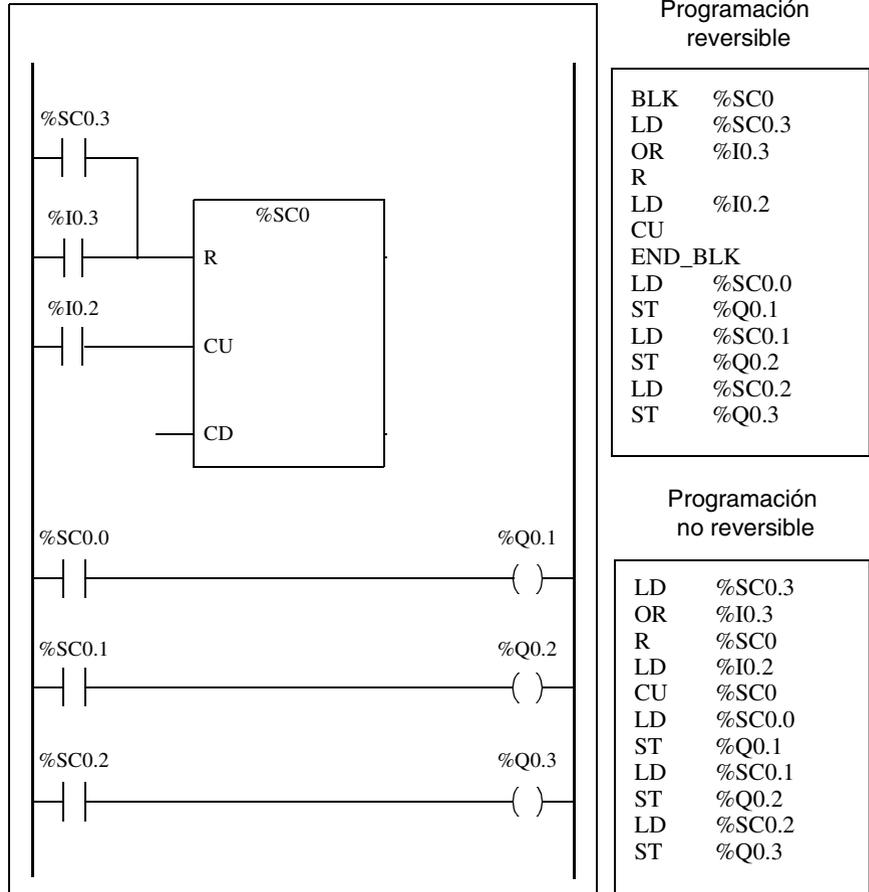


Programación

A continuación se muestra un ejemplo de un bloque de función del contador de pasos.

- La entrada %I0.2 aumenta el contador de pasos 0.
- El contador de pasos 0 se vuelve a poner a cero cuando llega al paso 3 o mediante la entrada %I0.3.
- El paso 0 controla la salida %Q0.1, el paso 1 controla la salida %Q0.2 y el paso 2 controla la salida %Q0.3.

La ilustración que aparece a continuación muestra tanto la programación reversible como la no reversible relativas al ejemplo.



Casos especiales

La siguiente tabla contiene una lista de casos especiales para programar el bloque de función del contador de pasos.

Caso especial	Descripción
Efecto de un reinicio en frío (%S0=1)	Inicializa el contador de pasos.
Efecto de un reinicio en caliente (%S1=1)	No tiene ningún efecto sobre el contador de pasos.

12.3 Procesamiento numérico

Introducción al procesamiento numérico

Vista general Esta sección ofrece una introducción al procesamiento numérico. Incluye descripciones y directrices de programación.

Contenido Esta sección contiene los siguientes apartados:

Apartado	Página
Introducción a las instrucciones numéricas	251
Instrucciones de asignación	252
Instrucciones de comparación	256
Instrucciones aritméticas	258
Instrucciones de lógica	262
Instrucciones de desplazamiento	264
Instrucciones de conversión	266

Introducción a las instrucciones numéricas

Vista general

Normalmente, las instrucciones numéricas se aplican a palabras de 16 bits (consulte *Objetos de palabra*, p. 28). Se escriben entre corchetes. Si el resultado de la operación de lógica anterior era verdadero (acumulador booleario = 0), se ejecuta la instrucción numérica. Si el resultado de la operación lógica anterior era falso, (acumulador booleario = 0), la instrucción numérica no se ejecuta y el operando permanece intacto.

Instrucciones de asignación

Introducción Las instrucciones de asignación se utilizan para cargar el operando Op2 en el operando Op1.

Asignación Sintaxis para las instrucciones de asignación.

$[Op1 := Op2]$ \Leftrightarrow $Op2 \rightarrow Op1$

Pueden realizarse operaciones de asignación en:

- Cadenas de bits
 - Palabras
 - Tablas de palabras
-

Asignación de cadenas de bits Pueden realizarse operaciones en las siguientes cadenas de bits (consulte "*Objetos estructurados*, p. 37"):

- Cadena de bits \rightarrow cadena de bits (Ejemplo 1)
 - Cadena de bits \rightarrow palabra (Ejemplo 2)
 - Palabra \rightarrow cadena de bits (Ejemplo 3)
 - Valor inmediato \rightarrow cadena de bits
-

Ejemplos

Ejemplos de asignaciones de cadenas de bits.

	<pre>LD 1 [%Q0:8:=%M64:8] (Ex. 1) LD %I0.2 [%MW100:=%I0:16] (Ex. 2) LDR %I0.3 [%M104:16:=%KW0] (Ex. 3)</pre>
--	---

Normas de uso:

- Para cadena de bits -> asignación de palabra: Los bits de la cadena se transfieren a la palabra que comienza a la derecha (primer bit de la cadena al bit 0 de la palabra) y los bits de palabra no implicados en la transferencia (longitud<16) se ponen a 0.
- Para palabra -> asignación de cadena de bits: Los bits de palabra se transfieren desde la derecha (bit de palabra 0 al primer bit de la cadena).

Asignaciones de cadenas de bits

Sintaxis para asignaciones de cadenas de bits.

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

Nota: La abreviatura %BLK.x (por ejemplo, %C0.P) se utiliza para describir cualquier palabra de bloque de función.

Asignación de palabras

Pueden realizarse operaciones de asignación en las siguientes palabras:

- Palabra -> palabra (Ejemplo 1)
- Palabra indexada -> palabra
- Valor inmediato -> palabra (Ejemplo 3)
- Cadena de bits -> palabra
- Palabra -> palabra indexada
- Palabra indexada -> palabra indexada (Ejemplo 2)
- Valor inmediato -> palabra indexada
- Palabra -> cadena de bits

Ejemplos

Ejemplos de asignaciones de palabras.

	<pre>LD 1 [%SW112:=%MW100] (Ex. 1) LD %I0.2 [%MW0[%MW10]:=] (Ex. 2) %KW0[%MW20] LDR %I0.3 [%MW10:=100] (Ex. 3)</pre>
--	---

Sintaxis

Sintaxis para asignaciones de palabras.

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

Nota: La abreviatura %BLK.x (por ejemplo, R3.I) se utiliza para describir cualquier palabra de bloque de función. Para las cadenas %Mi:L, %Si:L y %Xi:L, la dirección de base de la primera cadena de bits debe ser un múltiplo de 8 (0, 8, 16, ..., 96, ...).

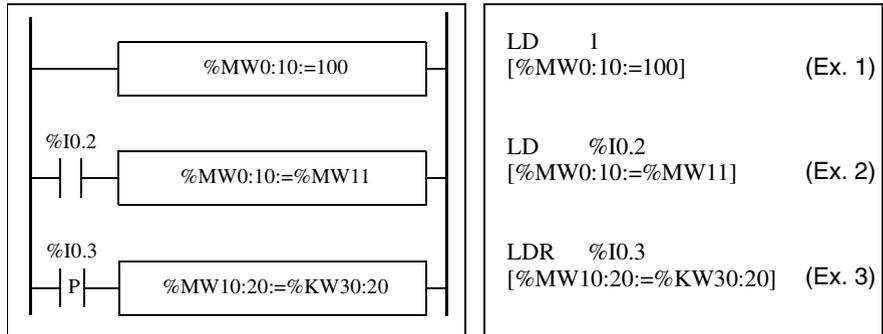
Asignación de tablas de palabras

Pueden realizarse operaciones de asignación en las siguientes tablas de palabras (consulte "*Tablas de palabras, p. 38*"):

- Valor inmediato -> tabla de palabras (Ejemplo 1)
 - Palabra -> tabla de palabras (Ejemplo 2)
 - Tabla de palabras -> tabla de palabras (Ejemplo 3)
- La longitud de tabla (L) debe ser igual para ambas tablas.

Ejemplos

Ejemplos de asignaciones de tablas de palabras.

**Sintaxis**

Sintaxis para asignaciones de tablas de palabras:

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

Nota: La abreviatura %BLK.x (por ejemplo, R3.I) se utiliza para describir cualquier palabra de bloque de función. Para las cadenas %Mi:L, %Si:L y %Xi:L, la dirección de base de la primera cadena de bits debe ser un múltiplo de 8 (0, 8, 16, ..., 96, ...).

Instrucciones de comparación

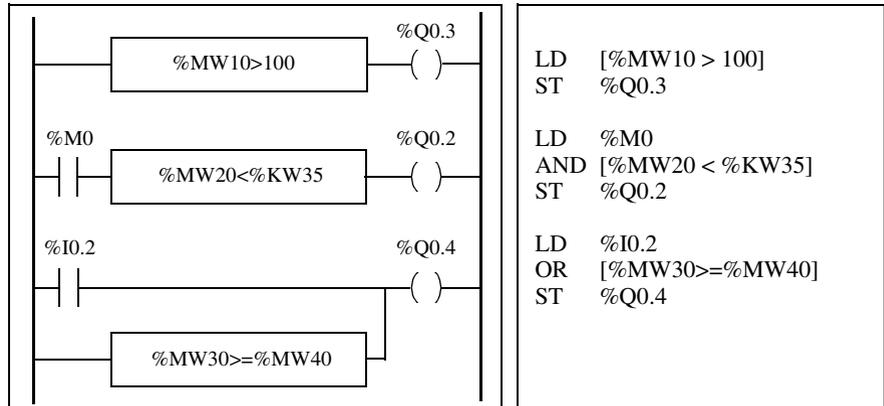
Introducción

Las instrucciones de comparación se utilizan para comparar dos operandos. La siguiente tabla enumera los tipos de instrucciones de comparación.

Instrucción	Función
>	Prueba si el operando 1 es mayor que el operando 2
>=	Prueba si el operando 1 es mayor o igual que el operando 2
<	Prueba si el operando 1 es menor que el operando 2
<=	Prueba si el operando 1 es menor o igual que el operando 2
=	Prueba si el operando 1 es igual que el operando 2
<>	Prueba si el operando 1 es diferente del operando 2

Estructura

La comparación se ejecuta entre corchetes siguiendo las instrucciones LD, AND y OR. El resultado es 1 cuando la comparación solicitada es verdadera. Ejemplos de instrucciones de comparación.



Sintaxis

Sintaxis para las instrucciones de comparación.

Operador	Sintaxis	Operando 1 (Op1)	Operando 2 (Op2)
>, >=, <, <=, =, <>	LD [Op1 Operador Op2] AND [Op1 Operador Op2] OR [Op1 Operador Op2]	%MWi, %KW _i , %INW _i , %IW, %QNW _i , %QW _i , %QNW _i , %SW _i , %BLK.x	Valor inmediato, %MW _i , %KW _i , %INW _i , %IW, %QNW _i , %QW, %SW _i , %BLK.x, %MW _i [%MW _i], %KW _i [%MW _i]

Nota: Las instrucciones de comparación pueden utilizarse entre paréntesis.

Ejemplo de utilización de una instrucción de comparación entre paréntesis:

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

Instrucciones aritméticas

Introducción

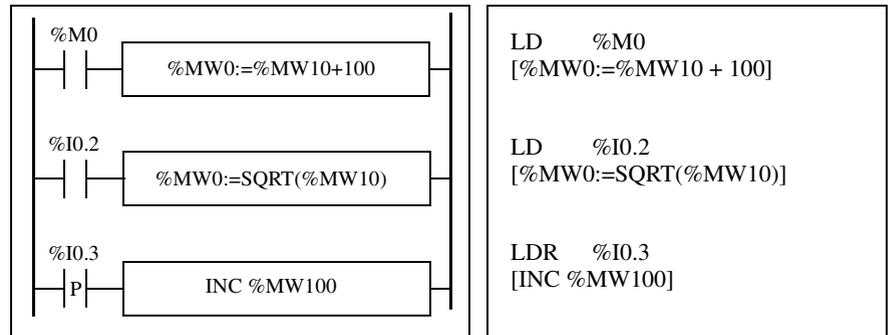
Las instrucciones aritméticas se utilizan para realizar operaciones aritméticas en un operando o entre dos.

La siguiente tabla enumera los tipos de instrucciones aritméticas.

Instrucción	Función
+	Agregar dos operandos
-	Sustraer dos operandos
*	Multiplicar dos operandos
/	Dividir dos operandos
REM	Resto de la división de dos operandos
SQRT	Raíz cuadrada de un operado
INC	Aumentar un operando
DEC	Disminuir un operando

Estructura

Las operaciones aritméticas se realizan del siguiente modo:



Sintaxis

La sintaxis depende de los operadores utilizados como se muestra en la siguiente tabla.

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

Nota: (1) Con SQRT, Op2 no puede ser un valor inmediato.

Condiciones de desborde y error

Adición

- Desborde durante la operación
Si el resultado supera los límites de -32768 ó +32767, el bit %S18 (desborde) se establece en 1. De este modo, el resultado no es correcto (consulte el ejemplo 1 en la siguiente página). El programa de aplicación gestiona el bit %S18.
- Desborde absoluto del resultado (aritmética sin signo)
Durante algunos cálculos, puede ser necesario interpretar un operando en aritmética sin signo (el bit 15 representa el valor 32768). El valor máximo de un operando es 65535. La adición de dos valores absolutos (sin signo) cuyo resultado sea superior a 65535 provoca un desborde. Esto se marca cambiando el bit del sistema %S17 (carry) a 1, lo que representa el valor 65536.

Sustracción

- Resultado negativo
Si el resultado de una sustracción es inferior a 0, el bit del sistema %S17 se establece a 1.

Multiplificación

- Desborde durante la operación
Si el resultado supera la capacidad de la palabra de resultados, el bit %S18 (desborde) se establece en 1 y el resultado no es significativo.

División/Resto

- División por 0
Si el divisor es 0, la división es imposible y el bit del sistema %S18 se establece a 1. El resultado es incorrecto.
- Desborde durante la operación
Si el cociente de la división supera la capacidad de la palabra de resultados, el bit %S18 se establece en 1.

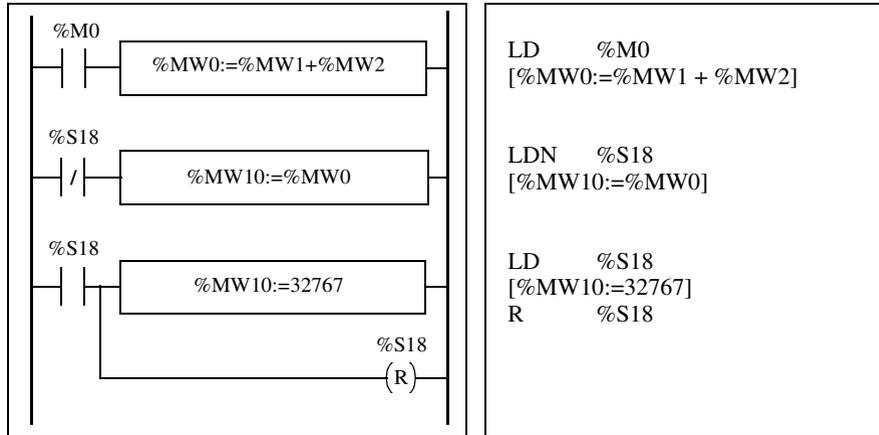
Extracción de la raíz cuadrada

- Desborde durante la operación
La extracción de la raíz cuadrada sólo se realiza en valores positivos. De este modo, el resultado siempre es positivo. Si el operando de la raíz cuadrada es negativo, el bit del sistema %S18 se establece en 1 y el resultado es incorrecto.

Nota: El programa de aplicación es responsable de la gestión de los bits del sistema %S17 y %S18. El controlador los establece a 1 y el programa debe restablecerlos para que puedan volver a utilizarse (para ver un ejemplo, consulte la página anterior).

Ejemplos

Ejemplo 1: desborde durante la adición.



Si $\%MW1 = 23241$ y $\%MW2 = 21853$, el resultado real (45094) no puede expresarse en una palabra de 16 bits, el bit $\%S18$ se establece en 1 y el resultado obtenido (-20442) es incorrecto. En este ejemplo, cuando el resultado es superior a 32767, su valor se fijará en 32767.

Ejemplo 2: $[\%MW2 := \%MW0 + \%MW1]$ donde $\%MW0 = 65086$, $\%MW1 = 65333$. La palabra $\%MW2$ contiene el número 64883. El bit $\%S17$ se establece en 1 y representa el valor 65536. El resultado aritmético sin signo es igual a: $65536 + 64883 = 130419$.

Ejemplo 3: $[\%MW2 := \%MW0 + \%MW1]$ donde $\%MW0 = 45736$ (es decir, un valor con signo de -19800), $\%MW1 = 38336$ (es decir, un valor con signo de 27200). Los dos bits del sistema $\%S17$ y $\%S18$ se establecen en 1. El resultado aritmético con signo (+18536) es incorrecto. En aritmética sin signo, el resultado (18536 + el valor de $\%S17$, que es 84072) es correcto.

Instrucciones de lógica

Introducción

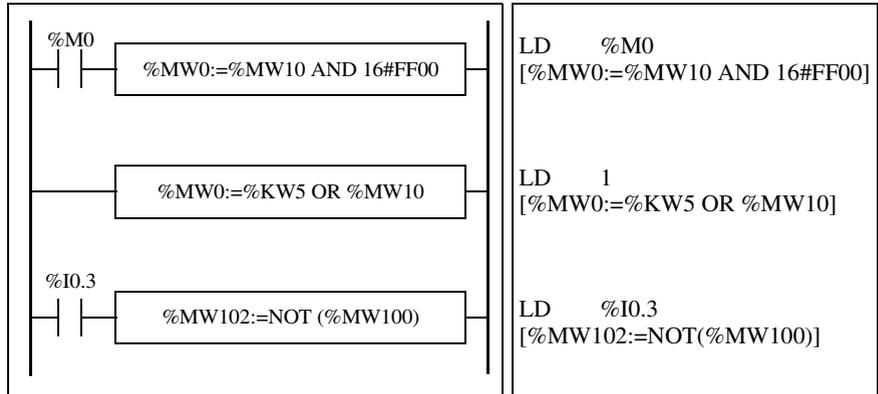
Las instrucciones de lógica se utilizan para realizar operaciones lógicas entre dos operandos de palabra o en un operando de palabra.

La siguiente tabla enumera los tipos de instrucciones de lógica.

Instrucción	Función
AND	AND (ámbito de bit) entre dos operandos
OR	OR lógica (ámbito de bit) entre dos operandos
XOR	OR exclusiva (ámbito de bit) entre dos operandos
NOT	Complemento de lógica (ámbito de bit) de un operando

Estructura

Las operaciones de lógica se realizan tal y como se muestra a continuación.



Sintaxis

La sintaxis varía según el operando que se utilice.

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

Nota: (1) Con NOT, Op2 no puede ser un valor inmediato.

Ejemplo

A continuación se muestra un ejemplo de instrucción de lógica AND.

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

Instrucciones de desplazamiento

Introducción

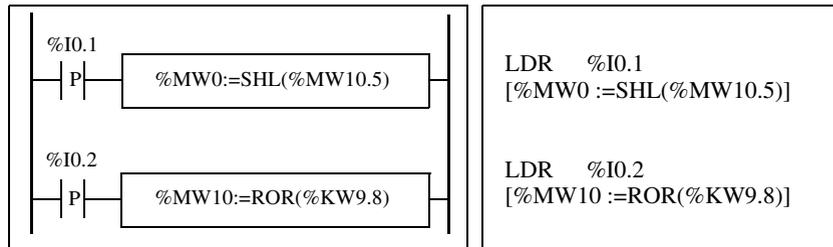
Las instrucciones de desplazamiento llevan los bits de un operando determinado número de posiciones hacia la izquierda o hacia la derecha.

La siguiente tabla enumera los tipos de instrucciones de desplazamiento.

Instrucción	Función	
Desplazamiento lógico		
SHL(op2,i)	desplazamiento lógico de i posiciones hacia la izquierda.	
SHR(op2,i)	desplazamiento lógico de i posiciones hacia la derecha.	
Desfase de rotación		
ROL(op2,i)	desfase de rotación de i posiciones hacia la izquierda.	
ROR(op2,i)	desfase de rotación de i posiciones hacia la derecha.	

Estructura

Las operaciones de desplazamiento se realizan tal y como se muestra a continuación.

**Sintaxis**

La sintaxis depende de los operadores utilizados, como se muestra en la siguiente tabla.

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

Instrucciones de conversión

Introducción

Las instrucciones de conversión realizan conversiones entre distintas representaciones de números.

La siguiente tabla enumera los tipos de instrucciones de conversión.

Instrucción	Función
BTI	Conversión BCD --> binario
ITB	Conversión binario --> BCD

Revisión del código BCD

Decimal codificado en binario (BCD) representa un dígito decimal (0 a 9) mediante la codificación de cuatro bits binarios. Un objeto de palabra de 16 bits puede contener un número expresado en cuatro dígitos (0000 a 9999).

Durante la conversión, si el valor no es BCD, el bit de sistema %S18 se establecerá a 1. El programa debe comprobar este bit y restablecerlo a 0.

Representación BCD de número decimales.

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

Ejemplos:

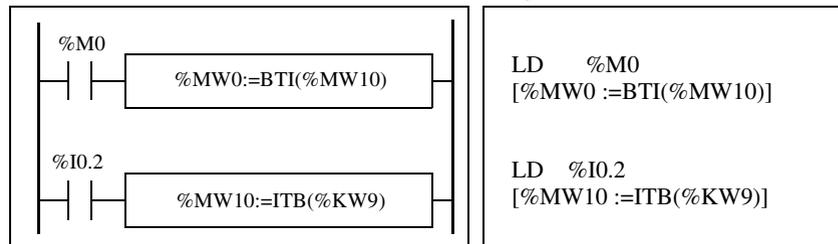
- La palabra %MW5 expresa el valor BCD "2450", que corresponde al valor binario: 0010 0100 0101 0000.
- La palabra %MW12 expresa el valor decimal "2450", que corresponde al valor binario: 0000 1001 1001 0010.

