

SIEMENS

SIMATIC

S7 Controlador programable S7-1200

Manual de sistema

Prólogo

Sinopsis del producto

1

Montaje

2

Principios básicos del PLC

3

Configuración de
dispositivos

4

Principios básicos de
programación

5

Instrucciones de
programación

6

PROFINET

7

Comunicación punto a punto
(PtP)

8

Herramientas online y
diagnóstico

9

Datos técnicos

A

Calcular la corriente
necesaria

B

Referencias

C

Notas jurídicas

Filosofía en la señalización de advertencias y peligros

Este manual contiene las informaciones necesarias para la seguridad personal así como para la prevención de daños materiales. Las informaciones para su seguridad personal están resaltadas con un triángulo de advertencia; las informaciones para evitar únicamente daños materiales no llevan dicho triángulo. De acuerdo al grado de peligro las consignas se representan, de mayor a menor peligro, como sigue.

 DANGER
Significa que, si no se adoptan las medidas preventivas adecuadas se producirá la muerte, o bien lesiones corporales graves.
 WARNING
Significa que, si no se adoptan las medidas preventivas adecuadas puede producirse la muerte o bien lesiones corporales graves.
 CAUTION
con triángulo de advertencia significa que si no se adoptan las medidas preventivas adecuadas, pueden producirse lesiones corporales.
CAUTION
sin triángulo de advertencia significa que si no se adoptan las medidas preventivas adecuadas, pueden producirse daños materiales.
NOTICE
significa que puede producirse un resultado o estado no deseado si no se respeta la consigna de seguridad correspondiente.

Si se dan varios niveles de peligro se usa siempre la consigna de seguridad más estricta en cada caso. Si en una consigna de seguridad con triángulo de advertencia se alarma de posibles daños personales, la misma consigna puede contener también una advertencia sobre posibles daños materiales.

Personal cualificado

El producto/sistema tratado en esta documentación sólo deberá ser manejado o manipulado por **personal cualificado** para la tarea encomendada y observando lo indicado en la documentación correspondiente a la misma, particularmente las consignas de seguridad y advertencias en ella incluidas. Debido a su formación y experiencia, el personal cualificado está en condiciones de reconocer riesgos resultantes del manejo o manipulación de dichos productos/sistemas y de evitar posibles peligros.

Uso previsto o de los productos de Siemens

Considere lo siguiente:

 WARNING
Los productos de Siemens sólo deberán usarse para los casos de aplicación previstos en el catálogo y la documentación técnica asociada. De usarse productos y componentes de terceros, éstos deberán haber sido recomendados u homologados por Siemens. El funcionamiento correcto y seguro de los productos exige que su transporte, almacenamiento, instalación, montaje, manejo y mantenimiento hayan sido realizados de forma correcta. Es preciso respetar las condiciones ambientales permitidas. También deberán seguirse las indicaciones y advertencias que figuran en la documentación asociada.

Marcas registradas

Todos los nombres marcados con ® son marcas registradas de Siemens AG. Los restantes nombres y designaciones contenidos en el presente documento pueden ser marcas registradas cuya utilización por terceros para sus propios fines puede violar los derechos de sus titulares.

Exención de responsabilidad

Hemos comprobado la concordancia del contenido de esta publicación con el hardware y el software descritos. Sin embargo, como es imposible excluir desviaciones, no podemos hacernos responsable de la plena concordancia. El contenido de esta publicación se revisa periódicamente; si es necesario, las posibles las correcciones se incluyen en la siguiente edición.

Prólogo

Objeto del manual

La gama S7-1200 abarca distintos controladores lógicos programables (PLCs) que pueden utilizarse para numerosas tareas. Gracias a su diseño compacto, bajo costo y amplio juego de instrucciones, los PLCs S7-1200 son idóneos para controlar una gran variedad de aplicaciones. Los modelos S7-1200 y el software de programación basado en Windows ofrecen la flexibilidad necesaria para solucionar las tareas de automatización.

Este manual contiene información sobre cómo montar y programar los PLCs S7-1200 y está dirigido a ingenieros, programadores, técnicos de instalación y electricistas que dispongan de conocimientos básicos sobre los controladores lógicos programables.

Nociones básicas

Para comprender este manual se requieren conocimientos básicos en el campo de la automatización y de los controladores lógicos programables.

Objeto del manual

Este manual es válido para el software STEP 7 Basic V10.5 y la gama de productos S7-1200. En los datos técnicos (Página 319) encontrará una lista completa de los productos S7-1200 descritos en el manual.

Homologaciones, marcado CE, C-Tick y otras normas

Para más información, consulte los datos técnicos (Página 319).

Service & Support

Además de nuestra documentación, ponemos nuestros conocimientos técnicos a su disposición en Internet:

<http://www.siemens.com/automation/support-request>
(<http://www.siemens.com/automation/support-request>)

Contacte con el representante de Siemens más próximo si tiene consultas de carácter técnico, así como para obtener información sobre los cursillos de formación o para pedir productos S7. Puesto que los representantes de Siemens han sido debidamente aleccionados y tienen conocimientos detallados sobre las operaciones, los procesos y la industria, así como sobre los distintos productos de Siemens empleados, pueden solucionar cualquier problema de forma rápida y eficiente.

Índice

	Prólogo	3
1	Sinopsis del producto	11
1.1	Introducción al PLC S7-1200	11
1.2	Signal Boards	13
1.3	Módulos de señales	14
1.4	Módulos de comunicación	14
1.5	STEP 7 Basic	15
1.5.1	Diferentes vistas que facilitan el trabajo	16
1.5.2	Acceso fácil a la ayuda	17
1.6	Visualizadores	21
2	Montaje	23
2.2	Procedimientos de montaje y desmontaje	26
2.2.1	Montaje y desmontaje de la CPU	29
2.2.2	Montaje y desmontaje de un módulo de señales	31
2.2.3	Montaje y desmontaje de un módulo de comunicación	33
2.2.4	Montaje y desmontaje de una Signal Board	34
2.2.5	Extraer y reinsertar el conector del bloque de terminales del S7-1200	35
2.3	Directrices de cableado	36
3	Principios básicos del PLC	41
3.1	Ejecución del programa de usuario	41
3.1.1	Estados operativos de la CPU	44
3.1.2	Prioridades y colas de espera para la ejecución de eventos	48
3.1.3	Memoria de la CPU	54
3.1.4	Protección por contraseña de la CPU S7-1200	59
3.1.5	Recuperación si se olvida la contraseña	60
3.2	Almacenamiento de datos, áreas de memoria y direccionamiento	60
3.3	Tipos de datos	65
3.4	Utilizar una Memory Card	69
3.4.1	Insertar una Memory Card en la CPU	70
3.4.2	Configurar los parámetros de arranque de la CPU antes de copiar el proyecto en la Memory Card	71
3.4.3	Tarjeta de transferencia	71
3.4.4	Tarjeta de programa	74

4	Configuración de dispositivos	77
4.1	Insertar una CPU	78
4.2	Detectar la configuración de una CPU sin especificar	79
4.3	Configurar el funcionamiento de la CPU.....	80
4.4	Agregar módulos a la configuración	81
4.5	Configurar los parámetros de los módulos	82
4.6	Crear una conexión de red.....	83
4.7	Configurar una dirección IP en el proyecto.....	84
5	Principios básicos de programación	87
5.1	Diretrizes para diseñar un sistema PLC.....	87
5.2	Estructurar el programa de usuario	88
5.3	Utilizar bloques para estructurar el programa.....	90
5.3.1	Bloque de organización (OB).....	92
5.3.2	Función (FC)	93
5.3.3	Bloque de función (FB)	94
5.3.4	Bloque de datos (DB).....	95
5.4	Principios básicos de la coherencia de datos	96
5.5	Seleccionar el lenguaje de programación.....	97
5.6	Protección anticopia	99
5.7	Cargar los elementos del programa en la CPU	100
5.8	Cargar los elementos del programa desde un dispositivo.....	101
5.9	Depurar y comprobar el programa	102
6	Instrucciones de programación.....	103
6.1	Instrucciones básicas.....	103
6.1.1	Instrucciones lógicas con bits	103
6.1.1.1	Instrucciones "Activar salida" y "Desactivar salida"	106
6.1.1.2	Instrucciones "Consultar flanco de señal ascendente de un operando" y "Consultar flanco de señal descendente de un operando"	109
6.1.2	Temporizadores	112
6.1.3	Contadores.....	116
6.1.3.1	Contadores.....	116
6.1.3.2	Instrucción CTRL_HSC.....	119
6.1.3.3	Funcionamiento del contador rápido.....	121
6.1.3.4	Configuración del HSC.....	124
6.1.4	Comparación.....	126
6.1.5	Funciones matemáticas	128
6.1.5.1	Instrucción "Obtener resto de división"	129
6.1.6	Desplazamiento	137
6.1.6.1	Instrucción "Cambiar disposición"	140
6.1.7	Convertir.....	141
6.1.7.1	Instrucciones "Escalar" y "Normalizar".....	143
6.1.8	Control del programa	145
6.1.9	Operaciones lógicas.....	147
6.1.10	Instrucciones de desplazamiento y rotación	151

6.2	Instrucciones avanzadas	153
6.2.1	Parámetros de error comunes de las instrucciones avanzadas.....	153
6.2.2	Instrucciones de reloj y calendario	154
6.2.3	Instrucciones con cadenas y caracteres.....	157
6.2.3.1	Sinopsis del tipo de datos String.....	157
6.2.3.2	Instrucciones de conversión de cadenas.....	159
6.2.3.3	Instrucciones con cadenas	167
6.2.4	Instrucciones de control del programa.....	174
6.2.4.1	Instrucción "Reiniciar la vigilancia del tiempo de ciclo"	174
6.2.4.2	Instrucción "Parar ciclo del PLC"	175
6.2.4.3	Instrucciones GET_ERROR.....	175
6.2.5	Instrucciones de comunicación.....	178
6.2.5.1	Comunicación Ethernet abierta.....	178
6.2.5.2	Instrucciones de comunicación punto a punto.....	192
6.2.6	Instrucciones con alarmas	193
6.2.6.1	Instrucciones ATTACH y DETACH.....	193
6.2.6.2	Instrucciones para iniciar y cancelar alarmas de retardo	196
6.2.6.3	Instrucciones para inhibir y habilitar el procesamiento de alarmas.....	199
6.2.7	Control PID.....	199
6.2.8	Instrucciones de Motion Control	200
6.2.9	Instrucción "Impulso"	202
6.2.9.1	Instrucción CTRL_PWM.....	202
6.3	Instrucciones de la librería global	205
6.3.1	USS.....	205
6.3.1.1	Requisitos para utilizar el protocolo USS	206
6.3.1.2	Instrucción USS_DRV.....	209
6.3.1.3	Instrucción USS_PORT	212
6.3.1.4	Instrucción USS_RPM	213
6.3.1.5	Instrucción USS_WPM.....	214
6.3.1.6	Códigos de estado USS.....	216
6.3.2	MODBUS	217
6.3.2.1	MB_COMM_LOAD.....	217
6.3.2.2	MB_MASTER.....	219
6.3.2.3	MB_SLAVE	231
7	PROFINET	241
7.1	Comunicación con una programadora.....	242
7.1.1	Establecer la conexión de hardware.....	243
7.1.2	Configurar los dispositivos	243
7.1.3	Asignar direcciones IP (Internet Protocol)	244
7.1.3.1	Asignar direcciones IP a los dispositivos de programación y red.....	244
7.1.3.2	Asignar una dirección IP online	247
7.1.3.3	Configurar una dirección IP en el proyecto.....	249
7.1.4	Comprobar la red PROFINET.....	251
7.2	Comunicación entre dispositivos HMI y el PLC.....	253
7.2.1	Configurar las conexiones de red lógicas entre un dispositivo HMI y una CPU	255
7.3	Comunicación entre PLCs	256
7.3.1	Configurar las conexiones de red lógicas entre dos CPUs	257
7.3.2	Configurar los parámetros de transmisión y recepción	257
7.3.2.1	Configurar los parámetros de transmisión de la instrucción TSEND_C.....	258
7.3.2.2	Configurar los parámetros de recepción de la instrucción TRCV_C	262
7.4	Información de referencia	266
7.4.1	Localizar la dirección Ethernet (MAC) en la CPU.....	266
7.4.2	Configurar la sincronización del Network Time Protocol (NTP)	268

8	Comunicación punto a punto (PtP)	269
8.1	Utilizar los módulos de comunicación RS232 y RS485	270
8.2	Configurar los puertos de comunicación.....	271
8.3	Gestionar el control de flujo	272
8.4	Configurar los parámetros de transmisión y recepción	274
8.5	Programar la comunicación PtP	280
8.5.1	Arquitectura de sondeo	281
8.6	Instrucciones de comunicación punto a punto.....	282
8.6.1	Parámetros comunes de las instrucciones de comunicación punto a punto.....	282
8.6.2	Instrucción PORT_CFG	284
8.6.3	Instrucción SEND_CFG	286
8.6.4	Instrucción RCV_CFG.....	288
8.6.5	Instrucción SEND_PTP	294
8.6.6	Instrucción RCV_PTP	298
8.6.7	Instrucción RCV_RST	299
8.6.8	Instrucción SGN_GET.....	300
8.6.9	Instrucción SGN_SET	301
8.7	Errores.....	302
9	Herramientas online y diagnóstico	307
9.1	LEDs de estado.....	307
9.2	Establecer una conexión online con una CPU.....	309
9.3	Ajustar la dirección IP y la hora	310
9.4	Panel de control de la CPU online	310
9.5	Vigilar el tiempo de ciclo y la carga de la memoria.....	311
9.6	Visualizar los eventos de diagnóstico de la CPU	312
9.7	Tablas de observación del programa de usuario.....	313
A	Datos técnicos	319
A.1	Datos técnicos generales.....	319
A.2	CPUs	325
A.2.1	Datos técnicos de la CPU 1211C	325
A.2.2	Datos técnicos de la CPU 1212C	330
A.2.3	Datos técnicos de la CPU 1214C	335
A.3	Módulos de señales digitales (SMs)	340
A.3.1	Datos técnicos del módulo de entradas digitales SM 1221	340
A.3.2	Datos técnicos del módulo de salidas digitales SM 1222.....	342
A.3.3	Datos técnicos del módulo de entradas/salidas digitales SM 1223.....	344
A.4	Módulos de señales analógicos (SMs)	346
A.4.1	Datos técnicos de los módulos de señales analógicos SM 1231, SM 1232, SM 1234	346
A.5	Signal Boards (SBs).....	353
A.5.1	Datos técnicos de la SB 1223 2 entradas x 24 V DC / 2 salidas x 24 V DC	353
A.5.2	Datos técnicos de la SB 1232 de 1 salida analógica.....	355

A.6	Módulos de comunicación (CMs).....	357
A.6.1	Datos técnicos del CM 1241 RS485.....	357
A.6.2	Datos técnicos del CM 1241 RS232.....	358
A.7	SIMATIC Memory Cards.....	359
A.8	Simuladores de entradas.....	359
A.9	Cable para módulos de ampliación.....	360
B	Calcular la corriente necesaria.....	361
B.2	Ejemplo de cálculo del consumo de corriente.....	363
B.3	Calcular el consumo de corriente.....	364
C	Referencias.....	365
	Índice alfabético.....	369

Sinopsis del producto

1.1 Introducción al PLC S7-1200

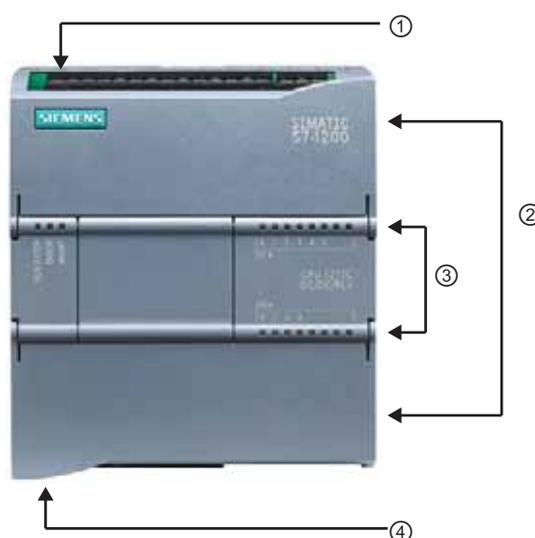
El controlador lógico programable (PLC) S7-1200 ofrece la flexibilidad y capacidad de controlar una gran variedad de dispositivos para las distintas tareas de automatización. Gracias a su diseño compacto, configuración flexible y amplio juego de instrucciones, el S7-1200 es idóneo para controlar una gran variedad de aplicaciones.

La CPU incorpora un microprocesador, una fuente de alimentación integrada, así como circuitos de entrada y salida en una carcasa compacta, conformando así un potente PLC. Una vez cargado el programa en la CPU, ésta contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de usuario, que puede incluir lógica booleana, instrucciones de conteo y temporización, funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes.

Numerosas funciones de seguridad protegen el acceso tanto a la CPU como al programa de control:

- Toda CPU ofrece protección por contraseña que permite configurar el acceso a sus funciones.
- Es posible utilizar la "protección de know-how" para ocultar el código de un bloque específico. Encontrará más detalles en el capítulo "Principios básicos de programación" (Página 99).

La CPU incorpora un puerto PROFINET para la comunicación en una red PROFINET. Los módulos de comunicación están disponibles para la comunicación en redes RS485 o RS232.



- ① Conector de corriente
- ② Conectores extraíbles para el cableado de usuario (detrás de las tapas)
- ② Ranura para Memory Card (debajo de la tapa superior)
- ③ LEDs de estado para las E/S integradas
- ④ Conector PROFINET (en el lado inferior de la CPU)

Los diferentes modelos de CPUs ofrecen una gran variedad de funciones y prestaciones que permiten crear soluciones efectivas destinadas a numerosas aplicaciones. Para más información sobre una CPU en particular, consulte los datos técnicos (Página 319).

Sinopsis del producto

1.1 Introducción al PLC S7-1200

Función	CPU 1211C	CPU 1212C	CPU 1214C
Dimensiones físicas (mm)	90 x 100 x 75		110 x 100 x 75
Memoria de usuario			
<ul style="list-style-type: none"> • Memoria de trabajo • Memoria de carga • Memoria remanente 	<ul style="list-style-type: none"> • 25 KB • 1 MB • 2 KB 		<ul style="list-style-type: none"> • 50 KB • 2 MB • 2 KB
E/S integradas locales			
<ul style="list-style-type: none"> • Digitales • Analógicas 	<ul style="list-style-type: none"> • 6 entradas/4 salidas • 2 entradas 	<ul style="list-style-type: none"> • 8 entradas/6 salidas • 2 entradas 	<ul style="list-style-type: none"> • 14 entradas/10 salidas • 2 entradas
Tamaño de la memoria imagen de proceso	1024 bytes para entradas (I) y 1024 bytes para salidas (Q)		
Área de marcas (M)	4096 bytes		8192 bytes
Ampliación con módulos de señales	Ninguna	2	8
Signal Board	1		
Módulos de comunicación	3 (ampliación en el lado izquierdo)		
Contadores rápidos			
<ul style="list-style-type: none"> • Fase simple • Fase en cuadratura 	<ul style="list-style-type: none"> • 3 • 3 a 100 kHz • 3 a 80 kHz 	<ul style="list-style-type: none"> • 4 • 3 a 100 kHz • 1 a 30 kHz • 3 a 80 kHz • 1 a 20 kHz 	<ul style="list-style-type: none"> • 6 • 3 a 100 kHz • 3 a 30 kHz • 3 a 80 kHz • 3 a 20 kHz
Salidas de impulsos	2		
Memory Card	SIMATIC Memory Card (opcional)		
Tiempo de respaldo del reloj de tiempo real	Típico: 10 días / Mínimo: 6 días a 40 °C		
PROFINET	1 puerto de comunicación Ethernet		
Velocidad de ejecución de funciones matemáticas con números reales	18 µs/instrucción		
Velocidad de ejecución booleana	0,1 µs/instrucción		

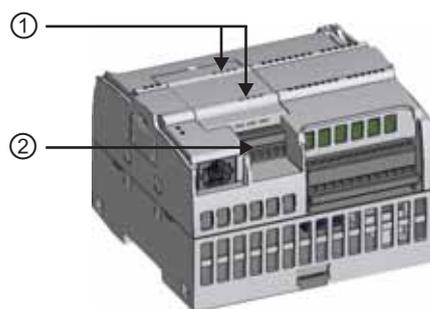
La gama S7-1200 ofrece una gran variedad de módulos de señales y Signal Boards que permiten ampliar las prestaciones de la CPU. También es posible instalar módulos de comunicación adicionales para soportar otros protocolos de comunicación. Para más información sobre un módulo en particular, consulte los datos técnicos (Página 319).

Módulo		Sólo entradas	Sólo salidas	Entradas y salidas
Módulo de señales (SM)	Digital	8 entradas DC	8 salidas DC 8 salidas de relé	8 entradas DC/8 salidas DC 8 entradas DC/8 salidas de relé
		16 entradas DC	16 salidas DC 16 salidas de relé	16 entradas DC/16 salidas DC 16 entradas DC/16 salidas de relé
	Analógico	4 entradas analógicas 8 entradas analógicas	2 salidas analógicas 4 salidas analógicas	4 entradas analógicas/2 salidas analógicas
Signal Board (SB)	Digital	-	-	2 entradas DC/2 salidas DC
	Analógico	-	1 salida analógica	-
Módulo de comunicación (CM)				
<ul style="list-style-type: none"> • RS485 • RS232 				

1.2 Signal Boards

Una Signal Board (SB) permite agregar E/S a la CPU. Es posible agregar una SB con E/S digitales o analógicas. Una SB se conecta en el frente de la CPU.

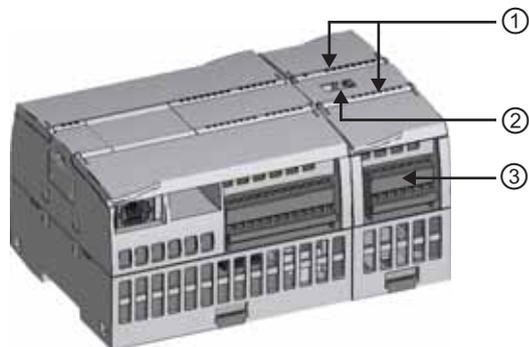
- SB con 4 E/S digitales (2 entradas DC y 2 salidas DC)
- SB con 1 entrada analógica



- ① LEDs de estado en la SB
- ② Conector extraíble para el cableado de usuario

1.3 Módulos de señales

Los módulos de señales se pueden utilizar para agregar funciones a la CPU. Los módulos de señales se conectan a la derecha de la CPU.

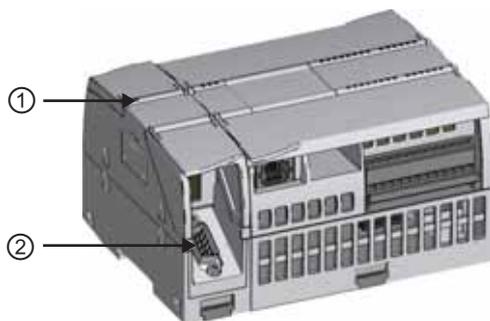


- ① LEDs de estado para las E/S del módulo de señales
- ② Conector de bus
- ③ Conector extraíble para el cableado de usuario

1.4 Módulos de comunicación

La gama S7-1200 provee módulos de comunicación (CMs) que ofrecen funciones adicionales para el sistema. Hay dos módulos de comunicación, a saber: RS232 y RS485.

- La CPU soporta como máximo 3 módulos de comunicación
- Todo CM se conecta en lado izquierdo de la CPU (o en lado izquierdo de otro CM)



- ① LEDs de estado del módulo de comunicación
- ② Conector de comunicación

1.5 STEP 7 Basic

El software STEP 7 Basic ofrece un entorno amigable que permite desarrollar, editar y observar la lógica del programa necesaria para controlar la aplicación, incluyendo herramientas para gestionar y configurar todos los dispositivos del proyecto, tales como PLCs y dispositivos HMI. STEP 7 Basic ofrece dos lenguajes de programación (KOP y FUP) que permiten desarrollar el programa de control de la aplicación de forma fácil y eficiente. Asimismo, incluye las herramientas para crear y configurar los dispositivos HMI en el proyecto.

Para poder encontrar la información necesaria, STEP 7 Basic ofrece un completo sistema de ayuda en pantalla.

Para instalar STEP 7 Basic, inserte el CD en la unidad de CDROM del equipo. El asistente de instalación arranca automáticamente y le guía por el proceso de instalación. Encontrará más información en el archivo Léame.

Nota

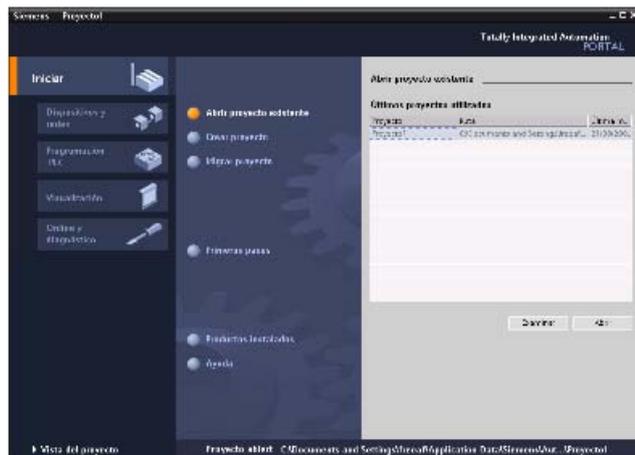
Para instalar el software STEP 7 Basic en un equipo con el sistema operativo Windows 2000, Windows XP o Windows Vista, es preciso iniciar la sesión con derechos de administrador.

1.5.1 Diferentes vistas que facilitan el trabajo

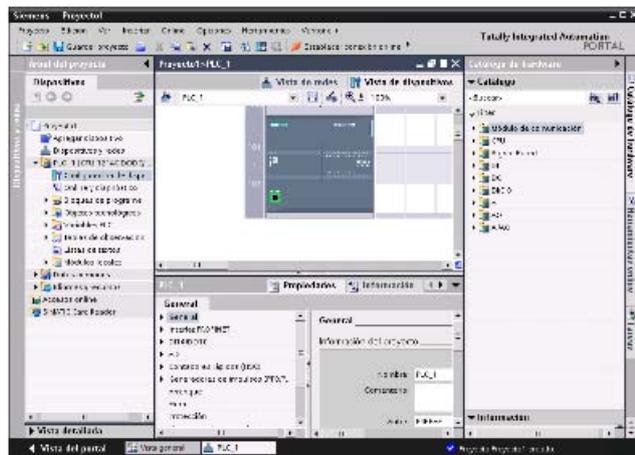
Para aumentar la productividad, el Totally Integrated Automation Portal (TIA Portal) ofrece dos vistas diferentes de las herramientas disponibles, a saber: distintos portales orientados a tareas organizados según las funciones de las herramientas (vista del portal) o una vista orientada a los elementos del proyecto (vista del proyecto). El usuario puede seleccionar la vista que considere más apropiada para trabajar eficientemente. Con un solo clic es posible cambiar entre la vista del portal y la vista del proyecto.

La vista del portal ofrece una vista funcional de las tareas del proyecto y organiza las funciones de las herramientas según las tareas que deben realizarse, p. ej. configurar los componentes de hardware y las redes.

Es posible determinar fácilmente el procedimiento y la tarea que debe seleccionarse.



La vista del proyecto proporciona acceso a todos los componentes del proyecto. Puesto que todos estos componentes se encuentran en un solo lugar, es posible acceder fácilmente a todas las áreas del proyecto. El proyecto contiene todos los elementos que se han creado o finalizado.



1.5.2 Acceso fácil a la ayuda

Respuestas rápidas a las preguntas

Para poder solucionar las tareas de forma rápida y eficiente, STEP 7 Basic proporciona asistencia inteligente donde se necesite:

- En los campos de entrada se ofrece ayuda "roll-out" (desplegable) que facilita la entrada de la información correcta (rangos válidos y tipo de datos). Por ejemplo, si se introduce un valor incorrecto, aparecerá un texto de aviso en el que se indica el rango de valores válidos.
- Algunos de los tooltips de la interfaz de usuario (p. ej. en las instrucciones) se abren "en cascada", ofreciendo información adicional. Algunos de los tooltips en cascada contienen enlaces a temas específicos del sistema de información (ayuda en pantalla).

Además, STEP 7 Basic ofrece un completo sistema de información que describe detalladamente las funciones de las herramientas SIMATIC.

Ayuda "roll-out" y tooltips en cascada

En los campos de entrada de los diferentes diálogos y Task Cards se ofrece asistencia en forma de un cuadro de texto desplegable que informa proporciona información acerca del rango o los tipos de datos requeridos.

A red rectangular box with a white 'X' icon on the left. The text inside reads: "Valor fuera del rango admisible: [0..8191]."

Los elementos de la interfaz de usuario ofrecen tooltips que explican la función del elemento en cuestión. Algunos de los elementos, tales como los botones "Abrir" o "Guardar", no requieren información adicional. No obstante, algunos de los elementos ofrecen un mecanismo que permite ver una descripción adicional del elemento en cuestión. Esta información adicional se visualiza en un tooltip "en cascada". (Un triángulo negro junto al tooltip indica que hay más información disponible.)

El tooltip aparece cuando se sitúa el puntero del ratón sobre un elemento de la interfaz de usuario. Para visualizar información adicional, el puntero del ratón se debe situar sobre el tooltip. Algunos de los tooltips en cascada también ofrecen enlaces a temas específicos del sistema de información. Al hacer clic en el enlace se visualiza el tema en cuestión.