La palabra %MW5 se convierte en la palabra %MW12 mediante la instrucción BTI.

La palabra %MW12 se convierte en la palabra %MW5 mediante la instrucción ITB.

Estructura

Las operaciones de conversión se realizan del siguiente modo:



Sintaxis

La sintaxis depende de los operadores utilizados como se muestra en la siguiente tabla.

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

Ejemplo de aplicación

La instrucción BTI se utiliza para procesar un valor teórico en las entradas del controlador a través de mandos rotatorios con codificación BCD.

La instrucción ITB se utiliza para mostrar valores numéricos (por ejemplo, el resultado de un cálculo, el valor actual de un bloque de función) en pantallas con codificación BCD.

12.4 Instrucciones del programa

Introducción a instrucciones del programa

Vista general Esta sección proporciona una introducción a las instrucciones del programa.

Contenido Esta sección contiene los siguientes apartados:

Apartado	Página
Instrucciones END	269
Instrucción NOP	271
Instrucciones de salto	272
Instrucciones de subrutina	273

Instrucciones END

Introducción Las instrucciones END definen el final de la ejecución de un ciclo de programa.

END, ENDC y ENDCN

Hay tres instrucciones de fin disponibles:

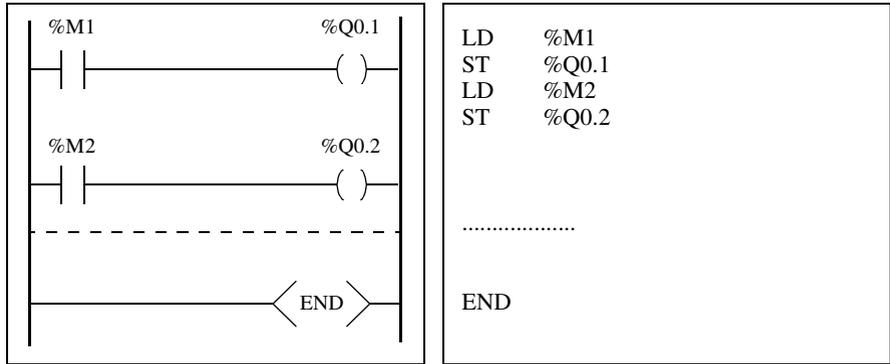
- END: fin incondicional del programa.
- ENDC: fin de programa si el resultado booleario de la instrucción de prueba precedente es 1.
- ENDCN: fin de programa si el resultado booleario de la instrucción de prueba precedente es 0.

De forma predeterminada (modo normal), cuando se activa el fin de un programa, las salidas se actualizan y se inicia el siguiente ciclo.

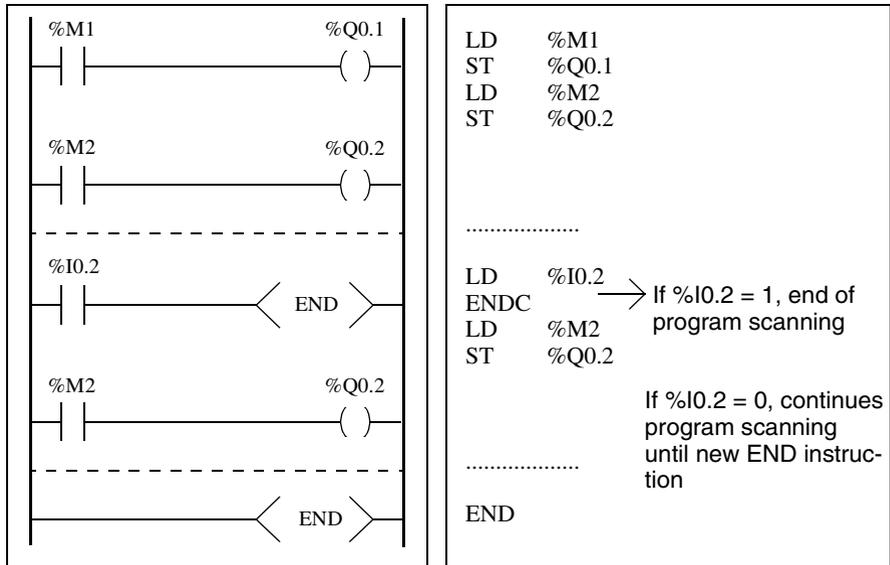
Si el ciclo es periódico, cuando se alcanza el final del periodo las salidas se actualizan y se inicia el siguiente ciclo.

Ejemplos

Ejemplo de una instrucción END incondicional.



Ejemplo de una instrucción END condicional.



Instrucción NOP

NOP

La instrucción NOP no realiza ninguna operación. Utilízela para "reservar" líneas en un programa para que pueda insertar instrucciones más adelante sin modificar los números de línea.

Instrucciones de salto

Introducción

Las instrucciones de salto provocan que la ejecución de un programa se interrumpa inmediatamente y que continúe a partir de la línea después de la línea del programa que contiene la etiqueta %Li (i = 0 a 15).

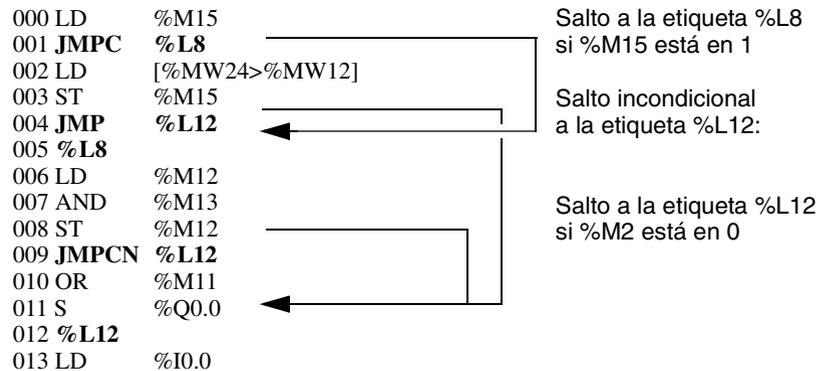
JMP, JMPC y JMPCN

Existen tres instrucciones de salto diferentes disponibles:

- JMP: salto de programa incondicional
- JMPC: salto de programa si el resultado booleario de la lógica precedentes es 1
- JMPCN: salto de programa si el resultado booleario de la lógica precedentes es 0

Ejemplos

Ejemplos de instrucciones de salto.



Directrices

- Las instrucciones de salto no están permitidas entre paréntesis y no deben situarse entre las instrucciones AND(, OR(, y una instrucción de cierre de paréntesis ")".
- La etiqueta sólo puede situarse antes de una instrucción LD, LDN, LDR, LDF or BLK.
- El número de etiqueta de la etiqueta %Li debe definirse únicamente una vez en un programa.
- El salto de programa se realiza en una línea de programación ubicada delante o detrás. Cuando el salto está ubicado detrás, debe prestarse especial atención al tiempo de ciclo del programa. Un tiempo de ciclo extendido puede utilizar el temporizador watchdog para finalizar.

Instrucciones de subrutina

Introducción Las instrucciones de subrutina hacen que un programa realice una subrutina y regrese al programa principal.

SRn, SRn: y RET Las subrutinas constan de tres pasos:

- La instrucción **SRn** llama a la subrutina a la que hace referencia la etiqueta SRn si el resultado de la instrucción boolearia precedente es 1.
- La etiqueta **SRn:** hace referencia a la subrutina con n= 0 a 15 TWDLCAA10DRF, TWDLCAA16DRF y 0 a 63 para los otros controladores.
- La instrucción **RET** situada al final de la subrutina devuelve el flujo de programas al programa principal.

Ejemplo Ejemplos de instrucciones de subrutina.

```

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

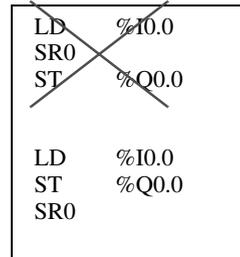
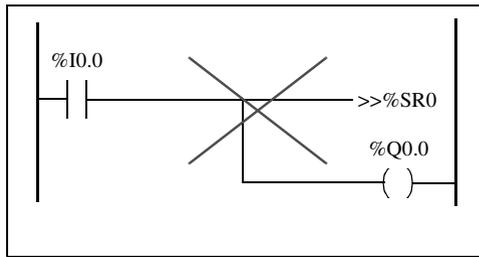
011 SR8:
012 LD     1
013 IN     %TM0
014 LD     [%TM0.Q]
015 ST     %M15
016 RET

```

Directrices

- Una subrutina no debe llamar a otra subrutina.
- Las instrucciones de subrutina no están permitidas entre paréntesis y no deben situarse entre las instrucciones AND(, OR(, y una instrucción de cierre de paréntesis ")".
- La etiqueta no sólo puede situarse antes de una instrucción LD o BLK marcando el inicio de una ecuación boolearia (o escalón).
- La llamada a la subrutina no debe ir seguida por una instrucción de asignación. Esto se debe a que es posible que la subrutina modifique el contenido del acumulador booleario. Por lo tanto, es posible que, durante la respuesta, tenga un valor diferente al que tenía antes de la llamada (consulte el siguiente ejemplo).

Ejemplo de programación de una subrutina.



Instrucciones avanzadas

13

Presentación

Vista general

Este capítulo proporciona detalles acerca de los bloques de función e instrucciones utilizados para crear programas de control avanzados para controladores programables Twido.

Contenido:

Este capítulo contiene las siguientes secciones:

Sección	Apartado	Página
13.1	Bloques de función avanzados	276
13.2	Funciones de reloj	323

13.1 Bloques de función avanzados

Presentación

Vista general Esta sección contiene una introducción a los bloques de función avanzados, incluyendo ejemplos de programación.

Contenido Esta sección contiene los siguientes apartados:

Apartado	Página
Objetos de palabra y de bit asociados a bloques de función avanzados	277
Principios de programación para bloques de función avanzados	279
Bloque de función de registro LIFO/FIFO (%Ri)	282
Operación LIFO	284
Operación FIFO	285
Programación y configuración de registros	286
Bloque de función de modulación de ancho de pulsos (%PWM)	289
Bloque de función de la salida del generador de pulsos (%PLS)	293
Bloque de función del controlador del conmutador de tambor (%DR)	296
Operación de bloque de función del controlador del conmutador de tambor	298
Programación y configuración de los controladores del conmutador de tambor	300
Bloque de función de contador rápido (%FC)	302
Bloque de función de contador muy rápido (%VFC)	306
Transmisión/recepción de mensajes - La instrucción de intercambio (EXCH)	318
Bloque de función de control de intercambio (%MSG)	319

Objetos de palabra y de bit asociados a bloques de función avanzados

Introducción

Los bloques de función avanzados utilizan tipos similares de palabras y bits especializados que los bloques de función básicos, pero requieren una mayor experiencia de programación que éstos últimos. Los bloques de función avanzados incluyen:

- Registros LIFO/FIFO (%R)
- Controladores del conmutador de tambor (%DR)
- Contadores rápidos (%FC)
- Contadores muy rápidos (%VFC)
- Salida de modulación de ancho de pulsos (%PWM)
- Salida del generador de pulsos (%PLS)
- Registro de bits de desplazamiento (%SBR)
- Contador de desplazamiento (%SC)
- Bloque de control de mensajes (%MSG)

Objetos accesibles a través del programa

La siguiente tabla contiene una vista general de las palabras y bits asociados con los diversos bloques de función avanzados. Tenga en cuenta que el acceso de escritura que figura en la tabla depende del ajuste "Ajustable" seleccionado durante la configuración. Con este ajuste se permite o se deniega el acceso a las palabras o bits desde TwidoSoft o la interfase del operador.

Bloque de función avanzado	Palabras y bits asociados		Dirección	Acceso de escritura
%R	Palabra	Acceso al registro	%Ri.I	Sí
	Palabra	Salida de registro	%Ri.O	Sí
	Bit	Salida de registro llena	%Ri.F	No
	Bit	Salida de registro vacía	%Ri.E	No
%DR	Palabra	Número del paso actual	%DRi.S	Sí
	Bit	Último paso equivale a paso actual	%DRi.F	Sí
%FC	Palabra	Valor actual	%FCi.V	No
	Palabra	Valor preestablecido	%FCi.P	Sí
	Bit	Finalización	%FCi.D	No

Bloque de función avanzado	Palabras y bits asociados		Dirección	Acceso de escritura
%VFC	Palabra	Valor actual	%VFCi.V	No
	Palabra	Valor preestablecido	%VFCi.P	Sí
	Bit	Dirección de conteo	%VFCi.U	No
	Palabra	Valor rápido	%VFCi.C	No
	Palabra	Valor de umbral 0	%VFCi.SO	Sí
	Palabra	Valor de umbral 0	%VFCi.S1	Sí
	Bit	Desborde	%VFCi.F	No
	Bit	Frecuencia lista	%VFCi.M	Sí
	Bit	Habilitar salida refleja 0	%VFCi.R	Sí
	Bit	Habilitar salida refleja 1	%VFCi.S	Sí
	Bit	Salida de umbral 0	%VFCi.TH0	No
	Bit	Base de tiempo de medida de frec.	%VFCi.T	Sí
%PWM	Palabra	Porcentaje de pulsos en 1 con relación al periodo total	%PWMi.R	Sí
	Palabra	Periodo preestablecido	%PWMi.P	Sí
%PLS	Palabra	Número de pulsos	%PLSi.N	Sí
	Palabra	Valor preestablecido	%PLSi.P	Sí
	Bit	Salida actual habilitada	%PLSi.Q	No
	Bit	Generación lista	%PLSi.D	No
%SBR	Bit	Bit de registro	%SBRi.J	No
%SC	Bit	Bit del contador de pasos	%SCi.J	Sí
%MSG	Bit	Finalización	%MSGi.D	No
	Bit	Error	%MSGi.E	No

Principios de programación para bloques de función avanzados

Vista general

Todas las aplicaciones Twido se almacenan en forma de programas de lista, incluso si se han escrito en el editor de Ladder Logic y, por lo tanto, los controladores Twido se pueden denominar "máquinas" de lista. El término "reversibilidad" se refiere a la capacidad de TwidoSoft de representar una aplicación de lista en formato Ladder Logic y de nuevo como aplicación de lista. De forma predeterminada, todos los programas de Ladder Logic son reversibles.

Al igual que los bloques de función básicos, los bloques de función avanzados también deben tener en cuenta las reglas de reversibilidad. Las instrucciones que aparecen a continuación son necesarias para la estructura de los bloques de función reversibles en lenguaje de lista.

- **BLK**: indica el inicio del bloque y la parte de entrada del bloque de función.
- **OUT_BLK**: indica el comienzo de la parte de salida del bloque de función.
- **END_BLK**: indica el final del bloque de función.

<p>Nota: El uso de estas instrucciones de bloque de función reversible no es obligatorio para que el programa de lista funcione correctamente. En lenguaje de lista se pueden programar algunas instrucciones como no reversibles.</p>

Entradas y salidas especializadas

Las funciones avanzadas contador rápido, contador muy rápido, PLS y PWM utilizan entradas y salidas especializadas, pero estos bits no están reservados para el uso exclusivo por parte de ningún bloque individual. De hecho, se debe gestionar el uso de estos recursos especializados.

Cuando utilice estas funciones avanzadas, deberá gestionar la asignación de las entradas y salidas especializadas. Para ayudarle a configurar estos recursos, TwidoSoft muestra detalles sobre la configuración de las entradas/salidas y avisa al usuario si una entrada o salida especializada ya está siendo utilizada por otro bloque de función configurado (consulte el Manual de funcionamiento de TwidoSoft).

Las tablas siguientes resumen las dependencias de las entradas y salidas especializadas y las funciones específicas.

Si se utilizan con funciones de conteo:

Entradas	Uso
%I0.0.0	%VFC0: administración progresiva/regresiva o fase B
%I0.0.1	%VFC0: entrada de pulsos o fase A
%I0.0.2	%FC0: entrada de pulsos o entrada preestablecida %VFC0
%I0.0.3	%FC1: entrada de pulsos o entrada de captura %VFC0
%I0.0.4	%FC2: entrada de pulsos o entrada de captura %VFC1
%I0.0.5	%VFC1: entrada preestablecida
%I0.0.6	%VFC1: administración progresiva/regresiva o fase B
%I0.0.7	%VFC1: entrada de pulsos o fase A

Si se utilizan con funciones especiales o de conteo:

Salidas	Uso
%Q0.0.0	Salida PWM0 o %PLS0
%Q0.0.1	Salida PWM1 o %PLS1
%Q0.0.2	Salidas reflejas para %VFC0
%Q0.0.3	
%Q0.0.4	Salidas reflejas para %VFC1
%Q0.0.5	

**Utilización de las
entradas y
salidas
especializadas**

TwidoSoft aplica las siguientes reglas para el uso de entradas y salidas especializadas.

- Cada bloque de función que utilice E/S debe ser configurado y referenciado en la aplicación. La E/S especializada sólo se asigna cuando se configura un bloque de función, y no cuando se referencia en un programa.
- Una vez configurado un bloque de función, su entrada y salida especializadas no pueden ser utilizadas por la aplicación o por otro bloque de función. Por ejemplo, si configura %PLS0, no podrá utilizar %Q0.0.0 en %DR0 (controlador del conmutador de tambor) o en la lógica de la aplicación (es decir, ST %Q0.0.0).
- Si un bloque de función necesita una entrada o salida especializada que ya está siendo utilizada por la aplicación o por otro bloque de función, dicho bloque de función no se podrá configurar. Por ejemplo, si configura %FC0 como contador progresivo, no podrá configurar %VFC0 para que utilice %I0.0.2 como entrada de captura.

Nota: Para modificar el uso de la E/S especializada, deberá deshacer la configuración del bloque de función estableciendo el tipo de objeto en "no utilizado" y, a continuación, eliminar las referencias al bloque de función en su aplicación.

Bloque de función de registro LIFO/FIFO (%Ri)

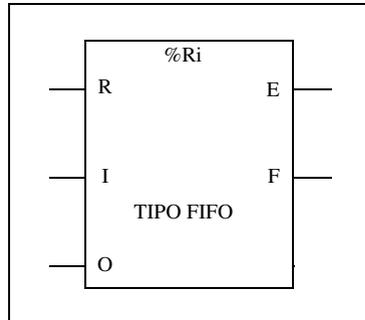
Introducción

Un registro es un bloque de memoria que puede almacenar hasta 16 palabras de 16 bits respectivamente de dos modos distintos:

- Cola (First In, First Out) conocida como FIFO
- Stack (Last In, First Out) conocida como LIFO

Ilustración

A continuación se muestra una ilustración del bloque de función de registro.



Bloque de función de registro

Parámetros

El bloque de función de registro tiene los siguientes parámetros:

Parámetro	Etiqueta	Valor
Número de registro	%Ri	0 a 3
Tipo	FIFO LIFO	Cola (selección predeterminada) Stack
Palabra de entrada	%Ri.I	Palabra de entrada de registro. Se puede leer, verificar y escribir.
Palabra de salida	%Ri.O	Palabra de salida de registro. Se puede leer, verificar y escribir.
Entrada de almacenamiento (o instrucción)	I (entrada)	Con cada flanco ascendente, almacena el contenido de la palabra %Ri.I en el registro.
Entrada de recuperación (o instrucción)	O (salida)	Con cada flanco ascendente, carga una palabra de datos en la palabra %Ri.O.
Entrada de restablecimiento (o instrucción)	R (restablecimiento)	Con el estado 1, inicializa el registro.
Salida vacía	E (vacía)	El bit asociado %Ri.E indica que el registro está vacío. Se puede verificar.
Salida llena	F (llena)	El bit asociado %Ri.F indica que el registro está lleno. Se puede verificar.

Operación LIFO

Introducción

En la operación LIFO (Last In, First Out), el último elemento de datos introducido es el primero que se recupera.

Operación

En la siguiente tabla se describe la operación LIFO.

Paso	Descripción	Ejemplo
1	Cuando se recibe una solicitud de almacenamiento (flanco ascendente en la entrada I o activación de la instrucción I), el contenido de la palabra de entrada %Ri.I (que ya está cargada) se almacena en la parte superior del stack (fig. a). Cuando el stack está lleno (salida F=1) no es posible continuar el almacenamiento.	<p>Storage of the contents of %Ri.I at the top of the stack.</p> <p>%Ri.I (a)</p>
2	Cuando se recibe una solicitud de recuperación (flanco ascendente en la entrada O o activación de la instrucción O), la palabra de datos superior (la última palabra introducida) se carga en la palabra %Ri.O (fig. b). Cuando el registro está vacío (salida E=1) no es posible continuar la recuperación.	<p>Retrieval of the data word highest in the stack.</p> <p>%Ri.O (b)</p>
3	La palabra de salida %Ri.O no se modifica y conserva su último valor. El stack se puede restablecer en cualquier momento (estado 1 en la entrada R o activación de la instrucción R). El elemento señalado por el pointer ocupa el lugar superior en el stack.	

Operación FIFO

Introducción

En la operación FIFO (First In, First Out), el primer elemento de datos introducido es el primero que se recupera.

Operación

En la siguiente tabla se describe la operación FIFO.

Paso	Descripción	Ejemplo
1	Cuando se recibe una solicitud de almacenamiento (flanco ascendente en la entrada I o activación de la instrucción I), el contenido de la palabra de entrada %Ri.I (que ya está cargada) se almacena en la parte superior de la cola (fig. a). Cuando la cola está llena (salida F=1) no es posible continuar el almacenamiento.	<p>Storage of the contents of %Ri.I at the top of the queue.</p>
2	Cuando se recibe una solicitud de recuperación (flanco ascendente en la entrada O o activación de la instrucción O), la palabra de datos inferior de la cola se carga en la palabra de salida %Ri.O y el contenido del registro se desplaza una posición hacia abajo en la cola (fig. b). Cuando el registro está vacío (salida E=1) no es posible continuar la recuperación.	<p>Retrieval of the first data item which is then loaded into %Ri.O.</p>
3	La palabra de salida %Ri.O no se modifica y conserva su valor. La cola se puede restablecer en cualquier momento (estado 1 en la entrada R o activación de la instrucción R).	

Programación y configuración de registros

Introducción

El siguiente ejemplo de programación contiene una palabra de memoria (%MW34) que se carga en un registro (%R2.I) con la solicitud de almacenamiento %I0.2 si el registro %R2 no está lleno (%R2.F = 0). La solicitud de almacenamiento en el registro se realiza mediante %M1. La solicitud de recuperación se realiza mediante la entrada %I0.3, y %R2.O se carga en %MW20 si el registro no está vacío (%R2.E = 0).

1. Una solicitud de almacenamiento en el registro se realiza mediante %M1.
 2. Una palabra de memoria (%MW34) se carga en un registro (%R2.I). Una solicitud de almacenamiento con %I0.2 si el registro %R2 no está lleno (%R2.F = 0).
 3. Una solicitud de almacenamiento con %I0.2 si el registro %R2 no está lleno (%R2.F = 0).
-

Ejemplo de programación

La siguiente ilustración es un bloque de función de registro con ejemplos de programación reversibles y no reversibles.

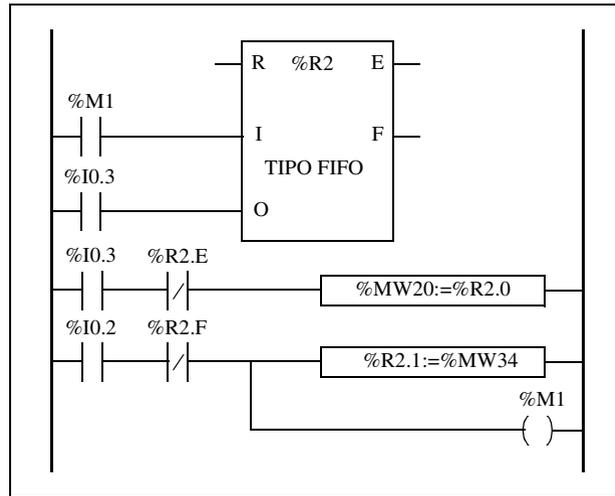


Diagrama Ladder Logic

```

BLK      %R2
LD       %M1
I
LD       %I0.3
O
END_BLK
LD       %I0.3
ANDN    %R2.E
[%MW20:=%R2.0]
LD       %I0.2
ANDN    %R2.F
[%R2.1:=%MW34]
ST      %M1

```

Programa reversible

```

LD      %M1
I      %R2
LD      %I0.3
O       %R2
ANDN    %R2.E
[%MW20:=%R2.0]
LD      %I0.2
ANDN    %R2.F
[%R2.1:=%MW34]
ST      %M1

```

Programa no reversible

Configuración

El único parámetro que se debe introducir durante la configuración es el tipo de registro:

- FIFO (predeterminado) o
 - LIFO
-

Casos especiales

La siguiente tabla contiene una lista de casos especiales de programación y configuración de registros.

Caso especial	Descripción
Efecto de un reinicio en frío (%S0=1)	Inicializa el contenido del registro. El bit de salida %Ri.E asociado a la salida E se pone a 1.
Efecto de un reinicio en caliente (%S1=1) de una detención del controlador	No tiene ningún efecto sobre el valor actual del registro ni sobre el estado de sus bits de salida.

Bloque de función de modulación de ancho de pulsos (%PWM)

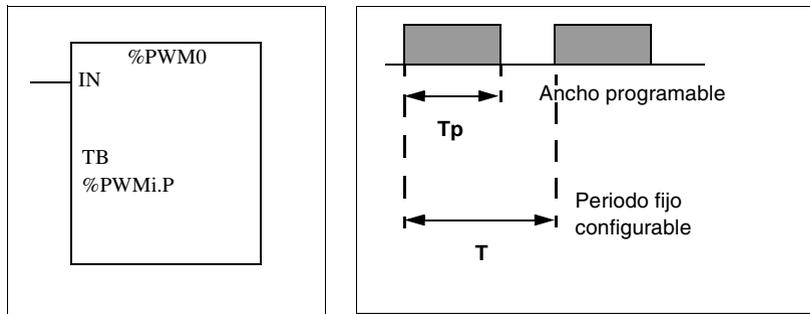
Introducción

EL bloque de función de modulación de ancho de pulsos (%PWM) genera una señal de ondas cuadradas en los canales de salidas especializadas %Q0.0.0 o %Q0.0.1. El bloque %PWM permite modificar el ancho de la señal, o ciclo de servicio. Los controladores con salidas de relé para estos dos canales no admiten esta función debido a una limitación de frecuencia.

Hay dos bloques %PWM disponibles. El bloque %PWM0 utiliza la salida especializada %Q0.0.0, mientras que el bloque %PWM1 utiliza la salida especializada %Q0.0.1. Los bloques de función %PLS compiten para utilizar estas mismas salidas especializadas, por lo tanto, deberá elegir entre las dos funciones.

Ilustración

Bloque PWM y cronograma.



Parámetros

En la tabla siguiente se enumeran los parámetros del bloque de función PWM.

Parámetro	Etiqueta	Descripción
Base de tiempo	TB	0,1 ms ¹ , 10 ms, 1 s (valor predeterminado).
Periodo preestablecido	%PwMi.P	0 < %PwMi.P ≤ 32.767 con base de tiempo de 10 ms o 1 s. 0 < %PwMi.P ≤ 255 con base de tiempo de 0,57 ms o 0,142 ms. 0 = Función no utilizada.
Ratio de pulsos (ciclo de servicio)	%PwMi.R	Este valor indica el porcentaje de la señal en estado 1 en un periodo. Por lo tanto, el ancho T_p es igual a: $T_p = T * (\%PwMi.R/100)$. La aplicación de usuario escribe el valor para %PwMi.R. Esta palabra es la que controla la modulación de ancho. Para obtener la definición de T, consulte "Rango de periodos" más adelante. El valor predeterminado es 0. Los valores superiores a 100 se consideran iguales a 100.
Entrada de generación de pulsos	IN	En estado 1, la señal de modulación de ancho de pulsos se genera en el canal de salida. En estado 0, el canal de salida se pone a 0.

Nota:

1. Esta base de tiempo no es aconsejable para controladores Twido con salidas de relé.

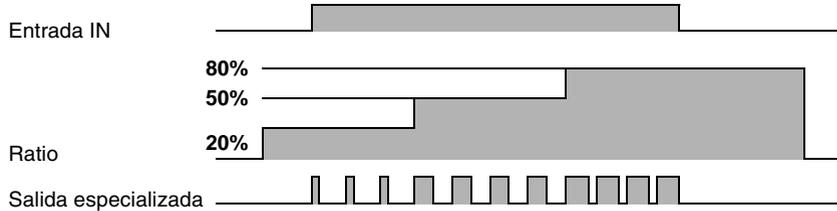
Rango de periodos

El valor preestablecido y la base de tiempo se pueden modificar durante la configuración. Se utilizan para fijar el periodo de señal $T = \%PwMi.P * TB$. Cuanto menores sean los ratios que se deban obtener, mayor deberá ser el %PwMi.P seleccionado. Rangos de periodos disponibles:

- De 0,142 ms a 36,5 ms en pasos de 0,142 ms (de 27,4 Hz a 7 kHz)
- De 0,57 ms a 146 ms en pasos de 0,57 ms (de 6,84 Hz a 1,75 kHz)
- De 20 ms a 5,45 min en pasos de 10 ms
- De 2 s a 9,1 horas en pasos de 1 s

Operación

La frecuencia de la señal de salida se ajusta durante la configuración seleccionando la base de tiempo TB y el preajuste %PWMi.P. Si se modifica el ratio % PWMi.R en el programa se modula el ancho de la señal. A continuación se incluye un diagrama de pulsos para el bloque de función PWM con ciclos de servicio cambiantes. Diagrama de pulsos para el bloque de función PWM.



Programación y configuración

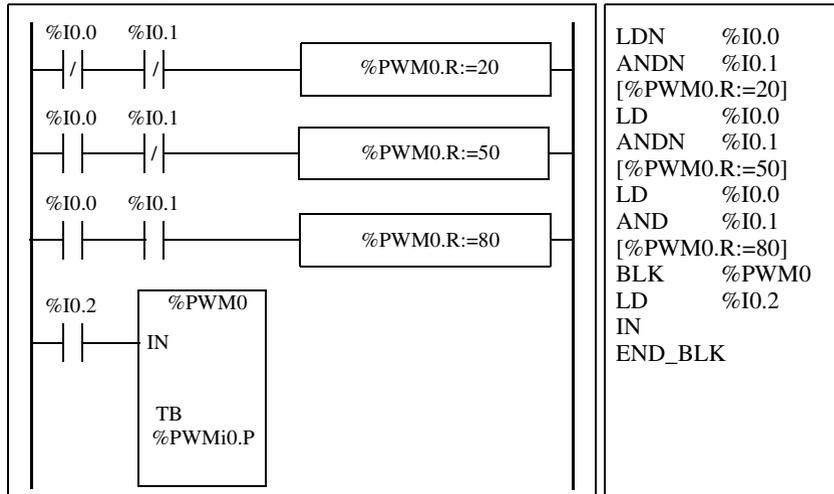
En este ejemplo, el programa modifica el ancho de señal de acuerdo con el estado de las entradas del controlador %I0.0.0 y %I0.0.1.

Si %I0.0.1 y %I0.0.2 se ponen a 0 y el ratio %PWM0.R se ajusta al 20%, la duración de la señal en estado 1 será: 20% x 500 ms = 100 ms.

Si %I0.0.0 se pone a 0 y %I0.0.1 se pone a 1, el ratio %PWM0.R se ajusta al 50% (duración 250 ms).

Si %I0.0.0 y %I0.0.1 se ponen a 1, el ratio %PWM0.R se pone al 80% (duración 400 ms).

Ejemplo de programación



Casos especiales

La siguiente tabla contiene una lista de casos especiales para programar el bloque de función PWM.

Caso especial	Descripción
Efecto de un reinicio en frío (%S0=1)	Ajusta el ratio %PWMi.R a 0. Además, el valor de %PWMi.P se restablece al valor configurado, y esto prevalecerá sobre cualquier cambio efectuado con el editor de tablas de animación o el monitor de operación opcional.
Efecto de un reinicio en caliente (%S1=1)	No tiene ningún efecto.
Utilización de una base de tiempo de 0,142 ms o 0,57 ms	Si se fuerza la salida %Q0.0.0 o %Q0.0.1 mediante un dispositivo de programación, no se detiene la generación de señal.

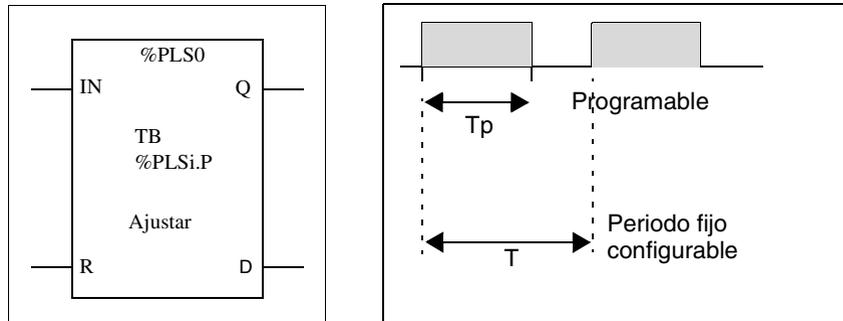
Bloque de función de la salida del generador de pulsos (%PLS)

Introducción

EL bloque de función %PWM se utiliza para generar una señal de ondas cuadradas en los canales de una salida especializada %Q0.0.0 o %Q0.0.1. La %PWM permite modificar el ancho de la señal, o ciclo de servicio. El controlador con salidas de relé para estos dos canales no son compatibles con esta función debido a una limitación de frecuencia.

Hay disponibles dos bloques %PWM. %PWM0 utiliza la salida especializada %Q0.0.0, y %PWM1 utiliza la salida especializada %Q0.0.1. Los bloques de función %PLS compiten para utilizar estas mismas salidas especializadas; por lo tanto, deberá elegir entre las dos funciones.

Representación



Características

La tabla que aparece a continuación contiene las características del bloque de función PLS:

Función	Objeto	Descripción
Base de tiempo	TB	0,142 ms, 0,57 ms, 10 ms, 1 s
Periodo de preajuste	%PLSi.P	$0 < \%PLSi.P \leq 32767$ con tiempo base de 10 ms o $1 s0 < \%PLSi.P \leq 255$ con tiempo base 0,57 ms o 0,142 ms 0 = Función no utilizada
Número de pulsos	%PLSi.N	La salida %PLS1 no deja de emitir pulsos cuando se alcanza %PLS1.N. Esto sólo es válido para %PLS0. El número de pulsos que se van a generar en el periodo T puede limitarse a $0 < \%PLSi.N < 32767$. El valor predeterminado se establece en 0. Para generar un número ilimitado de pulsos, %PLSi.N se establece a cero. El número de pulsos siempre puede modificarse independientemente del valor ajustable.
Ajustable	S/N	Si se establece a S, es posible modificar el valor preestablecido %PLSi.P mediante la HMI o el editor de tablas de animación. Si se establece a N indica que no existe acceso al valor preestablecido.
Entrada de generación de pulsos	IN	En estado 1, la generación de pulsos se crea en el canal de salida especializada. En estado 0, el canal de salida se pone a 0.
Restablecer entrada	R	En estado 1, se restablece el número de pulsos de entradas %PLSi.Q y %PLSi.D a cero.
Generación de salida de pulsos actuales	%PLSi.Q	El estado 1 indica que la generación de pulsos se crea en el canal de salida especializada configurada.
Salida realizada de generación de pulsos	%PLSi.D	En estado 1, la generación de señal está completa. Se ha alcanzado el número de pulsos deseados.

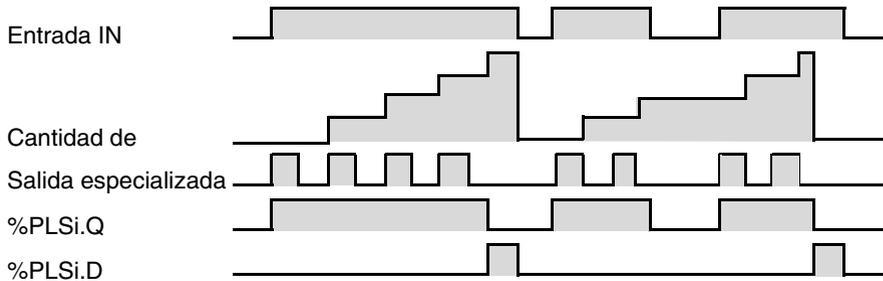
Rango de periodos

El valor de preajuste y la base de tiempo se pueden modificar durante la configuración. Se utilizan para fijar el periodo de señal $T = \%PLSi.P * TB$. Cuanto menores sean los ratios que se deban obtener, mayor deberá ser el $\%PLSi.P$ seleccionado. Rangos de periodos disponibles:

- de 0,142 ms a 36,5 ms en pasos de 0,142 ms (de 27,4 Hz a 7 kHz)
- de 0,57 ms a 146 ms en pasos de 0,57 ms (de 6,84 Hz a 1,75 kHz)
- de 20 ms a 5,45 min en pasos de 10 ms
- de 2 s a 9,1 horas en pasos de 1 s

Operación

A continuación se muestra una ilustración del diagrama de pulsos el bloque de función $\%PLS$.

**Casos especiales**

Caso especial	Descripción
Efecto de un reinicio en frío ($\%S0=1$)	Establece el $\%PLSi.P$ a aquél que aparece definido durante la configuración
Efecto de un reinicio en caliente ($\%S1=1$)	(No tiene ningún efecto)
Efecto de una detención del controlador	La salida $\%Q0.0.0$ o $\%Q0.0.1$ se establece a 0 independientemente del estado del bit del sistema $\%S8$.
Efecto de modificar el valor preestablecido ($\%PLSi.P$)	Entra en vigor inmediatamente
Uso de una base de tiempo de 0,142 ms o 0,57 ms	Forzando la salida $\%Q0.0.0$ o $\%Q0.0.1$ mediante un equipo de programación no se detiene la generación de señal.

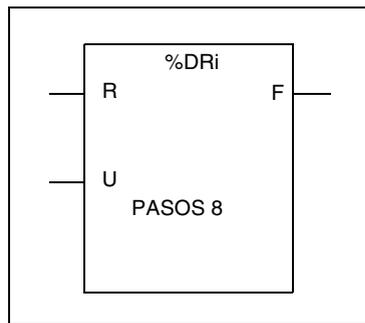
Bloque de función del controlador del conmutador de tambor (%DR)

Introducción

El controlador de tambor funciona según un principio similar a un controlador del conmutador de tambor electromecánico con cambios de pasos asociados a eventos externos. En cada paso, el punto superior de una CAM proporciona un comando que es ejecutado por el sistema de control. En el caso de un conmutador de tambor, estos puntos superiores se simbolizan mediante el estado 1 para cada paso y se asignan a bits de salida %Qi.j o bits internos %Mi, conocidos como bits de control.

Ilustración

A continuación se muestra una ilustración del bloque de función del controlador del conmutador de tambor.



Bloque de función del controlador del conmutador de tambor

Parámetros

El bloque de función del controlador de tambor tiene los siguientes parámetros:

Parámetro	Etiqueta	Valor
Número	%DRi	Controlador compacto 0 a 3 Controladores modulares 0 a 7
Número del paso actual	%DRi.S	0-%DRi.S-7. Palabra que se puede leer y escribir. El valor escrito debe ser un valor inmediato decimal. Cuando se escribe, el efecto se produce en la siguiente ejecución del bloque de función.
Número de pasos		1 a 8 (predeterminado)
Retorno a la entrada (o instrucción) del paso 0	R (restablecimiento)	En estado 1, ajusta el controlador del conmutador de tambor al paso 0.
Avance de entrada (o instrucción)	U (arriba)	Con un flanco ascendente, hace que el controlador del conmutador de tambor avance un paso y actualiza los bits de control.
Salida	F (llena)	Indica que el paso actual equivale al último paso definido. El bit asociado %DRi.F se puede verificar (por ejemplo, %DRi.F=1, si %DRi.S= número de pasos configurado - 1).
Bits de control		Salidas o bits internos asociados al paso (16 bits de control) y definidos en el editor de configuración.

Operación de bloque de función del controlador del conmutador de tambor

Introducción

El controlador del conmutador de tambor está compuesto por:

- Una matriz de datos constantes (CAM) organizada en ocho pasos (de 0 a 7) y 16 bits de datos (estado del paso) distribuidos en columnas numeradas (de 0 a F).
- Una lista de bits de control (uno por columna) correspondientes a las salidas %Q0.i o %Q1.i o a los bits internos %Mi. Durante el paso actual, los bits de control toman los estados binarios definidos para dicho paso.

El ejemplo de la tabla siguiente resume las principales características del controlador del conmutador de tambor.

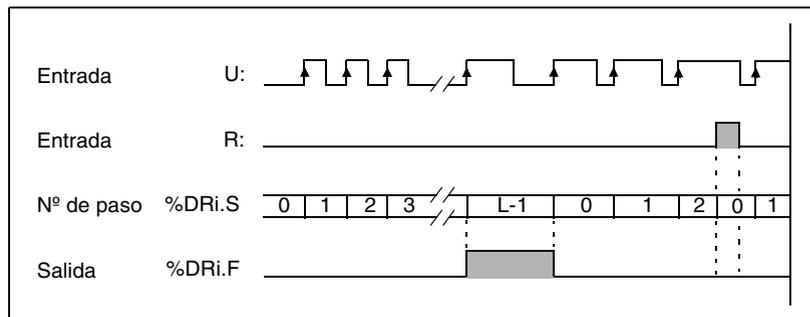
Columna	0	1	2		D	E	F
Bits de control	%Q0.1	%Q0.3	%Q1.5		%Q0.6	%Q0.5	%Q1.0
Paso 0	0	0	1		1	1	0
Paso 1	1	0	1		1	0	0
Paso 5	1	1	1		0	0	0
Paso 6	0	1	1		0	1	0
Paso 7	1	1	1		1	0	0

Operación

En el ejemplo anterior, el paso 5 es el paso actual; los bits de control %Q0.1, %Q0.3 y %Q1.5 se ajustan al estado 1; los bits de control %Q0.6, %Q0.5 y %Q1.0 se ajustan al estado 0. El número del paso actual se incrementa con cada flanco ascendente en la entrada U (o con la activación de la instrucción U). El programa puede modificar el paso actual.

Cronograma

El siguiente diagrama muestra la secuencia de la operación del controlador del conmutador de tambor.



Casos especiales

La tabla siguiente contiene una lista de casos especiales para la operación del controlador del conmutador de tambor.

Caso especial	Descripción
Efectos de un reinicio en frío (%S0=1)	Restablece el controlador del conmutador de tambor al paso 0 (actualizando los bits de control).
Efecto de un reinicio en caliente (%S1=1)	Actualiza los bits de control después del paso actual.
Efecto de un salto del programa	Si el controlador del conmutador de tambor no se explora, los bits de control no se restablecen a 0.
Actualización de los bits de control	Sólo ocurre cuando se produce un cambio de paso o un reinicio en frío o en caliente.

Programación y configuración de los controladores del conmutador de tambor

Introducción

En el siguiente ejemplo de programación y configuración del controlador de tambor, las seis primeras salidas, de %Q0.0 a %Q0.5, se activan sucesivamente cada vez que la entrada %I0.1 se pone a 1. La entrada I0.0 pone las salidas a 0.

Ejemplo de programación

La siguiente ilustración es un bloque de función del controlador del conmutador de tambor con ejemplos de programación reversibles y no reversibles.

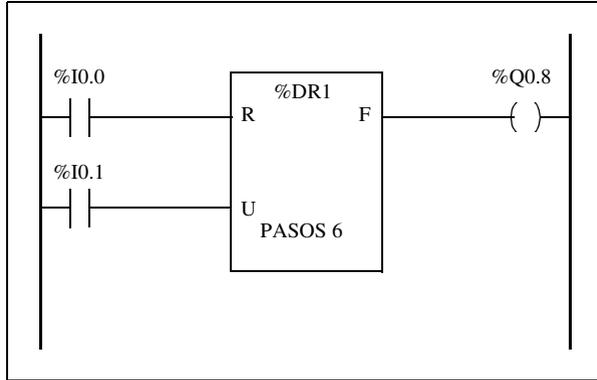


Diagrama Ladder Logic

```

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

Programa reversible

```

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

Programa no reversible

Configuración

Durante la configuración se define la siguiente información:

- El número de pasos: 6
- Los estados de salida (bits de control) para cada paso del controlador del conmutador de tambor.

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

- Asignación de los bits de control.