A yellow tooltip box with a black downward-pointing triangle at the top left. The text inside reads: "Conectar dispositivos en red", "Permite la conexión gráfica en red de interfaces.", and a blue underlined link: "Conectar dispositivos en red en la vista de redes".

▼ Conectar dispositivos en red
Permite la conexión gráfica en red de interfaces.
⇒ [Conectar dispositivos en red en la vista de redes](#)

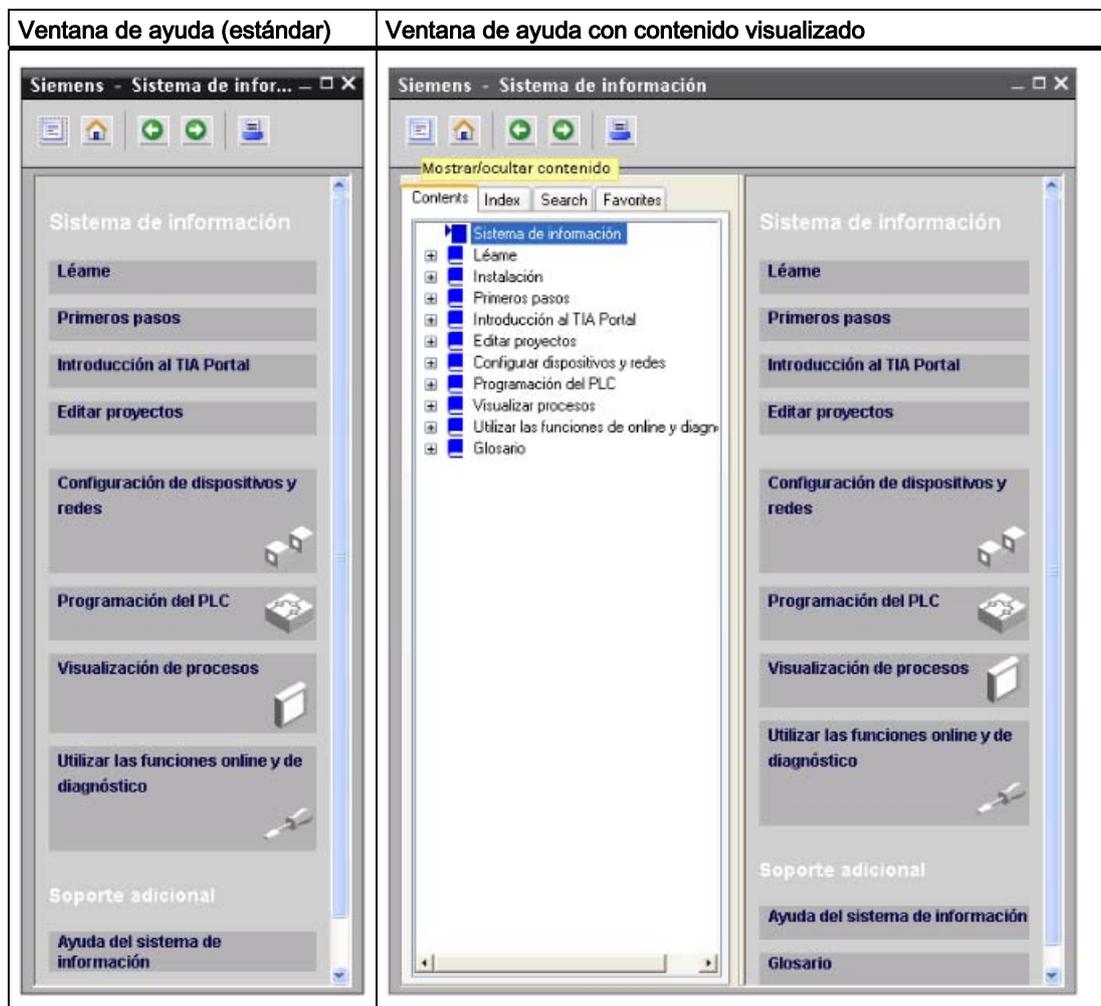
Sistema de información

STEP 7 Basic pone a disposición un completo sistema de información online y Ayuda en pantalla, en el que se describen todos los productos SIMATIC que se han instalado. El sistema de información incluye asimismo información de referencia y ejemplos. Para visualizar el sistema de información, seleccione uno de los puntos de acceso siguientes:

- En la vista del portal, seleccione el portal de inicio y haga clic en "Ayuda".
- En la vista del proyecto, elija el comando "Mostrar ayuda" del menú "Ayuda".
- En un tooltip en cascada, haga clic en un enlace para ver más información sobre ese tema.

El sistema de información se abre en una ventana que no oculta las áreas de trabajo.

Haga clic en el botón "Mostrar/ocultar contenido" del sistema de información para ver el contenido y desacoplar la ventana de ayuda. Entonces se puede cambiar el tamaño de la ventana de ayuda. Utilice las fichas "Contenido" o "Índice" para buscar un tema o palabra clave en el sistema de información.

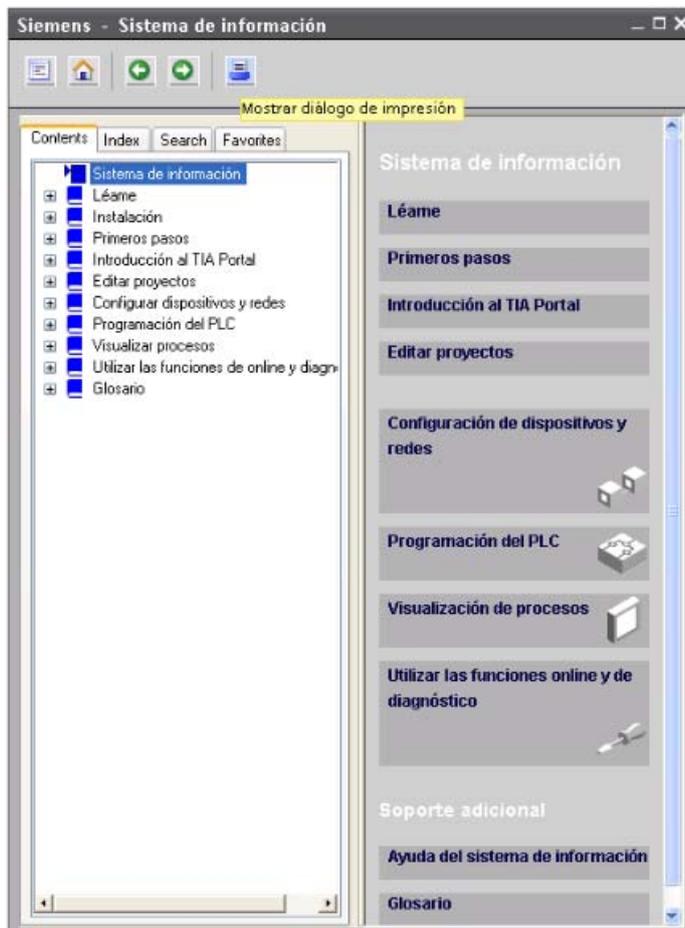


Nota

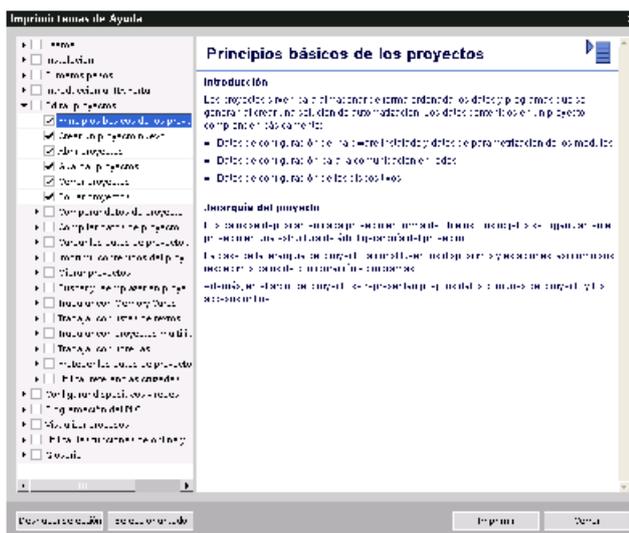
Si STEP 7 Basic está maximizado y se hace clic en el botón "Mostrar/ocultar contenido", no se desacoplará la ventana de ayuda. Haga clic en el botón "Acoplar" para desacoplar la ventana de ayuda. Entonces se puede desplazar la ventana de ayuda y cambiar su tamaño.

Imprimir temas del sistema de información

Para imprimir desde el sistema de información, haga clic en el botón "Imprimir" de la ventana de ayuda.



Para imprimir desde el sistema de información, haga clic en el botón "Imprimir" de la ventana de ayuda.



El diálogo "Imprimir" permite seleccionar los temas que se desean imprimir. Asegúrese de que se visualiza un tema en la ventana. Luego es posible seleccionar cualquier otro tema para imprimirlo. Haga clic en el botón "Imprimir" para enviar los temas seleccionados a la impresora.

1.6 Visualizadores

Puesto que la visualización se está convirtiendo cada vez más en un componente estándar de la mayoría de las máquinas, los Basic Panels SIMATIC HMI ofrecen dispositivos con pantalla táctil para tareas básicas de control y supervisión. Todos los paneles tienen el tipo de protección IP65 y certificación CE, UL, cULus y NEMA 4x.



KTP 400 Basic PN

- Mono (STN, escala de grises)
- Pantalla táctil de 4 pulgadas con 4 teclas táctiles
- Vertical u horizontal
- Tamaño: 3.8"
- Resolución: 320 x 240
- 128 variables
- 50 pantallas de proceso
- 200 alarmas
- 25 curvas
- 32 KB memoria de recetas
- 5 recetas, 20 registros, 20 entradas



KTP 600 Basic PN

- Color (TFT, 256 colores) o monocromo (STN, escala de grises)
- Pantalla táctil de 6 pulgadas con 6 teclas táctiles
- Vertical u horizontal
- Tamaño: 5.7"
- Resolución: 320 x 240
- 128 variables
- 50 pantallas de proceso
- 200 alarmas
- 25 curvas
- 32 KB memoria de recetas
- 5 recetas, 20 registros, 20 entradas



KTP1000 Basic PN

- Color (TFT, 256 colores)
- Pantalla táctil de 10 pulgadas con 8 teclas táctiles
- Tamaño: 10.4"
- Resolución: 640 x 480
- 256 variables
- 50 pantallas de proceso
- 200 alarmas
- 25 curvas
- 32 KB memoria de recetas
- 5 recetas, 20 registros, 20 entradas



TP1500 Basic PN

- Color (TFT, 256 colores)
- Pantalla táctil de 15 pulgadas
- Tamaño: 15.1"
- Resolución: 1024 x 768
- 256 variables
- 50 pantallas de proceso
- 200 alarmas
- 25 curvas
- 32 KB memoria de recetas (memoria flash integrada)
- 5 recetas, 20 registros, 20 entradas

Los equipos S7-1200 son fáciles de montar. El S7-1200 puede montarse en un panel o en un raíl DIN, bien sea horizontal o verticalmente. El tamaño pequeño del S7-1200 permite ahorrar espacio.

 **ADVERTENCIA**

Los PLCs S7-1200 SIMATIC son controladores abiertos. Por este motivo, el S7-1200 debe montarse en una carcasa, un armario eléctrico o una sala de control. Sólo el personal autorizado debe tener acceso a la carcasa, el armario eléctrico o la sala de control.

Si no se cumplen los requisitos de montaje, pueden producirse la muerte, lesiones corporales graves y/o daños materiales.

Vigile siempre los requisitos de montaje de los PLCs S7-1200.

Alejar los dispositivos S71200 de fuentes de calor, alta tensión e interferencias

Como regla general para la disposición de los dispositivos del sistema, los aparatos que generan altas tensiones e interferencias deben mantenerse siempre alejados de los equipos de baja tensión y de tipo lógico, tales como el S71200.

Al configurar la disposición del S7-1200 en el panel, se deben tener en cuenta los aparatos que generan calor y disponer los equipos electrónicos en las zonas más frías del armario eléctrico. Si se reduce la exposición a entornos de alta temperatura, aumentará la vida útil de cualquier dispositivo electrónico.

También se debe considerar la ruta del cableado de los dispositivos montados en el panel. Evite tender las líneas de señales de baja tensión y los cables de comunicación en un mismo canal junto con los cables AC y DC de alta energía y conmutación rápida.

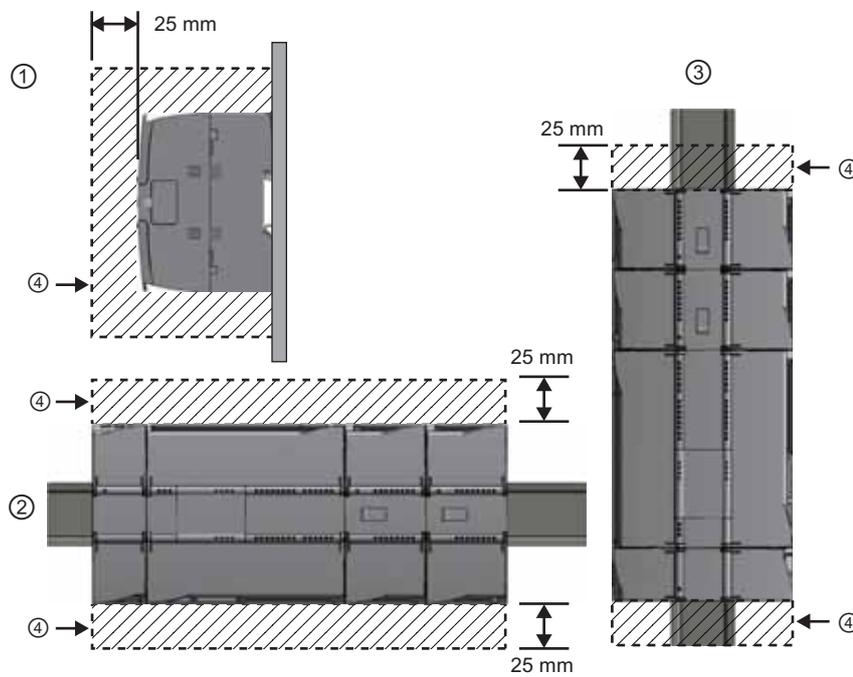
Prever espacio suficiente para la refrigeración y el cableado

La refrigeración de los dispositivos S71200 se realiza por convección natural. Para la refrigeración correcta es preciso dejar un espacio mínimo de 25 mm por encima y por debajo de los dispositivos. Asimismo, se deben prever como mínimo 25 mm de profundidad entre el frente de los módulos y el interior de la carcasa.

⚠ PRECAUCIÓN

En el montaje vertical, la temperatura ambiente máxima admisible se reduce en 10 grados centígrados. Un sistema S7-1200 montado verticalmente debe orientarse de manera que la CPU se encuentre en el extremo inferior del conjunto.

Al planificar la disposición del sistema S71200, prevea espacio suficiente para el cableado y la conexión de los cables de comunicación.



① Vista lateral

② Montaje horizontal

③ Montaje vertical

④ Espacio libre

Corriente necesaria

La CPU dispone de una fuente de alimentación interna que suministra energía eléctrica a la CPU, los módulos de señales, la Signal Board y los módulos de comunicación, así como otros equipos consumidores de 24 V DC.

En los datos técnicos (Página 319) encontrará más información sobre la corriente de 5 V DC que suministra la CPU y la corriente de 5 V DC que requieren los módulos de señales, la Signal Board y los módulos de comunicación. En "Calcular la corriente necesaria" (Página 361) encontrará más información sobre cómo determinar cuánta energía (o corriente) puede proveer la CPU para la configuración.

La CPU provee una alimentación de sensores de 24 V DC que puede suministrar 24 V DC a las entradas y bobinas de relé de los módulos de señales, así como a otros equipos consumidores. Si los requisitos de corriente de 24 V DC exceden la capacidad de la alimentación de sensores, es preciso añadir una fuente de alimentación externa de 24 V DC al sistema. En los datos técnicos (Página 319) se indica la corriente necesaria para la alimentación de sensores de 24 V DC de las distintas CPUs S7-1200.

Si se requiere una fuente de alimentación externa de 24 V DC, vigile que no se conecte en paralelo con la alimentación de sensores de la CPU. Para aumentar la protección contra interferencias, se recomienda conectar los cables neutros (M) de las distintas fuentes de alimentación.

 ADVERTENCIA

<p>Si se conecta una fuente de alimentación externa de 24 V DC en paralelo con la fuente de alimentación de sensores de 24 V DC, puede surgir un conflicto entre ambas fuentes, ya que cada una intenta establecer su propio nivel de tensión de salida.</p>
--

<p>Este conflicto puede reducir la vida útil u ocasionar la avería inmediata de una o ambas fuentes de alimentación y, en consecuencia, el funcionamiento imprevisible del sistema PLC. El funcionamiento imprevisible puede producir la muerte, lesiones corporales graves y/o daños materiales.</p>

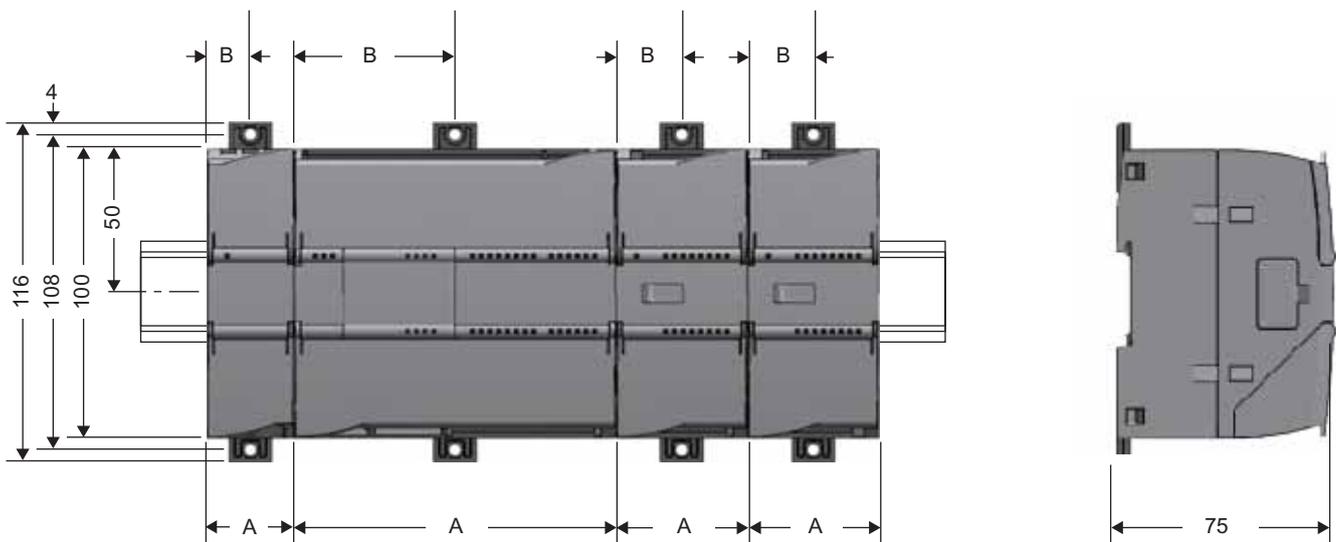
<p>La fuente de alimentación DC de sensores y cualquier fuente de alimentación externa deben alimentar diferentes puntos.</p>

Algunos puertos de entrada de alimentación de 24 V DC del sistema S7-1200 están interconectados, teniendo un circuito lógico común que conecta varios bornes M. Por ejemplo, los circuitos siguientes están interconectados si no tienen aislamiento galvánico según las hojas de datos técnicos: la fuente de alimentación de 24 V DC de la CPU, la entrada de alimentación de la bobina de relé de un SM, o bien la fuente de alimentación de una entrada analógica sin aislamiento galvánico. Todos los bornes M sin aislamiento galvánico deben conectarse al mismo potencial de referencia externo.

 ADVERTENCIA
<p>Si los bornes M sin aislamiento galvánico se conectan a diferentes potenciales de referencia, circularán corrientes indeseadas que podrían averiar o causar reacciones inesperadas en el PLC y los equipos conectados.</p> <p>Si no se cumplen estas directrices, es posible que se produzcan averías o reacciones inesperadas que podrían causar la muerte, lesiones corporales graves y/o daños materiales.</p> <p>Asegúrese que todos los bornes M sin aislamiento galvánico de un sistema S7-1200 están conectados al mismo potencial de referencia.</p>

2.2 Procedimientos de montaje y desmontaje

Dimensiones de montaje (mm)



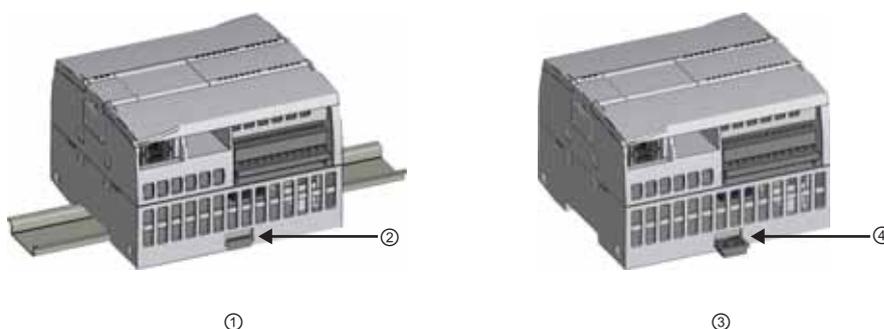
Dispositivos S7-1200		Ancho A	Ancho B
CPUs:	CPU 1211C y CPU 1212C	90 mm	45 mm
	CPU 1214C	110 mm	55 mm
Módulos de señales:	8 y 16 E/S, DC y relé (8I, 16I, 8Q, 16Q, 8I/8Q) Análogos (4AI, 8AI, 4AI/4AQ, 2AQ, 4AQ)	45 mm	22,5 mm
	16I/16Q relé (16I/16Q)	70 mm	35 mm
Módulos de comunicación:	CM 1241 RS232 y CM 1241 RS485	30 mm	15 mm

Las CPUs, los SMs y CMs pueden montarse en un perfil DIN o en un panel. Utilice los clips del módulo previstos para el perfil DIN para fijar el dispositivo al perfil. Estos clips también pueden extenderse a otra posición para poder montar la unidad directamente en un panel. La dimensión interior del orificio para los clips de fijación en el dispositivo es 4,3 mm.

Es preciso prever una zona de disipación de 25 mm por encima y por debajo de la unidad para que el aire pueda circular libremente.

Montaje y desmontaje de dispositivos S7-1200

La CPU se puede montar fácilmente en un perfil estándar o en un panel. Los clips de fijación permiten fijar el dispositivo al perfil DIN. Estos clips también encajan en una posición extendida para proveer orificios de montaje que permiten montar el dispositivo directamente en un panel.



- | | |
|--|---|
| ① Montaje en perfil DIN | ③ Montaje en panel |
| ② Clip de fijación al perfil enclavado | ④ Clip de fijación en posición extendida para el montaje en panel |

Antes de montar o desmontar cualquier dispositivo eléctrico, asegúrese que se ha desconectado la alimentación. Asegúrese también que está desconectada la alimentación eléctrica de todos los dispositivos conectados.

ADVERTENCIA

Si el S7-1200 o los dispositivos conectados se montan o desmontan estando conectada la alimentación, puede producirse un choque eléctrico o un funcionamiento inesperado de los dispositivos.

Si la alimentación del S7-1200 y de los dispositivos conectados no se desconecta por completo antes del montaje o desmontaje, podrían producirse la muerte, lesiones corporales graves y/o daños materiales debidos a choques eléctricos o al funcionamiento inesperado de los equipos.

Respete siempre las medidas de seguridad necesarias y asegúrese que la alimentación del S7-1200 está desconectada antes de montar o desmontar las CPUs S7-1200 o los equipos conectados.

Al sustituir o montar un dispositivo S7-1200, vigile que se utilice siempre el módulo correcto o un dispositivo equivalente.

 **ADVERTENCIA**

El montaje incorrecto de un módulo S7-1200 puede ocasionar el funcionamiento impredecible del programa del S7-1200.

Si un dispositivo S7-1200 no se sustituye por el mismo modelo o si no se monta con la orientación correcta y en el orden previsto, podrían producirse la muerte, lesiones corporales graves y/o daños materiales debido al funcionamiento inesperado del equipo.

Sustituya un dispositivo S7-1200 por el mismo modelo y móntelo con la orientación y posición correctas.

2.2.1 Montaje y desmontaje de la CPU

Montaje

La CPU se puede montar en un panel o en un perfil DIN.

Nota

Conecte los módulos de comunicación necesarios a la CPU y monte el conjunto en forma de unidad. Los módulos de señales se montan por separado una vez montada la CPU.

Para montar la CPU en un panel, proceda del siguiente modo:

1. Posicione y taladre los orificios de montaje (M4 o estándar americano n.º 8) según las dimensiones de montaje indicadas en la tabla.
2. Extienda los clips de fijación del módulo. Asegúrese que los clips de fijación al perfil DIN en los lados superior e inferior de la CPU están en posición extendida.
3. Atornille el módulo al panel utilizando tornillos dispuestos en los clips.

Nota

Si el sistema está sometido a vibraciones fuertes o si se monta verticalmente, el montaje en panel ofrece mayor protección al S7-1200.

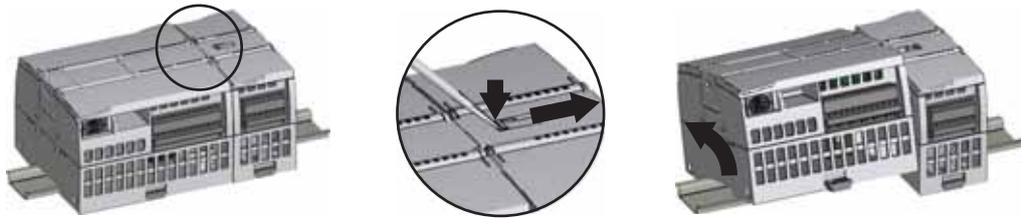
Para montar la CPU en un perfil DIN, proceda del siguiente modo:



1. Monte el perfil DIN. Atornille el perfil al panel de montaje dejando un espacio de 75 mm entre tornillo y tornillo.
2. Enganche la CPU por el lado superior del perfil.
3. Extraiga el clip de fijación en el lado inferior de la CPU de manera que asome por encima del perfil.
4. Gire la CPU hacia abajo para posicionarla correctamente en el perfil.
5. Oprima los clips hasta que la CPU encaje en el perfil.

Desmontaje

Para preparar la CPU para el desmontaje, desconecte la alimentación eléctrica y los conectores de E/S y retire el cableado y demás cables de la CPU. Desmonte la CPU y los módulos de comunicación conectados en forma de conjunto. Todos los módulos de señales deben permanecer montados.



Si un módulo de señales está conectado a la CPU, retraiga el conector de bus:

1. Coloque un destornillador junto a la lengüeta en el lado superior del módulo de señales.
2. Oprima hacia abajo para desenclavar el conector de la CPU.
3. Desplace la lengüeta por completo hacia la derecha.

Desmonte la CPU:

1. Extraiga el clip de fijación para desenclavar la CPU del perfil DIN.
2. Gire la CPU hacia arriba, extráigala del perfil y retírela del sistema.

2.2.2 Montaje y desmontaje de un módulo de señales

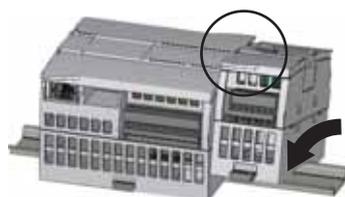
Montaje

El SM se monta una vez montada la CPU.



Retire la tapa del conector en el lado derecho de la CPU.

- Inserte un destornillador en la ranura arriba de la tapa.
- Haga palanca suavemente en el lado superior de la tapa y retírela. Guarde la tapa para poder reutilizarla.



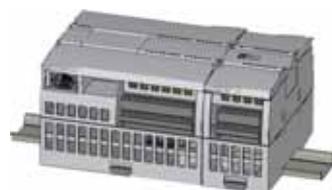
Coloque el SM junto a la CPU.

1. Enganche el SM por el lado superior del perfil DIN.
2. Extraiga el clip de fijación inferior para colocar el SM sobre el perfil.
3. Gire el SM hacia abajo hasta su posición junto a la CPU y oprima el clip de fijación inferior para enclavar el SM en el perfil.



Extienda el conector de bus.

1. Coloque un destornillador junto a la lengüeta en el lado superior del SM.
2. Desplace la lengüeta por completo hacia la izquierda para extender el conector de bus hacia la CPU.



Al extender el conector de bus se crean las conexiones mecánicas y eléctricas para el SM.

Siga el mismo procedimiento para montar un módulo de señales en otro módulo de señales.

Desmontaje

Cualquier SM se puede desmontar sin necesidad de desmontar la CPU u otros SMs. Para preparar el SM para el desmontaje, desconecte la alimentación eléctrica de la CPU y los conectores de E/S y retire el cableado del SM.

Retraiga el conector de bus.

1. Coloque un destornillador junto a la lengüeta en el lado superior del SM.
2. Oprima hacia abajo para desenclavar el conector de la CPU.
3. Desplace la lengüeta por completo hacia la derecha.



Si hay otro SM en el lado derecho, repita este procedimiento para ese SM.

Desmunte el SM:

1. Extraiga el clip de fijación inferior para desenclavar el SM del perfil DIN.
2. Gire el SM hacia arriba y extráigalo del perfil. Retire el SM del sistema.
3. En caso necesario, cubra el conector de bus de la CPU para impedir que se ensucie.