1 :	%Q0.0	4 :	%Q0.1
2 :	%Q0.2	5 :	%Q0.3
3 :	%Q0.4	6 :	%Q0.5

Bloque de función de contador rápido (%FC)

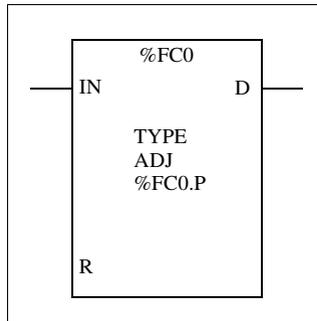
Introducción

El bloque de función de contador rápido (%FC) se puede utilizar como contador progresivo o regresivo. Puede contar el flanco ascendente de las entradas digitales con una frecuencia de hasta 5 kHz. Dado que los contadores rápidos son gestionados por interrupts de hardware específicos, mantener las tasas de muestreo a frecuencia máxima puede variar en función de la configuración de hardware y la aplicación específica.

Los controladores compactos se pueden configurar para utilizar un máximo de tres contadores rápidos, mientras que los controladores modulares sólo pueden usar un máximo de dos. Los bloques de función de contador rápido %FC0, %FC1 y %FC2 utilizan las entradas especializadas %I0.0.2, %I0.0.3 y %I0.0.4 respectivamente. Estos bits no están reservados para su uso exclusivo. Para su ubicación se debe tener en cuenta el uso de otros bloques de función en cuanto a estos recursos especializados.

Ilustración

A continuación se muestra un ejemplo de un bloque de función de contador rápido.



Parámetros

En la tabla siguiente se enumeran los parámetros del bloque de función de contador rápido.

Parámetro	Etiqueta	Descripción
Dirección	TYPE	Ajustado durante la configuración. Se puede ajustar a contador regresivo o progresivo.
Valor preestablecido	%FCi.P	Valor inicial ajustado entre 1 y 65535.
Ajustable	Y/N	Si se pone a Y, es posible modificar el valor preestablecido %FCi.P y %FCi.V con el monitor de operación o el editor de tablas de animación. Si se pone a N no es posible acceder al valor preestablecido.
Valor actual	%FCi.V	El valor actual cuenta de forma progresiva o regresiva de acuerdo con la función de conteo seleccionada. Para el conteo progresivo, el valor actual se restablece a cero y cuenta hasta 65536. Para el conteo regresivo, el valor actual se restablece al valor preestablecido %FCi.P y cuenta hasta cero.
Entrada de habilitación	IN	Con estado 1, el valor actual se actualiza de acuerdo con los pulsos aplicados a la entrada física. Con estado 0, el valor actual se mantiene con su último valor.
Restablecer	%FCi.R	Utilizado para inicializar el bloque. Con estado 1, el valor actual se restablece a 0 si está configurado como un contador progresivo, o a %FCi.P si está configurado como un contador regresivo. El bit de finalización %FCi.D se restablece a su valor predeterminado.
Finalización	%FCi.D	Este bit se pone a 1 cuando %FCi.V alcanza el %FCi.P si está configurado como contador progresivo o %FCi.V alcanza cero si está configurado como contador regresivo. Este bit de sólo lectura sólo se restablece ajustando %FCi.R a 1.

Nota especial

Si se configura como ajustable, la aplicación puede cambiar el valor preestablecido %FCi.P y el valor actual %FCi.V en cualquier momento. Sin embargo, los valores nuevos sólo se tienen en cuenta si está activo el restablecimiento de entrada o con un flanco ascendente de la salida %FCi.D. Esto permite que se realicen conteos sucesivos diferentes sin perder ni un solo pulso.

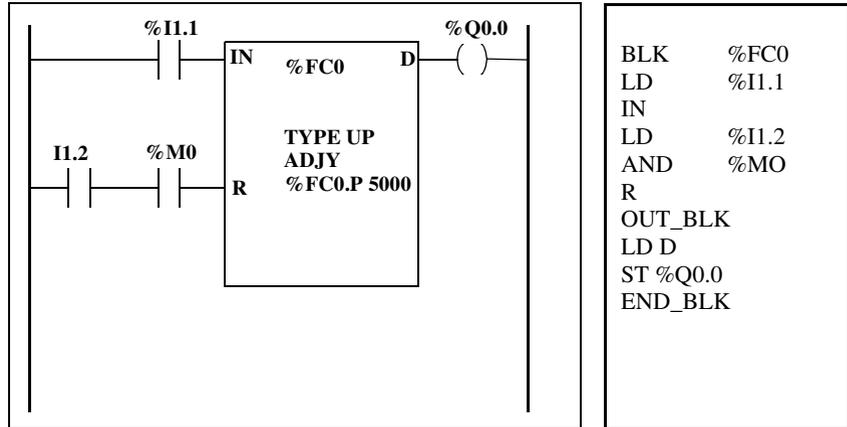
Operación

Si está configurado como contador progresivo, el valor actual se incrementa en 1 con cada flanco ascendente que aparezca en la entrada especializada. Si el valor es igual que el valor preestablecido %FCi.P, el bit de salida Finalización %FCi.D se pone a 1 y en el valor actual %FCi.V se carga cero.

Si está configurado como contador regresivo, el valor actual se reduce en 1 con cada flanco ascendente que aparezca en la entrada especializada. Si el valor es igual a cero, el bit de salida Finalización %FCi.D se pone a 1 y en el valor actual %FCi.P se carga el valor preestablecido.

Configuración y programación

En este ejemplo, la aplicación cuenta un número de elementos hasta 5000 mientras %I1.1 se pone a 1. La entrada para %FC0 es la entrada especializada %I0.0.2. Cuando se alcanza el valor preestablecido, %FC0.D se activa y permanece así hasta que se restablece %FC0.R mediante el resultado de agregar %I1.2 y %M0 con un operador AND lógico.



Casos especiales

La siguiente tabla contiene una lista de casos especiales para programar el bloque de función %FC.

Caso especial	Descripción
Efecto de un reinicio en frío (%S0=1)	Restablece todos los atributos %FC con los valores configurados por el usuario o la aplicación de usuario.
Efecto de un reinicio en caliente (%S1=1)	No tiene ningún efecto.
Efecto de una detención del controlador	El %FC continúa contando con los ajustes de atributo efectivos en el momento en que se detiene el controlador.

Bloque de función de contador muy rápido (%VFC)

Introducción

El bloque de función de contador muy rápido (%VFC) se puede configurar mediante TwidoSoft y realiza cualquiera de las siguientes funciones:

- Contador progresivo/regresivo
- Contador progresivo/regresivo bifásico
- Contador progresivo
- Contador regresivo
- Frecuencímetro

El %VFC proporciona el conteo de entradas digitales con una frecuencia de hasta 20 kHz. Los controladores compactos pueden configurar un contador muy rápido; los controladores modulares pueden configurar hasta dos contadores muy rápidos.

Asignaciones de E/S especializadas

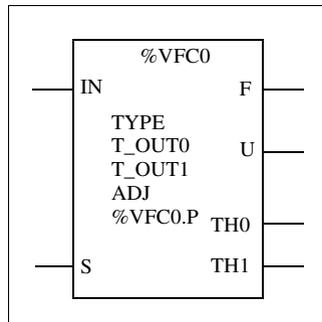
Los bloques de función de contador muy rápido utilizan entradas especializadas y entradas y salidas auxiliares. Estas entradas y salidas no están reservadas para su uso exclusivo. Para su ubicación se debe tener en cuenta el uso de otros bloques de función en cuanto a estos recursos especializados. La siguiente tabla resume estas asignaciones.

		Entradas principales		Entradas auxiliares		Salidas reflejas	
%VFC	Uso seleccionado	Primera entrada (pulsos) IA	Segunda entrada (pulsos o PRO/REG) IB	Entrada preestablecida Ipres	Entrada rápida Ica	Primera salida refleja	Segunda salida refleja
	Contador PROGRESIVO/REGRESIVO	%I0.0.1 (pulsos)	%I0.0.0 (indica PROG=1/RE=0G)	%I0.0.2 opcional	%I0.0.3 opcional	%Q0.0.2 opcional	%Q0.0.3 opcional
	Contador PROGRESIVO/REGRESIVO bifásico	%I0.0.1 (pulsos)	%I0.0.0 (pulso, fase B)	%I0.0.2 opcional	%I0.0.3 opcional	%Q0.0.2 opcional	%Q0.0.3 opcional
	Contador PROGRESIVO	%I0.0.1 (pulsos)	Sin utilizar	%I0.0.2 opcional	%I0.0.3 opcional	%Q0.0.2 opcional	%Q0.0.3 opcional
	Contador REGRESIVO	%I0.0.1 (pulsos)	Sin utilizar	%I0.0.2 opcional	%I0.0.3 opcional	%Q0.0.2 opcional	%Q0.0.3 opcional
	Frecuencímetro	%I0.0.1 (pulsos)	Sin utilizar	Sin utilizar	Sin utilizar	Sin utilizar	Sin utilizar
	Contador PROGRESIVO/REGRESIVO	%I0.0.7 (pulsos)	%I0.0.6 (indica PROG=1/REG=0)	%I0.0.5 opcional	%I0.0.4 opcional	%Q0.0.4 opcional	%Q0.0.5 opcional
	Contador PROGRESIVO/REGRESIVO bifásico	%I0.0.7 (pulsos)	%I0.0.6 (pulso, fase B)	%I0.0.5 opcional	%I0.0.4 opcional	%Q0.0.4 opcional	%Q0.0.5 opcional
	Contador PROGRESIVO	%I0.0.7 (pulsos)	Sin utilizar	%I0.0.5 opcional	%I0.0.4 opcional	%Q0.0.4 opcional	%Q0.0.5 opcional
	Contador REGRESIVO	%I0.0.7 (pulsos)	Sin utilizar	%I0.0.5 opcional	%I0.0.4 opcional	%Q0.0.4 opcional	%Q0.0.5 opcional
	Frecuencímetro	%I0.0.7 (pulsos)	Sin utilizar	Sin utilizar	Sin utilizar	Sin utilizar	Sin utilizar

		Entradas principales		Entradas auxiliares		Salidas reflejas	
%VFC	Uso seleccionado	Primera entrada (pulsos) IA	Segunda entrada (pulsos o PRO/REG) IB	Entrada preestablecida lpres	Entrada rápida lca	Primera salida refleja	Segunda salida refleja
<p>Comentarios:</p> <p>PRO/REG = PROGRESIVO/REGRESIVO Uso opc. = Uso opcional</p> <p>Si no se utiliza, la entrada o salida permanece como E/S digital normal disponible que puede ser gestionada por la aplicación en el ciclo principal.</p> <p>Si se utiliza %I0.0.2, no está disponible %FC0. Si se utiliza %I0.0.3, no está disponible %FC2. Si se utiliza %I0.0.4, no está disponible %FC3.</p>							

Ilustración

A continuación se ofrece una representación de bloque del contador muy rápido



Parámetros

En la tabla siguiente se enumeran las características del bloque de función de contador muy rápido.

Función	Descripción	Valores	Uso de VFC ⁴	Acceso de tiempo de ejecución
Valor actual (%VFCi.V)	Valor actual que se incrementa o se reduce de acuerdo con las entradas físicas y la función seleccionadas. Este valor se puede ajustar o restablecer mediante Establecer entrada (%VFCi.S).	0 -> 65535	CM	Lectura
Valor preestablecido (%VFCi.P)	Sólo se utiliza con la función de conteo progresivo/regresivo, el conteo progresivo y el conteo regresivo.	0 -> 65535	CM o FM	Lectura y escritura ¹
Valor de captura	Sólo se utiliza con la función de conteo progresivo/regresivo, el conteo progresivo y el conteo regresivo.	0 -> 65535	CM	Lectura
Dirección de conteo (%VFCi.U)	Ajustado por el sistema, este bit se utiliza con la función de conteo progresivo/regresivo para indicar al usuario la dirección de conteo. Si se pone a 1, el conteo es progresivo; si se pone a 0, el conteo es regresivo. Como contador regresivo o contador progresivo, %I0.0.0 decide la dirección de %VFC0, y %I0.0.6 la de %VFC1. Para un contador progresivo/regresivo bifásico, la diferencia de fase entre las dos señales determina la dirección. Para %VFC0, %I0.0 está especializada para IB, y %I0.1 para IA. Para %VFC1, %I0.6 está especializada para IB, y %I0.7 para IA.	0 (regresivo) 1 (progresivo)	CM	Lectura
Habilitar salida refleja 0 (%VFCi.R)	Habilitar salida refleja 0	0 (bloqueado) 1 (habilitado)	CM	Lectura y escritura ²
Habilitar salida refleja 1 (%VFCi.S)	Habilitar salida refleja 1	0 (bloqueado) 1 (habilitado)	CM	Lectura y escritura ²
Valor de umbral S0 (%VFCi.S0)	Esta palabra contiene el valor de umbral 0. El significado se define durante la configuración del bloque de función. Tenga en cuenta que este valor debe ser inferior a %VFCi.S1.	0 -> 65535	CM	Lectura y escritura ²
Valor de umbral S1 (%VFCi.S1)	Esta palabra contiene el valor de umbral 0. El significado se define durante la configuración del bloque de función. Tenga en cuenta que este valor debe ser superior a %VFCi.S0.	0 -> 65535	CM	Lectura y escritura ¹
Medida de frecuencia válida (%VFCi.M)	Bit utilizado para determinar si el controlador ha terminado una medición de frecuencia.	0 (no válido) 1 (válido)	FM	Lectura y escritura

Función	Descripción	Valores	Uso de VFC ⁴	Acceso de tiempo de ejecución
Base de tiempo de medida de frecuencia (%VFCi.T)	Elemento de configuración para una base de tiempo de 100 o 1000 milisegundos.	1000 ó 100	FM	Lectura y escritura ¹
Ajustable (Y/N)	Elemento configurable que, cuando está seleccionado, permite al usuario modificar los valores de base de tiempo de medida de frecuencia, de umbral y preestablecidos durante la ejecución.	0 (no) 1 (sí)	CM o FM	No
Habilitar entrada (IN)	Se utiliza para validar o bloquear la función actual.	0 (no)	CM o FM	Lectura y escritura ³
Establecer entrada (S)	Dependiendo de la configuración, con estado 1: <ul style="list-style-type: none"> ● Conteo progresivo/regresivo o conteo regresivo: ajusta el valor actual al valor preestablecido. ● Conteo progresivo: restablece el valor actual a cero. Además, inicializa la operación de las salidas de umbral y tiene en cuenta cualquier modificación por parte del usuario a los valores de umbral ajustados por el monitor de operación o el programa de usuario.	0 ó 1	CM o FM	Lectura y escritura
Salida de desborde (F)	Se ajusta a 1 si %VFCi.V pasa de 0 a 65535. Este valor se borra ajustando el valor preestablecido mediante una entrada digital, la instrucción S o un reinicio en frío.	0 ó 1	CM	Lectura
Umbral Bit 0 (%VFCi.TH0)	Se pone a 1 cuando el valor actual es mayor o igual que el valor de umbral %VFCi.S0. Es recomendable comprobar este bit una sola vez en el programa porque se actualiza a tiempo real. La aplicación de usuario es responsable de la validez del valor en el momento de su uso.	0 ó 1	CM	Lectura
Umbral Bit 1 (%VFCi.TH1)	Se pone a 1 cuando el valor actual es mayor o igual que el valor de umbral %VFCi.S1. Es recomendable comprobar este bit una sola vez en el programa porque se actualiza a tiempo real. La aplicación de usuario es responsable de la validez del valor en el momento de su uso.	0 ó 1	CM	Lectura

Nota:

1. Sólo se puede escribir si Ajustable se pone a uno.
2. El acceso sólo está disponible si se ha configurado.
3. El acceso de lectura y escritura sólo es posible a través de la aplicación, no a través del monitor de operación o del editor de tablas de animación.
4. CM = modo de conteo y FM = modo de frecuencímetro.

Descripción de la función de conteo

La función de conteo muy rápido funciona a una frecuencia máxima de 20 kHz, con un rango de 0 a 65535. Los pulsos que se van a contar se aplican del siguiente modo.

Función	Descripción	%VFC0		IA ... IB	
		IA ... IB		IA ... IB	IA ... IB
Contador progresivo/ regresivo	Los pulsos se aplican a la entrada física; la operación actual (incremento/reducción) está determinada por el estado de la entrada física IB.	%I0.0.1	%I0.0.0	%I0.0.7	%I0.0.6
Contador progresivo/ regresivo bifásico	Las dos fases del codificador se aplican a las entradas físicas IA e IB.	%I0.0.1	%I0.0.0	%I0.0.7	%I0.0.6
Contador progresivo	Los pulsos se aplican a la entrada física IA (IB no se utiliza).	%I0.0.1	NA	%I0.0.7	NA
Contador regresivo	Los pulsos se aplican a la entrada física IA (IB no se utiliza).	%I0.0.1	NA	%I0.0.7	NA

Notas sobre los bloques de función

Las operaciones de incremento o reducción se realizan con el flanco ascendente de los pulsos y sólo si la función de conteo está habilitada.

Hay dos entradas opcionales que se utilizan en el modo de conteo: Ica e IPres. Ica se utiliza para capturar el valor actual (%VFCi.V) y almacenarlo en %VFCi.C. Las entradas Ica están especificadas como %I0.0.3 para %VFC0 y %I0.0.4 para %VFC1 si existe.

Si la entrada IPres está activa, el valor actual resulta afectado de los siguientes modos:

- Para el conteo progresivo, %VFCi.V se restablece a 0.
- Para el conteo regresivo, %VFCi.V se pone a %VFCi.P.
- Para el conteo de frecuencia, %VFCi.V y VFCi.M se ponen a 0.

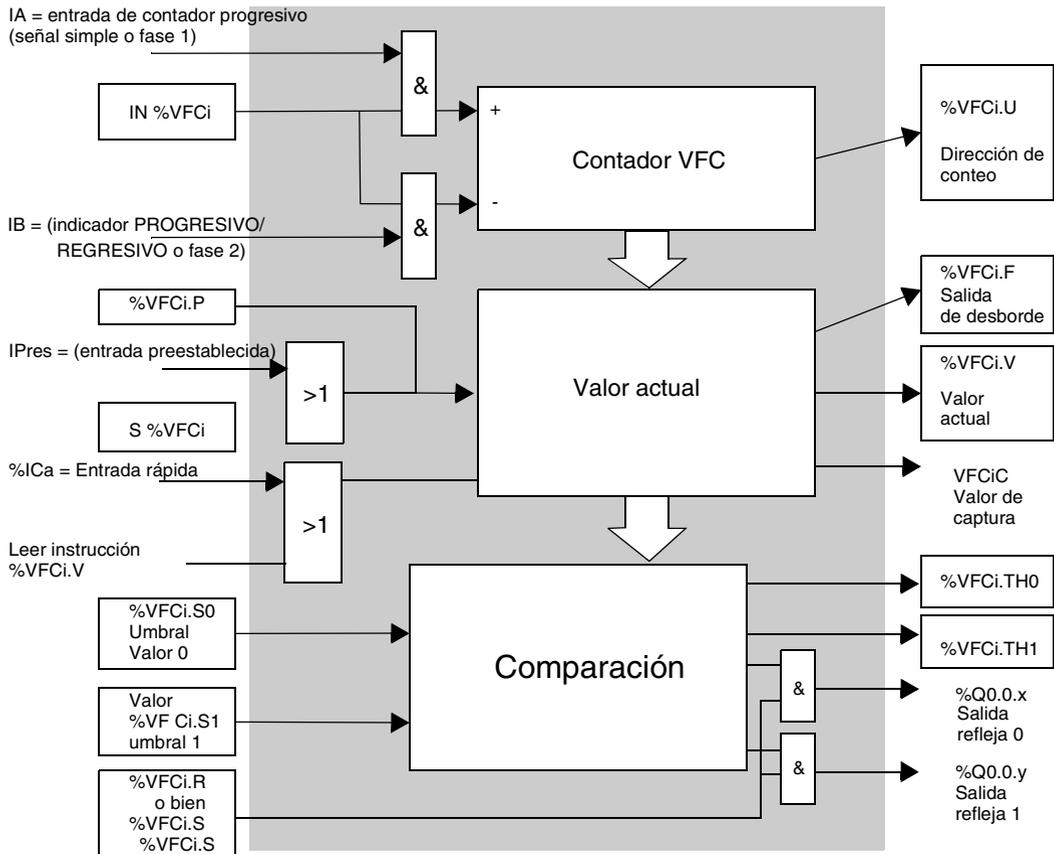
Tenga en cuenta también que %VFCi.F se restablecerá a cero. Las entradas IPres están especificadas como %I0.0.2 para %VFC0 y %I0.0.5 para %VFC1 si existe.

Notas sobre las salidas de los bloques de función

Para todas las funciones, el valor actual se compara con dos umbrales (%VFCi.S0 y % VFCi.S1). De acuerdo con el resultado de esta comparación, dos objetos de bit (%VFCi.TH0 y %VFCi.TH1) se ponen a 1 si el valor actual es mayor o igual que el umbral correspondiente; de lo contrario se restablecen a 0. Las salidas reflejas (si están configuradas) se ajustan de acuerdo con esta comparación. Tenga en cuenta que se puede configurar una, dos o ninguna salida.

%VFC.U es una salida del FB. Indica la dirección de la variación de contador asociada (1 para PROGRESIVO, 0 para REGRESIVO).

Diagrama de función de conteo



Operación de contador progresivo

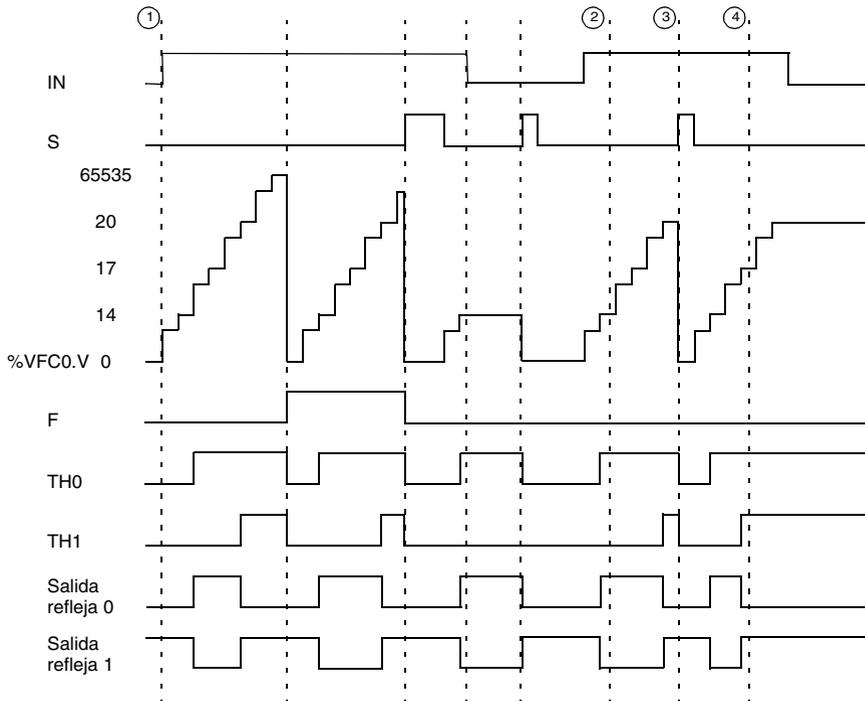
A continuación se incluye un ejemplo del uso de %VFC en modo de contador progresivo. Los siguientes elementos de configuración se han ajustado para este ejemplo.