Siga el mismo procedimiento para desmontar un módulo de señales de otro módulo de señales.

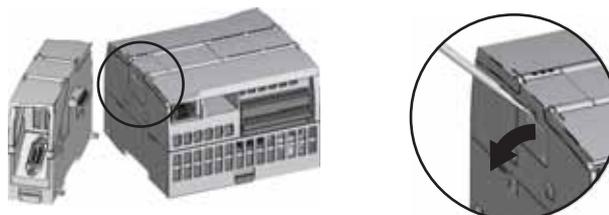
2.2.3 Montaje y desmontaje de un módulo de comunicación

Montaje

Acople el CM a la CPU antes de montar el conjunto en forma de unidad en el perfil DIN o panel.

Retire la tapa de bus en el lado izquierdo de la CPU:

1. Inserte un destornillador en la ranura arriba de la tapa de bus.
2. Haga palanca suavemente en el lado superior de la tapa.



Retire la tapa de bus. Guarde la tapa para poder reutilizarla.

Conecte las unidades:

1. Alinee el conector de bus y las clavijas del CM con los orificios de la CPU.
2. Empuje firmemente una unidad contra la otra hasta que encajen las clavijas.



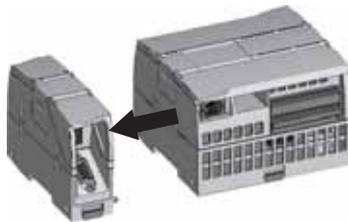
Montar las unidades en un perfil DIN o panel.

1. Para el montaje en un raíl DIN, asegúrese de que el clip de fijación superior está en la posición enclavada (interior) y que el clip de fijación inferior está extendido, tanto en la CPU como en los CMs acoplados.
2. Monte la CPU y los CMs acoplados de la forma descrita en Montaje y desmontaje de la CPU (Página 29).
3. Una vez montados los dispositivos en el perfil DIN, enclave los clips de sujeción para sujetar los dispositivos al raíl.

Para el montaje en un panel, asegúrese de que los clips de fijación al raíl DIN están en posición extendida.

Desmontaje

Desmonte la CPU y el CM en forma de unidad del raíl DIN o panel.



Prepare el CM para el desmontaje.

1. Desconecte la alimentación eléctrica de la CPU.
2. Desconecte los conectores de E/S y retire el cableado y demás cables de la CPU y los CMs.
3. Para el montaje en un raíl DIN, extienda los clips de sujeción inferiores de la CPU y los CMs.
4. Desmonte la CPU y los CMs del raíl DIN o panel.

Desmonte el CM.

1. Sujete la CPU y los CMs con las manos.
2. Sepárelos.

No utilice herramientas para separar los módulos, puesto que podrían deteriorarse.

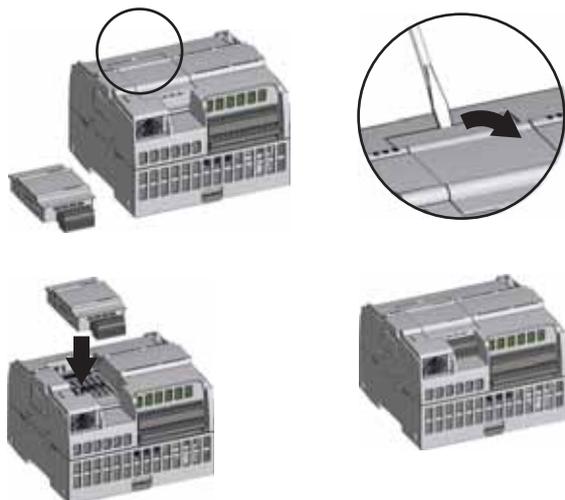
2.2.4 Montaje y desmontaje de una Signal Board

Montaje

Prepare la CPU para el montaje de la SB desconectando la alimentación de la CPU y retirando las tapas superior e inferior de los bloques de terminales de la CPU.

Para montar el SB, proceda del siguiente modo:

1. Inserte un destornillador en la ranura arriba de la CPU en el lado posterior de la tapa.
2. Haga palanca suavemente para levantar la tapa y retirela de la CPU.
3. Coloque la SB rectamente en su posición de montaje en el lado superior de la CPU.
4. Oprima firmemente la SB hasta que encaje en su posición.
5. Coloque nuevamente las tapas de los bloques de terminales.

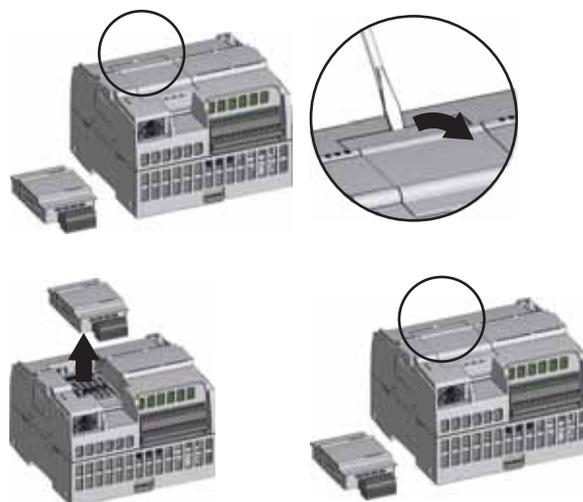


Desmontaje

Prepare la CPU para el desmontaje de la SB desconectando la alimentación de la CPU y retirando las tapas superior e inferior de los bloques de terminales de la CPU.

Para desmontar la SB, proceda del siguiente modo:

1. Inserte un destornillador en la ranura en el lado superior de la SB.
2. Haga palanca suavemente para desacoplar la SB de la CPU.
3. Retire la SB rectamente desde arriba de su posición de montaje en el lado superior de la CPU.
4. Coloque nuevamente la tapa de la SB.
5. Coloque nuevamente las tapas de los bloques de terminales.



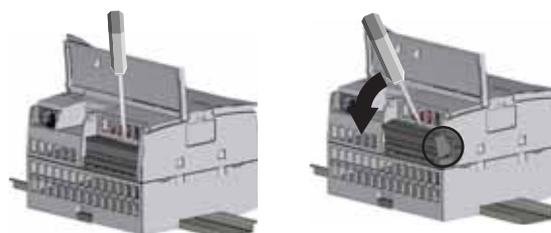
2.2.5 Extraer e insertar el conector del bloque de terminales del S7-1200

La CPU, la SB y los módulos SM incorporan conectores extraíbles que facilitan la conexión del cableado. Para preparar el sistema para la extracción del conector del bloque de terminales:

- Desconecte la alimentación eléctrica de la CPU.
- Abra la tapa por encima del conector.

Para desmontar el conector, proceda del siguiente modo:

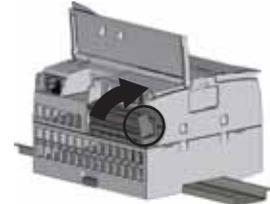
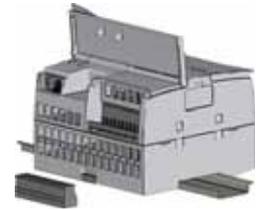
1. Busque la ranura para insertar la punta del destornillador en el lado superior del conector.
2. Inserte un destornillador en la ranura.
3. Haga palanca suavemente en el lado superior del conector para extraerlo de la CPU. El conector se desenclava audiblemente.
4. Sujete el conector con las manos y extráigalo de la CPU.



Para montar el conector, proceda del siguiente modo:

1. Prepare los componentes para el montaje del bloque de terminales desconectando la alimentación de la CPU y abriendo la tapa del bloque de terminales.
2. Alinee el conector a los pines del dispositivo.
3. Alinee el lado de cableado del conector en el zócalo.
4. Con un movimiento giratorio, empuje firmemente el conector hacia abajo hasta que encaje.

Compruebe si el conector está bien alineado y encajado correctamente.



2.3 Directrices de cableado

La puesta a tierra y el cableado correctos de todos los equipos eléctricos es importante para garantizar el funcionamiento óptimo del sistema y aumentar la protección contra interferencias de la aplicación y del S7-1200. Encontrará los diagramas de cableado del S7-1200 en los datos técnicos (Página 319).

Requisitos

Antes de poner a tierra o cablear cualquier dispositivo eléctrico, asegúrese que la alimentación está desconectada. Asegúrese también que está desconectada la alimentación eléctrica de todos los equipos conectados.

Vigile que se respeten todos los reglamentos eléctricos vinculantes al cablear el S7-1200 y los equipos conectados. El equipo se debe montar y operar conforme a todas las normas nacionales y locales vigentes. Contacte con las autoridades locales para determinar qué reglamentos y normas rigen en su caso específico.

ADVERTENCIA

Si el S7-1200 o los equipos conectados se montan o cablean estando conectada la alimentación, puede producirse un choque eléctrico o un funcionamiento inesperado de los equipos. Si la alimentación del S7-1200 y de los equipos conectados no se desconecta por completo antes del montaje o desmontaje, pueden producirse la muerte, lesiones corporales graves y/o daños debidos a choques eléctricos o al funcionamiento inesperado de los equipos.

Respete siempre las medidas de seguridad necesarias y asegúrese que la alimentación eléctrica del S7-1200 está desconectada antes de montar o desmontar el S7-1200 o los equipos conectados.

Considere siempre los aspectos de seguridad al configurar la puesta a tierra y el cableado del sistema S7-1200. Los dispositivos de control electrónicos, tales como el S7-1200, pueden fallar y causar reacciones inesperadas de los equipos que se están controlando o vigilando. Por este motivo, se recomienda prever medidas de seguridad independientes del S7-1200 para evitar lesiones corporales y/o daños materiales.

 ADVERTENCIA
<p>Los dispositivos de control pueden fallar y provocar condiciones no seguras, causando a su vez reacciones inesperadas de los equipos controlados. Las reacciones inesperadas podrían producir la muerte, lesiones corporales graves y/o daños materiales.</p> <p>Prevea dispositivos de parada de emergencia, dispositivos de protección electromecánicos y otras medidas redundantes de seguridad que sean independientes del S7-1200.</p>

Directrices de aislamiento galvánico

Los límites de la alimentación AC del S7-1200 y de las E/S a los circuitos AC se han diseñado y aprobado para proveer un aislamiento galvánico seguro entre las tensiones de línea AC y los circuitos de baja tensión. Estos límites incluyen un aislamiento doble o reforzado, o bien un aislamiento básico más uno adicional, según las distintas normas. Los componentes que cruzan estos límites, tales como optoacopladores, condensadores, transformadores y relés se han aprobado, ya que proveen un aislamiento galvánico seguro. Los límites de aislamiento que cumplen estos requisitos se identifican en las hojas de datos de los productos S7-1200, indicando que tienen un aislamiento de 1500 V AC o superior. Esta indicación se basa en una prueba de fábrica rutinaria de $(2U_e + 1000 \text{ V AC})$ o equivalente, según los métodos aprobados. Los límites de aislamiento galvánico seguro del S7-1200 se han comprobado hasta 4242 V DC.

La salida de la fuente de alimentación de sensores, los circuitos de comunicación y los circuitos lógicos internos de un S7-1200 con fuente de alimentación AC incluida tienen una fuente SELV (pequeña tensión de seguridad) conforme a EN 61131-2.

Para conservar el carácter seguro de los circuitos de baja tensión del S7-1200, las conexiones externas a puertos de comunicación, circuitos analógicos y todas las fuentes de alimentación nominales de 24 V y circuitos E/S deben ser alimentados por fuentes aprobadas que cumplan los requisitos de SELV, PELV, clase 2, tensión limitada o intensidad limitada, según distintas normas.

 ADVERTENCIA
<p>La utilización de fuentes de alimentación no aisladas o con aislamiento simple para abastecer los circuitos de baja tensión desde un conductor AC pueden causar tensiones peligrosas en circuitos considerados no peligrosos (seguros al tacto), tales como los circuitos de comunicación y el cableado de sensores de baja tensión.</p> <p>Las altas tensiones inesperadas podrían causar choques eléctricos que pueden producir la muerte, lesiones corporales graves y/o daños materiales.</p> <p>Utilice sólo convertidores de alta a baja tensión aprobados como fuentes de circuitos de tensión limitada seguros al tacto.</p>

Directrices de puesta a tierra del S7-1200

La mejor forma de poner a tierra la aplicación es garantizar que todos los conductores neutros y de masa del S7-1200 y de los equipos conectados se pongan a tierra en un mismo punto. Este punto debería conectarse directamente a la toma de tierra del sistema.

Todos los cables de puesta a tierra deberían tener la menor longitud posible y una sección grande, p. ej. 2 mm² (14 AWG).

Al definir físicamente las tierras es necesario considerar los requisitos de puesta a tierra de protección y el funcionamiento correcto de los dispositivos protectores.

Directrices de cableado del S7-1200

Al diseñar el cableado del S7-1200, prevea un interruptor unipolar para cortar simultáneamente la alimentación de la CPU S7-1200, de todos los circuitos de entrada y de todos los circuitos de salida. Prevea dispositivos de protección contra sobrecorriente (p. ej. fusibles o cortacircuitos) para limitar las corrientes de fallo en el cableado de alimentación. Para mayor protección es posible disponer un fusible u otro limitador de sobrecorriente en todos los circuitos de salida.

Utilice dispositivos de supresión de sobretensiones apropiados en el cableado sujeto a perturbaciones por descargas atmosféricas.

Evite colocar las líneas de señales de baja tensión y los cables de comunicación en una misma canalización junto con los cables AC y los cables DC de alta energía y conmutación rápida. El cableado deberá efectuarse por pares; con el cable de neutro o común combinado con el hilo caliente o de señal.

Utilice el cable más corto posible y vigile que tenga una sección suficiente para conducir la corriente necesaria. El conector acepta cables con una sección de 2 mm² a 0,3 mm² (14 AWG a 22 AWG). Utilice cables apantallados para obtener una protección óptima contra interferencias. Por lo general, los mejores resultados se obtienen poniendo a tierra la pantalla del S7-1200.

Al cablear circuitos de entrada alimentados por una fuente externa, prevea dispositivos protectores contra sobrecorriente en estos circuitos. La protección externa no se requiere en los circuitos alimentados por la alimentación de sensores de 24 V DC del S7-1200, puesto que la alimentación de sensores ya está protegida contra sobrecorriente.

Todos los módulos S7-1200 incorporan conectores extraíbles para el cableado de usuario. Para evitar conexiones flojas, asegúrese que el conector está encajado correctamente y que el cable está insertado de forma segura en el conector. No apriete excesivamente los tornillos para impedir que se deteriore el conector. El par máximo de apriete de los tornillos del conector es de 0,56 Nm (5 pulgadas-libra).

Para impedir flujos de corriente indeseados en la instalación, el S7-1200 provee límites de aislamiento galvánico en ciertos puntos. Tenga en cuenta estos límites de aislamiento al planificar el cableado del sistema. En los datos técnicos encontrará más información acerca de la ubicación de los puntos de aislamiento galvánico y la capacidad que ofrecen. Los aislamientos con valores nominales inferiores a 1500 V AC no deben tomarse para definir barreras de seguridad.

Directrices relativas a las cargas inductivas

Se recomienda equipar las cargas inductivas con circuitos de supresión que limiten el incremento de tensión producido al desactivarse las salidas. Los circuitos de supresión protegen las salidas contra fallos prematuros debidos a altas tensiones al desconectar las cargas inductivas. Además, estos circuitos limitan las interferencias generadas al conmutar las cargas inductivas. La manera más efectiva de reducir las interferencias es disponer un circuito de supresión externo paralelo eléctricamente a la carga y ubicado físicamente cerca de la carga.

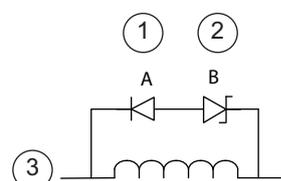
Nota

La eficacia de un determinado circuito de supresión depende de la aplicación. Por tanto, debe verificarse para cada caso en particular. Asegúrese que los todos componentes utilizados en el circuito de supresión se adecúan para la aplicación en cuestión.

Controlar cargas inductivas DC

Las salidas DC del S7-1200 incluyen circuitos de supresión adecuados para las cargas inductivas en la mayoría de las aplicaciones. Puesto que los relés pueden utilizarse para cargas tanto DC como AC, no proporcionan protección interna. La figura siguiente muestra un ejemplo de un circuito de supresión para una carga DC.

En la mayoría de las aplicaciones es suficiente prever adicionalmente un diodo (A) paralelo a la carga inductiva. No obstante, si la aplicación requiere tiempos de desconexión más rápidos, se recomienda utilizar un diodo Zener (B).



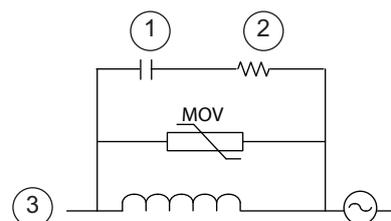
- ① Diodo 1N4001 o equivalente
- ② Zener de 8,2 V (salidas DC)
Zener de 36 V (salidas de relé)
- ③ Salida

Vigile que el diodo Zener tenga suficiente capacidad para la cantidad de corriente en el circuito de salida.

Salidas de relé que controlan cargas AC

Si se utiliza una salida de relé para conmutar cargas de 115 V/230 V AC, es preciso disponer redes de resistores/condensadores paralelas a la carga AC como se muestra en esta figura.

También es posible utilizar un varistor de óxido metálico (MOV) para limitar la tensión de pico. Vigile que la tensión de trabajo del varistor MOV sea como mínimo un 20% superior a la tensión de línea nominal.



- ① 0,1 μ F
- ② 100 a 120 Ω
- ③ Salida

Directrices para las cargas de lámpara

Las cargas de lámpara pueden averiar los contactos de relé, debido a la elevada sobrecorriente momentánea de conexión. Esta sobrecorriente momentánea es nominalmente 10 a 15 veces superior a la corriente en régimen permanente de una lámpara de tungsteno. Se recomienda intercalar un relé sustituable o un limitador de sobretensión para las cargas de lámparas que deben conmutarse con frecuencia durante la vida útil de la aplicación.

Principios básicos del PLC

3.1 Ejecución del programa de usuario

La CPU soporta los siguientes tipos de bloques lógicos que permiten estructurar eficientemente el programa de usuario:

- Los bloques de organización (OBs) definen la estructura del programa. Algunos OBs tienen reacciones y eventos de arranque predefinidos. No obstante, también es posible crear OBs con eventos de arranque personalizados.
- Las funciones (FCs) y los bloques de función (FBs) contienen el código de programa correspondiente a tareas específicas o combinaciones de parámetros. Cada FC o FB provee parámetros de entrada y salida para compartir datos con el bloque invocante. Un FB utiliza también un bloque de datos asociado (denominado DB instancia) para conservar el estado de valores durante la ejecución que pueden utilizar otros bloques del programa.
- Los bloques de datos (DBs) almacenan datos que pueden ser utilizados por los bloques del programa.

La ejecución del programa de usuario comienza con uno o varios bloques de organización (OBs) de arranque que se ejecutan una vez al cambiar a estado operativo RUN, seguidos de uno o varios OBs de ciclo que se ejecutan cíclicamente. También es posible asociar un OB a un evento de alarma que puede ser un evento estándar o de error y que se ejecuta cada vez que ocurre el evento en cuestión.

Una función (FC) o un bloque de función (FB) es un bloque de código del programa que puede llamarse desde un OB, o bien desde otra FC u otro FB. Son posibles los niveles siguientes:

- 16 desde OBs de ciclo o de arranque
- 4 desde OBs de alarma de retardo, alarma cíclica, alarma de proceso, alarma de error de tiempo o alarma de diagnóstico

Las FCs no están asociadas a ningún bloque de datos (DB) en particular, mientras que los FBs están vinculados directamente a un DB que utilizan para transferir parámetros, así como para almacenar valores intermedios y resultados.

El tamaño del programa de usuario, los datos y la configuración está limitado por la memoria de carga disponible y la memoria de trabajo de la CPU. El número de bloques soportado no está limitado dentro de la cantidad de memoria de trabajo disponible.

En cada ciclo se escribe en las salidas, se leen las entradas, se ejecutan las instrucciones del programa de usuario y se realiza el mantenimiento del sistema o procesamiento en segundo plano. En inglés, el ciclo también se llama "scan cycle" o "scan".

La Signal Board, así como los módulos de señales y de comunicación, se detectan y se dan de alta durante el arranque.

Nota

No se soporta la sustitución en caliente (hot swap) de la Signal Board ni de los módulos de señales ni de comunicación. La única excepción es la SIMATIC Memory Card que se puede insertar o extraer estando conectada la alimentación de la CPU.

De forma predeterminada, todas las E/S digitales y analógicas se actualizan de forma síncrona con el ciclo, utilizando un área de memoria interna denominada memoria imagen de proceso. La memoria imagen de proceso contiene una instantánea de las entradas y salidas físicas (es decir, las E/S físicas de la CPU, de la Signal Board y de los módulos de señales).

La CPU ejecuta las siguientes tareas:

- La CPU escribe las salidas desde la memoria imagen de proceso de las salidas en las salidas físicas.
- La CPU lee las entradas físicas inmediatamente antes de ejecutar el programa de usuario y almacena los valores de entrada en la memoria imagen de proceso de las entradas. Así se garantiza que estos valores sean coherentes durante la ejecución de las instrucciones programadas.
- La CPU ejecuta la lógica de las instrucciones programadas y actualiza los valores de salida en la memoria imagen de proceso de las salidas, en vez de escribirlos en las salidas físicas reales.

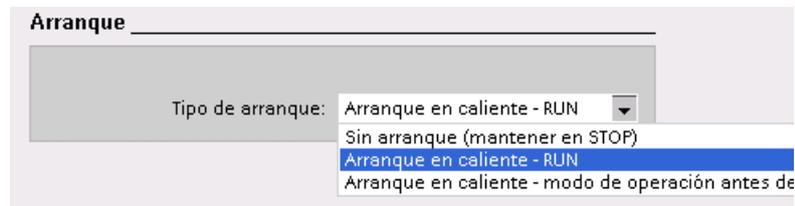
Este proceso ofrece una lógica coherente al ejecutar las instrucciones programadas durante un ciclo determinado y previene la fluctuación de las salidas físicas cuyo estado puede cambiar varias veces en la memoria imagen de proceso de las salidas.

Es posible definir si las E/S digitales y analógicas deben guardarse en la memoria imagen de proceso. Si un módulo se inserta en la vista de dispositivos, sus datos se encontrarán en la memoria imagen de proceso de la CPU S7-1200 (ajuste predeterminado). La CPU procesa el intercambio de datos entre el módulo y el área de la memoria imagen de proceso automáticamente durante la actualización de ésta. Para excluir E/S digitales o analógicas de la actualización automática de la memoria imagen de proceso, seleccione el dispositivo en cuestión en la "Configuración de dispositivos", abra la ficha "Propiedades", expanda en caso necesario para localizar las E/S deseadas y seleccione luego "Direcciones I/O/identificador HW". A continuación, cambie la entrada en "Memoria imagen de proceso:" de "IP cíclica" a "---". Para volver a incluir las E/S en la actualización automática de la memoria imagen de proceso, cambie de nuevo esta selección a "IP cíclica".

Es posible leer inmediatamente los valores de las entradas físicas y escribir inmediatamente los valores de las salidas físicas cuando se ejecuta una instrucción. Una lectura inmediata accede al estado actual de la entrada física y no actualiza la memoria imagen de proceso de las entradas, independientemente de si se ha configurado que la entrada se almacene en la memoria imagen de proceso. Una lectura inmediata en una salida física actualiza tanto la memoria imagen de proceso de las salidas (si se ha configurado que la salida se almacene en la memoria imagen de proceso) y la salida física. Añada el sufijo ":P" a la dirección E/S si desea que el programa acceda inmediatamente a los datos E/S directamente desde la E/S física, en vez de utilizar la memoria imagen de proceso.

Configurar los parámetros de arranque

En las propiedades de la CPU se puede configurar cómo debe arrancar tras desconectar y conectar la alimentación.



Seleccione si la CPU debe arrancar en estado operativo STOP o RUN, o bien en el estado operativo anterior (antes de desconectar y conectar la alimentación).

La CPU ejecuta un arranque en caliente antes de pasar al estado operativo RUN. El arranque en caliente inicializa toda la memoria no remanente a los valores iniciales predeterminados. No obstante, se conservan los valores actuales almacenados en la memoria remanente.

Nota

La CPU ejecuta siempre un re arranque tras una carga

Cada vez que un elemento del proyecto (p. ej. bloque de programa, bloque de datos o configuración hardware) se carga en la CPU, ésta ejecuta un re arranque en la próxima transición al estado operativo RUN. Además de borrar las entradas e inicializar las salidas y la memoria no remanente, el re arranque inicializa asimismo las áreas de memoria remanente.

Tras el re arranque posterior a la carga en la CPU, todas las transiciones de STOP a RUN subsiguientes ejecutan un arranque en caliente (que no inicializa la memoria remanente).

3.1.1 Estados operativos de la CPU

La CPU tiene tres estados operativos, a saber: STOP, ARRANQUE y RUN. Los LEDs de estado en el frente de la CPU indican el estado operativo actual.

- En estado operativo STOP, la CPU no ejecuta el programa. Entonces es posible cargar un proyecto en la CPU.
- En estado operativo ARRANQUE, los OBs de arranque (si existen) se ejecutan una vez. Los eventos de alarma no se procesan durante la fase de arranque del estado operativo RUN.
- El ciclo se ejecuta repetidamente en estado operativo RUN. Los eventos de alarma pueden ocurrir y procesarse en cualquier fase del ciclo del programa.

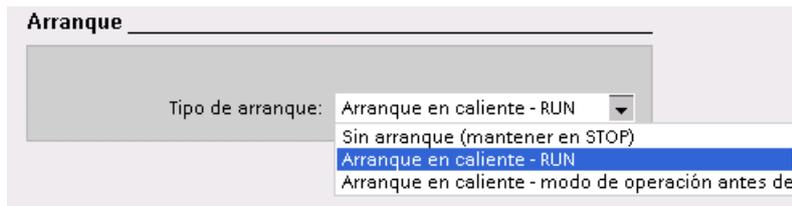
En estado operativo RUN no es posible cargar proyectos en la CPU.

La CPU soporta el arranque en caliente para pasar al estado operativo RUN. El arranque en caliente no incluye la inicialización de la memoria. Los datos de sistema no remanentes y los datos de usuario se inicializan en un arranque en caliente. Se conservan los datos de usuario remanentes.

El borrado total borra toda la memoria de trabajo, así como las áreas de memoria remanentes y no remanentes. Además, copia la memoria de carga en la memoria de trabajo. El borrado total no borra el búfer de diagnóstico ni tampoco los valores almacenados permanentemente de la dirección IP.

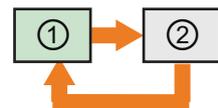
El tipo de arranque y método de re arranque de la CPU se pueden determinar mediante el software de programación. Este ajuste se encuentra en la "Configuración de dispositivos" de la CPU en "Arranque". Cuando se aplica tensión, la CPU ejecuta una secuencia de tests de diagnóstico de arranque e inicialización del sistema. A continuación, la CPU conmuta al tipo de arranque configurado. Determinados errores impiden que la CPU pase a estado operativo RUN. La CPU soporta los tipos de arranque siguientes:

- Estado operativo STOP
- Cambio a estado operativo RUN después del arranque en caliente
- Cambio al estado operativo anterior después del arranque en caliente



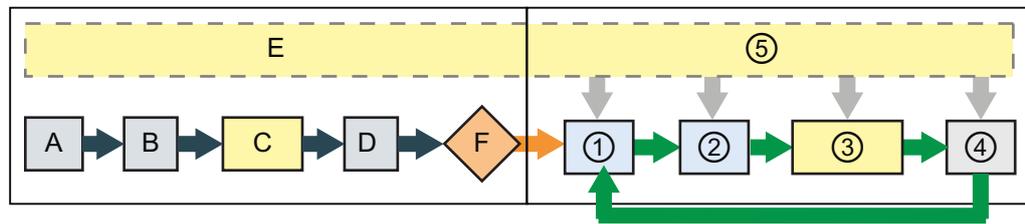
El estado operativo actual se puede cambiar mediante los comandos "STOP" o "RUN" de las herramientas online del software de programación. También se puede insertar una instrucción STP en el programa para cambiar la CPU a estado operativo STOP. Esto permite detener la ejecución del programa en función de la lógica.

En estado operativo STOP, la CPU ① procesa las peticiones de comunicación (según sea necesario) y ② realiza el autodiagnóstico. En estado operativo STOP, la CPU no ejecuta el programa de usuario y la memoria imagen de proceso no se actualiza automáticamente.



El proyecto sólo se puede cargar en la CPU si está se encuentra en estado operativo STOP.

En estado operativo RUN, la CPU ejecuta las tareas que muestra la figura siguiente.