El valor preestablecido %VFC0.P es 17; el valor de umbral inferior %VFC0.S0 es 14; el umbral superior %VFC0.S1 es 20.

Salida refleja	<%VFC.S0	%VFC0.S0 <= < %VFC0.S1	>= %VFC0.S1
%Q0.0.2		X	
%Q0.0.3	X		X

Gráfico de tiempo

%VFC0.P = 17
 %VFC0.S0 = 14
 %VFC0.S1 = 20



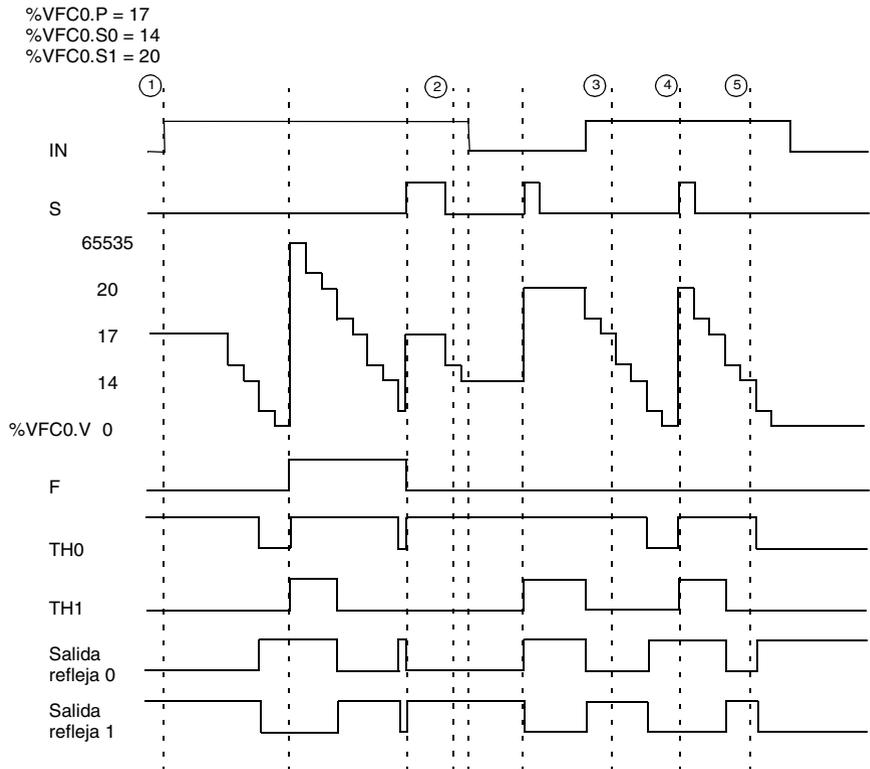
- ① : %VFC0.U = 1 porque VFC es un contador progresivo.
- ② : Cambiar %VFC0.S1 a 17.
- ③ : La entrada activa S hace que el nuevo valor de umbral S1 esté garantizado en el siguiente conteo.
- ④ : Se realiza una captura del valor actual de forma que %VFC0.C = 17.

Operación de contador regresivo

A continuación se incluye un ejemplo del uso de %VFC en modo de contador regresivo. Los siguientes elementos de configuración se han ajustado para este ejemplo.

El valor preestablecido %VFC0.P es 17; el valor de umbral inferior %VFC0.S0 es 14; el umbral superior %VFC0.S1 es 20.

Salida refleja	<%VFC.S0	%VFC0.S0 <= < %VFC0.S1	>= %VFC0.S1
%Q0.0.2		X	
%Q0.0.3	X		X



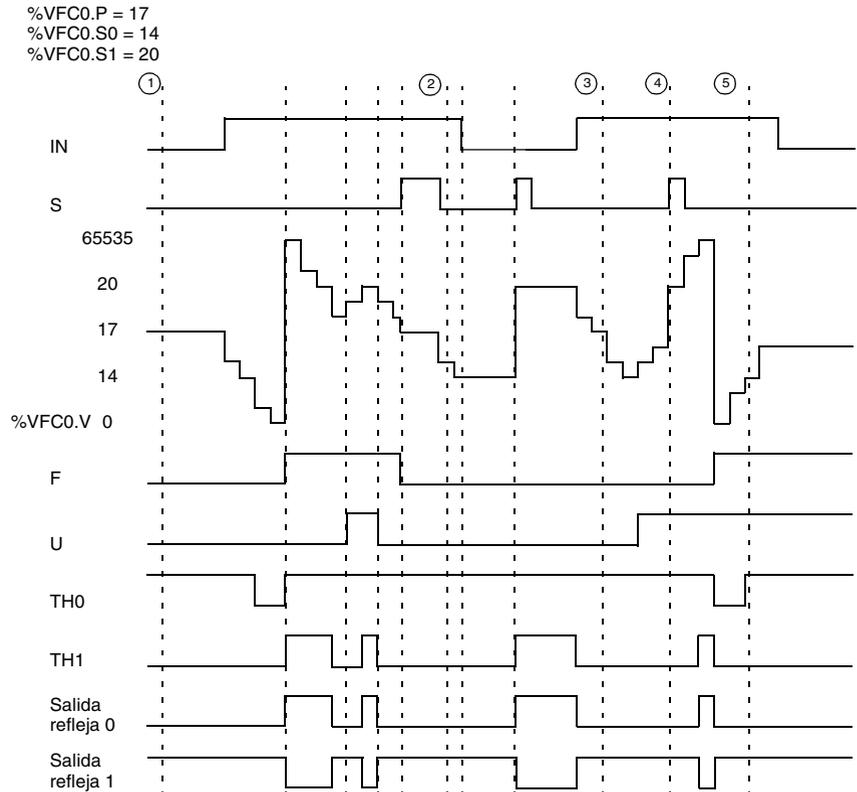
- ① : %VFC0.U = 1 porque VFC es un contador regresivo.
- ② : Cambiar %VFC0.P a 20.
- ③ : Cambiar %VFC0.S1 a 17.
- ④ : La entrada activa S hace que el nuevo valor de umbral S1 esté garantizado en el siguiente conteo.
- ⑤ : Se realiza una captura del valor actual de forma que %VFC0.C = 17.

Operación de contador progresivo/regresivo

A continuación se incluye un ejemplo del uso de %VFC en modo de contador progresivo/regresivo. Los siguientes elementos de configuración se han ajustado para este ejemplo.

El valor preestablecido %VFC0.P es 17; el valor de umbral inferior %VFC0.S0 es 14; el umbral superior %VFC0.S1 es 20.

Salida refleja	<%VFC.S0	%VFC0.S0 <= < %VFC0.S1	%VFC0.S1
%Q0.0.2		X	
%Q0.0.3	X		X



- ① : %VFC0.U = 1 porque VFC es un contador regresivo.
- ② : Cambiar %VFC0.P a 20.
- ③ : Cambiar %VFC0.S1 a 17.
- ④ : La entrada activa S hace que el nuevo valor de umbral S1 esté garantizado en el siguiente conteo.
- ⑤ : Se realiza una captura del valor actual de forma que %VFC0.C = 17.

Descripción de la función de frecuencímetro

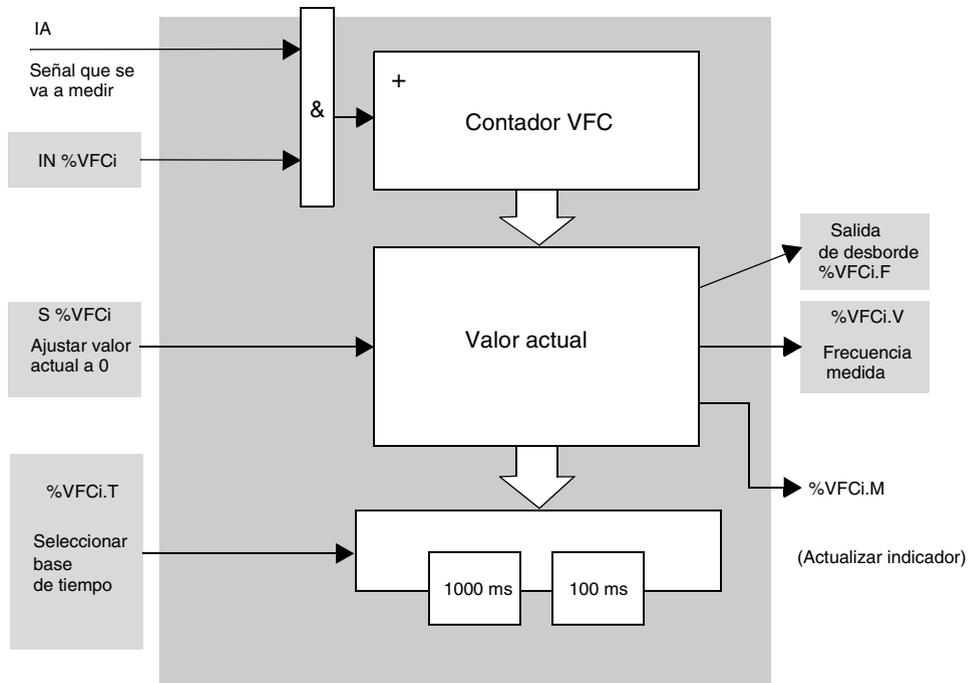
La función de frecuencímetro de un %VFC se utiliza para medir la frecuencia de una señal periódica en Hz en la entrada IA. El rango de frecuencia que se puede medir oscila entre 10 Hz y 20 kHz. El usuario puede elegir entre dos bases de tiempo. La elección se realiza mediante un objeto nuevo %VFC.T (base de tiempo). El valor 100 equivale a una base de tiempo de 100 ms; el valor 1000, a una base de tiempo de 1 segundo.

Base de tiempo	Rango de medición	Precisión	Actualización
100 ms	100 Hz a 20 kHz	0,05% para 20 kHz; 10% para 100 Hz	10 veces por segundo
1 s	10 Hz a 20 kHz	0,005% para 20 kHz; 10% para 10 Hz	Una vez por segundo

El objeto %VFC.M (medida de frecuencia válida) se pone a 1 para indicar que la medición ha concluido.

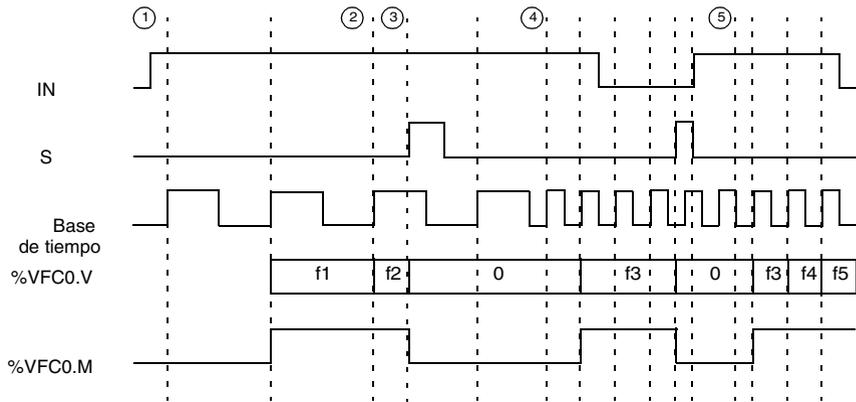
Diagrama de la función de frecuencímetro

Diagrama de función de frecuencímetro



Operación del frecuencímetro

A continuación se incluye un ejemplo de cronograma en el que se utiliza el %VFC en modo de frecuencímetro.



- ① : La primera medición de frecuencia comienza aquí.
- ② : El valor de frecuencia actual se actualiza.
- ③ : La entrada S activa ajusta %VFC0.V a 0.
- ④ : Cambiar %VFC0.T a 100 ms: este cambio cancela la medición actual e inicia otra.
- ⑤ : %VFC0.M se pone a 0 por el usuario.

Casos especiales

La siguiente tabla contiene una lista de casos especiales para programar el bloque de función %VFC.

Caso especial	Descripción
Efecto de un reinicio en frío (%S0=1)	Restablece todos los atributos %VFC con los valores configurados por el usuario o la aplicación de usuario.
Efecto de un reinicio en caliente (%S1=1)	No tiene ningún efecto.
Efecto de una detención del controlador	El %VFC deja de funcionar y las salidas permanecen en su estado actual.

Transmisión/recepción de mensajes - La instrucción de intercambio (EXCH)

Introducción

Un controlador Twido se puede configurar para comunicarse con dispositivos slave de Modbus o puede enviar y/o recibir mensajes en modo carácter (ASCII).

TwidoSoft ofrece las siguientes funciones para este tipo de comunicaciones:

- Instrucción EXCH para transmitir/recibir mensajes
- Bloque de función de control de intercambio (%MSG) para controlar el intercambio de datos

Cuando se procesa una instrucción EXCH, el controlador Twido utiliza el protocolo configurado para el puerto especificado. Cada puerto de comunicaciones se puede configurar para protocolos distintos o para el mismo. A la instrucción EXCH o al bloque de función %MSG de cada puerto de comunicaciones se accede añadiendo el número de puerto (1 ó 2).

Instrucción EXCH

La instrucción EXCH permite a los controladores Twido enviar o recibir información dirigida a o procedente de dispositivos ASCII. El usuario define una tabla de palabras (%MWi:L o %KW i:L) que contiene los datos que se van a enviar o recibir (hasta 64 palabras de datos en la transmisión o recepción). El formato de la tabla de palabras se describe en los apartados correspondientes a cada protocolo. Un intercambio de mensajes se realiza utilizando la instrucción EXCH.

Sintaxis

A continuación se muestra el formato de la instrucción EXCH.

[EXCHx %MWi:L] o [EXCHx %KW i:L]

Donde: x = número de puerto (1 ó 2); L = número de palabras de la tabla de palabras. Los valores de la tabla de palabras internas %MWi:L son del tipo de i+L - 255.

El controlador Twido debe finalizar el intercambio desde la primera instrucción EXCHx antes de que se pueda iniciar una segunda instrucción de intercambio. El bloque de función %MSG debe utilizarse cuando se envíen varios mensajes.

Bloque de función de control de intercambio (%MSG)

Introducción

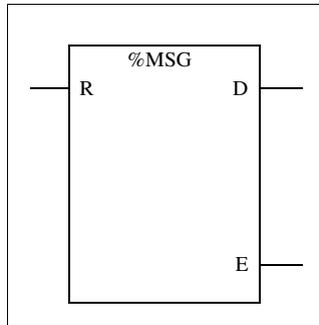
El bloque de función %MSG gestiona el intercambio de datos. Tiene tres funciones:

- Comprobación de errores de comunicación
La comprobación de errores verifica que la longitud de bloque (tabla de palabras) programada con la instrucción EXCH sea lo suficientemente larga como para contener la longitud del mensaje que se va a enviar (compárela con la longitud programada en el byte de menor valor de la primera palabra de la tabla de palabras).
- Coordinación de varios mensajes
Para asegurar la coordinación cuando se envíen varios mensajes, el bloque de función %MSG proporciona la información requerida para determinar cuándo está completo un mensaje anterior.
- Transmisión de mensajes prioritarios
El bloque de función %MSG permite la detención de la transmisión del mensaje actual para permitir el envío inmediato de un mensaje urgente.

La programación del bloque de función %MSG es opcional.

Ilustración

A continuación se muestra un ejemplo de un bloque de función %MSG.



Parámetros

En la tabla siguiente se enumeran los parámetros del bloque de función %MSG.

Parámetro	Etiqueta	Valor
Entrada de restablecimiento (o instrucción)	R	Con estado 1, reinicializa la comunicación: %MSG.E = 0 y %MSG.D = 1.
Salida de com. finalizada	%MSG.D	Con estado 1, com. lista si: <ul style="list-style-type: none"> ● Final de la transmisión (si hay transmisión) ● Final de la recepción (carácter final recibido) ● Error ● Restablecimiento del bloque Con estado 0, solicitud en curso
Salida de fallo (error)	%MSG.E	Con estado 1, com. lista si: <ul style="list-style-type: none"> ● Comando incorrecto ● Tabla mal configurada ● Carácter incorrecto recibido (velocidad, paridad, etc.) ● Tabla de recepción llena (no actualizada) Con estado 0, longitud de mensaje correcta, conexión correcta

Si se produce un error durante el uso de una instrucción EXCH, los bits %MSG.D y %MSG.E se ponen a 1 y la palabra de sistema %SW63 contiene el código de error para el puerto 1, y %SW64 contiene el código de error para el puerto 2. Consulte "*Palabras de sistema (%SW), p. 342*".

Entrada de restablecimiento (R)

Si la entrada de restablecimiento se pone a 1:

- Se detiene la transmisión de todos los mensajes.
- La salida de fallo (error) se restablece a 0.
- El bit de finalización se pone a 1.

Se puede enviar un mensaje nuevo.

Salida de fallo (error) (%MSG.E)

La salida de error se pone a 1 debido a un error de programación de comunicaciones o a un error de transmisión de mensajes. La salida de error se pone a 1 si el número de bytes definido en el bloque de datos asociado a la instrucción EXCH (palabra 1, byte de menor valor) es mayor que 128 (80 en formato hexadecimal). La salida de error también se pone a 1 si existe algún problema al enviar un mensaje Modbus a un dispositivo Modbus. En este caso, el usuario deberá comprobar el cableado y asegurarse de que el dispositivo de destino admita la comunicación Modbus.

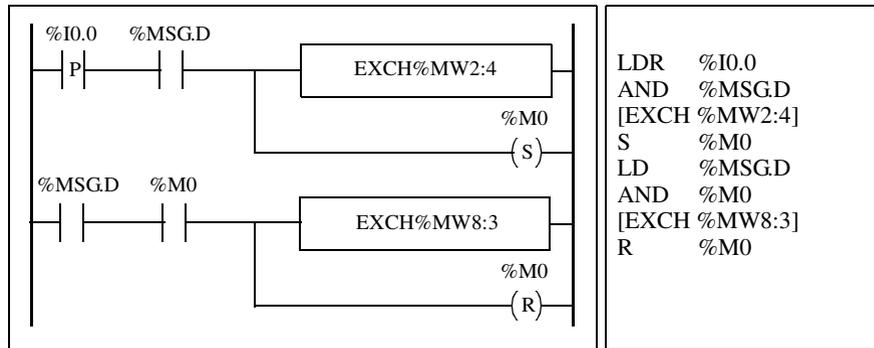
Salida de comunicación finalizada (%MSG.D)

Cuando la salida de comunicación finalizada se pone a 1, el controlador Twido está listo para enviar otro mensaje. Se recomienda utilizar el bit %MSG.D cuando se envían varios mensajes. Si no se utiliza, se puede perder algún mensaje.

Transmisión de varios mensajes sucesivos

La ejecución de la instrucción EXCH activa un bloque de mensaje en el programa de aplicación. El mensaje se transmite si el bloque de mensaje no está activo (%MSG.D = 1). Si se envían varios mensajes en el mismo ciclo, sólo se transmite el primer mensaje. El usuario es responsable de gestionar la transmisión de varios mensajes utilizando el programa.

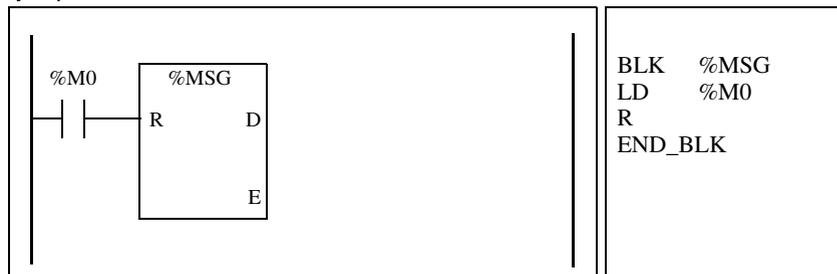
Ejemplo de transmisión de dos mensajes sucesivos.



Reinicialización de intercambios

Un intercambio se cancela activando la entrada (o instrucción) R. Esta entrada inicializa la comunicación y restablece a salida %MSG.E a 0 y la salida %MSG.D a 1. Es posible reinicializar un intercambio si se detecta un error.

Ejemplo de reinicialización de un intercambio.



Casos especiales

En la tabla siguiente se enumeran los casos especiales para programar el bloque de función %MSG.

Caso especial	Descripción
Efecto de un reinicio en frío (%S0=1)	Fuerza la reinicialización de la comunicación.
Efecto de un reinicio en caliente (%S1=1)	No tiene ningún efecto.
Efecto de una detención del controlador	Si se está transmitiendo un mensaje, el controlador detiene la transferencia y reinicializa las salidas %MSG.D y %MSG.E.

13.2 Funciones de reloj

Presentación

Vista general En esta sección se describen las funciones de gestión de tiempo para los controladores Twido.

Contenido Esta sección contiene los siguientes apartados:

Apartado	Página
Funciones de reloj	324
Fecha y hora	325
Fijación de la fecha y la hora	328
Establecimiento de la fecha y la hora	330

Funciones de reloj

Introducción

Los controladores Twido disponen de una función de reloj de fecha/hora que requiere la opción Reloj de tiempo real (RTC) y que ofrece lo siguiente:

- **Fechadores:** se utilizan para controlar acciones a horas predefinidas o calculadas.
- **Fijación de fecha y hora:** se utiliza para asignar fechas y horas a eventos y para medir la duración de los eventos.

Al reloj de fecha/hora de Twido se puede acceder seleccionando **Fechadores** del menú **Software** de TwidoSoft. Además, el reloj de fecha/hora se puede ajustar mediante un programa. Los ajustes del reloj siguen funcionando hasta 30 días después de desconectar el controlador si la batería se ha cargado durante seis horas consecutivas como mínimo antes de desconectar el controlador.

El reloj de fecha/hora tiene formato de 24 horas y tiene en cuenta los años bisiestos.

Valor de corrección RTC

El valor de corrección RTC es necesario para que el RTC funcione de forma adecuada. Cada unidad RTC dispone de su propio valor de corrección escrito en la unidad. Este valor se puede configurar en TwidoSoft mediante la opción **Configurar RTC** del cuadro de diálogo **Operaciones del controlador**.