ARRANQUE

- A Borra el área de memoria I
- B Inicializa las salidas con el último valor o el valor sustitutivo
- C Ejecuta los OBs de arranque
- D Copia el estado de las entradas físicas en la memoria I
- E Almacena los eventos de alarma en la cola de espera que deben procesarse en estado operativo RUN
- F Habilita la escritura de la memoria Q en las salidas físicas

RUN

- ① Escribe la memoria Q en las salidas físicas
- ② Copia el estado de las entradas físicas en la memoria I
- ③ Ejecuta los OBs de ciclo
- ④ Realiza autodiagnóstico
- ⑤ Procesa alarmas y comunicaciones en cualquier parte del ciclo

Procesamiento del ARRANQUE

Cada vez que el estado operativo cambia de STOP a RUN, la CPU borra las entradas de la memoria imagen de proceso, inicializa las salidas de la memoria imagen de proceso y procesa los OBs de arranque. En los accesos de lectura a las entradas de la memoria imagen de proceso realizados por instrucciones en los OBs de arranque se lee cero, en vez del valor actual de la entrada física. Por tanto, para leer el estado actual de una entrada física durante el estado operativo ARRANQUE, es preciso realizar una lectura inmediata. Luego se ejecutan los OBs de arranque, así como los FBs y FCs asociados. Si existe más de un OB de arranque, cada uno de ellos se ejecuta en el orden correspondiente al número de OB, comenzando con el número de OB más bajo.

Todo OB de arranque incluye información de arranque que ayuda a determinar la validez de los datos remanentes y el reloj en tiempo real. Es posible programar instrucciones dentro de los OBs de arranque para examinar estos valores de arranque y realizar las acciones apropiadas. Los OBs de arranque soportan las siguientes ubicaciones de arranque:

Entrada	Tipos de datos	Descripción
LostRetentive	BOOL	Este bit es verdadero (TRUE) si se han perdido las áreas de almacenamiento de datos remanentes
LostRTC	BOOL	Este bit es verdadero (TRUE) si se ha perdido el reloj en tiempo real

La CPU también ejecuta las siguientes tareas durante el procesamiento del arranque.

- Las alarmas se ponen en cola de espera, pero no se procesan durante la fase de arranque
- El tiempo de ciclo no se vigila durante la fase de arranque
- La configuración de HSC (contadores rápidos), PWM (modulación del ancho de pulso) y módulos PtP (comunicación punto a punto) se puede modificar durante el arranque
- Los HSC, la PWM y los módulos de comunicación punto a punto sólo funcionan en estado operativo RUN

Una vez finalizada la ejecución de los OBs de arranque, la CPU pasa a estado operativo RUN y procesa las tareas de control en un ciclo continuo.

Procesamiento del ciclo en estado operativo RUN

En cada ciclo, la CPU escribe en las salidas, lee las entradas, ejecuta el programa de usuario, actualiza los módulos de comunicación, realiza tareas de mantenimiento internas y reacciona a los eventos de alarma de usuario y peticiones de comunicación. Las peticiones de comunicación se procesan periódicamente durante el ciclo.

Estas acciones (excepto los eventos de alarma de usuario) se procesan con regularidad y en orden secuencial. Los eventos de alarma de usuario habilitados se procesan según su prioridad en el orden en que aparecen.

El sistema garantiza que el ciclo se procese dentro de un periodo denominado tiempo de ciclo máximo. De lo contrario, se generará un evento de error de tiempo.

- Todo ciclo comienza con la consulta de los valores actuales de las salidas digitales y analógicas de la memoria imagen de proceso. Estos valores se escriben luego en las salidas físicas de la CPU, la SB y los módulos SM configurados para la actualización automática de E/S (configuración predeterminada). Cuando una instrucción accede a una salida física, se actualizan tanto la memoria imagen de proceso de las salidas como la salida física.
- El ciclo continúa con la lectura de los valores actuales de las entradas digitales y analógicas de la CPU, la SB y los SMs configurados para la actualización automática de E/S (configuración predeterminada). Estos valores se escriben luego en la memoria imagen de proceso. Cuando una instrucción accede a una entrada física, se modifica el valor de ésta, pero no se actualiza la memoria imagen de proceso de las entradas.
- Tras leer las entradas, el programa de usuario se ejecuta desde la primera hasta la última instrucción. Esto incluye todos los OBs de ciclo, así como sus FCs y FBs asociados. Los OBs de ciclo se ejecutan en el orden correspondiente al número de OB, comenzando con el número de OB más bajo.

Las comunicaciones se procesan periódicamente durante todo el ciclo, siendo posible que se interrumpa la ejecución del programa de usuario.

El autodiagnóstico incluye comprobaciones periódicas del sistema y de estado de los módulos de E/S.

Las alarmas pueden ocurrir en cualquier parte del ciclo y son controladas por eventos. Cuando ocurre un evento, la CPU interrumpe el ciclo y llama el OB configurado para procesar ese evento. Una vez que el OB haya finalizado el procesamiento del evento, la CPU reanuda la ejecución del programa de usuario en el punto de interrupción.

Bloques de organización (OBs)

Los OBs controlan la ejecución del programa de usuario. Todo OB debe tener un número de OB unívoco. Algunos números menores que 200 están reservados para OBs predeterminados. La numeración de los demás OBs debe comenzar a partir de 200.

Determinados eventos de la CPU disparan la ejecución de un bloque de organización. Un OB no puede llamar a otro. Tampoco es posible llamar un OB desde una FC o un FB. Sólo un evento de arranque, p. ej. una alarma de diagnóstico o un intervalo, puede iniciar la ejecución de un OB. La CPU procesa los OBs según su clase de prioridad. Los OBs de mayor prioridad se ejecutan antes que los de menor prioridad. La clase de prioridad más baja es 1 (para el ciclo del programa principal) y la más alta es 27 (para las alarmas de error de tiempo).

Los OBs controlan los siguientes procesos:

- Los OBs de ciclo se ejecutan cíclicamente cuando la CPU se encuentra en estado operativo RUN. El bloque principal del programa es un OB de ciclo. Éste contiene las instrucciones que controlan el programa y permite llamar otros bloques de usuario. Es posible utilizar varios OBs de ciclo. Éstos se ejecutan en orden numérico. El OB 1 es el bloque predeterminado. Los demás OBs de ciclo deben identificarse como OB 200 o superior.
- Los OBs de arranque se ejecutan una vez cuando el estado operativo de la CPU cambia de STOP a RUN, al arrancar a estado operativo RUN y en una transición ordenada de STOP a RUN. Una vez finalizado, se comienza a ejecutar el OB de ciclo. Es posible utilizar varios OBs de arranque. El OB 100 es el bloque predeterminado. El número de los demás OBs debe ser 200 o superior.
- Los OBs de alarma de retardo se ejecutan al cabo de un intervalo posterior a un evento configurado en la instrucción de alarma de arranque (SRT_DINT). El tiempo de retardo se especifica en el parámetro de entrada de la instrucción avanzada SRT_DINT. Los OBs de alarma de retardo interrumpen la ejecución cíclica del programa una vez transcurrido un tiempo de retardo especificado. Es posible configurar como máximo 4 eventos de retardo en cualquier momento. Por cada evento de retardo configurado se permite un OB. El número del OB de alarma de retardo debe ser 200 o superior.
- Los OBs de alarma cíclica se ejecutan en intervalos periódicos. Los OBs de alarma cíclica interrumpen la ejecución cíclica del programa en intervalos definidos, p. ej. cada 2 segundos. Es posible configurar como máximo 4 eventos de alarma cíclica. Por cada evento de alarma cíclica configurado se permite un OB. El número del OB debe ser 200 o superior.
- Los OBs de alarma de proceso se ejecutan cuando ocurre el evento de hardware correspondiente, incluyendo flancos ascendentes y descendentes en las entradas digitales integradas y eventos de contadores rápidos (HSC). Los OBs de alarma de proceso interrumpen la ejecución cíclica del programa como reacción a una señal de un evento de hardware. Los eventos se definen en las propiedades de la configuración hardware. Por cada evento de hardware configurado se permite un OB. El número del OB debe ser 200 o superior.
- Los OBs de error de tiempo se ejecutan cuando se detecta un error de tiempo. Los OBs de error de tiempo interrumpen la ejecución cíclica del programa cuando se rebasa el tiempo de ciclo máximo. El tiempo de ciclo máximo se define en las propiedades del PLC. El OB 80 es el único número de OB soportado para el evento de error de tiempo. Es posible configurar la acción que debe realizarse si no existe el OB 80: ignorar el error o cambiar a STOP.
- Los OBs de alarma de diagnóstico se ejecutan cuando se detecta y notifica un error de diagnóstico. Los OBs de alarma de diagnóstico interrumpen la ejecución cíclica del programa cuando el módulo apto para diagnóstico detecta un error (si se ha habilitado la alarma de diagnóstico para ese módulo). El OB 82 es el único número de OB soportado para el evento de error de diagnóstico. Si el programa no contiene ningún OB de diagnóstico, la CPU se puede configurar para que ignore el error o cambie a STOP.

3.1.2 Prioridades y colas de espera para la ejecución de eventos

El procesamiento de la CPU es controlado por eventos. Los eventos disparan OBs de alarma que deben ejecutarse. El OB de alarma de un evento se especifica durante la creación del bloque, durante la configuración de dispositivos, o bien con una instrucción ATTACH o DETACH. Algunos eventos ocurren con regularidad, tales como los eventos de ciclo o cíclicos. Otros eventos ocurren una sola vez, tales como el evento de arranque y los eventos de alarma de retardo. Algunos eventos ocurren cuando se produce un cambio disparado por hardware, p. ej. un flanco en una entrada o un evento de contador rápido. Asimismo, hay eventos p. ej. de error de diagnóstico o de error de tiempo que ocurren solamente cuando se produce un error. Las prioridades de eventos, los grupos de prioridad y las colas de espera sirven para determinar el orden de procesamiento de los OBs de alarma.

El evento de ciclo ocurre una vez por ciclo del programa. Durante el ciclo del programa, la CPU escribe en las salidas, lee las entradas y ejecuta los OBs de ciclo. El evento de ciclo es necesario y siempre está habilitado. Es posible no tener OBs de ciclo, o bien tener varios OBs de ciclo seleccionados para el evento de ciclo. Una vez disparado el evento de ciclo, se ejecuta el OB de ciclo con el número más bajo (normalmente el OB1). Los demás OBs de ciclo se ejecutan secuencialmente, en orden numérico, dentro del ciclo de programa.

Los eventos de alarma cíclica permiten configurar la ejecución de un OB de alarma en un intervalo configurado. El intervalo se configura al crear y seleccionar el OB para que sea de alarma cíclica. Los eventos cíclicos interrumpen el ciclo del programa y ejecutan el OB de alarma cíclica (el evento cíclico está en un grupo de mayor prioridad que el evento de ciclo). Un solo OB de alarma cíclica puede asignarse a un evento cíclico. La CPU soporta cuatro eventos de alarma cíclica. Los OBs de alarma cíclica tienen una propiedad de desfase. Por tanto, las alarmas cíclicas con el mismo periodo pueden ejecutarse con un offset la una de la otra por la cantidad de desfase.

El evento de arranque ocurre una vez al producirse un cambio de STOP a RUN y lanza la ejecución de los OBs de arranque. Es posible seleccionar varios OBs para el evento de arranque. Los OBs de arranque se ejecutan en orden numérico.

Los eventos de alarma de retardo permiten configurar la ejecución de un OB de alarma transcurrido un tiempo de retardo definido. El tiempo de retardo se especifica con la instrucción SRT_DINT. Los eventos de alarma de retardo interrumpen el ciclo del programa, con el fin de ejecutar el OB de alarma de retardo. Un solo OB de alarma de retardo puede asignarse a un evento de retardo. La CPU soporta cuatro eventos de retardo.

Los eventos de alarma de proceso son disparados por un cambio en el hardware, p. ej. un flanco ascendente o descendente en una entrada, o bien un evento de contador rápido (HSC). Sólo un OB de alarma puede estar seleccionado para cada evento de alarma de proceso. Los eventos de alarma de proceso se habilitan en la "Configuración de dispositivos". Los OBs se definen para el evento en la "Configuración de dispositivos" o con una instrucción ATTACH en el programa de usuario. La CPU soporta varios eventos de alarma de proceso. Los eventos exactos dependen del modelo de CPU y del número de entradas.

Los eventos de error de tiempo y diagnóstico son disparados cuando la CPU detecta un error. Estos eventos están en un grupo de mayor prioridad que los demás eventos de alarma y pueden interrumpir la ejecución de los eventos de alarma de retardo, alarma cíclica y alarma de proceso. Es posible definir un OB de alarma para cada uno de los eventos de error de tiempo y diagnóstico.

Prioridades y colas de espera para la ejecución de eventos

El número de eventos pendientes (en cola de espera) de una sola fuente se limita utilizando una cola diferente para cada tipo de evento. Al alcanzar el límite de eventos pendientes de un determinado tipo, se pierde el evento siguiente. Para más información sobre el desbordamiento de colas de espera, consulte el apartado "Eventos de error de tiempo".

Todo evento de la CPU tiene una prioridad asociada y las prioridades de eventos se clasifican en clases de prioridad. La tabla siguiente ofrece una sinopsis de la profundidad de las colas de espera, las clases de prioridad y las prioridades de los eventos de CPU soportados.

Nota

No es posible modificar la prioridad, ni la asignación a las clases de prioridad, ni tampoco la profundidad de las colas de espera.

Generalmente, los eventos se procesan según su prioridad (primero los de mayor prioridad). Los eventos de igual prioridad se procesan según su orden de aparición.

Tipo de evento (OB)	Cantidad	Números de OB válidos	Profundidad de la cola de espera	Clase de prioridad	Prioridad
Ciclo	1 evento de ciclo Se permiten varios OBs	1 (estándar) 200 o superior	1	1	1
Arranque	1 evento de arranque ¹ Se permiten varios OBs	100 (estándar) 200 o superior	1		1
Retardo	4 eventos de retardo 1 OB por evento	200 o superior	8	2	3
Cíclico	4 eventos cíclicos 1 OB por evento	200 o superior	8		4
Flancos	16 eventos de flanco ascendente 16 eventos de flanco descendente 1 OB por evento	200 o superior	32		5
HSC	6 eventos CV = PV 6 eventos de cambio de sentido 6 eventos de reset externo 1 OB por evento	200 o superior	16		6
Error de diagnóstico	1 evento	Sólo 82	8		9
Evento de error de tiempo/evento de tiempo MaxCycle	1 evento de error de tiempo 1 evento de tiempo MaxCycle	Sólo 80	8	3	26

Tipo de evento (OB)	Cantidad	Números de OB válidos	Profundidad de la cola de espera	Clase de prioridad	Prioridad
2 eventos de tiempo MaxCycle	1 2 eventos de tiempo MaxCycle	No se llama ningún OB	-	3	27
¹ Casos especiales del evento de arranque <ul style="list-style-type: none"> • Los eventos de arranque y de ciclo no ocurren nunca simultáneamente, ya que el evento de arranque se debe haber finalizado antes de poder iniciar el evento de ciclo (controlado por el sistema operativo). • Ningún evento puede interrumpir el evento de arranque. Los eventos que ocurran durante el arranque se pondrán en cola de espera para su posterior procesamiento una vez finalizado el evento de arranque. 					

Tras iniciarse la ejecución de un OB, su procesamiento no se puede interrumpir al aparecer otro evento cuya clase de prioridad sea igual o inferior. Estos eventos se ponen en cola de espera para su procesamiento posterior, permitiendo que finalice el OB actual.

No obstante, un evento de una clase de prioridad superior sí interrumpe el OB actual. La CPU ejecuta entonces el OB correspondiente al evento de mayor prioridad. Tras finalizar el OB de mayor prioridad, la CPU ejecuta los OBs de los demás eventos en cola de espera en esta clase de prioridad superior, según la prioridad en la clase. Si no hay otros eventos pendientes (en cola de espera) en esta clase de prioridad superior, la CPU regresa a la clase de prioridad inferior y reanuda el procesamiento del OB interrumpido en el punto en el que se interrumpió el procesamiento de ese OB.

Latencia de alarmas

La latencia de los eventos de alarma (es decir, el tiempo que transcurre desde que la CPU notifica que ha ocurrido un evento hasta que comienza la ejecución de la primera instrucción en el OB que procesa este evento) es de aproximadamente 210 µs, siempre que un OB de ciclo sea el único subprograma activo que procese el evento de alarma en el momento de su aparición.

Eventos de error de tiempo

La aparición de cualquiera de las condiciones de error de tiempo diferentes causa un evento de error de tiempo. Se soportan los siguientes errores de tiempo:

- Rebase del tiempo de ciclo máximo
- No se puede iniciar el OB solicitado
- Desbordamiento de la cola de espera

El error de rebase del tiempo de ciclo máximo ocurre si el ciclo de programa no finaliza dentro del tiempo de ciclo máximo especificado. Encontrará más información acerca de este error y sobre cómo configurar el tiempo de ciclo máximo e inicializar la vigilancia del tiempo de ciclo en el apartado "Vigilancia del tiempo de ciclo (Página 48)".

El OB solicitado no se puede iniciar si una alarma cíclica o una alarma de retardo solicita un OB, pero éste ya se está ejecutando.

La cola de espera se desborda si las alarmas ocurren más rápidamente de lo que pueden procesarse. El número de eventos pendientes (en cola de espera) se limita utilizando una cola diferente para cada tipo de evento. Si ocurre un evento estando llena la cola de espera correspondiente, se genera un evento de error de tiempo.

Todos los eventos de error de tiempo disparan la ejecución del OB 80 (si existe). Si el OB 80 no existe, la CPU ignora el error. Si el tiempo de ciclo máximo se rebasa dos veces en un mismo ciclo del programa sin que se inicialice el temporizador de ciclo, la CPU pasará a STOP, independientemente de si existe el OB 80. Consulte el apartado "Vigilancia del tiempo de ciclo". (Página 48)

El OB 80 incluye información de arranque que permite determinar qué evento y OB ha generado el error de tiempo. Es posible programar instrucciones dentro del OB 80 para examinar estos valores de arranque y realizar las acciones apropiadas. El OB 80 soporta las siguientes ubicaciones de arranque:

Entrada	Tipo de datos	Descripción
fault_id	BYTE	16#01 - rebase del tiempo de ciclo máximo 16#02 - no se puede iniciar el OB solicitado 16#07 y 16#09 - desbordamiento de la cola de espera
csg_OBnr	OB_ANY	Número de OB que se estaba ejecutando cuando ocurrió el error
csg_prio	UINT	Prioridad del OB que ha causado el error

Cuando se crea un proyecto nuevo, no existe ningún OB 80 de error de tiempo. Si desea agregar un OB 80 de error de tiempo al proyecto, haga doble clic en "Agregar nuevo bloque" en "Bloques de programa" en el árbol del proyecto, seleccione luego "Bloque de organización" y después "OB de error de tiempo".

Eventos de error de diagnóstico

Algunos dispositivos pueden detectar y notificar errores de diagnóstico. La aparición o eliminación de cualquiera de las diferentes condiciones de error de diagnóstico ocasiona un evento de error de diagnóstico. Se soportan los siguientes errores de diagnóstico:

- Falta alimentación externa
- Límite alto excedido
- Límite bajo excedido
- Rotura de hilo
- Cortocircuito

Todos los eventos de error de diagnóstico disparan la ejecución del OB 82 (si existe). Si el OB 82 no existe, la CPU ignora el error. Cuando se crea un proyecto nuevo, no existe ningún OB 82 de error de diagnóstico. Si desea agregar un OB 82 de error de diagnóstico al proyecto, haga doble clic en "Agregar nuevo bloque" en "Bloques de programa" en el árbol del proyecto, seleccione luego "Bloque de organización" y después "OB de alarma de diagnóstico".

El OB 82 incluye información de arranque que ayuda a determinar si el evento se debe a la aparición o eliminación de un error, así como el dispositivo y canal que han notificado el error. Es posible programar instrucciones dentro del OB 82 para examinar estos valores de arranque y realizar las acciones apropiadas. El OB 82 soporta las siguientes ubicaciones de arranque:

Entrada	Tipo de datos	Descripción
IOstate	WORD	Estado de E/S del dispositivo
laddr	HW_ANY	ID de hardware del dispositivo o unidad funcional que ha notificado el error
channel	UINT	Número de canal
multierror	BOOL	TRUE (verdadero) si ha ocurrido más de un error (<i>no se soporta en versiones anteriores</i>)

El bit 4 de IO_state indica si el evento se debe a la aparición o eliminación de un error. El bit 4 es "1" si existe un error (p. ej. rotura de hilo), o bien "0" si se ha solucionado el error.

La entrada en KOP contiene el identificador de hardware (ID HW) del dispositivo o unidad funcional que ha devuelto el error. La ID HW se asigna automáticamente cuando se insertan componentes en la vista de dispositivos o redes y aparece en la ficha "Constantes" de "Variables PLC". También se asigna automáticamente un nombre a la ID HW. Estas entradas de la ficha "Constantes" de "Variables PLC" no se pueden modificar.

El número de canal comienza en 0 para la primera entrada (analógica o digital) y en 64 para la primera salida (analógica o digital). Los diferentes offsets son necesarios para distinguir las entradas de las salidas, en caso de que el dispositivo disponga de ambas. Si un error afecta al dispositivo o unidad funcional en su totalidad (p. ej. falta de alimentación externa), se activa el bit más significativo de la palabra del número de canal (número de canal 32768).

Vigilancia del tiempo de ciclo

El tiempo de ciclo es el tiempo que requiere el sistema operativo de la CPU para ejecutar la fase cíclica del estado operativo RUN. La CPU ofrece dos métodos para vigilar el tiempo de ciclo:

- Tiempo de ciclo máximo
- Tiempo de ciclo mínimo fijo

La vigilancia del tiempo de ciclo comienza una vez finalizado el evento de arranque. Esta función se configura en la "Configuración de dispositivos" de la CPU en "Tiempo de ciclo".

La CPU vigila siempre el ciclo y reacciona si se rebasa el tiempo de ciclo máximo. Si se rebasa el tiempo de ciclo máximo configurado, se generará un error que se procesa de dos maneras posibles:

- Si no existe el OB 80 de error de tiempo, la CPU generará un error y continuará ejecutando el programa de usuario.
- Si existe un OB 80 de error de tiempo, la CPU ejecutará el OB 80.

La instrucción RE_TRIGR (Volver a lanzar la vigilancia del tiempo de ciclo) permite resetear el temporizador que mide el tiempo de ciclo. No obstante, esta instrucción funciona únicamente si se ejecuta en un OB de ciclo. La instrucción RE_TRIGR se ignorará si se ejecuta en el OB 80. Si el tiempo de ciclo máximo se rebasa dos veces en un mismo ciclo del programa, sin que la instrucción RE_TRIGR se ejecute entre los dos rebases, la CPU cambiará inmediatamente a estado operativo STOP. Si la instrucción RE_TRIGR se ejecuta repetidas veces, ello puede ocasionar un bucle infinito o un ciclo muy prolongado.

Generalmente, el ciclo se ejecuta tan rápido como sea posible y el ciclo siguiente comienza cuando finaliza el ciclo actual. En función del programa de usuario y las tareas de comunicación, el tiempo de ciclo puede fluctuar de ciclo en ciclo. Para eliminar esta fluctuación, la CPU soporta un tiempo de ciclo mínimo fijo opcional (o "ciclo fijo"). Si está habilitada esta función opcional y se ha definido un tiempo de ciclo mínimo fijo en ms, la CPU mantendrá el tiempo de ciclo mínimo con una tolerancia de ± 1 ms para la finalización de cada ciclo.

Si la CPU finaliza el ciclo normal antes del tiempo de ciclo mínimo especificado, aprovechará el tiempo restante para realizar tareas de diagnóstico en runtime y/o procesar peticiones de comunicación. De esta manera, la CPU utiliza siempre un tiempo fijo para finalizar un ciclo.

Si la CPU no finaliza el ciclo normal dentro del tiempo de ciclo mínimo especificado, lo finalizará normalmente (incluyendo el procesamiento de las peticiones de comunicación), sin que el rebase del tiempo de ciclo mínimo cause una reacción del sistema. La tabla siguiente muestra los rangos y valores predeterminados para las funciones de vigilancia del tiempo de ciclo.

Tiempo de ciclo	Rango (ms)	Valor predeterminado
Tiempo de ciclo máximo ¹	1 a 6000	150 ms
Tiempo de ciclo mínimo fijo ²	1 hasta tiempo de ciclo máximo	Inhibido

¹ El tiempo de ciclo máximo siempre está habilitado. Configure un tiempo de ciclo comprendido entre 1 y 6000 ms. El valor predeterminado es 150 ms.

² El tiempo de ciclo mínimo fijo es opcional y está inhibido de forma predeterminada. En caso necesario, configure un tiempo de ciclo comprendido entre 1 ms y el tiempo de ciclo máximo.

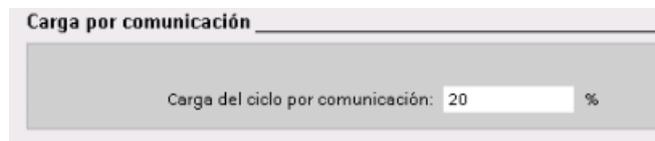
Configurar el tiempo de ciclo y la carga de comunicación

Las propiedades de la CPU en la "Configuración de dispositivos" permiten configurar los siguientes parámetros:

- Tiempo de ciclo: Es posible introducir un tiempo de ciclo máximo. También se puede definir un tiempo de ciclo mínimo fijo.



- Carga de comunicación: Es posible configurar un porcentaje del tiempo que debe dedicarse a las tareas de comunicación.



Encontrará más información sobre el ciclo en el apartado "Vigilancia del tiempo de ciclo". (Página 48)

3.1.3 Memoria de la CPU

Gestión de la memoria

La CPU provee las áreas de memoria siguientes para almacenar el programa de usuario, los datos y la configuración:

- La memoria de carga permite almacenar de forma no volátil el programa de usuario, los datos y la configuración. Cuando un proyecto se carga en la CPU, se almacena primero en el área de memoria de carga. Esta área se encuentra bien sea en una Memory Card (si está disponible) o en la CPU. Esta área de memoria no volátil se conserva incluso tras una pérdida de potencia. La Memory Card ofrece mayor espacio de almacenamiento que el integrado en la CPU.
- La memoria de trabajo ofrece almacenamiento volátil para algunos elementos del proyecto mientras se ejecuta el programa de usuario. La CPU copia algunos elementos del proyecto desde la memoria de carga en la memoria de trabajo. Esta área volátil se pierde si se desconecta la alimentación. La CPU la restablece al retornar la alimentación.
- La memoria remanente permite almacenar de forma no volátil un número limitado de valores de la memoria de trabajo. El área de memoria remanente se utiliza para almacenar los valores de algunas posiciones de memoria durante una pérdida de potencia. Si ocurre un corte de alimentación, la CPU dispone de suficiente tiempo de retención para respaldar los valores de un número limitado de posiciones de memoria definidas. Estos valores remanentes se restablecen al retornar la alimentación.

Para ver el uso de memoria del proyecto actual, haga clic con el botón derecho del ratón en la CPU (o uno de sus bloques) y elija el comando "Carga de la memoria" del menú contextual. Para ver el uso de memoria de la CPU actual, haga doble clic en "Online y diagnóstico", expanda "Diagnóstico" y seleccione "Memoria".

Memoria remanente

Para impedir la pérdida de datos tras un corte de alimentación, es posible definir que ciertos datos sean remanentes. Los siguientes datos pueden configurarse para que sean remanentes:

- Área de marcas (M): El ancho preciso de la memoria para el área de marcas puede definirse en la tabla de variables PLC o el plano de ocupación. El área de marcas remanente comienza siempre en MB0, abarcando consecutivamente un determinado número de bytes. Para definir este valor, haga clic en el botón "Remanencia" de la barra de herramientas de la tabla de variables PLC o del plano de ocupación. Introduzca el número de bytes M que deben ser remanentes a partir de MB0.
- Variables de un bloque de función (FB): Si un FB se ha creado estando activada la casilla "Sólo con direccionamiento simbólico", la interfaz del bloque de este FB incluirá una columna "Remanencia". En esta columna es posible seleccionar "Remanente" o "No remanente" individualmente para cada una de las variables. Un DB instancia que haya sido creado al insertar este FB en el editor de programas muestra asimismo la columna "Remanencia", aunque no permite editarla. El estado remanente no se puede modificar desde la interfaz del bloque del DB de instancia para un FB que haya sido configurado "Sólo con direccionamiento simbólico".

Si un FB ha sido creado estando desactivada la casilla "Sólo con direccionamiento simbólico", la interfaz del bloque de este FB no incluirá la columna "Remanencia". Un DB instancia que haya sido creado al insertar este FB en el editor de programas muestra y permite editar la columna "Remanencia". En este caso, si se activa la opción "Remanente" para alguna de las variables, se seleccionarán **todas** las variables. Por analogía, si se desactiva la opción "Remanente" para alguna de las variables, se deseleccionarán **todas** las variables. Si un FB se ha configurado **sin** el atributo "Sólo con direccionamiento simbólico", el estado remanente se puede cambiar desde la interfaz del bloque del DB instancia, pero todas las variables se ajustan conjuntamente al mismo estado remanente.

Tras haber creado el FB no es posible modificar la opción "Sólo con direccionamiento simbólico". Esta opción sólo se puede seleccionar cuando se crea el FB. Para determinar si un FB existente se ha configurado "Sólo con direccionamiento simbólico", haga clic con el botón derecho del ratón en el FB en el árbol del proyecto, elija "Propiedades" y seleccione luego "Atributos".

- Variables de un bloque de datos global: El comportamiento de un DB global respecto a la asignación del estado remanente es similar al de un FB. En función del ajuste de direccionamiento simbólico, es posible definir el estado remanente de algunas o todas las variables de un bloque de datos global.
 - Si el atributo "Sólo con direccionamiento simbólico" está activado para el DB, el estado remanente se podrá ajustar para cada una de las variables.
 - Si el atributo "Sólo con direccionamiento simbólico" está desactivado para el DB, el ajuste de remanencia se aplicará a todas las variables del DB. Por tanto, todas o ninguna de las variables serán remanentes.

Un total de 2048 bytes de datos pueden ser remanentes. Para ver cuánto espacio está disponible, haga clic en el botón "Remanencia" de la barra de herramientas de la tabla de variables PLC o del plano de ocupación. Aunque aquí se especifica el rango remanente para la memoria M, la segunda fila indica la memoria restante disponible en total para M y DB conjuntamente.

Búfer de diagnóstico

La CPU soporta un búfer de diagnóstico que contiene una entrada para cada evento de diagnóstico. Toda entrada incluye la fecha y hora del evento, así como su categoría y descripción. Las entradas se visualizan en orden cronológico. El evento más reciente aparece en primer lugar. Estando conectada la alimentación de la CPU, los 50 eventos más recientes están disponibles en este búfer. Cuando se llena el búfer, un evento nuevo reemplaza al evento más antiguo. Cuando se corta la alimentación, se almacenan los diez eventos más recientes.

Los siguientes tipos de eventos se registran en el búfer de diagnóstico:

- Todo evento de diagnóstico del sistema, p. ej. errores de la CPU y de los módulos
- Todo cambio de estado de la CPU (todo arranque, toda transición a STOP, toda transición a RUN)

Para acceder al búfer de diagnóstico es preciso estar online. El búfer se encuentra en "Online y diagnóstico / Diagnóstico / Búfer de diagnóstico". Encontrará más información acerca de la búsqueda y eliminación de errores en el capítulo "Online y diagnóstico".

Reloj en tiempo real

La CPU soporta un reloj en tiempo real. Un condensador de alto rendimiento suministra la energía necesaria para que el reloj pueda seguir funcionando mientras está desconectada la alimentación de la CPU. El condensador de alto rendimiento se carga mientras está conectada la alimentación de la CPU. Tras haber estado conectada la alimentación de la CPU como mínimo 2 horas, la carga del condensador de alto rendimiento alcanzará para que el reloj pueda funcionar 10 días por lo general.

El reloj en tiempo real sirve para ajustar la hora del sistema que es el tiempo universal coordinado (UTC). STEP 7 Basic ajusta el reloj en tiempo real a la hora del sistema. Se dispone de instrucciones que permiten leer la hora del sistema (RD_SYS_T) o la hora local (RD_LOC_T). La hora local se calcula según la diferencia con respecto a la zona horaria y al horario de verano que se han ajustado en la "Configuración de dispositivos" del reloj de la CPU.

El reloj en tiempo real de la CPU se configura en la propiedad "Hora". También es posible habilitar el horario de verano y determinar su fecha y hora de inicio y fin. Para ajustar el reloj en tiempo real es preciso estar online y en la vista "Online y diagnóstico" de la CPU. Utilice la función "Ajustar hora".

Marcas de sistema y de ciclo

Los bytes de "marcas de sistema" y "marcas de ciclo" se habilitan en las propiedades de la CPU. La lógica del programa puede referenciar los distintos bits de estas funciones.

- Un byte del área de marcas (M) se puede asignar a las marcas de sistema. El byte de marcas de sistema pone a disposición los siguientes cuatro bits que puede referenciar el programa de usuario:
 - El bit "Siempre 0 (low)" está siempre puesto a 0.
 - El bit "Siempre 1 (high)" está siempre puesto a 1.
 - El bit "Diagrama de diagnóstico modificado" se pone a 1 durante un ciclo, una vez que la CPU registre un evento de diagnóstico. Puesto que la CPU no activa el bit "Diagrama de diagnóstico modificado" hasta el final de la primera ejecución de los OBs de ciclo, el programa de usuario no puede detectar si ha cambiado el diagnóstico durante la ejecución de los OBs de arranque, o bien durante la primera ejecución de los OBs de ciclo.
 - El bit "Primer ciclo" se pone a 1 durante el primer ciclo tras finalizar el OB de arranque. (Una vez finalizada la ejecución del primer ciclo, el bit "Primer ciclo" se pone a 0.)
- Es posible asignar un byte de marcas de ciclo en el área de marcas. Todo bit del byte de marcas de ciclo genera un impulso de onda cuadrada. El byte de marcas de ciclo ofrece 8 frecuencias diferentes, comprendidas entre 0,5 Hz (lenta) hasta 10 Hz (rápida). Estos bits pueden utilizarse como bits de control para disparar acciones cíclicas en el programa de usuario, especialmente si se combinan con instrucciones de detección de flancos.

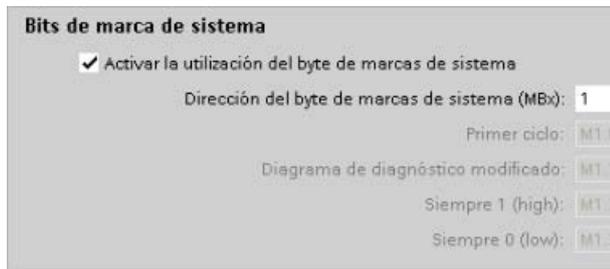
La CPU inicializa estos bytes cuando el estado operativo cambia de STOP a ARRANQUE. Los bits de las marcas de ciclo cambian de forma síncrona al reloj de la CPU durante los estados operativos ARRANQUE y RUN.

PRECAUCIÓN

Si se sobrescriben los bits de marcas de sistema o de ciclo, se podrían corromper los datos en estas funciones. Debido a ello, el programa de usuario funcionará incorrectamente, lo que podría ocasionar daños materiales y lesiones corporales.

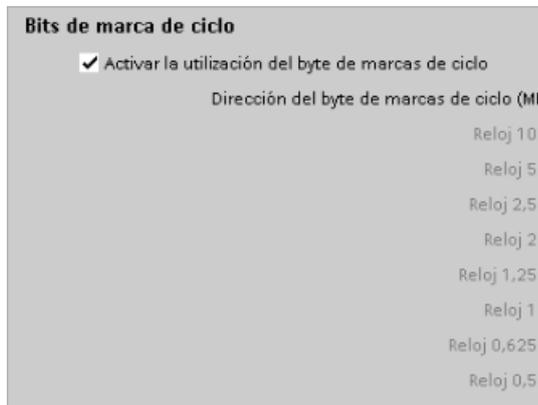
Puesto que las marcas de ciclo y de sistema forman no están reservadas en la memoria M, las instrucciones o comunicaciones pueden escribir en estas posiciones de memoria y corromper los datos.

Evite escribir datos en estas direcciones para garantizar el funcionamiento correcto de estas funciones y prevea siempre un circuito de parada de emergencia para el proceso o la máquina.



Las marcas de sistema configuran un byte que se activa (se pone a 1) en las siguientes condiciones.

- Primer ciclo: Se activa en el primer ciclo en modo RUN
- Diagrama de diagnóstico modificado:
- Siempre 1 (high): Siempre activado
- Siempre 0 (low): Siempre desactivado



Las marcas de ciclo configuran un byte que activa y desactive los distintos bits en intervalos fijos.

Las marcas de ciclo generan un impulso de onda cuadrada en el bit correspondiente del área de marcas. Estos bits pueden utilizarse como bits de control para disparar acciones cíclicas en el programa de usuario, especialmente si se combinan con instrucciones de detección de flancos.

Configurar la reacción de las salidas cuando la CPU esté en STOP

Es posible configurar la reacción de las salidas digitales y analógicas cuando la CPU se encuentre en estado operativo STOP. Es posible congelar los valores de las salidas o aplicar un valor sustitutivo a cualquier salida de una CPU, SB o SM:

- Sustituir un valor de salida específico (ajuste predeterminado): Para cada salida (canal) de la CPU, de la SB o del SM se define un valor sustitutivo.

El valor sustitutivo predeterminado de los canales de salida digitales es OFF y el de los canales de salida analógicos es 0.

- Congelar las salidas a su último estado: Las salidas conservan su valor actual en el momento de la transición de RUN a STOP. Después del arranque, las salidas se ajustan al valor sustitutivo predeterminado.

La reacción de las salidas se configura en la "Configuración de dispositivos". Seleccione los dispositivos individuales y utilice la ficha "Propiedades" para configurar las salidas de cada dispositivo.

Cuando la CPU cambia de RUN a STOP, conserva la memoria imagen de proceso y escribe los valores correspondientes en las salidas digitales y analógicas según la configuración.

3.1.4 Protección por contraseña de la CPU S7-1200

La CPU ofrece 3 niveles de protección para restringir el acceso a determinadas funciones. Al configurar el nivel de protección y la contraseña de una CPU, se limitan las funciones y áreas de memoria accesibles sin introducir una contraseña.

Para configurar la contraseña, proceda del siguiente modo:

1. Seleccione la CPU en la "Configuración de dispositivos".
2. Seleccione la ficha "Propiedades" en la ventana de inspección.
3. Elija la propiedad "Protección" para seleccionar el nivel de protección e introducir una contraseña.

La contraseña distingue entre mayúsculas y minúsculas.

Todo nivel permite acceder a ciertas funciones sin introducir una contraseña. El ajuste predeterminado de la CPU es "sin restricción" y "sin protección por contraseña". Para restringir el acceso a una CPU, es preciso configurar sus propiedades e introducir la contraseña.

Si la contraseña se introduce a través de una red, ésto no afecta la protección por contraseña de la CPU. Una CPU protegida por contraseña permite el acceso ilimitado sólo a un usuario a la vez. La protección por contraseña no es aplicable a la ejecución de las instrucciones del programa de usuario incluyendo las funciones de comunicación. Si se introduce la contraseña correcta es posible acceder a todas las funciones.

El nivel de protección de la CPU no restringe la comunicación entre PLCs (mediante instrucciones de comunicación en los bloques lógicos). Tampoco se restringen las funciones HMI.

Nivel de protección	Restricciones de acceso
Sin protección	Permite el acceso completo sin protección por contraseña.
Protección contra escritura	Ofrece acceso a los dispositivos HMI y permite toda la comunicación entre PLCs sin protección por contraseña. La contraseña se requiere para modificar (escribir en) la CPU y cambiar su estado operativo (RUN/STOP).
Protección contra lectura/escritura	Ofrece acceso a los dispositivos HMI y permite toda la comunicación entre PLCs sin protección por contraseña. La contraseña se requiere para leer los datos de la CPU, modificar (escribir en) la CPU y cambiar su estado operativo (RUN/STOP).

3.1.5 Recuperación si se olvida la contraseña

Si se ha olvidado la contraseña de una CPU protegida por contraseña, es preciso utilizar una tarjeta de transferencia vacía para borrar el programa protegido por contraseña. La tarjeta de transferencia vacía borra la memoria de carga interna de la CPU. Luego es posible cargar un programa de usuario nuevo desde STEP 7 Basic en la CPU.

Encontrará más información sobre cómo crear y utilizar una tarjeta de transferencia vacía en el apartado Tarjeta de transferencia (Página 71).

 ADVERTENCIA
<p>Si se inserta una tarjeta de transferencia en una CPU que esté funcionando, la CPU pasará a STOP. Los dispositivos de control pueden fallar y provocar condiciones no seguras, causando a su vez reacciones inesperadas de los equipos controlados. Las reacciones inesperadas podrían producir la muerte, lesiones corporales graves y/o daños materiales.</p>

Extraiga la tarjeta de transferencia antes de cambiar la CPU a estado operativo RUN.

3.2 Almacenamiento de datos, áreas de memoria y direccionamiento

La CPU ofrece varias opciones para almacenar datos durante la ejecución del programa de usuario:

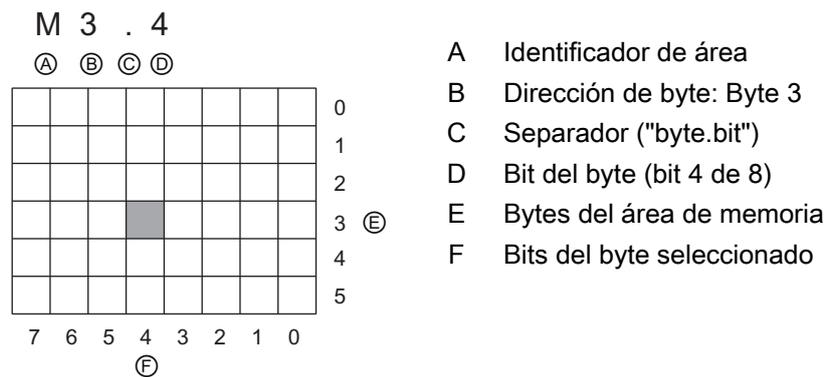
- Memoria global: La CPU ofrece distintas áreas de memoria, incluyendo entradas (I), salidas (Q) y marcas (M). Todos los bloques lógicos pueden acceder sin restricción alguna a esta memoria.
- Bloque de datos (DB): Es posible incluir DBs en el programa de usuario para almacenar los datos de los bloques lógicos. Los datos almacenados se conservan cuando finaliza la ejecución del bloque lógico asociado. Un DB "global" almacena datos que pueden ser utilizados por todos los bloques lógicos, mientras que un DB instancia almacena datos para un bloque de función (FB) específico y está estructurado según los parámetros del FB.
- Memoria temporal: Cada vez que se llama un bloque lógico, el sistema operativo de la CPU asigna la memoria temporal o local (L) que debe utilizarse durante la ejecución del bloque. Cuando finaliza la ejecución del bloque lógico, la CPU reasigna la memoria local para la ejecución de otros bloques lógicos.

Toda posición de memoria diferente tiene una dirección unívoca. El programa de usuario utiliza estas direcciones para acceder a la información de la posición de memoria.

Área de memoria	Descripción	Forzado permanente	Remanente
I Memoria imagen de proceso de las entradas I:P (entrada física)	Se copia de las entradas físicas al inicio del ciclo	No	No
	Lectura inmediata de las entradas físicas de la CPU, SB y SM	Sí	No

Área de memoria	Descripción	Forzado permanente	Remanente
Q Memoria imagen de proceso de las salidas	Se copia en las salidas físicas al inicio del ciclo	No	No
Q:P (salida física)	Escritura inmediata en las salidas físicas de la CPU, SB y SM	Sí	No
M Área de marcas	Control y memoria de datos	No	Sí
L Memoria temporal	Datos locales temporales de un bloque	No	No
DB Bloque de datos	Memoria de datos y de parámetros de FBs	No	Sí

Toda posición de memoria diferente tiene una dirección unívoca. El programa de usuario utiliza estas direcciones para acceder a la información de la posición de memoria. La figura muestra cómo acceder a un bit (lo que también se conoce como direccionamiento "byte.bit"). En este ejemplo, el área de memoria y la dirección del byte (1 = entrada y 3 = byte 3) van seguidas de un punto (".") que separa la dirección del bit (bit 4).



A los datos de la mayoría de las áreas de memoria (I, Q, M, DB y L) se puede acceder como bytes, palabras o palabras dobles utilizando el formato "dirección de byte". Para acceder a un byte, una palabra o una palabra doble de datos en la memoria, la dirección debe especificarse de forma similar a la dirección de un bit. Esto incluye un identificador de área, el tamaño de los datos y la dirección de byte inicial del valor de byte, palabra o palabra doble. Los designadores de tamaño son B (byte), W (palabra) y D (palabra doble), p. ej. IB0, MW20 ó QD8. Las direcciones tales como I0.3 y Q1.7 acceden a la memoria imagen de proceso. Para acceder a la entrada o salida física es preciso añadir ":P" a la dirección (p. ej. I0.3:P, Q1.7:P o "Stop:P").

Acceder a los datos en las áreas de memoria de la CPU

STEP 7 Basic facilita la programación simbólica. Normalmente, las variables se crean en variables PLC, en un bloque de datos o en la interfaz arriba de un OB, FC o FB. Estas variables incluyen un nombre, tipo de datos, offset y comentario. Además, es posible definir un valor inicial en un bloque de datos. Estas variables pueden utilizarse durante la programación, introduciendo el nombre de la variable en el parámetro de la instrucción. Opcionalmente se puede introducir el operando absoluto (memoria, área, tamaño y offset) en el parámetro de la instrucción. Los ejemplos de los apartados siguientes muestran cómo introducir operandos absolutos. El editor de programación antepone automáticamente el carácter % al operando absoluto. Es posible cambiar entre las siguientes vistas del editor de programación: simbólica, simbólica y absoluta y absoluta.

I (memoria imagen de proceso de las entradas): La CPU consulta las entradas de periferia (físicas) inmediatamente antes de ejecutar el OB de ciclo en cada ciclo y escribe estos valores en la memoria imagen de proceso de las entradas. A la memoria imagen de proceso de las entradas se puede acceder en formato de bit, byte, palabra o palabra doble. Aunque se permiten accesos de lectura y escritura, generalmente sólo se leen las entradas de la memoria imagen de proceso.

Bit	I[dirección de byte].[dirección de bit]	I0.1
Byte, palabra o palabra doble	I[tamaño][dirección de byte inicial]	IB4, IW5 o ID12

Añadiendo una ":P" a la dirección es posible leer inmediatamente las entradas digitales y analógicas de la CPU, SB o SM. La diferencia entre un acceso que utiliza I_:P en vez de I es que los datos provienen directamente de las entradas direccionadas, en vez de la memoria imagen de proceso de las entradas. El acceso I_:P también se denomina "lectura inmediata", puesto que los datos se leen inmediatamente del origen y no de una copia creada la última vez que se actualizó la memoria imagen de proceso de las entradas.

Puesto que las entradas físicas reciben sus valores directamente de los aparatos de campo conectados a ellas, está prohibido escribir en estas entradas. Por tanto, los accesos I_:P son de sólo lectura, a diferencia de los accesos I que pueden ser de lectura o escritura.

Los accesos I_:P también están restringidos por el tamaño de las entradas que soporta una única CPU, SB o SM, redondeado al byte más próximo. Por ejemplo, si las entradas de una SB de 2 DI / 2 DQ se configuran de manera que comiencen en I4.0, las entradas se podrán direccionar como I4.0:P e I4.1:P, o bien IB4:P. Aunque no se rechazan los accesos a I4.2:P hasta I4.7:P, no tienen sentido ya que estas entradas no se utilizan. Los accesos a IW4:P y ID4:P están prohibidos, puesto que exceden el offset de bytes asociado a la SB.

Los accesos mediante I_:P no afectan el valor correspondiente almacenado en la memoria imagen de proceso de las entradas.

Bit	I[dirección de byte].[dirección de bit]:P	I0.1:P
Byte, palabra o palabra doble	I[tamaño][dirección de byte inicial]:P	IB4:P, IW5:P o ID12:P

Q (memoria imagen de proceso de las salidas): La CPU copia los valores almacenados en la imagen de proceso de las salidas en las salidas físicas. A la memoria imagen de proceso de las salidas se puede acceder en formato de bit, byte, palabra o palabra doble. Se permiten accesos de lectura y escritura a la memoria imagen de proceso de las salidas.

Bit	Q[dirección de byte].[dirección de bit]	Q1.1
Byte, palabra o palabra doble	Q[tamaño][dirección de byte inicial]	QB5, QW10, QD40

Añadiendo una ":P" a la dirección es posible escribir inmediatamente en las salidas digitales y analógicas físicas de la CPU, SB o SM. La diferencia entre un acceso que utiliza Q_:P en vez de Q es que los datos se escriben directamente en las salidas direccionadas y también en la memoria imagen de proceso de las salidas. El acceso Q_:P se denomina a veces "escritura inmediata", puesto que los datos se escriben inmediatamente en la salida de destino. Por tanto, ésta no tiene que esperar hasta la siguiente actualización desde la memoria imagen de proceso de las salidas.

Puesto que las salidas físicas controlan directamente los aparatos de campo conectados a ellas, está prohibido leer de estas salidas. Por tanto, los accesos Q_:P son de sólo escritura, a diferencia de los accesos Q que pueden ser de lectura o escritura.

Los accesos Q_:P también están restringidos por el tamaño de las salidas que soporta una única CPU, SB o SM, redondeado al byte más próximo. Por ejemplo, si las salidas de una SB de 2 DI / 2 DQ se configuran de manera que comiencen en Q4.0, las salidas se podrán direccionar como Q4.0:P y Q4.1:P, o bien QB4:P. Aunque no se rechazan los accesos a QB4.2:P hasta QB4.7:P, no tienen sentido ya que estas salidas no se utilizan. Los accesos a QW4:P y QD4:P están prohibidos, puesto que exceden el offset de bytes asociado a la SB.

Los accesos mediante Q_:P afectan tanto la salida física como el valor correspondiente almacenado en la memoria imagen de proceso de las salidas.

Bit	Q[dirección de byte].[dirección de bit]:P	Q1.1:P
Byte, palabra o palabra doble	Q[tamaño][dirección de byte inicial]:P	QB5:P, QW10:P o QD40:P

M (área de marcas): El área de marcas (memoria M) puede utilizarse para relés de control y datos para almacenar el estado intermedio de una operación u otra información de control. Al área de marcas se puede acceder en formato de bit, byte, palabra o palabra doble. Se permiten accesos de lectura y escritura al área de marcas.

Bit	M[dirección de byte].[dirección de bit]	M26.7
Byte, palabra o palabra doble	M[tamaño][dirección de byte inicial]	MB20, MW30, MD50

Temp (memoria temporal): La CPU asigna la memoria temporal según sea necesario. La CPU asigna la memoria temporal al bloque lógico cuando éste se inicia (en caso de un OB) o se llama (en caso de una FC o un FB). La asignación de la memoria temporal a un bloque lógico puede reutilizar las mismas posiciones de memoria temporal usadas anteriormente por un OB, FC o FB diferente. La CPU no inicializa la memoria temporal durante la asignación. por lo que esta memoria puede contener un valor cualquiera.

La memoria temporal es similar al área de marcas, con una excepción importante: el área de marcas tiene un alcance "global", en tanto que la memoria temporal tiene un alcance "local".

- Área de marcas: Cualquier OB, FC o FB puede acceder a los datos del área de marcas. Esto significa que los datos están disponibles globalmente para todos los elementos del programa de usuario.
- Memoria temporal: El acceso a los datos de la memoria temporal está restringido al OB, FC o FB que ha creado o declarado la posición de memoria temporal. Las posiciones de memoria temporal son siempre locales y no son compartidas por diferentes bloques lógicos, incluso si un bloque lógico llama otro bloque lógico. Ejemplo: Cuando un OB llama una FC, ésta no puede acceder a la memoria temporal del OB que ha efectuado la llamada.

La CPU pone a disposición memoria temporal (local) para cada una de las tres clases de prioridad de OBs:

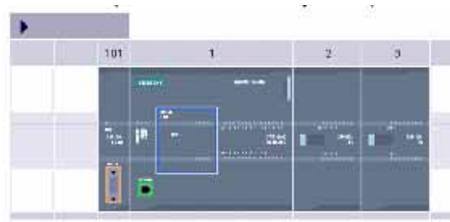
- 16 KB para arranque y ciclo, incluyendo los FBs y FCs asociados
- 4 KB para eventos de alarma estándar, incluyendo FBs y FCs
- 4 KB para eventos de alarma de error, incluyendo FBs y FCs

A la memoria temporal se puede acceder sólo con direccionamiento simbólico.

DB (bloque de datos): Los bloques de datos se utilizan para almacenar diferentes tipos de datos, incluyendo el estado intermedio de una operación u otros parámetros de control de FBs, así como estructuras de datos requeridas para numerosas instrucciones, p. ej. temporizadores y contadores. Es posible determinar que un bloque de datos sea de lectura/escritura o de sólo lectura. A los bloques de datos se puede acceder en formato de bit, byte, palabra o palabra doble. A los bloques de datos que se pueden leer y escribir se permiten accesos de lectura y escritura. A los bloques de datos de sólo lectura se permiten sólo los accesos de lectura.

Bit	DB[número de bloque de datos].DBX[dirección de byte].[dirección de bit]	DB1.DBX2.3
Byte, palabra o palabra doble	DB[número de bloque de datos].DB[tamaño][dirección de byte inicial]	DB1.DBB4, DB10.DBW2, DB20.DBD8

Direccionar las E/S de la CPU y los módulos de E/S



Vista general de dispositivos						
Módulo	Slot	Dirección...	Dirección...	Tipo	Referencia	
	103					
	102					
RS485_1	101			DM 1241 (RS485)	6ES7	
PLC_1	1			CPU 1214C D0DO	6ES7	
DI16/DO16	1.1	0...1	0...1	DI16/DO16		
A12	1.2	64...67		A12		
A01 x12bit	1.3		80...81	A01 Signal Board	6ES7	
HSC_1	1.16	1000....		Contador rápido (H)		
HSC_2	1.17			Contador rápido (H)		
HSC_3	1.18			Contador rápido (H)		
HSC_4	1.19			Contador rápido (H)		
HSC_5	1.20			Contador rápido (H)		
HSC_6	1.21			Contador rápido (H)		
Pulse_1	1.32			Generador de imp.		
Pulse_2	1.33			Generador de imp.		
Interfaz PR. X1				Interfaz PROFINET		
D18 x DC24V	2	8		SM 1221 D18 x DC	6ES7	

Al agregar una CPU y módulos de E/S en la ventana de configuración, se asignan automáticamente direcciones I y Q.

El direccionamiento predeterminado puede cambiarse seleccionando el campo de dirección en la ventana de configuración y tecleando números nuevos. Las entradas y salidas digitales se asignan en bytes completos (de 8 bits), sin importar si el módulo utiliza todas las E/S o no. Las entradas y salidas analógicas se asignan en grupos de 2 (4 bytes). En este ejemplo, la dirección de DI16 se podría cambiar a 2..3 en vez de 8..9. La herramienta cambia los rangos de direcciones cuyo tamaño sea incorrecto o que causen conflictos con otras direcciones.

La figura muestra un ejemplo de una CPU 1214C con dos SMs.

3.3 Tipos de datos

Los tipos de datos se utilizan para determinar el tamaño de un elemento de datos y cómo deben interpretarse los datos. Todo parámetro de instrucción soporta como mínimo un tipo de datos. Algunos parámetros soportan varios tipos de datos. Sitúe el cursor sobre el campo de parámetro de una instrucción para ver qué tipos de datos soporta el parámetro en cuestión.

Un parámetro formal es el identificador en una instrucción que indica la ubicación de los datos que deben utilizarse (ejemplo: la entrada IN1 de una instrucción ADD). Un parámetro actual es la posición de memoria o constante que contiene los datos que debe utilizar la instrucción (ejemplo: %MD400 "Número_de_widgets"). El tipo de datos del parámetro actual definido por el usuario debe concordar con uno de los tipos de datos que soporta el parámetro formal especificado por la instrucción.

Al definir un parámetro actual es preciso indicar una variable (símbolo) o una dirección absoluta. Las variables asocian un nombre simbólico (nombre de variable) con un tipo de datos, área de memoria, offset y comentario. Se pueden crear bien sea en el editor de variables PLC, o bien en la interfaz del bloque (OB, FC, FB o DB). Si se introduce una dirección absoluta que no tenga una variable asociada, es preciso utilizar un tamaño apropiado que coincida con el tipo de datos soportado. Al realizar la entrada se creará una variable predeterminada.

3.3 Tipos de datos

También es posible introducir un valor de constante para numerosos parámetros de entrada. La tabla siguiente muestra los tipos de datos simples soportados, incluyendo ejemplos de entrada de constantes. Todos los tipos de datos, excepto String, están disponibles en el editor de variables PLC y en la interfaz del bloque. String sólo está disponible en la interfaz del bloque. La tabla siguiente muestra los tipos de datos simples.

Tipo de datos	Tamaño (bits)	Rango	Ejemplos de entrada de constantes
Bool	1	0 a 1	TRUE, FALSE, 0, 1
Byte	8	16#00 a 16#FF	16#12, 16#AB
Word	16	16#0000 a 16#FFFF	16#ABCD, 16#0001
DWord	32	16#00000000 a 16#FFFFFFFF	16#02468ACE
Char	8	16#00 a 16#FF	'A', 't', '@'
Sint	8	128 a 127	123, -123
Int	16	32.768 a 32.767	123, -123
Dint	32	-2.147.483.648 a 2.147.483.647	123, -123
USInt	8	0 a 255	123
UInt	16	0 a 65.535	123
UDInt	32	0 a 4.294.967.295	123
Real	32	+/-1,18 x 10 ⁻³⁸ a +/-3,40 x 10 ³⁸	123,456, -3,4, -1,2E+12, 3,4E-3
LReal	64	+/-2,23 x 10 ⁻³⁰⁸ a +/-1,79 x 10 ³⁰⁸	12345.123456789 -1,2E+40
Time	32	T#-24d_20h_31m_23s_648ms a T#24d_20h_31m_23s_647ms Almacenado como: -2,147,483,648 ms a +2,147,483,647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String	Variable	0 a 254 caracteres en tamaño de byte	'ABC'

Aunque no están disponibles como tipos de datos, las instrucciones de conversión soportan el siguiente formato numérico BCD.

Formato	Tamaño (bits)	Rango numérico	Ejemplos de entrada de constantes
BCD16	16	-999 a 999	123, -123
BCD32	32	-9999999 a 9999999	1234567, -1234567

Formato de números reales

Los números reales (o en coma flotante) se representan como números de 32 bits de precisión simple (Real) o de 64 bits de precisión doble (LReal) según la norma ANSI/IEEE 7541985. Los números en coma flotante de precisión simple tienen una exactitud de hasta 6 dígitos significativos, en tanto que los de precisión doble tienen una exactitud de hasta 15 dígitos significativos. Al introducir una constante en coma flotante, pueden indicarse como máximo 6 (Real) o 15 (LReal) dígitos significativos para conservar la precisión.