Fechadores

Introducción

Los fechadores se utilizan para controlar las acciones en un mes, día u hora predefinidos. Puede utilizarse un máximo de 16 fechadores y no es necesario utilizar ninguna entrada del programa.

Nota: Compruebe el bit del sistema %S51 para confirmar que la opción Reloj de tiempo real (RTC) está instalada. Consulte *Bits del sistema (%S)*, p. 336. La opción RTC también es necesaria para utilizar fechadores.

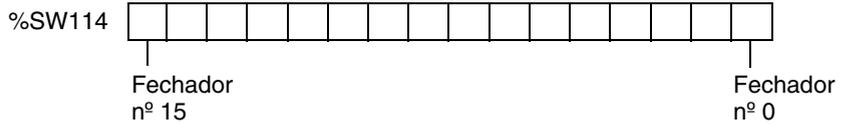
Parámetros

La siguiente tabla enumera los parámetros para un fechador:

Parámetro	Formato	Función/Rango
Número de fechador	n	n = 0 a 15
Configurado	Casilla de verificación	Marque esta casilla para configurar el número de fechador seleccionado.
Bit de salida	%Qx.y.z	El fechador activa la asignación de salida: %Mi o %Qj.k. Esta salida se establece como 1 cuando la fecha y la hora actuales están entre el ajuste del inicio del periodo activo y el ajuste del final del periodo activo.
Mes de inicio	Enero a diciembre	Mes para iniciar el fechador.
Mes de finalización	Enero a diciembre	Mes para finalizar el fechador.
Fecha de inicio	1 - 31	Día del mes para iniciar el fechador.
Fecha de finalización	1 - 31	Día del mes para finalizar el fechador.
Hora de inicio	hh:mm	Momento del día, hora (0 a 23) y minutos (0 a 59), para iniciar el fechador.
Tiempo de parada	hh:mm	Momento del día, hora (0 a 23) y minutos (0 a 59), para finalizar el fechador.
Día de la semana	Lunes - Domingo	Casillas de verificación que identifican el día de la semana para activar el fechador.

**Habilitación de
fechadores**

Los bits de la palabra del sistema %SW114 habilitan (bit establecido a 1) o inhabilitan (bit establecido a 0) el funcionamiento de cada uno de los 16 fechadores. Asignación de fechadores en %SW114:



De forma predeterminada (o después de un reinicio en frío) todos los bits de esta palabra del sistema se establecen a 1. La utilización de estos bits por parte del programa es opcional.

**Salida de los
fechadores**

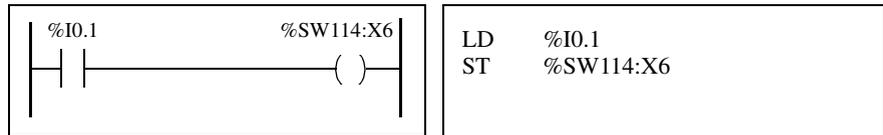
Si se asigna la misma salida a varios fechadores (%Mi o %Qj,k), se asignará el OR de los resultados de cada fechador a este objeto (es posible tener varios "rangos de funcionamiento" para la misma salida).

Ejemplo

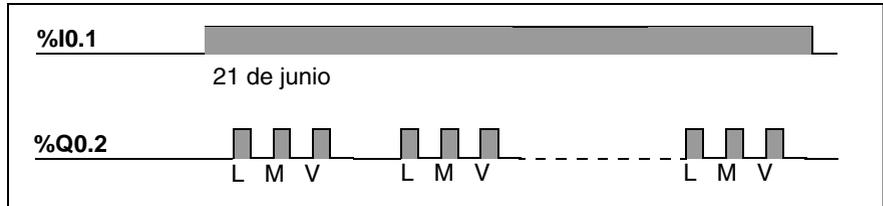
La siguiente tabla muestra los parámetros correspondientes a un ejemplo de un programa de vaporización para un mes veraniego.

Parámetro	Valor	Descripción
Fechador	6	Número 6 de fechador
Bit de salida	%Qx.y.z	Activa la salida %Qx.y.z
Mes de inicio	Junio	Inicia la actividad en junio
Mes de finalización	Septiembre	Detiene la actividad en septiembre
Fecha de inicio	21	Inicia la actividad el día 21 de junio
Fecha de finalización	21	Detiene la actividad el día 21 de septiembre
Día de la semana	Lunes, miércoles, viernes	Realiza la actividad en lunes, miércoles y viernes
Hora de inicio	21:00	Inicia la actividad a las 21:00
Tiempo de parada	22:00	Detiene la actividad a las 22:00

Utilizando el siguiente programa, el fechador puede inhabilitarse mediante un conmutador o un detector de humedad cableado a la entrada %I0.1.



El siguiente cronograma muestra la activación de la salida %Q0.2.

**Asignación de fecha y hora por parte del programa**

La fecha y la hora están disponibles en las palabras del sistema %SW50 a %SW53 (consulte *Palabras de sistema (%SW)*, p. 342). Por lo tanto, es posible realizar una marca de hora y fecha en el programa controlador estableciendo comparaciones aritméticas entre la fecha y hora actuales y los valores o palabras inmediatos %MWi (o %KWi), que pueden contener valores teóricos.

Fijación de la fecha y la hora

Introducción

Las palabras del sistema %SW50 a %SW53 contienen la fecha y la hora actuales en formato BCD (consulte *Revisión del código BCD*, p. 266, que resulta útil para realizar visualizaciones o transmisiones a un dispositivo periférico. Estas palabras del sistema pueden utilizarse para almacenar la fecha y la hora de un evento (consulte *Palabras de sistema (%SW)*, p. 342).

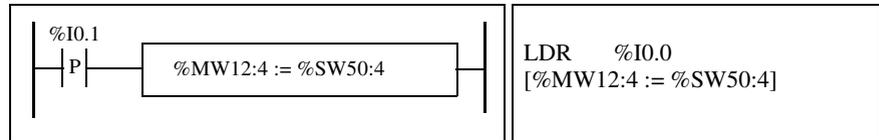
Nota: La fecha y la hora pueden establecerse utilizando la visualización del operador opcional (consulte *Reloj de fecha/hora*, p. 152).

Fechado de un evento

Para fechar un evento, es suficiente utilizar las operaciones de asignación, transferir el contenido de las palabras del sistema a palabras internas y, a continuación, procesar estas palabras internas (por ejemplo, la transmisión a una unidad de visualización mediante la instrucción EXCH).

Ejemplo de programación

El siguiente ejemplo muestra cómo fechar un flanco ascendente en una entrada %I0.1.



Una vez detectado un evento, la tabla de palabras contendrá:

Codificación	Byte de mayor valor	Byte de menor valor
%MW12	Segundo	Día de la semana (1)
%MW13	Hora	Minuto
%MW14	Mes	Día
%MW15	Siglo	Año

Nota: (1) 0 = Lunes, 1 = Martes, 2 = Miércoles, 3 = Jueves, 4 = Viernes, 5 = Sábado, 6 = Domingo.

Ejemplo de una tabla de palabras

Ejemplo de datos para el lunes 19 de abril de 2002 a las 13:40:30:

Palabra	Valor (hexadecimal)	Significado
%MW12	3000	30 segundos, 00 = Lunes
%MW13	1340	13 horas, 40 minutos
%MW14	0419	04 = 19 de abril
%MW15	2002	2002

Fecha y hora de la última parada

Las palabras del sistema %SW54 a %SW57 contienen la fecha y la hora de la última parada y la palabra %SW58 contiene el código que muestra la causa de la última parada, en formato BCD (consulte *Palabras de sistema (%SW)*, p. 342).

Establecimiento de la fecha y la hora

Introducción

Puede actualizar los ajustes de fecha y hora utilizando uno de los siguientes métodos:

- TwidoSoft

Utilice el cuadro de diálogo **Establecer hora**. Este diálogo está disponible en el cuadro de diálogo **Operaciones del controlador** que aparece seleccionando **Operaciones del controlador** en el menú **Controlador** (consulte el Manual de funcionamiento de TwidoSoft).

- Palabras de sistema

Utilice las palabras del sistema %SW50 a %SW53 o la palabra de sistema %SW59.

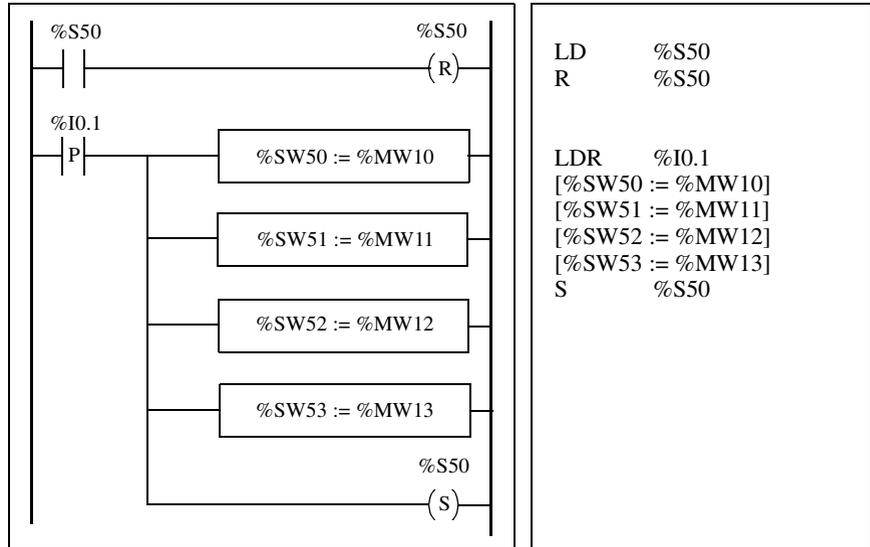
Los ajustes de fecha y hora sólo pueden actualizarse cuando está instalado el cartucho RTC opcional (TWDXCPRTC) en el controlador.

Utilización de %SW 50 a %SW53

Para establecer la fecha y la hora utilizando las palabras de sistema %SW50 a %SW53, el bit %S50 debe ponerse a 1, lo que produce lo siguiente:

- Cancela la actualización de las palabras %SW50 a %SW53 mediante el reloj interno.
- Transmite los valores escritos en las palabras %SW50 a %SW53 al reloj interno.

Ejemplo de programación



Las palabras %MW10 a %MW13 contendrán la nueva fecha y hora en formato BCD (consulte *Revisión del código BCD*, p. 266) y corresponderán a la siguiente codificación de las palabras %SW50 a 53.

La tabla de palabras debe contener la nueva fecha y hora.

Codificación	Byte de mayor valor	Byte de menor valor
%MW10	Segundo	Día de la semana (1)
%MW11	Hora	Minuto
%MW12	Mes	Día
%MW13	Siglo	Año

Nota: (1) 0 = Lunes, 1 = Martes, 2 = Miércoles, 3 = Jueves, 4 = Viernes, 5 = Sábado, 6 = Domingo.

Ejemplo de datos para el lunes 19 de abril de 2002.

Palabra	Valor (hexadecimal)	Significado
%MW10	3000	30 segundos, 00 = Lunes
%MW11	1340	13 horas, 40 minutos
%MW12	0419	04 = 19 de abril
%MW13	2002	2002

Utilización de %SW59

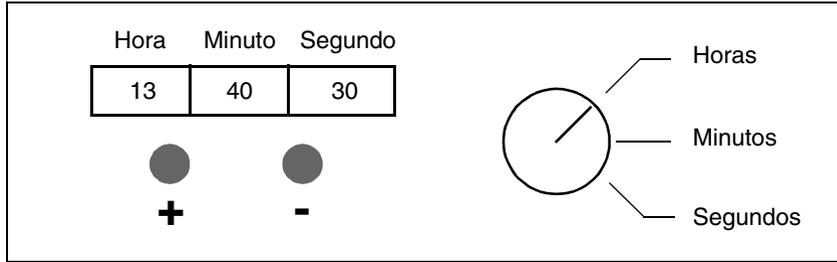
Otro método para actualizar la fecha y la hora es utilizar el bit de sistema %S59 y la palabra de sistema de ajuste de fecha %SW59.

El establecimiento del bit %S59 en 1 permite ajustar la fecha y la hora actuales mediante la palabra %SW59 (consulte *Palabras de sistema (%SW)*, p. 342).

%SW59 aumenta o reduce cada uno de los componentes de fecha y hora en un flanco ascendente.

Ejemplo de aplicación

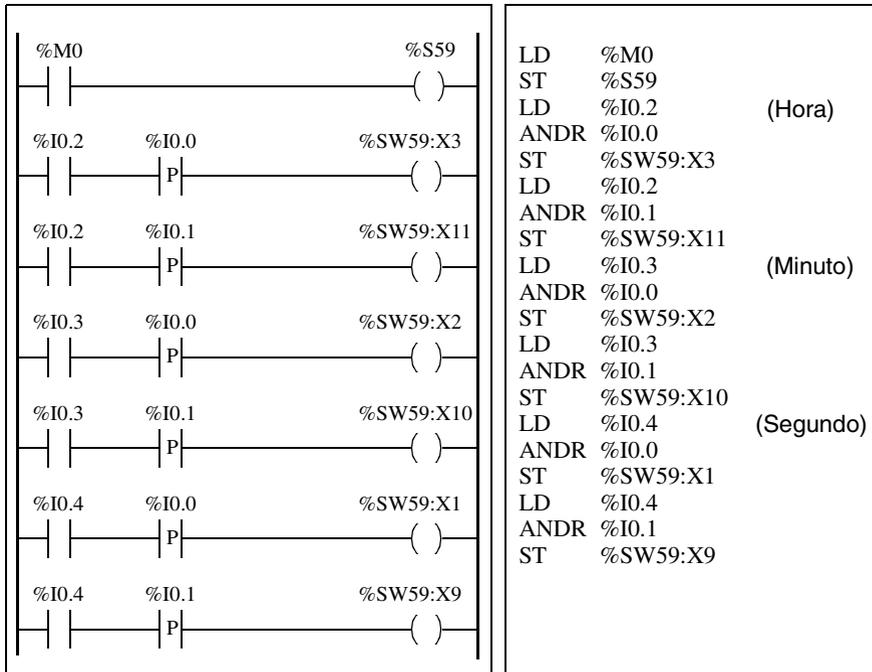
El siguiente panel frontal está creado para modificar la hora, los minutos y segundos del reloj interno.



Descripción de los controles:

- El conmutador de Horas/Minutos/Segundos selecciona la visualización de la hora para cambiarla utilizando las entradas %I0.2, %I0.3 y %I0.4 respectivamente.
- El botón de comando "+" aumenta la visualización del tiempo seleccionado utilizando la entrada %I0.0.
- El botón de comando "-" reduce la visualización del tiempo seleccionado utilizando la entrada %I0.1.

El siguiente programa lee las entradas del panel y establece el reloj interno.



Bits de sistema y palabras de sistema

14

Presentación

Vista general

Este capítulo contiene una vista general de los bits de sistema y las palabras de sistema que se pueden utilizar para crear programas de control para controladores Twido.

Contenido:

Este capítulo contiene los siguiente apartados:

Apartado	Página
Bits del sistema (%S)	336
Palabras de sistema (%SW)	342

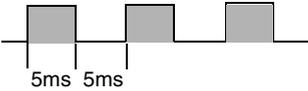
Bits del sistema (%S)

Introducción

La siguiente sección ofrece información detallada acerca de la función de los bits del sistema y el modo en el que se controlan.

Descripción detallada

La tabla siguiente contiene una vista general de los bits del sistema y del modo en el que se controlan.

Bit de sistema	Función	Descripción	Estado de inic.	Control
%S0	Inicio en frío	Normalmente ajustado a 0. Se pone a 1 por: <ul style="list-style-type: none"> Una recuperación de alimentación con pérdida de datos (fallo de batería) El programa de aplicación o el editor de tablas de animación El monitor de operación Este bit se pone a 1 durante el primer ciclo completo. El sistema lo restablece a 0 antes del siguiente ciclo.	0	S o U->S
%S1	Inicio en caliente	Normalmente ajustado a 0. Se restablece a 1 por: <ul style="list-style-type: none"> Una recuperación de alimentación sin pérdida de datos El programa de aplicación o el editor de tablas de animación El monitor de operación El sistema lo restablece a 0 al final de un ciclo completo.	0	S o U->S
%S4 %S5 %S6 %S7	Base de tiempo: 10 ms Base de tiempo: 100 ms Base de tiempo: 1 s Base de tiempo: 1 min	Los cambios de estado de estos bits se controlan mediante un reloj interno. No están sincronizados con el ciclo del controlador. Ejemplo: %S4 	-	S
%S8	Congelación de salida	Inicialmente se establece en 1, se puede ajustar a 0 mediante el programa o el terminal (en el editor de tablas de animación): <ul style="list-style-type: none"> En estado 1, borra las salidas durante el estado NO CONFIG. En estado 0, permite pruebas de cableado durante el estado NO CONFIG. 	1	U

Bit de sistema	Función	Descripción	Estado de inic.	Control
%S9	Restablecer salidas	Normalmente no ajustado. Se puede ajustar a 1 mediante el programa o el terminal (en el editor de tablas de animación): <ul style="list-style-type: none"> ● En estado 1, las salidas se fuerzan a 0 cuando el controlador se encuentra en modo de ejecución. ● En estado 0, las salidas se actualizan de forma normal. 	0	U
%S10	Error de E/S	Normalmente ajustado a 1. El sistema lo pone a 0 cuando detecta un error de E/S.	1	S
%S11	Desborde de watchdog	Normalmente está ajustado a 0. El sistema lo pone a 1 cuando el tiempo de ejecución del programa (tiempo de ciclo) supera el tiempo de ciclo máximo (watchdog de software). El desborde de watchdog hace que el controlador cambie a PARADA.	0	S
%S12	Controlador en ejecución	Este bit refleja el estado de ejecución del controlador. El sistema pone el bit a 1 cuando el controlador está en ejecución, y a 0 para detención, inicio o cualquier otro estado.	0	S
%S13	Primer ciclo	Normalmente está ajustado a 0. El sistema lo pone a 1 durante el primer ciclo después de que el controlador cambie a EJECUCIÓN.	1	S
%S17	Desborde de carga	Normalmente ajustado a 0. El sistema lo pone a 1: <ul style="list-style-type: none"> ● En caso de desborde de carga durante una operación aritmética sin signo (resto). ● Durante una operación de desplazamiento o rotación indica que la salida de un bit está a 1. Se debe verificar mediante el programa de aplicación después de cada operación que entrañe un riesgo de desborde. Se debe poner a 0 si se produce un desborde. 	0	S->U
%S18	Error o desborde aritmético	Normalmente está ajustado a 0. Se pone a 1 en caso de desborde cuando se realiza una operación de 16 bits, es decir: <ul style="list-style-type: none"> ● Un resultado mayor que + 32767 o menor que - 32768 ● División entre 0 ● La raíz cuadrada de un número negativo ● Conversión ITB o BTI no significativa: valor BCD fuera de los límites Se debe verificar mediante el programa de aplicación después de cada operación que entrañe un riesgo de desborde. El usuario deberá ponerlo a 0 si se produce un desborde.	0	S->U
%S19	Desborde de periodo de ciclo (ciclo periódico)	Normalmente ajustado a 0. El sistema lo pone a 1 en caso de desborde de periodo de ciclo (tiempo de ciclo mayor que el periodo definido por el usuario en la configuración o programado en %SW0). El usuario se encarga de restablecer este bit a 0.	0	S->U

Bit de sistema	Función	Descripción	Estado de inic.	Control
%S20	Desborde de índice	Normalmente ajustado a 0. Se pone a 1 cuando la dirección del objeto indexado es inferior a 0 o mayor que el tamaño máximo de un objeto. Se debe verificar mediante el programa de aplicación después de cada operación que entrañe un riesgo de desborde. Se debe poner a 0 si se produce un desborde.	0	S->U
%S21	Inicialización GRAFCET	Normalmente ajustado a 0. Se pone a 1 por: <ul style="list-style-type: none"> ● Un reinicio en frío, %S0=1. ● El programa de aplicación, sólo en la parte de procesamiento previo del programa, mediante una instrucción Establecer (S %S21) o una bobina de establecimiento -(S)- %S21. ● El terminal. En estado 1, causa la inicialización GRAFCET. Los pasos activos se desactivan y los pasos iniciales se activan. El sistema lo pone a 0 después de la inicialización GRAFCET.	0	U->S
%S22	RESTABLECIMIENTO GRAFCET	Normalmente ajustado a 0. Sólo se puede poner a 1 por el programa durante el procesamiento previo. En estado 1, causa la desactivación de los pasos activos de todo el proceso GRAFCET. El sistema lo pone a 0 cuando se inicia la ejecución del procesamiento secuencial.	0	U->S
%S23	Preajustar y congelar GRAFCET	Normalmente ajustado a 0. Sólo se puede poner a 1 por el programa de aplicación en el módulo de procesamiento previo del programa. En estado 1, valida el preajuste del diagrama GRAFCET. Si se mantiene este bit a 1 se congela GRAFCET (se congela el diagrama). El sistema lo pone a 0 cuando se inicia la ejecución del procesamiento secuencial para garantizar que el diagrama GRAFCET abandona el estado de congelación.	0	U->S
%S24	Monitor de operación	Normalmente está ajustado a 0. El usuario puede poner a 1 este bit. <ul style="list-style-type: none"> ● En estado 0, el monitor de operación funciona con normalidad. ● En estado 1, el monitor de operación está congelado, permanece en la pantalla actual, el parpadeo se bloquea y el procesamiento de la tecla de entrada se detiene. 	0	U->S
%S50	Actualización de la fecha y la hora mediante las palabras %SW50 a 53	Normalmente está ajustado a 0. Este bit se puede poner a 1 o a 0 por medio del programa o del monitor de operación. <ul style="list-style-type: none"> ● En estado 0 se pueden leer la fecha y la hora. ● En estado 1 se pueden actualizar la fecha y la hora. 	0	U->S