Los cálculos que comprenden una serie de valores prolongada, incluyendo números muy grandes y muy pequeños, pueden producir resultados inexactos. Esto puede suceder si los números difieren en 10 a la potencia de x, siendo x > 6 (Real) ó 15 (LReal). Ejemplo (Real): 100 000 000 + 1 = 100 000 000.

Formato del tipo de datos STRING

La CPU soporta el tipo de datos STRING para almacenar una secuencia de caracteres de un byte. El tipo de datos STRING contiene el número de caracteres total (número de caracteres de la cadena) y el número de caracteres actual. El tipo de datos STRING ofrece como máximo 256 bytes para almacenar el número máximo de caracteres total (1 byte), el número de caracteres actual (1 byte) y como máximo 254 caracteres. Cada carácter se almacena en 1 byte.

Es posible utilizar cadenas literales (constantes) para los parámetros de instrucción del tipo IN entre comillas sencillas. Por ejemplo, 'ABC' es una cadena de tres caracteres que podría utilizarse como entrada para el parámetro IN de la instrucción S_CONV. También es posible crear variables de cadena, seleccionando para ello el tipo de datos "String" en la interfaz de bloques OB, FC, FB y DB. En el editor de variables PLC no se pueden crear cadenas. El tamaño máximo de la cadena puede indicarse en bytes al declarar la cadena. Por ejemplo, "MiCadena[10]" especificaría un tamaño máximo de 10 bytes para MiCadena. Si se omiten los corchetes con un indicador de tamaño máximo, se supone que el tamaño es 254.

El ejemplo máximo define un STRING con un número máximo de caracteres de 10 y un número de caracteres actual de 3. Esto significa que el STRING contiene actualmente 3 caracteres de un byte, pero que podría ampliarse de manera que contenga como máximo 10 caracteres de un byte.

Número de caracteres total	Número de caracteres actual	Carácter 1	Carácter 2	Carácter 3	...	Carácter 10
10	3	'C' (16#43)	'A' (16#41)	'T' (16#54)	...	-
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	...	Byte 11

Matrices

Es posible crear una matriz que contenga varios elementos de un tipo de datos simple. Las matrices pueden crearse en las interfaces de bloques OB, FC, FB y DB. En el editor de variables PLC no se pueden crear matrices.

Para crear una matriz en la interfaz del bloque, seleccione el tipo de datos "Array [lo .. hi] of type" y modifique luego "lo", "hi" y "type" como se indica a continuación:

- lo - el índice inicial (más bajo) de la matriz
- hi - el índice final (más alto) de la matriz
- type - uno de los tipos de datos simples, p. ej. BOOL, SINT, UDINT

Se soportan los índices negativos. El nombre de la matriz se puede introducir en la columna "Nombre" de la interfaz del bloque. La tabla siguiente muestra ejemplos de matrices, tal y como podrían aparecer en la interfaz del bloque:

Nombre	Tipo de datos	Comentario
My_Bits	Array [1 .. 10] of BOOL	Esta matriz contiene 10 valores booleanos
My_Data	Array [-5 .. 5] of SINT	Esta matriz contiene 11 valores SINT, incluyendo el índice 0

Los elementos de matriz se referencian en el programa con la sintaxis siguiente:

- Array_name[*i*], donde *i* es el índice deseado.

Ejemplos que podrían aparecer en el editor de programación como entrada de parámetro:

- #My_Bits[3] - referencia el tercer bit de la matriz "My_Bits"
- #My_Data[-2] - referencia el cuatro SINT de la matriz "My_Data"

El editor de programación inserta automáticamente el símbolo #.

Tipo de datos DTL (Data and Time Long)

El tipo de datos DTL es una estructura de 12 bytes que almacena información de fecha y hora en una estructura predefinida. Un DTL se puede definir en la memoria temporal del bloque o en un DB.

Longitud (bytes)	Formato	Rango de valores	Ejemplo de un valor de entrada
12	Reloj y calendario (año-mes día:hora:minuto:segundo.na nosegundos)	Min.: DTL#1970-01-01- 00:00:00.0 Max.: DTL#2554-12-31- 23:59:59.999 999 999	DTL#2008-12-16- 20:30:20.250

Todo componente de DTL contiene un diferente tipo de datos y rango de valores. El tipo de datos de un valor especificado debe concordar con el tipo de datos de los componentes correspondientes.

Byte	Componente	Tipo de datos	Rango de valores
0	Año	UINT	1970 a 2554
1			
2	Mes	USINT	1 a 12
3	Día	USINT	1 a 31
4	Día de la semana	USINT	1(domingo) a 7(sábado) El día de la semana no se considera en la entrada del valor.
5	Hora	USINT	0 a 23
6	Minuto	USINT	0 a 59
7	Segundo	USINT	0 a 59
8	Nanosegundos	UDINT	0 a 999 999 999
9			
10			
11			

3.4 Utilizar una Memory Card

ATENCIÓN

La CPU sólo soporta la SIMATIC Memory Card (Página 359) preformateada. Si se utiliza un formateador de Windows para reformatear la SIMATIC Memory Card, la CPU no podrá utilizar la Memory Card reformateada.

Antes de copiar cualquier programa en la Memory Card formateada, borre todo programa almacenado previamente en ella.

Utilice la Memory Card como tarjeta de transferencia o de programa. Todo programa que se copie en la Memory Card contendrá todos los bloques lógicos y de datos, los objetos tecnológicos y la configuración del dispositivo. En cambio, **no** contendrá los valores de forzado permanente.

- Utilice una tarjeta de transferencia para copiar un programa en la memoria de carga interna de la CPU sin utilizar STEP 7 Basic. Una vez insertada la tarjeta de transferencia, la CPU borra primero el programa de usuario y los valores de forzado permanente de la memoria de carga interna y copia luego el programa desde la tarjeta de transferencia en la memoria de carga interna. Tras finalizar la transferencia es preciso extraer la tarjeta de transferencia.

Una tarjeta de transferencia vacía puede utilizarse para acceder a una CPU protegida por contraseña si se ha perdido u olvidado la contraseña (Página 60). Cuando se inserta una tarjeta de transferencia vacía, se borra el programa protegido por contraseña en la memoria de carga interna de la CPU. Luego es posible cargar un programa nuevo en la CPU.

- Utilice una tarjeta de programa como memoria de carga externa para la CPU. Cuando se inserta una tarjeta de programa, se borra toda la memoria de carga interna de la CPU (el programa de usuario y los valores de forzado permanente). La CPU ejecuta luego el programa en la memoria de carga externa (la tarjeta de programa). Si se realiza una carga en una CPU que tenga insertada una tarjeta de programa, se actualizará sólo la memoria de carga externa (la tarjeta de programa).

Puesto que la memoria de carga interna de la CPU se borró cuando se insertó la tarjeta de programa, ésta **debe** permanecer en la CPU. Si se extrae la tarjeta de programa, la CPU pasará a estado operativo STOP. (El LED de error parpadea para indicar que se ha extraído la tarjeta de programa.)

El programa contenido en una Memory Card incluye los bloques lógicos y de datos, los objetos tecnológicos y la configuración del dispositivo. La Memory Card **no** contiene valores de forzado permanente. Los valores de forzado permanente no forman parte del programa, pero se almacenan en la memoria de carga interna (en la CPU) o externa (en una tarjeta de programa). Si se inserta una tarjeta de programa en la CPU, STEP 7 Basic aplicará los valores de forzado permanente sólo a la memoria de carga externa en la tarjeta de programa.

3.4.1 Insertar una Memory Card en la CPU

⚠ ADVERTENCIA

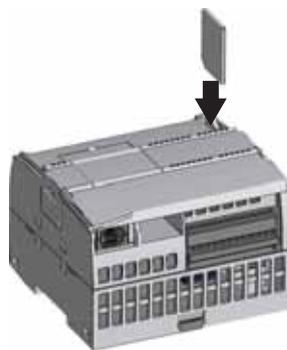
Si se inserta una Memory Card (configurada como tarjeta de programa o transferencia) en una CPU que esté funcionando, la CPU pasará inmediatamente a STOP. Los dispositivos de control pueden fallar y provocar condiciones no seguras, causando a su vez reacciones inesperadas de los equipos controlados. Las reacciones inesperadas podrían producir la muerte, lesiones corporales graves y/o daños materiales. Prevea siempre un circuito de parada de emergencia para la aplicación o proceso.

PRECAUCIÓN

Las descargas electrostáticas pueden deteriorar la Memory Card o la ranura para tarjetas en la CPU.

Al manejar la Memory Card deberá estar en contacto con una superficie conductiva puesta a tierra y/o llevar una muñequera antiestática. Guarde la Memory Card en una caja conductiva.

Para insertar una Memory Card, abra la tapa superior de la CPU e inserte la Memory Card en la ranura. Un conector de trinquete facilita la inserción y extracción. La Memory Card está diseñada de manera que pueda insertarse en un único sentido.



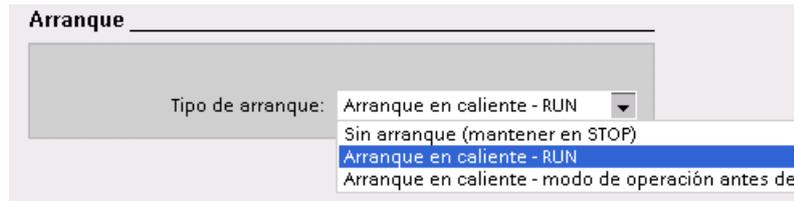
Asegúrese de que la Memory Card no está protegida contra escritura. Deslice el interruptor de protección fuera de la posición "Lock".

Nota

Si se inserta una Memory Card estando la CPU en estado operativo STOP, el búfer de diagnóstico mostrará un mensaje de que se ha iniciado la evaluación de la Memory Card. Ignore este mensaje. La evaluación de la Memory Card no comienza hasta que la CPU se conmuta a estado operativo RUN, se realiza un borrado total de la CPU (MRES) o se desconecta y vuelve a conectar la alimentación de la CPU.

3.4.2 Configurar los parámetros de arranque de la CPU antes de copiar el proyecto en la Memory Card

Cuando un programa se copia en una tarjeta de transferencia o de programa, incluye los parámetros de arranque de la CPU. Antes de copiar el programa en la Memory Card, asegúrese de que se ha configurado el estado operativo de la CPU posterior a la desconexión y conexión de la alimentación. Seleccione si la CPU debe arrancar en estado operativo STOP o RUN, o bien en el estado operativo anterior (a la desconexión y conexión de la alimentación).



3.4.3 Tarjeta de transferencia

PRECAUCIÓN

Las descargas electroestáticas pueden deteriorar la Memory Card o la ranura para tarjetas en la CPU.

Cuando maneje la Memory Card deberá estar en contacto con una superficie conductiva puesta a tierra y/o llevar una muñequera antiestática. Guarde la Memory Card en una caja conductiva.

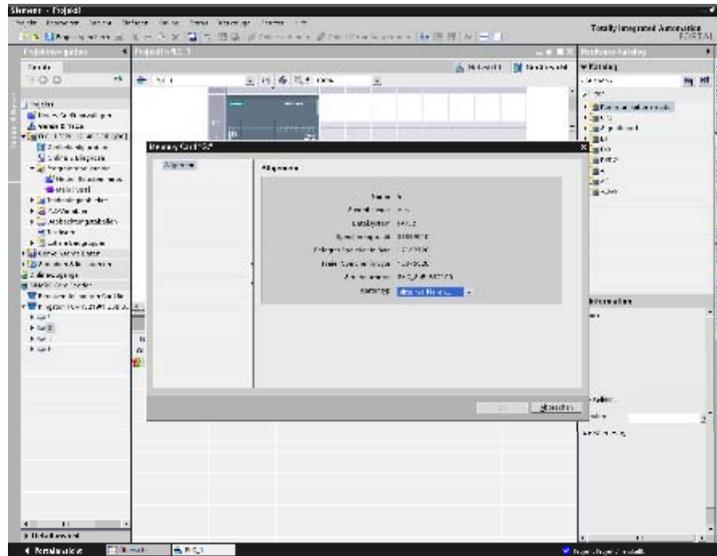
Crear una tarjeta de transferencia

Recuerde siempre que es necesario configurar los parámetros de arranque de la CPU (Página 71) antes de copiar un programa en la tarjeta de transferencia. Para crear una tarjeta de transferencia, proceda del siguiente modo:

1. Inserte una Memory Card vacía en el lector/grabador de tarjetas conectado a la programadora.
(Si la Memory Card no está vacía, borre la carpeta "SIMATIC.S7S" y el archivo "S7_JOB.S7S" de la Memory Card utilizando p. ej. el Explorador de Windows.)
2. En el árbol del proyecto (vista del proyecto), expanda la carpeta "SIMATIC Card Reader" y seleccione el lector de tarjetas deseado.
3. Para abrir el diálogo "Memory Card", haga clic con el botón derecho del ratón en la Memory Card en el lector de tarjetas y elija el comando "Propiedades" del menú contextual.

4. En el diálogo "Memory Card", seleccione "Transferencia" en la lista desplegable.

STEP 7 Basic crea entonces la tarjeta de transferencia vacía. Si está creando una tarjeta de transferencia vacía p. ej. para realizar una recuperación tras olvidar la contraseña de la CPU (Página 60), extraiga la tarjeta de transferencia del lector de tarjetas.



5. Agregue el programa seleccionando la CPU (p. ej. PLC_1 [CPU 1214 DC/DC/DC]) en el árbol del proyecto y arrastrándola hasta la Memory Card. (Como alternativa, copie la CPU e insértela en la Memory Card.) Cuando la CPU se copia en la Memory Card se abre el diálogo "Cargar vista preliminar".
6. En el diálogo "Cargar vista preliminar", haga clic en el botón "Cargar" para copiar la CPU en la Memory Card.
7. Cuando aparezca un mensaje indicando que la CPU (el programa) se ha cargado sin errores, haga clic en el botón "Finalizar".

Utilizar una tarjeta de transferencia

Para transferir el programa a una CPU, proceda del siguiente modo:

1. Inserte la tarjeta de transferencia en la CPU (Página 70). Si la CPU está en RUN, pasará a estado operativo STOP. (El LED de mantenimiento parpadea para indicar que debe evaluarse la Memory Card.)
2. Utilice una de las opciones siguientes para evaluar la Memory Card:
 - Desconecte y vuelva a conectar la alimentación de la CPU.
 - Cambie el estado operativo de STOP a RUN.
 - Realice un borrado total (MRES).
3. Tras rearmar y evaluar la Memory Card, la CPU copiará el programa en la memoria de carga interna de la CPU. Una vez finalizada la operación de copia, el LED de mantenimiento de la CPU parpadeará para indicar que la tarjeta de transferencia se puede extraer.
4. Extraiga la tarjeta de "Transferencia" de la CPU.
5. Utilice una de las opciones siguientes para evaluar el programa nuevo que se ha transferido a la memoria de carga interna:
 - Desconecte y vuelva a conectar la alimentación de la CPU.
 - Cambie el estado operativo de STOP a RUN.
 - Realice un borrado total (MRES).

La CPU pasa entonces al modo de arranque (RUN o STOP) configurado para el proyecto.

Nota

Extraiga la tarjeta de transferencia antes de cambiar la CPU a estado operativo RUN.

3.4.4 Tarjeta de programa

PRECAUCIÓN

Las descargas electrostáticas pueden deteriorar la Memory Card o la ranura para tarjetas en la CPU.

Al manejar la Memory Card deberá estar en contacto con una superficie conductiva puesta a tierra y/o llevar una muñequera antiestática. Guarde la Memory Card en una caja conductiva.



Asegúrese de que la Memory Card no está protegida contra escritura. Deslice el interruptor de protección fuera de la posición "Lock".

Antes de copiar elementos de programa en la tarjeta de programa, borre todo programa almacenado previamente en ella.

Crear una tarjeta de programa

Si se utiliza como tarjeta de programa, la Memory Card es la memoria de carga externa de la CPU. Si se extrae la tarjeta de programa, la memoria de carga interna de la CPU estará vacía.

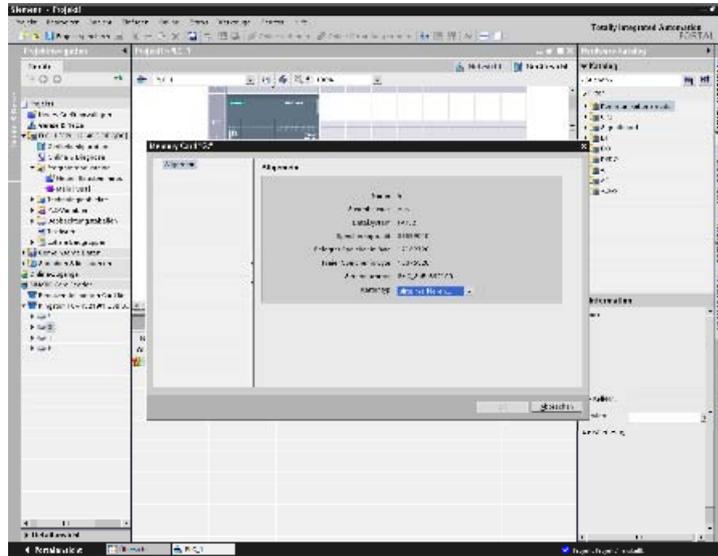
Nota

Si una Memory Card vacía se inserta en la CPU y se evalúa desconectando y conectando la alimentación de la CPU, cambiando el estado operativo de STOP a RUN o realizando un borrado total (MRES), el programa y los valores de forzado permanente contenidos en la memoria de carga interna de la CPU se copiarán en la Memory Card. (La Memory Card se convierte entonces en una tarjeta de programa.) Una vez finalizada la operación de copia, se borrará el programa en la memoria de carga interna de la CPU. La CPU pasa entonces al modo de arranque (RUN o STOP) configurado.

Recuerde siempre que es necesario configurar los parámetros de arranque de la CPU (Página 71) antes de copiar el proyecto en la tarjeta de programa. Para crear una tarjeta de programa con STEP 7 Basic, proceda del siguiente modo:

1. Inserte una Memory Card vacía en el lector/grabador de tarjetas conectado a la programadora.
(Si la Memory Card no está vacía, borre la carpeta "SIMATIC.S7S" y el archivo "S7_JOB.S7S" de la Memory Card utilizando p. ej. el Explorador de Windows.)
2. En el árbol del proyecto (vista del proyecto), expanda la carpeta "SIMATIC Card Reader" y seleccione el lector de tarjetas deseado.
3. Para abrir el diálogo "Memory Card", haga clic con el botón derecho del ratón en la Memory Card en el lector de tarjetas y elija el comando "Propiedades" del menú contextual.

4. En el diálogo "Memory Card", seleccione "Programa" en la lista desplegable.



5. Agregue el programa seleccionando la CPU (p. ej. PLC_1 [CPU 1214 DC/DC/DC]) en el árbol del proyecto y arrastrándola hasta la Memory Card. (Como alternativa, copie la CPU e insértela en la Memory Card.) Cuando la CPU se copia en la Memory Card se abre el diálogo "Cargar vista preliminar".
6. En el diálogo "Cargar vista preliminar", haga clic en el botón "Cargar" para copiar la CPU en la Memory Card.
7. Cuando aparezca un mensaje indicando que la CPU (el programa) se ha cargado sin errores, haga clic en el botón "Finalizar".

Utilizar una tarjeta de programa como memoria de carga para la CPU

PRECAUCIÓN

La CPU pasa a modo STOP si se inserta una Memory Card vacía. Al desconectar y volver a conectar la alimentación de la CPU, cambiar el estado operativo de la CPU de STOP a RUN o realizar un borrado total (MRES), la CPU copia su memoria de carga interna en la Memory Card (lo que configura la Memory Card como tarjeta de programa) y borra el programa de la memoria de carga interna. Si se extrae la tarjeta de programa, la memoria de carga interna de la CPU no contendrá ningún programa.

Para utilizar una tarjeta de programa en la CPU, proceda del siguiente modo:

1. Inserte la tarjeta de programa en la CPU. Si la CPU está en RUN, pasará a estado operativo STOP. El LED de mantenimiento parpadea para indicar que debe evaluarse la tarjeta de programa.
2. Utilice una de las opciones siguientes para evaluar la tarjeta de programa:
 - Desconecte y vuelva a conectar la alimentación de la CPU.
 - Cambie el estado operativo de STOP a RUN.
 - Realice un borrado total (MRES).
3. La CPU reanuda automáticamente. Tras reanudar y evaluar la tarjeta de programa, la CPU borrará su memoria de carga interna.

La CPU pasa entonces al modo de arranque (RUN o STOP) configurado para ella.

La tarjeta de programa debe permanecer en la CPU. Si se extrae la tarjeta de programa, la memoria de carga interna de la CPU no contendrá ningún programa.

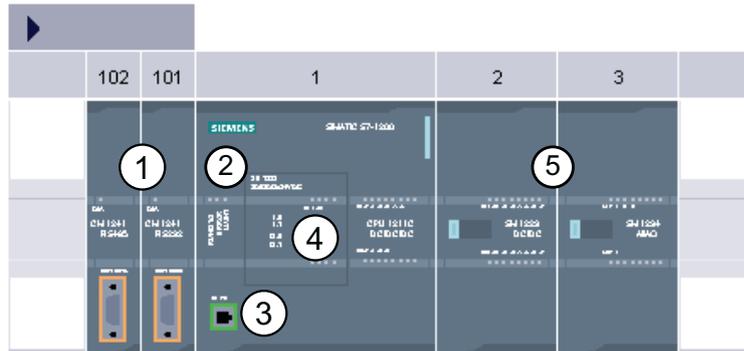
ADVERTENCIA

Si se extrae la tarjeta de programa, la CPU perderá su memoria de carga externa y generará un error. La CPU pasa a estado operativo STOP y el LED de error parpadea.

Los dispositivos de control pueden fallar y provocar condiciones no seguras, causando a su vez reacciones inesperadas de los equipos controlados. Las reacciones inesperadas podrían producir la muerte, lesiones corporales graves y/o daños materiales.

Configuración de dispositivos

Para crear la configuración de dispositivos del PLC es preciso agregar una CPU y módulos adicionales al proyecto.



- ① Módulo de comunicación (CM): máx. 3, insertados en los slots 101, 102 y 103
- ② CPU: Slot 1
- ③ Puerto Ethernet de la CPU
- ④ Signal Board (SB): máx. 1, insertada en la CPU
- ⑤ Módulo de señales (SM) para E/S digitales o analógicas: máx. 8, insertados en los slots 2 a 9 (la CPU 1214C permite 8, la CPU 1212C permite 2, la CPU 1211C no permite ninguno)

Para crear la configuración de dispositivos, agregue un dispositivo al proyecto.

- En la vista del portal, seleccione "Dispositivos y redes" y haga clic en "Agregar dispositivo".
- En la vista del proyecto, bajo el nombre del proyecto, haga doble clic en "Agregar nuevo dispositivo".



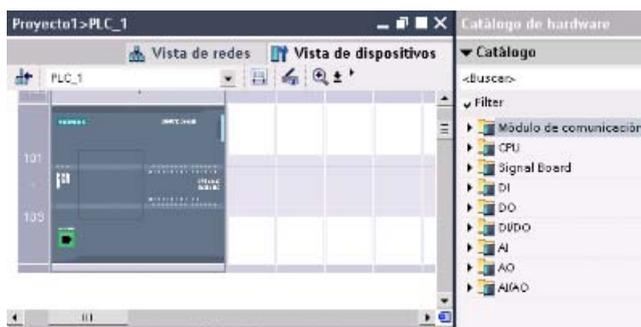
4.2 Insertar una CPU

La configuración de dispositivos se crea insertando una CPU en el proyecto. Al seleccionar la CPU en el diálogo "Agregar nuevo dispositivo" se crean el rack y la CPU.

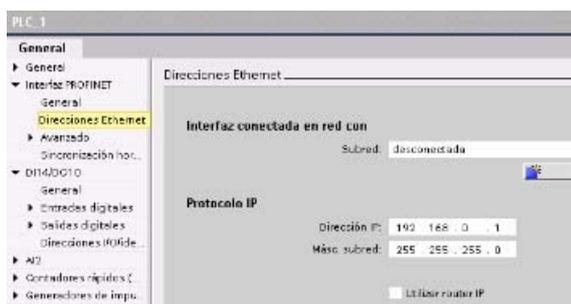
Diálogo "Agregar nuevo dispositivo"



Vista de dispositivos de la configuración de hardware



Al seleccionar la CPU en la vista de dispositivos se visualizan las propiedades de la CPU en la ventana de inspección.

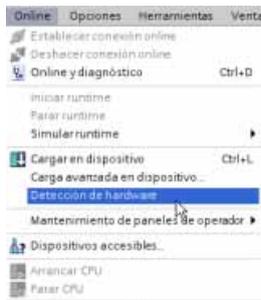


Nota

La CPU no tiene una dirección IP preconfigurada. La dirección IP de la CPU se debe asignar manualmente durante la configuración de dispositivos. Si la CPU está conectada a un router de la red, también es preciso introducir la dirección IP del router.

4.3 Detectar la configuración de una CPU sin especificar

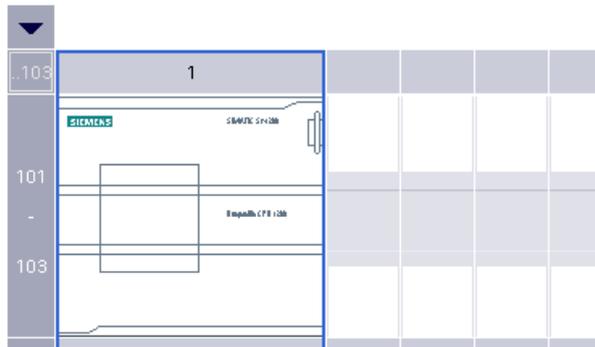
Cargar una configuración hardware existente es muy fácil



Si existe una conexión con una CPU, es posible cargar su configuración en el módulo (incluidos los módulos). Tan sólo hay que crear un proyecto nuevo y seleccionar la "CPU sin especificar" en lugar de una específica. (También es posible omitir la configuración de dispositivo por completo seleccionando "Crear un programa PLC" en "Primeros pasos". Entonces STEP 7 Basic crea automáticamente una CPU sin especificar.)

En el editor de programación, seleccione el comando "Detección de hardware" del menú "Online".

En el editor de configuración de dispositivos, seleccione la opción de detección del dispositivo conectado.



El dispositivo no está especificado.
→ Utilice el [catálogo de hardware](#) para especificar la CPU
→ o [determine](#) la configuración del dispositivo conectado.

Tras seleccionar la CPU en el cuadro de diálogo online, STEP 7 Basic carga la configuración hardware de la CPU, incluyendo todos los módulos (SM, SB o CM). Entonces pueden configurarse los parámetros de la CPU y de los módulos.



4.4 Configurar el funcionamiento de la CPU

Para configurar los parámetros operativos de la CPU, selecciónela en la vista de dispositivos (recuadro azul alrededor de la CPU) y utilice la ficha "Propiedades" de la ventana de inspección.



Edite las propiedades para configurar los siguientes parámetros:

- Interfaz PROFINET: Permite configurar la dirección IP de la CPU y la sincronización horaria
- DI, DO y AI: Permite configurar la reacción de las E/S locales (integradas) digitales y analógicas
- Contadores rápidos y generadores de impulsos: Permite habilitar y configurar los contadores rápidos (HSC) y generadores de impulsos utilizados para las operaciones de tren de impulsos (PTO) y modulación del ancho de pulso (PWM)

Si las salidas de la CPU o Signal Board se configuran como generadores de impulsos (para su utilización con la PWM o instrucciones de Motion Control básicas), las direcciones de las salidas correspondientes (Q0.0, Q0.1, Q4.0 y Q4.1) se eliminarán de la memoria Q y no podrán utilizarse para ningún otro fin en el programa de usuario. Si el programa de usuario escribe un valor en una salida utilizada como generador de impulsos, la CPU no escribirá ese valor en la salida física.

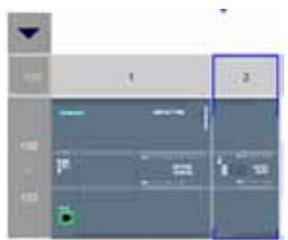
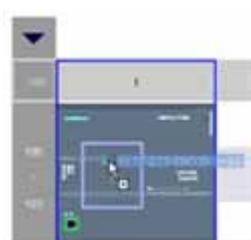
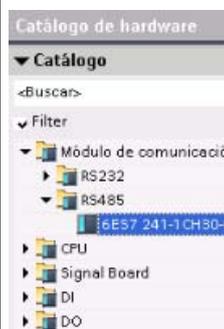
- Arranque: Permite configurar la reacción de la CPU a una transición de OFF a ON, p. ej. el arranque en estado operativo STOP o la transición a RUN tras un arranque en caliente
- Hora: Permite ajustar la hora, la zona horaria y el horario de verano/invierno
- Protección: Permite ajustar la protección de lectura/escritura y la contraseña para acceder a la CPU
- Marcas de sistema y de ciclo: Permite habilitar un byte para "marcas de sistema" (para los bits "Primer ciclo", "Siempre 1 (high)" y "Siempre 0 (low)") y un byte para "marcas de ciclo" (cada bit se activa y desactiva con una frecuencia predefinida).
- Tiempo de ciclo: Permite definir un tiempo de ciclo máximo o un tiempo de ciclo mínimo fijo
- Carga de comunicación: Permite asignar el porcentaje del tiempo de la CPU que debe dedicarse a las tareas de comunicación

4.5 Agregar módulos a la configuración

El catálogo de hardware se utiliza para agregar módulos a la CPU. Hay tres tipos de módulos, a saber:

- Los módulos de señales (SM) proveen E/S digitales o analógicas adicionales. Estos módulos se conectan a la derecha de la CPU.
- Las Signal Boards (SB) proveen unas pocas E/S adicionales a la CPU. La SB se inserta en el frente de la CPU.
- Los módulos de comunicación (CM) proveen un puerto de comunicación adicional (RS232 o RS485) a la CPU. Estos módulos se conectan a la izquierda de la CPU.

Para insertar un módulo en la configuración de hardware, selecciónelo en el catálogo de hardware y haga doble clic en él, o bien arrástrelo hasta el slot resaltado.

Módulo	Seleccionar el módulo	Insertar el módulo	Resultado
SM			
SB			
CM			

4.6 Configurar los parámetros de los módulos

Para configurar los parámetros operativos de un módulo, selecciónelo en la vista de dispositivos y utilice la ficha "Propiedades" de la ventana de inspección.

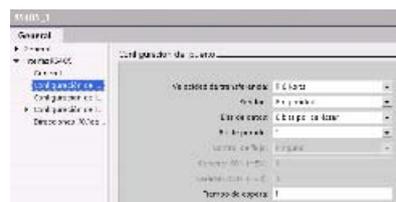
Configurar un módulo de señales (SM) o una Signal Board (SB)

- E/S digitales: Las entradas pueden configurarse para detectar flancos ascendentes o descendentes (asociando cada una de ellas a un evento y una alarma de proceso) y para la "captura de impulsos" (para que permanezcan activadas tras un impulso momentáneo) hasta la siguiente actualización de la memoria imagen de proceso de las entradas. Las salidas pueden congelarse o utilizar un valor sustitutivo.
- E/S analógicas: Es posible configurar los parámetros de las distintas entradas, tales como el tipo de medición (tensión o intensidad), el rango y el alisamiento, así como habitar el diagnóstico de rebase por defecto o por exceso. Las salidas ponen a disposición parámetros tales como el tipo de salida (tensión o intensidad) y para el diagnóstico, p. ej. cortocircuito (para salidas de tensión) o diagnóstico de límite superior/inferior.
- Direcciones E/S de diagnóstico: Permite configurar la dirección inicial de las entradas y salidas del módulo



Configurar un módulo de comunicación (CM)

- Configuración del puerto: Permite configurar los parámetros de comunicación, p. ej. velocidad de transferencia, paridad, bits de datos, bits de parada, control de flujo, caracteres XON y XOFF y tiempo de espera
- Configuración de la transferencia de mensajes: Permite habilitar y configurar las opciones relativas a la transferencia
- Configuración de la recepción de mensajes: Permite habilitar y configurar los parámetros de inicio y fin del mensaje



El programa de usuario puede cambiar estos parámetros de configuración.

4.7 Crear una conexión de red

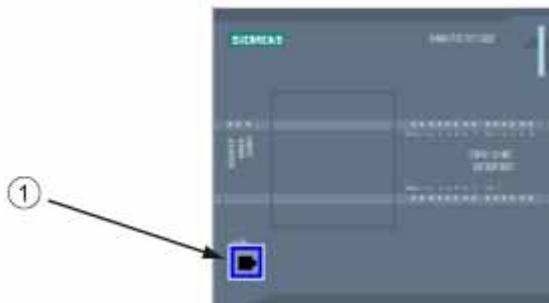
Utilice la "Vista de red" de la "Configuración de dispositivos" para crear las conexiones de red entre los dispositivos del proyecto. Tras crear la conexión de red, utilice la ficha "Propiedades" de la ventana de inspección para configurar los parámetros de la red.

Acción	Resultado
<p>Seleccione "Vista de red" para visualizar los dispositivos que deben conectarse.</p>	 <p>The screenshot shows the 'Vista de red' window with two PLC devices, PLC_1 and PLC_2, each labeled 'CPU 1214C'. No connections are visible between them.</p>
<p>Seleccione el puerto de uno de los dispositivos y arrastre la conexión hasta el puerto del otro dispositivo.</p>	 <p>The screenshot shows the same two PLC devices. A blue line is being dragged from a port on PLC_1 towards a port on PLC_2.</p>
<p>Suelte el botón del ratón para crear la conexión de red.</p>	 <p>The screenshot shows the two PLC devices with a completed network connection between them, labeled 'PNME_1'.</p>

4.8 Configurar una dirección IP en el proyecto

Configurar la interfaz PROFINET

Tras configurar el rack con la CPU (Página 80), es posible configurar los parámetros de la interfaz PROFINET. A este efecto, haga clic en la casilla PROFINET verde en la CPU para seleccionar el puerto PROFINET. La ficha "Propiedades" de la ventana de inspección muestra el puerto PROFINET.



① Puerto PROFINET

Configurar la dirección IP

Dirección Ethernet (MAC): Todo dispositivo de una red PROFINET recibe una dirección MAC (Media Access Control o control de acceso al medio) del fabricante para su identificación. Una dirección MAC consta de seis grupos de dos dígitos hexadecimales, separados por guiones (-) o dos puntos (:), en orden de transmisión (p. ej. 01-23-45-67-89-AB ó 01:23:45:67:89:AB).

Dirección IP: Todo dispositivo debe tener también una dirección IP (Internet Protocol o Protocolo Internet). Esta dirección permite al dispositivo transferir datos a través de una red enrutada y más compleja.

Toda dirección IP se divide en segmentos de ocho bits (octetos) y se expresa en formato decimal separado por puntos (p. ej. 211.154.184.16). La primera parte de la dirección IP se utiliza para la ID de red (¿en qué red se encuentra?) y, la segunda, para la ID del host (unívoca para cada dispositivo de la red). Una dirección IP 192.168.x.y es una designación estándar reconocida como parte de una red privada que no se enruta vía Internet.

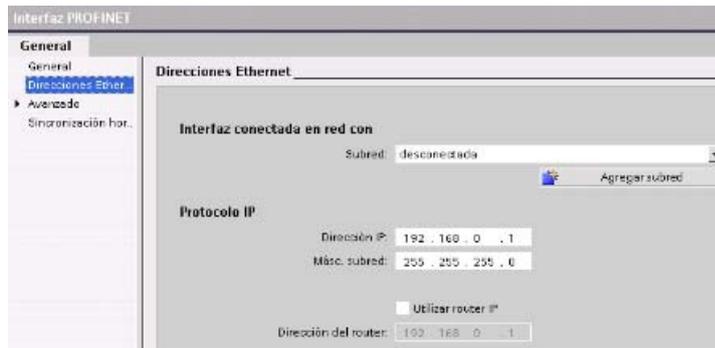
Máscara de subred: Una subred es una agrupación lógica de dispositivos de red conectados. Generalmente, los nodos de una subred están próximos físicamente en una red de área local (LAN). Una máscara (denominada "máscara de subred" o "máscara de red") define los límites de una subred IP.

Generalmente, una máscara de subred 255.255.255.0 se adecúa para una red local pequeña. Esto significa que los 3 primeros octetos de todas las direcciones IP de esta red deberían ser iguales. Los diferentes dispositivos de la red se identifican mediante el último octeto (campo de 8 bits). Por ejemplo, es posible asignar la máscara de subred 255.255.255.0 y direcciones IP comprendidas entre 192.168.2.0 y 192.168.2.255 a los dispositivos de una red local pequeña.

La única conexión entre las diferentes subredes se realiza a través de un router. Si se utilizan subredes, es preciso utilizar un router IP.

Router IP: Los routers interconectan las distintas LANs. Si se utiliza un router, un equipo de una LAN puede enviar mensajes a otras redes que, a su vez, pertenezcan a otras LANs. Si el destino de los datos se encuentra fuera de la LAN, el router reenvía los datos a otra red o grupo de redes desde donde pueden transferirse a su destino.

Los routers necesitan direcciones IP para poder transferir y recibir paquetes de datos.



Propiedades de direcciones IP: En la ventana de propiedades, seleccione la entrada de configuración "Dirección Ethernet". El TIA Portal visualiza el diálogo de configuración de direcciones Ethernet, en el que el proyecto de software se asocia a la dirección IP de la CPU que lo recibirá.

Nota

La CPU no tiene una dirección IP preconfigurada. La dirección IP de la CPU se debe asignar manualmente. Si la CPU está conectada a un router de la red, también es preciso introducir la dirección IP del router. Todas las direcciones IP se configuran al cargar el proyecto en el dispositivo.

Para más información, consulte el apartado "Asignar direcciones IP a los dispositivos de programación y red (Página 244)".

La tabla siguiente define los parámetros de la dirección IP:

Parámetro	Descripción	
Subred	Nombre de la subred a la que está conectada el dispositivo. Haga clic en el botón "Agregar nueva subred" para crear una subred nueva. El ajuste predeterminado es "no conectado". Hay dos tipos de conexión posibles: <ul style="list-style-type: none"> • El ajuste predeterminado "no conectado" ofrece una conexión local. • Una subred se requiere cuando la red comprende dos o más dispositivos. 	
Protocolo IP	Dirección IP	Dirección IP asignada a la CPU
	Máscara de subred	Máscara de subred asignada
	Utilizar router IP	Haga clic en esta casilla de verificación para indicar el uso de un router IP
	Dirección del router	Dirección IP asignada al router (si es aplicable)

Principios básicos de programación

5.1 Directrices para diseñar un sistema PLC

Al diseñar un sistema PLC es posible seleccionar entre diferentes métodos y criterios. Las directrices generales siguientes pueden aplicarse a un gran número de proyectos. Por supuesto que es necesario respetar las directrices corporativas y las prácticas usuales aprendidas y aplicadas.

Pasos recomendados	Tareas
Dividir el proceso o máquina	Divida el proceso o máquina en secciones independientes. Estas secciones determinan los límites entre los controladores e influyen en las especificaciones funcionales y la asignación de recursos.
Crear las especificaciones funcionales	Describa el funcionamiento de cada una de las secciones del proceso o máquina, tales como las entradas y salidas, la descripción funcional de la operación, los estados que deben adoptarse antes de que puedan entrar en acción los actuadores (como p. ej. electroválvulas, motores o accionamientos), la descripción de la interfaz de operador y cualquier interfaz con otras secciones del proceso o máquina.
Diseñar los circuitos de seguridad	<p>Determine los equipos que puedan requerir cableado fijo por motivos de seguridad. Recuerde que los dispositivos de control pueden fallar y provocar condiciones no seguras, causando a su vez un arranque inesperado o cambios de funcionamiento de la maquinaria. El funcionamiento inesperado o incorrecto de la maquinaria puede causar lesiones corporales o daños materiales considerables. Por tanto, prevea dispositivos de protección electromecánicos (que funcionen independientemente del PLC) para evitar las condiciones no seguras. Las siguientes tareas deben incluirse en el diseño de circuitos de seguridad:</p> <ul style="list-style-type: none"> • Definir el funcionamiento erróneo o inesperado de los actuadores que pudiera resultar peligroso. • Definir las condiciones que garanticen un funcionamiento seguro y determinar cómo detectar estas condiciones, independientemente del PLC. • Definir cómo el PLC y los módulos de ampliación deben influir en el proceso al conectarse y desconectarse la alimentación eléctrica, así como al detectarse errores. Utilice esta información sólo para proyectar el funcionamiento normal y el funcionamiento anormal esperado. Por motivos de seguridad, no conviene fiarse del supuesto más favorable. • Prever dispositivos de parada de emergencia manual o dispositivos de protección electromecánicos que impidan el funcionamiento peligroso, independientemente del PLC. • Proporcionar información de estado apropiada desde los circuitos independientes al PLC para que el programa y las interfaces de operador dispongan de la información necesaria. • Definir otros requisitos adicionales de seguridad para el funcionamiento seguro del proceso.
Determinar las estaciones de operador	<p>Según los requisitos de las especificaciones funcionales, cree los siguientes dibujos de las estaciones de operador:</p> <ul style="list-style-type: none"> • Dibujo general de la ubicación de todas las estaciones de operador con respecto al proceso o máquina • Dibujo de la disposición mecánica de los dispositivos de la estación de operador, p. ej. display, interruptores y lámparas • Esquemas eléctricos con las E/S asociadas del PLC y los módulos de señales

Pasos recomendados	Tareas
Crear los dibujos de configuración	Según los requisitos de las especificaciones funcionales, cree dibujos de configuración de los equipos de control: <ul style="list-style-type: none">• Dibujo general de la ubicación de todos los PLCs con respecto al proceso o máquina• Dibujo de la disposición mecánica de todos los PLCs y módulos de E/S, incluyendo los armarios y otros equipos.• Esquemas eléctricos de todos los PLCs y módulos de E/S, incluyendo los números de referencia de los dispositivos, las direcciones de comunicación y las direcciones de E/S.
Crear una lista de nombres simbólicos	Cree una lista de los nombres simbólicos correspondientes a las direcciones absolutas. Incluya no sólo las E/S físicas, sino también los demás elementos (p. ej. los nombres de variables) que se utilizarán en el programa.

5.2 Estructurar el programa de usuario

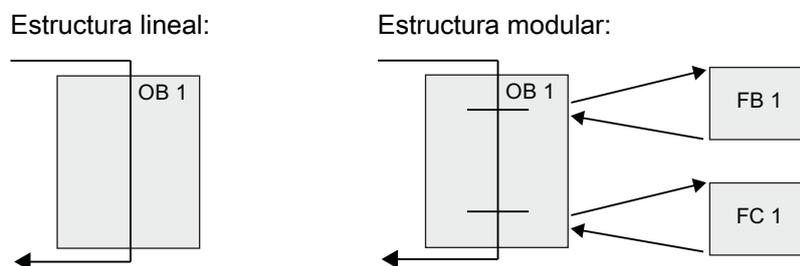
Al crear el programa de usuario para las tareas de automatización, las instrucciones del programa se insertan en bloques lógicos:

- Un bloque de organización (OB) reacciona a un evento específico en la CPU y puede interrumpir la ejecución del programa de usuario. El bloque predeterminado para la ejecución cíclica del programa de usuario (OB 1) ofrece la estructura básica y es el único bloque lógico que se requiere para el programa de usuario. Si se incluyen otros OBs en el programa, éstos interrumpen la ejecución del OB 1. Los demás OBs ejecutan funciones específicas, tales como tareas de arranque, procesamiento de alarmas y tratamiento de errores, o ejecución de un código de programa específico en determinados intervalos.
- Un bloque de función (FB) es una subrutina que se ejecuta cuando se llama desde otro bloque lógico (OB, FB o FC). El bloque que efectúa la llamada transfiere parámetros al FB e identifica un bloque de datos determinado (DB) que almacena los datos de la llamada o instancia específica de este FB. La modificación del DB instancia permite a un FB genérico controlar el funcionamiento de un conjunto de dispositivos. Por ejemplo, un solo FB puede controlar varias bombas o válvulas. Diferentes DBs de instancia contienen los parámetros operativos específicos de cada bomba o válvula.
- Una función (FC) es una subrutina que se ejecuta cuando se llama desde otro bloque lógico (OB, FB o FC). La FC no tiene un DB instancia asociado. El bloque que efectúa la llamada transfiere los parámetros a la FC. Los valores de salida de la FC deben escribirse en una dirección de la memoria o en un DB global.

Seleccionar el tipo de estructura del programa de usuario

Según los requisitos de la aplicación, es posible seleccionar una estructura lineal o modular para crear el programa de usuario:

- Un programa lineal ejecuta todas las instrucciones de la tarea de automatización de forma secuencial, es decir, una tras otra. Generalmente, el programa lineal deposita todas las instrucciones del programa en el OB encargado de la ejecución cíclica del programa (OB 1).
- Un programa modular llama bloques de función específicos que ejecutan determinadas tareas. Para crear una estructura modular, la tarea de automatización compleja se divide en tareas subordinadas más pequeñas, correspondientes a las funciones tecnológicas del proceso. Cada bloque lógico provee el segmento del programa para cada tarea subordinada. El programa se estructura llamando uno de los bloques lógicos desde otro bloque.



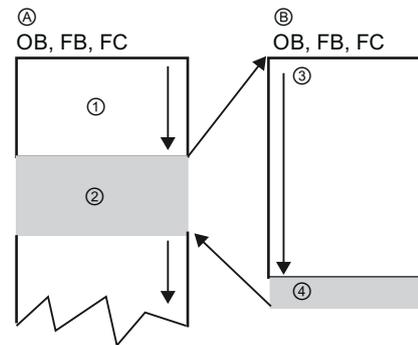
Creando bloques lógicos genéricos que pueden reutilizarse en el programa de usuario, es posible simplificar el diseño y la implementación del programa de usuario. La utilización de bloques lógicos genéricos ofrece numerosas ventajas:

- Es posible crear bloques lógicos reutilizables para tareas estándar, tales como el control de una bomba o motor. También es posible almacenar estos bloques lógicos genéricos en una librería, de manera que puedan ser utilizados por diferentes aplicaciones o soluciones.
- El programa de usuario puede dividirse en componentes modulares para las tareas funcionales, facilitando así su comprensión y gestión. Los componentes modulares ayudan no sólo a estandarizar el diseño del programa, sino que también pueden facilitar y agilizar la actualización o modificación del código del programa.
- La creación de componentes modular simplifica la depuración del programa. Dividiendo el programa completo en segmentos de programa modulares, es posible comprobar las funciones de cada bloque lógico a medida que se va desarrollando.
- La creación de componentes modulares para las distintas funciones tecnológicas permite simplificar y reducir el tiempo de puesta en marcha de la aplicación.

5.3 Utilizar bloques para estructurar el programa

Diseñando FBs y FCs que ejecuten tareas genéricas, se crean bloques lógicos modulares. El programa se estructura luego, de manera que otros bloques lógicos llamen estos bloques modulares reutilizables. El bloque que efectúa la llamada transfiere los parámetros específicos del dispositivo al bloque llamado.

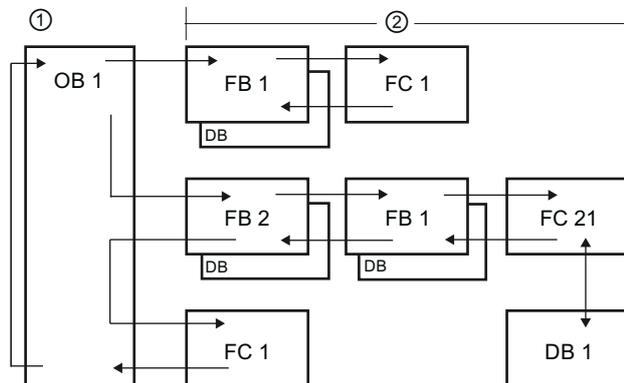
- A Bloque que llama
- B Bloque llamado (o que interrumpe)
- ① Ejecución del programa
- ② Operación que llama otro bloque
- ③ Ejecución del programa
- ④ Fin del bloque (regresa al bloque que llama)



Cuando un bloque lógico llama otro bloque lógico, la CPU ejecuta el código del programa en el bloque llamado. Una vez finalizada la ejecución del bloque llamado, la CPU reanuda la ejecución del bloque que ha efectuado la llamada.

El procesamiento continúa con la ejecución de la instrucción siguiente a la llamada de bloque. Las llamadas de bloque pueden anidarse para crear una estructura más modular.

- ① Inicio del ciclo
- ② Profundidad de anidamiento



Crear bloques lógicos reutilizables



Utilice el diálogo "Agregar nuevo bloque" en "Bloques de programa" en el árbol del proyecto para crear OBs, FBs, FCs y DBs globales.

Al crear el bloque lógico se selecciona el lenguaje de programación para el bloque. El lenguaje de un DB no se selecciona, puesto que éste sólo almacena datos.

5.3.1 Bloque de organización (OB)

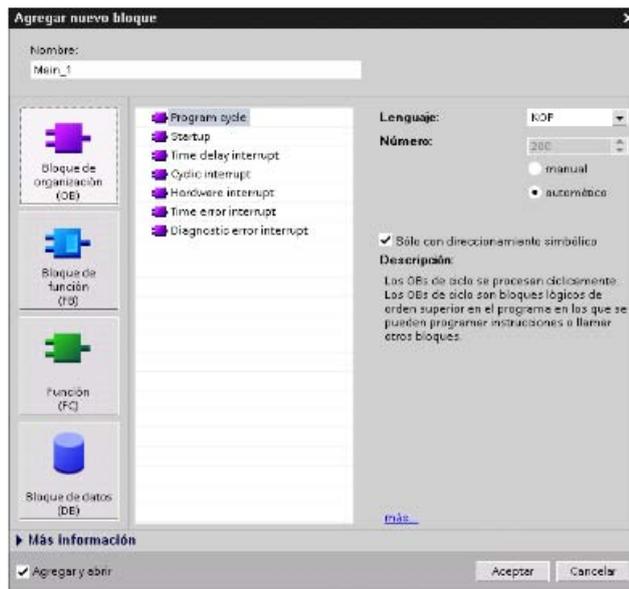
Los bloques de organización permiten estructurar el programa. Estos bloques sirven de interfaz entre el sistema operativo y el programa de usuario. Los OBs son controlados por eventos. Un evento, p. ej. una alarma de diagnóstico o un intervalo, hace que la CPU ejecute un OB. Algunos OBs tienen eventos de arranque y comportamiento en arranque predefinidos.

El OB de ciclo contiene el programa principal. Es posible incluir más de un OB de ciclo en el programa de usuario. En estado operativo RUN, los OBs de ciclo se ejecutan en el nivel de prioridad más bajo y pueden ser interrumpidos por todos los demás tipos de procesamiento del programa. El OB de arranque no interrumpe el OB de ciclo, puesto que la CPU ejecuta el OB de arranque antes de pasar al estado operativo RUN.

Tras finalizar el procesamiento de los OBs de ciclo, la CPU vuelve a ejecutarlos inmediatamente. Esta ejecución cíclica es el tipo de procesamiento "normal" que se utiliza para los controladores lógicos programables. En numerosas aplicaciones, el programa de usuario entero está contenido en un solo OB de ciclo.

Es posible crear otros OBs para ejecutar funciones específicas, tales como tareas de arranque, procesamiento de alarmas y tratamiento de errores, o ejecución de un código de programa específico en determinados intervalos. Estos OBs interrumpen la ejecución de los OBs de ciclo.

Utilice el diálogo "Agregar nuevo bloque" para crear OBs nuevos en el programa de usuario.



En función de su nivel de prioridad, un OB puede interrumpir a otro OB. Las alarmas se procesan siempre de forma controlada por eventos. Cuando ocurre un evento, la CPU interrumpe la ejecución del programa de usuario y llama el OB configurado para procesar ese evento. Una vez finalizada la ejecución del OB de alarma, la CPU reanuda la ejecución del programa de usuario en el punto de interrupción.

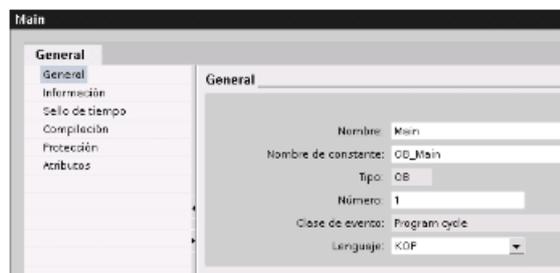
La CPU determina el orden de procesamiento de eventos de alarma según la prioridad asignada a cada OB. Todo evento tiene una prioridad de procesamiento propia. Varios eventos de alarma pueden combinarse en clases de prioridad. Para más información, consulte la sección relativa a la ejecución del programa de usuario en el capítulo "Principios básicos del PLC" (Página 41).

Crear un OB adicional en una clase de OB

Es posible crear varios OBs para el programa de usuario, incluso para las clases de OB correspondientes a los OBs de ciclo y de arranque. Utilice el diálogo "Agregar nuevo bloque" para crear un OB. Introduzca el nombre del OB y un número de OB mayor que 200.

Si se crean varios OBs de ciclo para el programa de usuario, la CPU ejecutará cada uno de ellos en una secuencia numérica, comenzando con el OB de ciclo principal (ajuste predeterminado: OB 1). Ejemplo: tras finalizar el primer OB de ciclo (OB1), la CPU ejecutará el segundo OB de ciclo (p. ej. OB 200).

Configurar el funcionamiento de un OB



Los parámetros operativos de un OB se pueden modificar. Por ejemplo, es posible configurar el parámetro de tiempo de un OB de alarma de retardo o de ciclo.

5.3.2 Función (FC)

Una función (FC) es un bloque lógico que, por lo general, realiza una operación específica en un conjunto de valores de entrada. La FC almacena los resultados de esta operación en posiciones de memoria.

Las FCs se utilizan para realizar las tareas siguientes:

- Para ejecutar operaciones estándar y reutilizables, p. ej. en cálculos matemáticos.
- Para ejecutar funciones tecnológicas, p. ej. controles individuales con operaciones lógicas binarias.

Una FC también se puede llamar varias veces en diferentes puntos de un programa. Esto facilita la programación de tareas que se repiten con frecuencia.

Una FC no tiene ningún bloque de datos instancia asociado (DB). La FC usa la pila de datos locales para los datos temporales utilizados para calcular la operación. Los datos temporales no se almacenan. Para almacenar los datos de forma permanente es preciso asignar el valor de salida a una posición de memoria global, p. ej. el área de marcas o un DB global.

5.3.3 Bloque de función (FB)

Un bloque de función (FB) es un bloque lógico que utiliza un bloque de datos instancia para sus parámetros y datos estáticos. Los FBs tienen una memoria variable ubicada en un bloque de datos (DB) o DB "instancia". El DB instancia ofrece un bloque de memoria asociado a esa instancia (o llamada) del FB y almacena datos una vez que haya finalizado el FB. Es posible asociar distintos DBs de instancia a diferentes llamadas del FB. Los DBs instancia permiten utilizar un FB genérico para controlar varios dispositivos. El programa se estructura de manera que un bloque lógico llame un FB y un DB instancia. La CPU ejecuta luego el código del programa en ese FB y almacena los parámetros del bloque y los datos locales estáticos en el DB instancia. Cuando finaliza la ejecución del FB, la CPU regresa al bloque lógico que ha llamado el FB. El DB instancia conserva los valores de esa instancia del FB. Estos valores están disponibles para las llamadas posteriores al bloque de función, bien sea en el mismo ciclo o en otros ciclos.

Bloques lógicos reutilizables con memoria asociada

Por lo general, los FBs se utilizan para controlar tareas o dispositivos cuya operación no finaliza dentro de un ciclo. Para almacenar los parámetros operativos de manera que sea posible acceder rápidamente a ellos de un ciclo a otro, todo FB del programa de usuario tiene uno o más DBs instancia. Cuando se llama un FB, se especifica también un DB instancia que contiene los parámetros del bloque y los datos locales estáticos de esa llamada o "instancia" del FB. El DB instancia conserva estos valores una vez finalizada la ejecución del FB.

Si el FB se diseña para realizar tareas de control genéricas, es posible reutilizarlo para varios dispositivos, seleccionando diferentes DB instancia para las distintas llamadas del FB.

Un FB almacena los parámetros de entrada (IN), salida (OUT) y entrada/salida (IN_OUT) en un DB instancia.

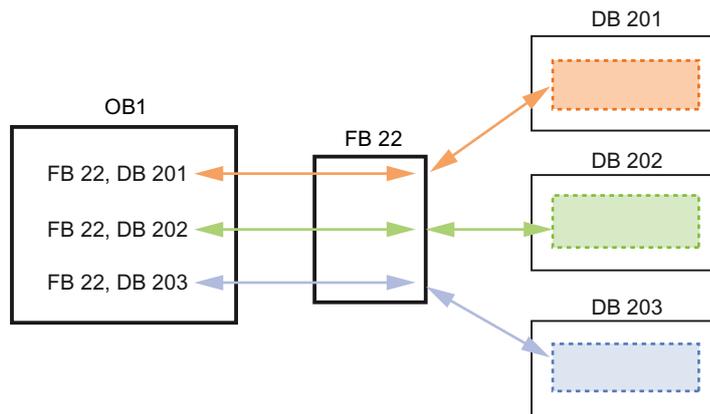
Asignar valores iniciales

Si no se asignan valores a los parámetros de entrada, salida o entrada/salida de un bloque de función (FB), se utilizan los valores almacenados en el bloque de datos (DB) instancia. En algunos casos es necesario asignar parámetros.

Los valores iniciales se asignan a los parámetros en la interfaz del FB. Estos valores se transfieren al DB instancia asociado. Si no se asignan parámetros, se utilizan los valores almacenados actualmente en el DB instancia.

Utilizar un solo FB con DBs

La figura siguiente muestra un OB que llama un FB tres veces, utilizando un bloque de datos diferente para cada llamada. Esta estructura permite que un FB genérico controle varios dispositivos similares (p. ej. motores), asignando un bloque de datos instancia diferente a cada llamada de los distintos dispositivos. Cada DB instancia almacena los datos (p. ej. velocidad, tiempo de aceleración y tiempo de operación total) de un dispositivo en particular. En este ejemplo, el FB 22 controla tres dispositivos diferentes. El DB 201 almacena los datos operativos del primer dispositivo, el DB 202, los del segundo y, el DB 203, los del tercero.



5.3.4 Bloque de datos (DB)

Los bloques de datos (DB) se crean en el programa de usuario para almacenar los datos de los bloques lógicos. Todos los bloques del programa de usuario pueden acceder a los datos en un DB global. En cambio, un DB instancia almacena los datos de un bloque de función (FB) específico. Un DB se puede definir de manera que sea de sólo lectura.

Los datos almacenados en un DB no se borran cuando finaliza la ejecución del bloque lógico asociado. Hay dos tipos de DBs, a saber:

- Un DB global almacena los datos de los bloques lógicos en el programa. Cualquier OB, FB o FC puede acceder a los datos en un DB global.
- Un DB instancia almacena los datos de un FB específico. La estructura de los datos en un DB instancia refleja los parámetros (Input, Output e InOut) y los datos estáticos del FB. (La memoria temporal del FB no se almacena en el DB instancia.)