Bit de sistema	Función	Descripción	Estado de inic.	Control
%S51	Estado del reloj de fecha/hora	Normalmente está ajustado a 0. Este bit se puede poner a 1 o a 0 por medio del programa o del monitor de operación. <ul style="list-style-type: none"> En estado 0 se ajustan la fecha y la hora. En estado 1, el usuario debe ajustar la fecha y la hora. Cuando este bit se pone a 1, los datos del reloj de fecha/hora no son válidos. Es posible que la fecha y la hora no se hayan configurado nunca, que la batería tenga poca carga o que la constante de corrección del controlador no sea válida. Cuando el estado 1 cambia al estado 0, se fuerza la escritura de la constante de corrección en el RTC.	0	U->S
%S59	Actualización de la fecha y la hora mediante la palabra %SW59	Normalmente está ajustado a 0. Este bit se puede poner a 1 o a 0 por medio del programa o del monitor de operación. <ul style="list-style-type: none"> En estado 0, la fecha y la hora permanecen sin cambios. En estado 1, la fecha y la hora se incrementan o se reducen de acuerdo con los bits de control ajustados en %SW59. 	0	U
%S69	Visualización del LED STAT de usuario	En estado 0, el LED STAT está apagado. En estado 1, el LED STAT está encendido.	0	U
%S70	Actualización de datos en el bus AS-i	El sistema ajusta este bit a 1 al final de cada ciclo del controlador o al final del ciclo de exploración del bus AS-i. En el arranque, indica que todos los datos se han actualizado al menos una vez y, por lo tanto, es significativo. El usuario debe restablecer este bit a 0.	0	S->U
%S73	Conmutación a modo de protección en el bus AS-i	Normalmente está ajustado a 0. El usuario pone este bit a 1 para conmutar al modo de protección en el bus AS-i. Antes de esta operación, el bit ya debe haber estado en 1. Este bit sólo se utiliza en una comprobación del sistema de cableado y no tiene ninguna aplicación dentro del controlador.	0	S
%S74	Guardar la configuración en el bus AS-i	Normalmente está ajustado a 0. El usuario pone este bit a 1 para guardar la configuración actual en el bus AS-i. Este bit sólo se utiliza en una comprobación del sistema de cableado y no tiene ninguna aplicación dentro del controlador.	0	S
%S96	Programa de copia de seguridad correcto	Este bit se puede leer en cualquier momento (ya sea por el programa o durante el ajuste), en particular después de un inicio en frío o un reinicio en caliente. <ul style="list-style-type: none"> En estado 0, el programa de copia de seguridad no es válido. En estado 1, el programa de copia de seguridad es válido. 	0	S

Bit de sistema	Función	Descripción	Estado de inic.	Control
%S97	Operación para guardar %MW correcta	Este bit se puede leer en cualquier momento (ya sea por el programa o durante el ajuste), en particular después de un inicio en frío o un reinicio en caliente. <ul style="list-style-type: none"> En estado 0, la operación para guardar %MW no es correcta. En estado 1, la operación para guardar %MW es correcta. 	0	S
%S100	Conexión del cable de comunicaciones TwidoSoft	Indica si el cable de comunicaciones TwidoSoft está conectado. <ul style="list-style-type: none"> En estado 1, el cable de comunicaciones TwidoSoft puede estar o no conectado. En estado 0, el cable de conexiones remotas TwidoSoft está conectado. 	-	S
%S110	Intercambio de conexión remota	Este bit se restablece a 0 por el programa o el terminal. <ul style="list-style-type: none"> En estado 1 para un master, se completan todos los intercambios de conexión remota (sólo E/S remotas). En estado 1 para un slave, se completa el intercambio con el master. 	0	S->U
%S111	Intercambio único de conexión remota	<ul style="list-style-type: none"> En estado 0 para un master, se completa un único intercambio de conexión remota. En estado 0 para un slave, se detecta un único intercambio de conexión remota. En estado 1 para un master, está activo un único intercambio de conexión remota. En estado 1 para un slave, se detecta un único intercambio de conexión remota. 	0	S
%S112	Activación de la conexión remota	<ul style="list-style-type: none"> En estado 0 para un master, la conexión remota está bloqueada. En estado 1 para un master, la conexión remota está habilitada. 	0	U
%S113	Configuración/ funcionamiento de conexión remota	<ul style="list-style-type: none"> En estado 0 para un master o slave, la configuración o el funcionamiento de la conexión remota es correcta. En estado 1 para un master, la configuración o el funcionamiento de la conexión remota tiene un error. En estado 1 para un slave, la configuración o el funcionamiento de la conexión remota tiene un error. 	0	S->U
%S118	Error de E/S remotas	Normalmente está ajustado a 1. Se pone a 0 cuando se detecta un error de E/S en la conexión remota.	1	S
%S119	Error de E/S locales	Normalmente está ajustado a 1. Se pone a 0 cuando se detecta un error de E/S locales (base o ampliación). %SW118 determina la naturaleza del error. Se restablece a 1 cuando desaparece el fallo.	1	S

**Tabla de
descripción de
abreviaturas**

Abreviatura	Descripción
S	Controlado por el sistema
U	Controlado por el usuario
U->S	Puesto a 1 por el usuario y restablecido a 0 por el sistema
S->U	Puesto a 1 por el sistema y restablecido a 0 por el usuario

Palabras de sistema (%SW)

Introducción

La siguiente sección ofrece información detallada acerca de la función de las palabras de sistema y el modo en el que se controlan.

Descripción detallada

La siguiente tabla proporciona información detallada acerca de la función de las palabras de sistema y el modo en el que se controlan.

Palabras de sistema	Función	Descripción	Control
%SW0	Periodo de ciclo del controlador (tarea periódica)	Modifica el periodo de ciclo del controlador definido en la configuración mediante el programa de aplicación en el editor de tablas de animación.	U
%SW6	Estado del controlador	Estado del controlador: 0 = NO CONFIG 2 = DETENIDO 3 = EJECUCIÓN 4 = PARADO	S

Palabras de sistema	Función	Descripción	Control
%SW7	Estado del controlador	<p>Bit [0] Copia de seguridad/restauración en curso</p> <p>Bit [1] Configuración del controlador correcta</p> <p>Bit [3..2] Bits de estado de EEPROM:</p> <ul style="list-style-type: none"> ● 00 = Ningún cartucho ● 01 = Cartucho EEPROM de 32 kB ● 10 = Cartucho EEPROM de 64 kB ● 11 = Reservado para un uso futuro <p>Bit [4] Aplicación de RAM diferente de EEPROM (1 = sí)</p> <p>Bit [5] Aplicación de RAM diferente del cartucho (1 = sí)</p> <p>Bit [6] Algunas tareas de equipos están en modo de detención</p> <p>Bit [7] Controlador reservado</p> <p>Bit [8] Aplicación en modo de protección de escritura</p> <p>Bit [9] No utilizado</p> <p>Bit [10] Segundo puerto serie instalado</p> <p>Bit [11] Segundo tipo de puerto serie (0 = EIA RS-232, 1 = EIA RS-485)</p> <p>Bit [12] Aplicación válida en la memoria interna (1 = sí)</p> <p>Bit [13] Aplicación válida en el cartucho (1 = sí)</p> <p>Bit [14] Aplicación válida en la memoria RAM (1 = sí)</p> <p>Bit [15] Listo para ejecutar</p>	S
%SW11	Tiempo de vigilancia watchdog del software	Se inicializa con el tiempo de vigilancia watchdog máximo. El valor (10 a 500 ms) se define mediante la configuración.	U
%SW18- %SW19	Contador temporizador absoluto de 100 ms	Contador temporizador absoluto de 100 ms. %SW18 representa los bytes de menor valor y %SW19 representa los bytes de mayor valor de la palabra doble.	S y U
%SW30	Último tiempo de ciclo	Muestra el tiempo de ejecución del último ciclo de exploración del controlador (en ms). Nota: Este tiempo corresponde al transcurrido entre el inicio (adquisición de entradas) y la finalización (actualización de salidas) de un ciclo de exploración.	S

Palabras de sistema	Función	Descripción	Control
%SW31	Tiempo máximo de ciclo	Muestra el tiempo de ejecución del ciclo de exploración más largo del controlador desde el último inicio en frío (en ms). Nota: Este tiempo corresponde al transcurrido entre el inicio (adquisición de entradas) y la finalización (actualización de salidas) de un ciclo de exploración.	S
%SW32	Tiempo mínimo de ciclo	Muestra el tiempo de ejecución del ciclo de exploración más corto del controlador desde el último inicio en frío (en ms). Nota: Este tiempo corresponde al transcurrido entre el inicio (adquisición de entradas) y la finalización (actualización de salidas) de un ciclo de exploración.	S

Palabras de sistema	Función	Descripción	Control										
%SW49 %SW50 %SW51 %SW52 %SW53	Función de los fechadores	Función de los fechadores (RTC): palabras que contienen los valores de fecha y hora actuales (en BCD). <table border="1"> <tr> <td>%SW49</td> <td>xN Día de la semana (N=0 para el lunes)</td> </tr> <tr> <td>%SW50</td> <td>00SS Segundos</td> </tr> <tr> <td>%SW51</td> <td>HHMM Hora y minuto</td> </tr> <tr> <td>%SW52</td> <td>MMDD Mes y día</td> </tr> <tr> <td>%SW53</td> <td>SSAA Siglo y año</td> </tr> </table> <p>El sistema controla estas palabras cuando el bit %S50 está ajustado a 0. Tanto el programa de aplicación como el terminal pueden escribir estas palabras cuando el bit %S50 se establece en 1.</p>	%SW49	xN Día de la semana (N=0 para el lunes)	%SW50	00SS Segundos	%SW51	HHMM Hora y minuto	%SW52	MMDD Mes y día	%SW53	SSAA Siglo y año	S y U
%SW49	xN Día de la semana (N=0 para el lunes)												
%SW50	00SS Segundos												
%SW51	HHMM Hora y minuto												
%SW52	MMDD Mes y día												
%SW53	SSAA Siglo y año												
%SW54 %SW55 %SW56 %SW57	Función de los fechadores	Función de los fechadores (RTC). Palabras de sistema que contienen la fecha y la hora del último fallo de alimentación o de la última detención del controlador (en BCD): <table border="1"> <tr> <td>%SW54</td> <td>SS Segundos</td> </tr> <tr> <td>%SW55</td> <td>HHMM Hora y minuto</td> </tr> <tr> <td>%SW56</td> <td>MMDD Mes y día</td> </tr> <tr> <td>%SW57</td> <td>SSAA Siglo y año</td> </tr> </table>	%SW54	SS Segundos	%SW55	HHMM Hora y minuto	%SW56	MMDD Mes y día	%SW57	SSAA Siglo y año	S		
%SW54	SS Segundos												
%SW55	HHMM Hora y minuto												
%SW56	MMDD Mes y día												
%SW57	SSAA Siglo y año												

Palabras de sistema	Función	Descripción	Control	
%SW58	Código de la última detención	Muestra el código que origina la causa de la última detención.	S	
		1 =	Flanco de la entrada Ejecutar/detener	
		2 =	Detención cuando falla el software (rebasamiento del ciclo del controlador)	
		3 =	Comando de detención	
		4 =	Corte de corriente	
		5 =	Detención cuando se produce un fallo del hardware	

Palabras de sistema	Función	Descripción	Control		
%SW59	Ajustar fecha actual	Ajusta la fecha actual. Contiene dos grupos de 8 bits para ajustar la fecha actual. La operación siempre se realiza en el flanco ascendente del bit. Esta palabra se habilita mediante el bit %S59.	U		
		Incrementar	Reducir	Parámetro	
		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	Meses	
		bit 6	bit 14	Años	
bit 7	bit 15	Siglos			
%SW60	Valor de corrección RTC	Valor de corrección del reloj de tiempo real (RTC)	U		

Palabras de sistema	Función	Descripción	Control
%SW63	Código de error del bloque EXCH1	<p>Si se produce un error al utilizar el bloque EXCH, los bits de salida %MSG.D y %MSG.E cambian a 1. Esta palabra de sistema contiene el código de error. Los valores posibles son los siguientes:</p> <ul style="list-style-type: none"> ● 0: Ningún error, intercambio correcto ● 1: Búfer de transmisión demasiado grande ● 2: Búfer de transmisión demasiado pequeño ● 3: Tabla demasiado pequeña ● 4:í ● 5: Timeout transcurrido ● 6: Error de transmisión ● 7: Comando ASCII incorrecto (sólo modo ASCII) ● 8: No utilizado ● 9: Error de recepción (sólo modo ASCII) ● 10: Tabla %KWi prohibida ● 11: Offset de transmisión mayor que tabla de transmisión ● 12: Offset de recepción mayor que tabla de recepción ● 13: Procesamiento EXCH detenido por el controlador <p>Esta palabra se establece en 0 cada vez que se utiliza el bloque EXCH.</p>	S
%SW64	Código de error del bloque EXCH2	Igual que %SW63	S
%SW67	Función y tipo de controlador	<p>Contiene la siguiente información:</p> <ul style="list-style-type: none"> ● Bits del tipo de controlador [0 -11] ● 8B0 = TWDLCAA10DRF ● 8B1 = TWDLCAA16DRF ● 8B2 = TWDLMDA20DUK/DTK ● 8B3 = TWDLCAA24DRF ● 8B4 = TWDLMDA40DUK/DTK ● 8B6 = TWDLMDA20DRT ● Bit 12 no utilizado = 0 ● Bits de direcciones de conexión remota [13-15] ● 000 = controlador master ● 001 - 111 = controlador remoto 1-7 ● 001 = dirección 1 ● 111 = dirección 7 	S

Palabras de sistema	Función	Descripción	Control
%SW76 a %SW79	Contadores regresivos 1-4	Estas 4 palabras sirven como temporizadores de 1 ms. El sistema hace que disminuyan individualmente cada ms si tienen un valor positivo. Esto ofrece un recuento regresivo de los 4 contadores regresivos, que es igual a un rango de funcionamiento de 1 ms a 32.767 ms. Si se establece el bit 15 en 1, se puede detener la reducción.	S and U

Palabras de sistema	Función	Descripción	Control
%SW96	Comando y/o diagnóstico de %MWs especificado en %SW97 es mayor que el número mún y %MW.	<p>Bit [0] Este bit se establece mediante la lógica de aplicación para indicar que las palabras de memoria, %MWi, deben almacenarse en la memoria EEPROM. El Executive vuelve a poner este bit a 0 cuando haya comenzado el proceso de almacenamiento de %MW, y no cuando haya finalizado.</p> <p>Bit [1] Este bit se establece mediante el firmware para indicar la finalización del proceso de guardado. Esto implica que cuando el bit es 1, se habrá completado cualquier solicitud de guardado en EEPROM. Este bit se pone a cero en la siguiente solicitud de guardado en EEPROM.</p> <p>Bit [2] Cuando se establece en 1, indica que se ha producido un error durante la última solicitud de guardado o restauración. Para obtener información adicional, consulte los bits 8, 9, 10 y 14.</p> <p>Bit [6] El controlador contiene una aplicación válida (1 = s).</p> <p>Bit [8] El número de %MWs especificado en %SW97 es mayor que el número máximo configurado realmente en la aplicación de usuario (1 = sí).</p> <p>Bit [9] El número de %MWs especificado en %SW97 es mayor que el número máximo de palabras de memoria permitido por TwidoSoft (1 = sí).</p> <p>Bit [10] Diferencia entre la memoria RAM interna y la memoria EEPROM interna (1 = sí).</p> <p>Bit [14] Se ha producido un fallo de escritura EEPROM (1 = sí).</p>	

Palabras de sistema	Función	Descripción	Control
%SW97	Comando o diagnósticos para la función guardar/restaurar	<p>Este valor representa el número físico de las palabras de memoria %MW para que se guarden únicamente en la memoria EEPROM interna. No se utiliza en una restauración de palabras de memoria. Cuando este número sea 0, no se guardarán las palabras de memoria. El usuario debe establecer el programa de lógica de aplicación pues, de lo contrario, se ajustará a 0 en la aplicación del controlador, excepto en el siguiente caso:</p> <p>En un inicio en frío, esta palabra se ajusta a -1 si la memoria Flash EEPROM interna no ha guardado el archivo de la palabra de memoria %MW. En el caso de un inicio en frío donde la memoria Flash EEPROM contiene un archivo de palabra de memoria %MW, el valor del número de palabras de memoria guardadas en el archivo debe establecerse en esta palabra de sistema %SW97.</p>	U

Palabras de sistema	Función	Descripción	Control
%SW111	Estado de conexión remota	<p>Dos bits para cada controlador remoto (sólo master):</p> <p>x0-5:0 - controlador remoto 1-6 ausente 1- controlador remoto 1-6 presente x6:0 - controlador remoto 7 ausente 1- controlador remoto 7 presente x8-13:0 - E/S remotas detectadas en el controlador remoto 1-6 1 - controlador peer detectado en el controlador remoto 1-6 x14:0 - E/S remotas detectadas en el controlador remoto 7 1 - controlador peer detectado en el controlador remoto 7</p>	S

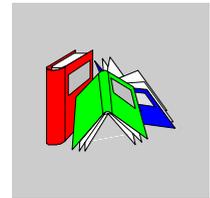
Palabras de sistema	Función	Descripción	Control
%SW112	Código de error de configuración/ funcionamiento de conexión remota	0 - operaciones correctas 1 - timeout detectado (slave) 2 - error de suma de control detectado (slave) 3 - discrepancia de configuración (slave) El sistema lo establece y el usuario es quien debe restablecerlo.	S
%SW113	Configuración de conexión remota	Dos bits para cada controlador remoto (sólo master): x0-5:0 - controlador remoto 1-6 no configurado 1- controlador remoto 1-6 configurado x6:0 - controlador remoto 7 no configurado 1- controlador remoto 7 configurado x8-13:0 - E/S remotas configuradas como controlador remoto 1-6 1 - controlador peer configurado como controlador remoto 1-6 x14:0 - E/S remotas configuradas como controlador remoto 7 1 - controlador peer configurado como controlador remoto 7	S
%SW114	Habilitar fechadores (RTC)	Habilita o bloquea el funcionamiento de los fechadores (RTC) por parte del programa de aplicación o el monitor de operación. Bit 0: 1 = habilita el fechador nº 0 Bit 15: 1 = habilita el fechador nº 15 Inicialmente, todos los fechadores están habilitados y su estado inicial es 0. En caso de que no haya ningún fechador configurado, el valor predeterminado será FFFF.	S y U
%SW118	Palabra de estado del controlador base	Muestra los fallos detectados en el controlador master. Bit 9: 0= error externo o de com. Bit 12: 0= RTC no instalado Bit 13: 0= fallo de configuración (extensión de E/S configurada, pero ausente o defectuosa). Los otros bits de esta palabra se ajustan a 1 y se reservan. Para un controlador sin errores, el valor de esta palabra es FFFFh.	S

Palabras de sistema	Función	Descripción	Control
%SW120	Estado funcional del módulo de E/S de ampliación	Un bit por módulo. Dirección 0 = Bit 0 1 = Incorrecto 0 = Correcto	S

Tabla de descripción de abreviaturas

Abreviatura	Descripción
S	Controlado por el sistema
U	Controlado por el usuario

Glosario



!

% Prefijo que identifica las direcciones de memoria interna en el controlador utilizadas para almacenar el valor de variables de programa, constantes, E/S, etc.

A

Administrador de recursos

Componente de TwidoSoft que controla los requisitos de memoria de una aplicación durante la programación y configuración realizando un seguimiento de las referencias a los objetos de software realizadas por una aplicación. Se considera que la aplicación hace referencia a un objeto si se utiliza como operando en una instrucción de lista o escalón de Ladder. Muestra la información de estado relativa al porcentaje de memoria total utilizada y proporciona una advertencia en caso de que la memoria está reduciéndose. Consulte "Indicador de uso de memoria".

Analizar programa

Comando que compila un programa y comprueba la existencia de errores en el mismo: errores de sintaxis y estructura, símbolos sin las correspondientes direcciones, recursos utilizados por el programa y que no están disponibles, y errores debidos a que el programa no se adapta a la memoria del controlador disponible. Los errores se muestran en el visualizador de errores de programa.

Aplicación

Una aplicación TwidoSoft se compone de un programa, datos de configuración, símbolos y documentación.

Archivo de aplicación

Las aplicaciones Twido se almacenan como archivos de tipo .twd.

ASCII Código estándar americano para el intercambio de información (del inglés "American Standard Code for Information Interchange"). Protocolo de comunicación que utiliza siete bits para representar caracteres alfanuméricos incluidos números, letras y algunos caracteres gráficos y de control.

Autómata programable Controlador Twido. Existen dos tipos de controladores: compacto y modular.

B

Bloque de función Unidad de programa de entradas y variables organizadas para calcular los valores de las salidas basadas en una función definida como un temporizador o un contador.

Bobina Elemento del diagrama Ladder que representa una salida del controlador.

Borrar Este comando elimina el almacenamiento de la aplicación y tiene dos opciones: elimina el contenido de la RAM del controlador, la EEPROM interna del controlador y un cartucho de copia de seguridad opcional instalado, o bien elimina el contenido de un cartucho de copia de seguridad opcional instalado.

Bus de ampliación Los módulos de ampliación de E/S se conectan al controlador base utilizando este bus.

C

Cabecera de escalón Panel que aparece directamente sobre un escalón Ladder y que puede utilizarse para documentar el propósito del escalón.

Carga automática Función siempre habilitada que permite transferir automáticamente una aplicación desde un cartucho de copias de seguridad a la RAM del controlador en caso de aplicaciones dañadas o perdidas. Durante el arranque, el controlador compara la aplicación presente en la RAM del controlador con la aplicación del cartucho de memoria de copias de seguridad opcional (si está instalado). En caso de que exista alguna diferencia, la copia del cartucho de copias de seguridad se copia en el controlador y en la EEPROM interna. Si no está instalado el cartucho de copias de seguridad, la aplicación de la EEPROM interna se copiará en el controlador.

Cartucho de memoria	Cartuchos de memoria de copias de seguridad que pueden utilizarse para realizar una copia de seguridad y restaurar una aplicación (datos de configuración y programa). Hay dos tamaños disponibles: 32 Kbytes y 64 Kbytes.
Comentarios	Los comentarios son el texto introducido para documentar el propósito de un programa. Para los programas Ladder, introduzca hasta tres líneas de texto en la cabecera de escalón para describir el propósito del escalón. Cada línea puede tener de 1 a 64 caracteres. Para los programas de lista, introduzca texto en una línea de programa no numerada n. Los comentarios deben introducirse entre paréntesis y asteriscos como: (*COMENTARIOS AQUÍ*).
Conexión remota	Bus master/slave de alta velocidad diseñado para transferir una pequeña cantidad de datos entre el controlador master y hasta siete controladores slave remotos. Hay dos tipos de controladores remotos que pueden configurarse para transferir datos a un controlador master: controlador peer que puede transferir datos de la aplicación o controlador remoto de E/S que puede transferir datos de E/S. Una red de conexión remota se compone de una mezcla de ambos tipos.
Constantes	Unidad de memoria como un bit o palabra cuyo contenido no puede ser modificado por el programa en ejecución.
Contacto	Elemento del diagrama Ladder que representa una entrada en el controlador.
Contador	Bloque de función utilizado para contar eventos (conteo progresivo o regresivo).
Contadores muy rápidos	Bloque de función que proporciona un conteo más rápido que el disponible con bloques de función de contadores y contadores rápidos. Un contador muy rápido puede contar a una velocidad de hasta 20 kHz.
Contadores rápidos	Bloque de función que proporciona un conteo progresivo y regresivo más rápido que el disponible en el bloque de función Contadores. Un contador rápido puede contar a una velocidad de hasta 5 kHz.
Controlador	Controlador programable Twido. Existen dos tipos de controladores: compacto y modular.
Controlador compacto	Tipo de controlador Twido que proporciona una configuración simple e integrada con ampliación limitada. Modular es el otro tipo de controlador Twido.
Controlador del conmutador de tambor	Bloque de función que funciona de un modo similar al de un controlador del conmutador de tambor electromecánico con cambios de pasos asociados a eventos externos.
Controlador master	Controlador Twido configurado para ser el master en una red de conexión remota.

Controlador modular	Tipo de controlador Twido que ofrece una configuración flexible con funciones de ampliación. Compacto es el otro tipo de controlador Twido.
Controlador peer	Controlador Twido configurado para ser el slave en una red de conexión remota. Una aplicación puede ejecutarse en la memoria del controlador peer y el programa puede acceder a los datos de E/S locales y de ampliación; sin embargo, los datos de E/S no pueden pasar al controlador master. El programa que está ejecutándose en el controlador peer pasa información al controlador master utilizando palabras de red (%INW y QNW).
Controlador remoto	Controlador Twido configurado para comunicarse con un controlador master en una red de conexión remota.
Copia de seguridad	Comando que copia la aplicación de la RAM del controlador en la EEPROM interna del controlador y en el cartucho de memoria de copias de seguridad opcional (si está instalado).

D

Datos variables	Consulte "variable".
Detener	Comando que hace que el controlador detenga la ejecución de un programa de aplicación.
Direcciones	Registros internos del controlador utilizados para almacenar valores para variables de programa, constantes, E/S, etc. Las direcciones se identifican con un prefijo con el símbolo de porcentaje (%). Por ejemplo, %I0.1 especifica una dirección de la memoria RAM del controlador que contiene el valor para el canal de entrada 1.

E

Editor de configuración	Ventana especializada de TwidoSoft utilizada para gestionar la configuración de hardware y software.
Editor de Ladder Logic	Ventana TwidoSoft especializada y utilizada para editar un programa Ladder.
Editor de lista	Sencillo editor de programas utilizado para crear y editar un programa de lista.

Editor de tablas de animación	Ventana especializada en la aplicación TwidoSoft para ver y crear tablas de animación.
EEPROM	Memoria de sólo lectura programable que se puede borrar de forma eléctrica. Twido tiene una EEPROM interna y un cartucho de memoria EEPROM externa opcional.
Entrada con retención	La aplicación captura y graba los pulsos entrantes para un posterior examen.
Escalón	Un escalón se introduce entre dos barras potenciales en una cuadrícula compuesta por un grupo de elementos gráficos unidos entre sí mediante conexiones horizontales y verticales. Las dimensiones máximas de un escalón son siete filas y once columnas.
Escalón de lista Ladder	Muestra partes de un programa de lista no reversibles a lenguaje Ladder.
Estado del monitor	El estado operativo de TwidoSoft que se muestra en la barra de estado cuando se conecta un PC a un controlador en modo de protección contra escritura.
Estado inicial	Estado de funcionamiento de TwidoSoft que aparece en la barra de estado cuando se inicia TwidoSoft o no tiene ninguna aplicación abierta.
Estado offline	El estado operativo de TwidoSoft que se muestra en la barra de estado cuando un PC no está conectado a un controlador.
Estado online	El estado operativo de TwidoSoft que se muestra en la barra de estado cuando un PC está conectado a un controlador.
Estados de funcionamiento	Indica el estado de TwidoSoft y se muestra en la barra de estado. Hay cuatro estados de funcionamiento: inicial, offline, online y supervisar.
Executive Loader	Aplicación Windows de 32 bits utilizada para descargar un nuevo programa de firmware Executive en un controlador Twido.
Exploración	Un controlador examina un programa y realiza básicamente tres funciones principales. En primer lugar, lee las entradas y sitúa estos valores en la memoria. A continuación, ejecuta una instrucción del programa de aplicación cada vez y almacena los resultados en memoria. Finalmente, utiliza los resultados para actualizar las salidas.

F

Fechadores	Bloque de función utilizado para programar funciones de fecha y hora con el fin de controlar eventos. Requiere la opción Reloj de tiempo real.
FIFO	First In, First Out. Bloque de función utilizado para operaciones de cola.
Firmware Executive	El firmware Executive es el sistema operativo gracias al cual se ejecutan las aplicaciones y que gestiona el funcionamiento del controlador.
Forzado	Establecer voluntariamente las entradas y salidas del controlador en 0 ó 1 aunque los valores reales sean diferentes. Se utiliza para depurar mientras se anima un programa.
Funciones de fecha y hora	Permiten el control de eventos por mes, día y hora. Consulte "Fechadores".

G

Grafcet	Un programa escrito en lenguaje Grafcet se compone de pasos que contienen una descripción gráfica y estructurada de la operación de automatización secuencial. Los símbolos gráficos sencillos se utilizan para describir la secuencia de pasos.
----------------	--

I

Indicador de uso de memoria	Parte de la barra de estado en la ventana principal de TwidoSoft que muestra un porcentaje de la memoria total del controlador utilizada por una aplicación. Proporciona una advertencia cuando la memoria es baja.
Inicio en caliente	Inicio por parte del controlador después de una pérdida de alimentación sin modificar la aplicación. El controlador regresa al estado existente antes de la pérdida de alimentación y completa la exploración en curso. Todos los datos de la aplicación quedan intactos. Esta función sólo está disponible en controladores modulares.

Inicio en frío o reinicio	Inicio por parte del controlador con todos los datos inicializados con los valores predeterminados y el programa iniciado desde el comienzo con todas las variables eliminadas. Todos los parámetros de software y hardware se inicializan. Un reinicio en frío puede producirse automáticamente debido a un fallo en la alimentación (sólo los controladores compactos) o a que se ha cargado una nueva aplicación en la RAM del controlador. Todos los controladores compactos o cualquier otro sin sostén de batería siempre se inician en frío.
Init	Comando que establece todos los valores de datos en estados iniciales. El controlador debe estar en modo Detener o Error.
Instancia	Objeto exclusivo de un programa que pertenece a un tipo específico de bloque de función. Por ejemplo, en formato de temporizador %T _{Mi} , i es un número que representa la instancia.
Instrucciones reversibles	Método de programación que permite visualizar las instrucciones de forma alternativa como instrucciones de lista o escalones de Ladder.

L

Lenguaje de lista de instrucciones	Programa escrito en el lenguaje de lista de instrucciones (IL), compuesto por una serie de instrucciones ejecutadas de forma secuencial por el controlador. Cada instrucción está compuesta por un número de línea, un código de instrucción y un operando.
Lenguaje Ladder	Programa escrito en lenguaje Ladder compuesto por una representación gráfica de instrucciones de un programa controlador con símbolos para contactos, bobinas y bloques en una serie de escalones ejecutados de forma secuencial por un controlador.
LIFO	Last In, First Out. Bloque de función utilizado para operaciones stack.
Líneas de comentarios	En los programas de lista, pueden introducirse comentarios en líneas separadas de las instrucciones. Las líneas de comentarios no tienen números de línea, y deben introducirse entre paréntesis y asteriscos como: (*COMENTARIOS AQUÍ*).

M

Modbus	Protocolo de comunicaciones master-slave que permite a un solo master solicitar respuestas de slaves.
Modo de exploración	Especifica el modo en el que el controlador explora un programa. Existen dos tipos de modos de exploración: Normal (cíclico), el controlador explora de forma continua, o periódico, el controlador explora durante el periodo seleccionado (2 a 150 milisegundos) antes de iniciar otra exploración.
Módulos de ampliación de E/S	Módulos de ampliación de E/S opcionales disponibles para agregar puntos de E/S a un controlador Twido. (No todos los modelos del controlador permiten la ampliación).

N

Navegador de aplicación	Ventana especializada en TwidoSoft que muestra una vista gráfica en forma de árbol de una aplicación. Ofrece una configuración y una visualización correctas de una aplicación.
--------------------------------	---

O

Operación offline	Modalidad de funcionamiento de TwidoSoft cuando un PC no está conectado al controlador y la aplicación de la memoria del PC no es la misma que la de la memoria del controlador. El usuario crea y desarrolla una aplicación en operación offline.
Operación online	Modo de funcionamiento de TwidoSoft cuando un PC está conectado al controlador y la aplicación de la memoria del PC es la misma que la de la memoria del controlador. El usuario depura y ajusta una aplicación en la operación online.
Operador	Símbolo o código que especifica la operación que va a realizar una instrucción.
Operando	Número, dirección o símbolo que representa un valor que puede manipular un programa en una instrucción.

P

Paso	Un paso Grafcet designa un estado de funcionamiento secuencial de automatización.
PC	Ordenador personal.
PLS	Generación de pulsos. Bloque de función que genera una onda cuadrada con un ciclo de servicio 50% activado y 50% desactivado.
Potenciómetro analógico	Tensión aplicada que puede ajustarse y convertirse en un valor digital para ser utilizado por una aplicación.
Preferencias	Cuadro de diálogo con opciones seleccionables para configurar los editores de programa Ladder y de lista.
Protección	Existen dos tipos de protección de aplicación diferentes: protección con contraseña, que proporciona control de acceso y protección de la aplicación del controlador, que evita la visualización y copia no autorizadas de una aplicación.
PWM	Modulación de ancho de pulsos. Bloque de función que genera una onda cuadrada con un ciclo de servicio variable que puede configurar un programa.

R

RAM	Memoria de acceso aleatorio (del inglés "Random Access Memory"). Las aplicaciones Twido se descargan en una memoria RAM interna y volátil que se va a ejecutar.
Referencias cruzadas	Generación de una lista de operandos, símbolos, números de red/línea y operadores utilizados en una aplicación para simplificar la creación y gestión de aplicaciones.
Registros	Registros especiales internos para el controlador dedicado a los bloques de función LIFO/FIFO.
Reloj de tiempo real	Opción que conservará la hora aunque el controlador no reciba alimentación durante un tiempo determinado.

RTC	Consulte "Reloj de tiempo real".
RTU	Remote Terminal Unit (unidad de terminal remota). Protocolo que utiliza ocho bits, empleado para establecer comunicación entre un controlador y un PC.
Run	Comando que hace que el controlador ejecute un programa de aplicación.

S

Salida refleja	En modo de conteo, el valor actual del contador rápido (%VFC.V) se compara con sus umbrales configurados para determinar el estado de estas salidas dedicadas.
Salidas de umbral	Bobinas controladas directamente por el contador rápido (%VFC) con arreglo a los ajustes establecidos durante la configuración.
Símbolo	Un símbolo es una cadena con un máximo de 32 caracteres alfanuméricos, de los cuales el primer carácter es alfabético. Permite personalizar un objeto del controlador para facilitar el mantenimiento de la aplicación.
Símbolos sin resolver	Símbolo sin una dirección variable.

T

Tabla de animación	Tabla creada con un editor de lenguaje o una pantalla de funcionamiento. Cuando un PC se conecta al controlador, proporciona una vista de las variables del mismo y permite que los valores se fuercen durante la depuración. Puede guardarse como archivo separado con una extensión .tat.
Tabla de símbolos	Tabla de los símbolos utilizados en una aplicación. Se muestra en el editor de símbolos.
Temporizador	Bloque de función utilizado para seleccionar la duración para controlar un evento.
Twido	Línea de controladores de Schneider Electric compuesta por dos tipos de controladores (compacto y modular), módulos de ampliación para agregar puntos de E/S y opciones como Reloj de tiempo real, comunicaciones, monitor de operación y cartuchos de memoria de copia de seguridad.

TwidoSoft Software de desarrollo gráfico de Windows de 32 bits para configurar y programar controladores Twido.

V

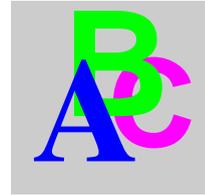
Validar línea automática Cuando se insertan o modifican instrucciones de lista, este parámetro opcional permite la validación de las líneas del programa a medida que se introduce cada una de ellas debido a símbolos no resueltos y errores. Cada error debe corregirse antes de que pueda abandonar la línea. Se selecciona utilizando el cuadro de diálogo Preferencias.

Variable Unidad de memoria que puede enviarse y modificarse mediante un programa.

Visualizador de errores de programa Ventana TwidoSoft especializada utilizada para ver errores de programa y advertencias.

Visualizador de referencias cruzadas Ventana especializada en la aplicación TwidoSoft para ver referencias cruzadas.

Índice



Symbols

%Ci, 238	%S50, 338
%DR, 296	%S51, 339
%FC, 302	%S59, 339
%INW, 35	%S6, 336
%MSG, 319	%S69, 339
%PLS, 293	%S7, 336
%QNW, 35	%S70, 339
%S, 336	%S73, 339
%S0, 336	%S74, 339
%S1, 336	%S8, 336
%S10, 337	%S9, 337
%S100, 340	%S96, 339
%S11, 337	%S97, 340
%S110, 340	%SW, 342
%S111, 340	%SW0, 342
%S112, 340	%SW11, 343
%S113, 340	%SW111, 349
%S118, 340	%SW112, 350
%S119, 340	%SW113, 350
%S12, 337	%SW114, 350
%S13, 337	%SW118, 350
%S17, 337	%SW120, 351
%S18, 337	%SW18, 343
%S19, 337	%SW19, 343
%S20, 338	%SW30, 343
%S21, 57, 338	%SW31, 344
%S22, 57, 338	%SW32, 344
%S23, 57, 338	%SW49, 344
%S24, 338	%SW50, 344
%S4, 336	%SW51, 344
%S5, 336	%SW52, 344
	%SW53, 344

- %SW54, 344
- %SW55, 344
- %SW56, 344
- %SW57, 344
- %SW58, 345
- %SW59, 345
- %SW6, 342
- %SW60, 345
- %SW63, 346
- %SW64, 346
- %SW67, 346
- %SW7, 343
- %SW76, 347
- %SW77, 347
- %SW78, 347
- %SW79, 347
- %SW96, 348
- %SW97, 349
- %TMI, 235
- %VFC, 306

A

- Acumulador, 184
- Acumulador booleano, 184
- Agregar, 258
- Área de actividad, 160
- Área de prueba, 160
- ASCII
 - comunicaciones, 72, 89
 - configuración de hardware, 90
 - configuración de software, 92
 - configuración del puerto, 93
- Aumentar, 258

B

- Bit Ejecutar/detener, 59
- Bits de memoria, 25
- Bits del sistema, 336
- BLK, 176
- Bloque de comparación
 - elemento gráfico, 167
- Bloque de función de contador rápido, 302
- Bloque de función de contadores muy rápidos, 306

- Bloque de función de intercambio, 319
- Bloque de función del controlador del conmutador de tambor, 296
- Bloques
 - en diagramas Ladder Logic, 162
- Bloques de comparación, 163
- Bloques de función
 - contador de pasos (%SCi), 246
 - contadores, 238
 - controlador del conmutador de tambor, 296, 300
 - elemento gráfico, 167
 - en reticulado de programación, 163
 - fechadores, 325
 - programación de bloques de función básicos, 228
 - PWM, 289
 - Registro de bits de desplazamiento (%SBR), 243
 - registros, 282
 - resumen de bloques de función básicos, 226
 - Temporizadores, 230
 - temporizadores, 235
- Bloques de función avanzados
 - objetos de palabra y de bit, 277
 - principios de programación, 279
- Bloques de función básicos, 226
- Bloques de operación, 164
 - elemento gráfico, 167
- Bobinas, 162
 - elementos gráficos, 166

C

- Cabecera de escalón, 161
 - comentarios, 179
- Cadenas de bits, 37
- Canal analógico, 128
- Ciclo de tarea master, 54
- Clavijas
 - Conector hembra del cable de comunicaciones, 75
 - conector macho del cable de comunicaciones, 75
- Cola, 282

Comentarios de la línea de lista, 178
Comprobación del tiempo de ciclo, 54
Comunicaciones
 ASCII, 89
 conexión remota, 76
 Modbus, 101
Conector inferior, 165
Conexión ASCII
 ejemplo, 98
Conexión del cable de comunicaciones, 74
Conexión Modbus
 ejemplo 1, 112
 ejemplo 2, 116
Conexión remota
 acceso de datos de E/S remotas, 82
 comunicaciones, 72, 76
 configuración de hardware, 77
 configuración de software, 79
 configuración del controlador master, 79
 configuración del controlador remoto, 80
 ejemplo, 86
 sincronización del ciclo del controlador remoto, 81
Conexión vertical, 165
Configuración
 búfer de transmisión/recepción para ASCII, 93
 puerto para ASCII, 93
 puerto para Modbus, 105
Consejos sobre programación, 169
Contactos, 162
 elemento gráfico, 165
Contador de pasos, 246
Contadores, 238
 programación y configuración, 242
Controlador
 inicialización, 67
Controlador del conmutador de tambor
 operación, 298
Controladores del conmutador de tambor
 programación y configuración, 300
Corrección RTC, 324
Corte de corriente, 58

D

Desborde, 260
 índice, 41
Desborde de índice, 41
Detección de flanco
 ascendente, 209
 descendente, 210
Diagramas Ladder Logic
 bloques, 162
 elementos gráficos, 165
 introducción, 158
 OPEN y SHORT, 168
 principios de programación, 160
Direccionamiento
 indexado, 40
Direccionamiento de E/S, 33
Direccionamiento de módulos de E/S analógicas, 131
Direccionamiento directo, 40
Disminuir, 258
Dividir, 258
Documentación del programa, 178

E

E/S
 direccionamiento, 33
Elementos de conexión
 elementos gráficos, 165
Elementos gráficos
 diagramas de Ladder Logic, 165
END_BLK, 176
Error, 260
Escalón de lista Ladder Logic, 177
Escalones
 incondicional, 177
Escalones incondicionales, 177
Escalones Ladder Logic, 159
EXCH, 318
Exploración
 cíclica, 48
 periódica, 51

F

Factor de corrección de tiempo real, 153

FIFO

introducción, 282

operación, 285

Funciones de reloj

establecimiento de la fecha y la hora, 330

fechadores, 325

fijación de la fecha y la hora, 328

vista general, 324

G

Generación de pulsos, 293

Graficet

acciones asociadas, 202

ejemplos, 196

instrucciones, 194

procesamiento previo, 199

procesamiento secuencial, 200

I

Inicialización de un controlador, 67

Inicio en frío, 58, 64

Instrucción EXCH, 318

Instrucción NOP, 271

Instrucción NOT, 224

Instrucción OR, 220

Instrucciones

AND, 218

aritméticas, 258

cargar, 214

comparación, 256

conversión, 266

END, 269

JMP, 272

lógica, 262

NOP, 271

NOT, 224

XOR, 222

instrucciones

RET, 273

SR, 273

Instrucciones AND, 218

Instrucciones aritméticas, 258

Instrucciones booleanas, 209

almacenar, 216

Comprensión del formato utilizado en este manual, 212

OR, 220

Instrucciones de almacenamiento, 216

Instrucciones de asignación

numérica, 252

Instrucciones de comparación, 256

Instrucciones de conversión, 266

Instrucciones de desplazamiento, 264

Instrucciones de lista, 185

Instrucciones de lógica, 262

Instrucciones de OR exclusivo, 222

Instrucciones de salto, 272

Instrucciones de stack, 191

Instrucciones de subrutina, 273

Instrucciones END, 269

Instrucciones numéricas

asignación, 252

desplazamiento, 264

J

JMP, 272

L

LD, 214

LDF, 210, 214

LDN, 214

LDR, 209, 214

Lenguaje de lista

vista general, 182

Lenguajes de programación

vista general, 19

LIFO

introducción, 282

operación, 284

M

Memoria

estructura, 43

Métodos Grafcet, 56

Modbus

- comunicaciones, 73, 101
- configuración de hardware, 102
- configuración de software, 104
- configuración del puerto, 105
- master, 73
- slave, 73
- solicitudes estándar, 119

Modos de funcionamiento, 56

Modulación de ancho de pulsos, 289

Módulo analógico

- ejemplo, 135
- funcionamiento, 130

Módulos analógicos

- Configuración de E/S, 133
- direccionamiento, 131

Monitor de operación

- ajustes del puerto serie, 151
- corrección de tiempo real, 153
- ID y estados del controlador, 141
- objetos y variables del sistema, 144
- reloj de fecha/hora, 152
- vista general, 138

MPP, 191

MPS, 191

MRD, 191

Multiplicar, 258

N

NOP, 271

O

Objetos

- bloques de función, 36
- estructurados, 37
- objetos de bit, 25
- palabras, 28

Objetos de bit, 277

- direccionamiento, 31
- vista general, 25

Objetos de palabra, 277

- direccionamiento, 32

Objetos de palabras

- vista general, 28

OPEN, 168

Operandos, 184

OUT_BLK, 176

P

Palabras de memoria, 28

Palabras de sistema, 342

Parámetros, 231

Parámetros de control

- ASCII, 93
- Modbus, 106

Paréntesis

- intercalado, 190
- modificadores, 190
- utilización en programas, 189

Potenciómetros, 126

Principios de programación, 279

Procesamiento numérico

- vista general, 251

Programa Ladder Logic

- reversibilidad a Lista, 174

Programación

- documentación del programa, 178

Programación no reversible, 279

Programación reversible, 279

Protocolos, 72

R

Raíz cuadrada, 258

Recepción de mensajes, 318

Recuperación de alimentación, 58

Red

- direccionamiento, 35

Registro de bits de desplazamiento, 243

Registros

- FIFO, 285
- LIFO, 284
- programación y configuración, 286

Reinicio en caliente, 58, 61

Resto, 258

RET, 273

Reticulado de programación, 160

Reversibilidad

- directrices, 176
- introducción, 174

S

- SHORT, 168
- Simbolización, 42
- SR, 273
- Stack, 282
- Sustraer, 258

T

- Tablas de palabras, 37
- Temporizador TOF, 232
- Temporizador TON, 233
- Temporizador TP, 234
- Temporizadores, 231
 - introducción, 230
 - programación y configuración, 235
 - tiempo base de 1 ms, 236
 - tipo TOF, 232
 - tipo TON, 233
 - tipo TP, 234
- Tiempo de ciclo, 54
- Transmisión de mensajes, 318
- TwidoSoft
 - introducción, 18

V

- Validación de objetos, 24
- Vista general de las comunicaciones, 72

W

- Watchdog del software, 54

X

- XOR, 222