Nota

Aunque el DB instancia refleja los datos de un FB específico, cualquier bloque lógico puede acceder a los datos en un DB instancia.

5.4 Principios básicos de la coherencia de datos

La CPU conserva la coherencia de datos de todos los tipos de datos simples (p. ej. Word o DWord) y de todas las estructuras definidas por el sistema (p. ej. IEC_TIMERS o DTL). La lectura o escritura de valores no se puede interrumpir. (Por ejemplo, la CPU protege el acceso a un valor de palabra doble (DWord) hasta que se hayan leído o escrito los cuatro bytes de la palabra doble.) Para garantizar que los OBs de ciclo y de alarma no puedan escribir simultáneamente en la misma posición de memoria, la CPU no ejecuta un OB de alarma hasta que no haya finalizado la operación de lectura o escritura en el OB de ciclo.

Si el programa de usuario comparte varios valores entre un OB de ciclo y un OB de alarma en la memoria, dicho programa debe garantizar asimismo que estos valores se modifiquen o lean de forma coherente. Las instrucciones DIS_AIRT y EN_AIRT se utilizan en el OB de ciclo para proteger cualquier acceso a los valores compartidos.

- Inserte una instrucción DIS_AIRT en el bloque lógico para garantizar que un OB de alarma no pueda ejecutarse durante la operación de lectura o escritura.
- Inserte las instrucciones que leen o escriben los valores que podrían ser modificados por un OB de alarma.
- Inserte una instrucción EN_AIRT al final de la secuencia para cancelar la instrucción DIS_AIRT y permitir la ejecución del OB de alarma.

Una petición de comunicación de un dispositivo HMI o de otra CPU puede interrumpir asimismo la ejecución del OB de ciclo. Las peticiones de comunicación también pueden causar problemas en relación con la coherencia de datos. La CPU garantiza que las instrucciones del programa de usuario lean y escriban coherentemente los tipos de datos simples. Puesto que las comunicaciones interrumpen el programa de usuario de forma periódica, no es posible garantizar que el dispositivo HMI actualice simultáneamente varios valores en la CPU. Por ejemplo, los valores visualizados en la pantalla de un HMI podrían provenir de diferentes ciclos de la CPU.

Las instrucciones PtP (punto a punto) y PROFINET (p. ej. TSEND_C y TRCV_C) transfieren búferes de datos que podrían interrumpirse. La coherencia de datos de los búferes debe asegurarse evitando operaciones de lectura y escritura en los búferes, tanto en el OB de ciclo como en un OB de alarma. Si es necesario modificar los valores de los búferes para estas instrucciones en un OB de alarma, utilice una instrucción DIS_AIRT para retardar las alarmas (un OB de alarma o una alarma de comunicación de un HMI u otra CPU) hasta que se ejecute una instrucción EN_AIRT.

Nota

La utilización de la instrucción DIS_AIRT retarda el procesamiento de los OBs de alarma hasta que se ejecuta la instrucción EN_AIRT, lo que afecta la latencia de alarmas (tiempo que transcurre desde un evento hasta que se ejecuta el OB de alarma) del programa de usuario.

5.5 Seleccionar el lenguaje de programación

Es posible elegir entre el lenguaje de programación KOP (Esquema de contactos) o FUP (Diagrama de funciones).

Lenguaje de programación KOP

KOP es un lenguaje de programación gráfico. Su representación es similar a los esquemas de circuitos.



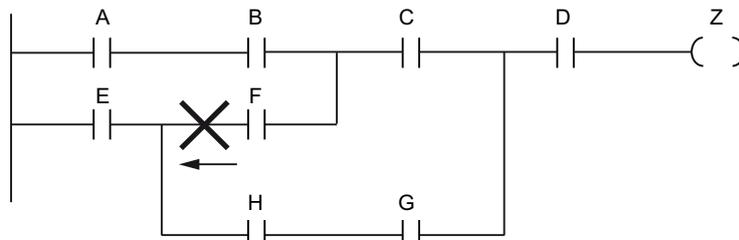
Los elementos de un esquema de circuitos, tales como los contactos normalmente cerrados y normalmente abiertos, así como las bobinas, se combinan para formar segmentos.

Para crear la lógica de operaciones complejas, es posible insertar ramas para los circuitos paralelos. Las ramas paralelas se abren hacia abajo o se conectan directamente a la barra de alimentación. Las ramas se terminan hacia arriba.

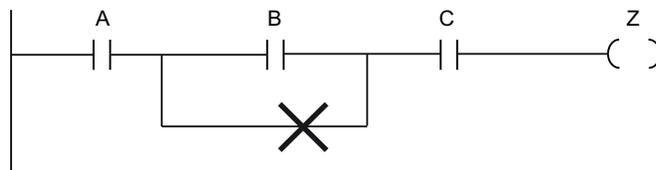
KOP ofrece instrucciones con cuadros para numerosas funciones, p. ej. matemáticas, temporizadores, contadores y transferencia.

Tenga en cuenta las reglas siguientes al crear segmentos KOP:

- Todo segmento KOP debe terminar con una bobina o cuadro. No termine un segmento con una instrucción de comparación ni de detección de flancos (ascendentes o descendentes).
- No se permite programar ramas que puedan ocasionar un flujo invertido de la corriente.



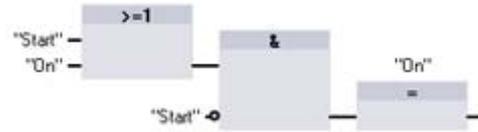
- No se permite programar ramas que causen cortocircuitos.



Lenguaje de programación Diagrama de funciones (FUP)

Al igual que KOP, FUP es un lenguaje de programación gráfico. La representación de la lógica se basa en los símbolos lógicos gráficos del álgebra booleana.

Las funciones matemáticas y otras operaciones complejas pueden representarse directamente en combinación con los cuadros lógicos. Para crear la lógica de operaciones complejas, inserte ramas paralelas entre los cuadros.



Significado de EN y ENO en las instrucciones con cuadros

Tanto KOP como FUP utilizan el flujo de corriente (EN y ENO) para algunas instrucciones con cuadros. Algunas instrucciones (p. ej. matemáticas y de transferencia) visualizan parámetros para EN y ENO. Estos parámetros se refieren al flujo de corriente y determinan si la instrucción se ejecuta en ese ciclo.

- EN (entrada de habilitación) es una entrada booleana para los cuadros KOP y FUP. Esta entrada debe tener flujo de corriente (EN = 1) para poder ejecutar el cuadro. Si la entrada EN de un cuadro KOP se conecta directamente a la barra de alimentación izquierda, el cuadro se ejecutará siempre.
- ENO (salida de habilitación) es una salida booleana para los cuadros KOP y FUP. Si el cuadro tiene flujo de corriente en la entrada EN y su función se ejecuta sin errores, la salida ENO conducirá corriente (ENO = 1) al siguiente elemento. Si se detecta un error en la ejecución del cuadro, el flujo de corriente se terminará (ENO = 0) en la instrucción de cuadro que ha generado el error.

Editor de programas	Entradas/salidas	Operandos	Tipo de datos
KOP	EN, ENO	Flujo de corriente	BOOL
FUP	EN	I, I:P, Q, M, DB, Temp, flujo de corriente	BOOL
	ENO	Flujo de corriente	BOOL

5.6 Protección anticopia

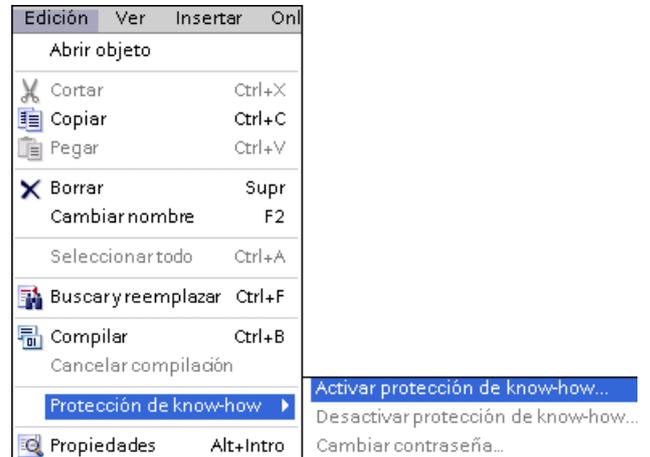
La protección anticopia o de "know-how" impide el acceso no autorizado a uno o más bloques lógicos (OB, FB o FC) del programa. Es posible crear una contraseña para limitar el acceso al bloque lógico.

Si un bloque se configura para que tenga protección de "know-how", no será posible acceder al código del bloque si no se introduce la contraseña.

Para proteger el bloque contra copia, elija el comando "Protección de know-how" del menú "Edición". Introduzca luego la contraseña que permite acceder al bloque.

La protección por contraseña impide que el bloque lógico sea leído o modificado sin autorización. Si no se introduce la contraseña, sólo es posible leer la siguiente información del bloque lógico:

- Título, comentario y propiedades del bloque
- Parámetros de transferencia (IN, OUT, IN_OUT, Return)
- Estructura de llamadas del programa
- Variables globales en las referencias cruzadas (sin información acerca de la ubicación); las variables locales se ocultan

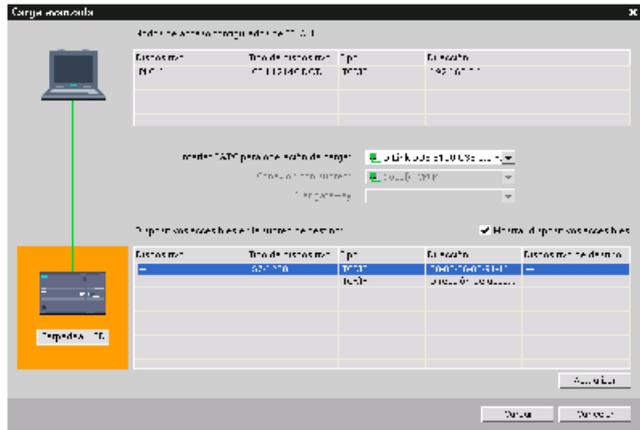


5.7 Cargar los elementos del programa en la CPU

Los elementos del proyecto se pueden cargar desde la programadora a la CPU. Al cargar un proyecto en la CPU, el programa de usuario (OBs, FCs, FBs y DBs) se almacena en la memoria no volátil de la CPU.

El proyecto se puede cargar de la programadora en la CPU desde cualquiera de las ubicaciones siguientes:

- "Árbol del proyecto": Haga clic con el botón derecho del ratón en el elemento de programa deseado y elija el comando "Cargar en dispositivo" del menú contextual.
- Menú "Online": Elija el comando "Cargar en dispositivo".
- Barra de herramientas: Haga clic en el botón "Cargar en dispositivo".



5.8 Cargar los elementos del programa desde un dispositivo

Todos los bloques de programa y la tabla de variables se pueden cargar desde una CPU online en un proyecto offline. No obstante, no es posible cargar la configuración de dispositivos ni tablas de observación. No es posible cargar en un proyecto vacío. Una CPU offline debe estar disponible para poder realizar la carga. No es posible cargar un solo bloque desde un dispositivo, sino sólo el programa entero. Al cargar desde un dispositivo, la CPU offline (es decir, todos los bloques y la tabla de variables) se "borrará" antes de realizar la carga y tras confirmar una pregunta de seguridad. No es posible editar un bloque en el área online. Primero es necesario cargarlo desde el dispositivo en el área offline, modificarlo allí y volver a cargarlo en el PLC.

La carga puede realizarse de dos maneras, a saber: mediante drag&drop en el árbol del proyecto o por sincronización en el editor de comparación.

Drag&drop en el árbol del proyecto

1. Cree un proyecto.
2. Agregue una CPU que concuerde con la CPU desde la que desea realizar la carga.
3. Expanda el nodo de la CPU una vez para visualizar la carpeta "Bloques de programa".
4. En el árbol del proyecto, expanda el nodo "Accesos online" y luego el nodo de la red deseada y haga doble clic en "Actualizar dispositivos accesibles".
5. Una vez listadas las CPUs disponibles, expanda el nodo de la CPU deseada.
6. Haga clic con el botón izquierdo del ratón en la carpeta "Bloques de programa" del área "Accesos online" y arrástrela hasta la carpeta "Bloques de programa" del área offline. Suelte luego el botón izquierdo del ratón. El puntero del ratón se convierte en un '+' cuando se encuentre en el área correcta.
7. Se abre el diálogo "Vista preliminar para cargar del dispositivo". Haga clic en la casilla para "Continuar" y luego en "Cargar de dispositivo".
8. Espere hasta que finalice la carga desde el dispositivo. Todos los bloques de programa, bloques tecnológicos y variables aparecen en el área offline.
9. Puesto que no es posible cargar la configuración del dispositivo, utilice la "Configuración de dispositivos" para configurar manualmente las propiedades de la CPU, incluyendo la dirección IP deseada y agregue todos los demás dispositivos al proyecto offline.

Es posible arrastrar desde el área online hasta el área "Bloques de programa" de un programa existente. Por tanto, el área offline "Bloques de programa" no tiene que estar vacía. En este caso, el programa existente se borrará y se sustituirá por el programa online.

Sincronización en el editor de comparación

1. Abra el proyecto que contiene el proyecto.
2. En el árbol del proyecto, seleccione la CPU offline que desea comparar.
3. Abra el editor de comparación haciendo clic con el botón derecho en la CPU offline o eligiendo el comando "Comparar offline/online" del menú "Herramientas".
4. El editor de comparación lista las diferencias en la carpeta "Bloques de programa". Haga clic en el símbolo en la columna "Acción". Para cargar el proyecto, seleccione "Cargar de dispositivo".
5. Haga clic en el botón "Sincronizar online y offline" para copiar el proyecto desde la CPU online en la CPU offline.

5.9 Depurar y comprobar el programa

Las "tablas de observación" se utilizan para observar y forzar los valores del programa de usuario que se está ejecutando en la CPU online. Es posible crear y guardar diferentes tablas de observación en el programa para soportar distintos entornos de test. Esto permite reproducir los tests durante la puesta en marcha, o bien para fines de servicio y mantenimiento.

Una tabla de observación permite observar e interactuar con la CPU mientras ésta ejecuta el programa de usuario. Es posible ver o cambiar los valores no sólo de las variables de los bloques lógicos y bloques de datos, sino también de las áreas de memoria de la CPU, incluyendo las entradas y salidas (I y Q), entradas y salidas de la periferia (I:P y Q:P), marcas (M) y bloques de datos (DB).

La tabla de observación permite habilitar las salidas físicas (Q:P) de una CPU en estado operativo STOP. Por ejemplo, es posible asignar valores específicos a las salidas al comprobar el cableado de la CPU.

La tabla de observación también permite "forzar permanentemente" o ajustar una variable a un determinado valor. Para más información sobre el forzado permanente, consulte el apartado Forzado permanente de valores en la CPU (Página 313) del capítulo "Online y diagnóstico".

Instrucciones de programación

6.1 Instrucciones básicas

6.1.1 Instrucciones lógicas con bits

Contactos KOP

	<p>"IN"</p> <p>Los contactos se pueden conectar a otros contactos, creando así una lógica combinacional propia. Si el bit de entrada indicado utiliza el identificador de memoria I (entrada) o Q (salida), el valor de bit se lee de la memoria imagen de proceso. Las señales de los contactos físicos del proceso controlado se cablean con los bornes de entrada del PLC. La CPU consulta las señales de entrada cableadas y actualiza continuamente los valores de estado correspondientes en la memoria imagen de proceso de las entradas.</p>
<p>Normalmente abierto</p>	
	<p>"IN"</p> <p>La lectura inmediata de una entrada física se indica introduciendo ":P" después del offset I (p. ej. "%I3.4:P"). En una lectura inmediata, los valores de datos de bit se leen directamente de la entrada física y no de la memoria imagen de proceso. La lectura inmediata no actualiza la memoria imagen de proceso.</p>
<p>Normalmente cerrado</p>	

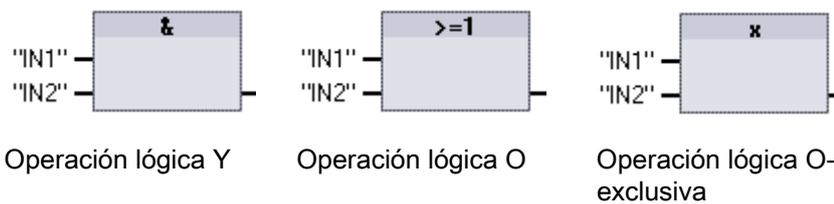
Parámetro	Tipo de datos	Descripción
IN	Bool	Bit asignado

- El contacto normalmente abierto se cierra (ON) cuando el valor de bit asignado es igual a 1.
- El contacto normalmente cerrado se cierra (ON) cuando el valor de bit asignado es igual a 0.
- Los contactos conectados en serie crean segmentos lógicos Y.
- Los contactos conectados en paralelo crean segmentos lógicos O.

Cuadros Y, O y O-exclusiva en FUP

En la programación FUP, los segmentos de los contactos KOP se transforman en segmentos de cuadros Y (&), O (>=1) y O-exclusiva OR (x), en los que pueden indicarse valores de bit para las entradas y salidas de los cuadros. También es posible interconectar cuadros lógicos y crear combinaciones lógicas propias. Tras disponer un cuadro en el segmento, es posible arrastrar la función "Insertar entrada binaria" desde la barra de herramientas "Favoritos" o desde el árbol de instrucciones y soltarla en el lado de entrada del cuadro para agregar entradas adicionales. También se puede hacer clic con el botón derecho del ratón en el conector de entrada del cuadro y seleccionar "Insertar entrada".

Es posible conectar las entradas y salidas de los cuadros con un cuadro lógico diferente, o bien introducir una dirección de bit o un nombre simbólico de bit para una entrada no conectada. Cuando se ejecuta el cuadro, los estados actuales de las entradas se aplican a la lógica del cuadro binario y, si se cumplen, la salida del cuadro será verdadera.



Parámetro	Tipo de datos	Descripción
IN1, IN2	Bool	Bit de entrada

- Todas las entradas de un cuadro Y tienen que cumplirse para que la salida sea TRUE (verdadera).
- Una entrada cualquiera de un cuadro O tiene que cumplirse para que la salida sea TRUE (verdadera).
- Un número impar de entradas de un cuadro O-exclusiva tiene que cumplirse para que la salida sea TRUE (verdadera).

Invertir resultado lógico (NOT)

En la programación FUP es posible arrastrar la función "Negar valor binario" desde la barra de herramientas "Favoritos" o desde el árbol de instrucciones y soltarla en una entrada o salida para crear un inversor lógico en ese conector del cuadro.



KOP: Contacto negado (NOT) FUP: Cuadro Y con una entrada lógica negada FUP: Cuadro Y con entrada y salida lógica negada

El contacto NOT KOP invierte el estado lógico de la entrada de flujo de corriente.

- Si no fluye corriente al contacto NOT, hay flujo de corriente en la salida.
- Si fluye corriente al contacto NOT, no hay flujo de corriente en la salida.

Bobina de salida KOP

"OUT"


Bobina de
relé, salida

"OUT"


Bobina de relé
negada

La instrucción "Bobina de salida, relé" escribe un valor en un bit de salida. Si el bit de salida indicado utiliza el identificador de memoria Q, la CPU activa o desactiva el bit de salida en la memoria imagen de proceso, poniendo el bit especificado al correspondiente estado de flujo de corriente. Las señales de salida de los actuadores se cablean con los bornes de salida del S7-1200. En estado operativo RUN, la CPU consulta continuamente las señales de entrada, procesa los estados de las entradas según la lógica del programa y escribe los nuevos valores de las salidas en la memoria imagen de proceso de las salidas. Tras cada ciclo del programa, la CPU transfiere el nuevo estado de las salidas almacenado en la memoria imagen de proceso a los bornes de salida cableados.

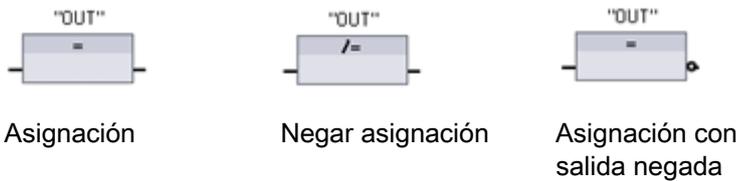
La escritura inmediata en una salida física se indica introduciendo ":P" después del offset Q (p. ej. "%Q3.4:P"). En una escritura inmediata, los valores de datos de bit se escriben en la memoria imagen de proceso de las salidas y directamente en la salida física.

Parámetro	Tipo de datos	Descripción
OUT	Bool	Bit asignado

- Si fluye corriente a través de una bobina de salida, el bit de salida se pone a 1.
- Si no fluye corriente a través de una bobina de salida, el bit de la bobina de salida se pone a 0.
- Si fluye corriente a través de una bobina de relé negada, el bit de salida se pone a 0.
- Si no fluye corriente a través de una bobina de relé negada, el bit de salida se pone a 1.

Cuadro de asignación de salidas FUP

En la programación FUP, las bobinas KOP se transforman en cuadros de asignación (= y /=), en los que se indica una dirección de bit para la salida del cuadro. Es posible conectar las entradas y salidas del cuadro con otros cuadros lógicos, o bien introducir una dirección de bit.



Parámetro	Tipo de datos	Descripción
OUT	Bool	Bit asignado

- Si la entrada del cuadro de salida es 1, el bit OUT se pone a 1.
- Si la entrada del cuadro de salida es 0, el bit OUT se pone a 0.
- Si la entrada del cuadro de salida negada es 1, el bit OUT se pone a 0.
- Si la entrada del cuadro de salida negada es 0, el bit OUT se pone a 1.

6.1.1.1 Instrucciones "Activar salida" y "Desactivar salida"

S y R: Activar y desactivar 1 bit

- Si se activa S (Set), el valor de datos de la dirección de salida OUT se pone a 1. Si no se activa S, no se modifica OUT.
- Si se activa R (Reset), el valor de datos de la dirección de salida OUT se pone a 0. Si no se activa R, no se modifica OUT.
- Estas instrucciones pueden disponerse en cualquier posición del segmento.



Parámetro	Tipo de datos	Descripción
IN (o conectar a contacto/puerta)	Bool	Dirección de bit que debe vigilarse
OUT	Bool	Dirección de bit que se debe activar o desactivar

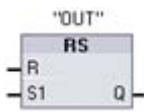
SET_BF y RESET_BF: Activar y desactivar mapa de bits



Parámetro	Tipo de datos	Descripción
n	Constante	Número de bits que deben escribirse
OUT	Elemento de una matriz booleana	Elemento inicial de un mapa de bits que se debe activar o desactivar Ejemplo: #MyArray[3]

- Cuando se activa SET_BF, el valor de datos 1 se asigna a "n" bits, comenzando en la dirección OUT. Si no se activa SET_BF, no se modifica OUT.
- RESET_BF escribe el valor de datos 0 en "n" bits, comenzando en la dirección OUT. Si no se activa RESET_BF, no se modifica OUT.
- Estas instrucciones sólo se pueden disponer en el extremo derecho de una rama.

RS y SR: Flipflop de activación/desactivación y flipflop de desactivación/activación



RS es un flipflop en el que domina la activación. Si las señales de activación (S1) y desactivación (R) son verdaderas, la dirección de salida OUT se pone a 1.



SR es un flipflop en el que domina la desactivación. Si las señales de activación (S) y desactivación (R1) son verdaderas, la dirección de salida OUT se pone a 0.

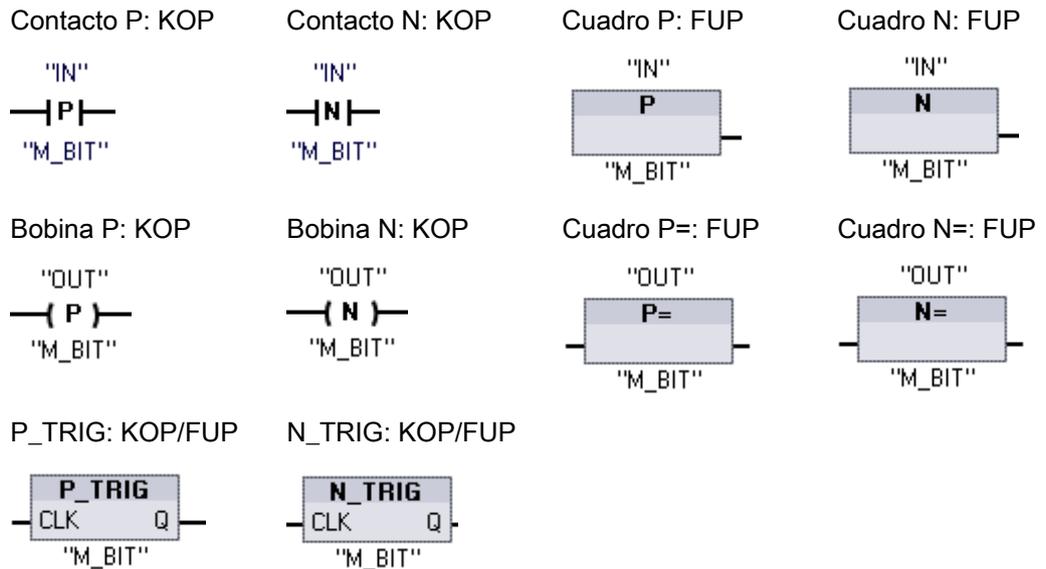
El parámetro OUT indica la dirección de bit que se activa o desactiva. La salida opcional OUT Q refleja el estado lógico de la dirección "OUT".

Parámetro	Tipo de datos	Descripción
S, S1	BOOL	Activar entrada; 1 indica dominancia
R, R1	BOOL	Desactivar entrada; 1 indica dominancia
OUT	BOOL	Salida de bit asignada "OUT"
Q	BOOL	Corresponde al estado del bit "OUT"

Instrucción	S1	R	Bit "OUT"
RS	0	0	Estado anterior
	0	1	0
	1	0	1
	1	1	1
SR	S	R1	
	0	0	Estado anterior
	0	1	0
	1	0	1
	1	1	0

6.1.1.2 Instrucciones "Consultar flanco de señal ascendente de un operando" y "Consultar flanco de señal descendente de un operando"

Consultar flanco de señal ascendente de un operando y Consultar flanco de señal descendente de un operando



Parámetro	Tipo de datos	Descripción
M_BIT	Bool	Marca en la que se almacena el estado anterior de la entrada
IN	Bool	Bit de entrada cuyo flanco debe detectarse
OUT	Bool	Bit de salida que indica que se ha detectado un flanco
CLK	Bool	Flujo de corriente o bit de entrada cuyo flanco debe detectarse
Q	Bool	Salida que indica que se ha detectado un flanco

Contacto P: El estado de este contacto es TRUE (verdadero) cuando se detecta un flanco ascendente (OFF a ON) en el bit "IN" asignado. El estado lógico del contacto se combina entonces con el estado de entrada del flujo de corriente para activar el estado de salida del flujo de corriente. El contacto P puede disponerse en cualquier posición del segmento, excepto al final de una rama.

Contacto N: El estado de este contacto es TRUE (verdadero) cuando se detecta un flanco descendente (ON a OFF) en el bit de entrada asignado. El estado lógico del contacto se combina entonces con el estado de entrada del flujo de corriente para activar el estado de salida del flujo de corriente. El contacto N puede disponerse en cualquier posición del segmento, excepto al final de una rama.

- Cuadro P:
FUP El estado lógico de la salida es TRUE (verdadero) cuando se detecta un flanco ascendente (OFF a ON) en el bit de entrada asignado. El cuadro P sólo se puede disponer al comienzo de una rama.
- Cuadro N:
FUP El estado lógico de la salida es TRUE (verdadero) cuando se detecta un flanco descendente (ON a OFF) en el bit de entrada asignado. El cuadro N sólo se puede disponer al comienzo de una rama.
- Bobina P:
KOP El bit asignado "OUT" es TRUE (verdadero) cuando se detecta un flanco ascendente (OFF a ON) en el flujo de corriente que entra a la bobina. El estado de entrada del flujo de corriente atraviesa la bobina como el estado de salida del flujo de corriente. La bobina P puede disponerse en cualquier posición del segmento.
- Bobina N:
KOP El bit asignado "OUT" es TRUE (verdadero) cuando se detecta un flanco descendente (ON a OFF) en el flujo de corriente que entra a la bobina. El estado de entrada del flujo de corriente atraviesa la bobina como el estado de salida del flujo de corriente. La bobina N puede disponerse en cualquier posición del segmento.
- Cuadro P=:
FUP El bit asignado "OUT" es TRUE (verdadero) cuando se detecta un flanco ascendente (OFF a ON) en el estado lógico de la conexión de entrada del cuadro, o bien en la asignación del bit de entrada si el cuadro está ubicado al comienzo de una rama. El estado lógico de la entrada atraviesa el cuadro como el estado lógico de la salida. El cuadro P= puede disponerse en cualquier posición de la rama.
- Cuadro N=:
FUP El bit asignado "OUT" es TRUE (verdadero) cuando se detecta un flanco descendente (ON a OFF) en el estado lógico de la conexión de entrada del cuadro, o bien en la asignación del bit de entrada si el cuadro está ubicado al comienzo de una rama. El estado lógico de la entrada atraviesa el cuadro como el estado lógico de la salida. El cuadro N= puede disponerse en cualquier posición de la rama.
- P_TRIG:
KOP/FUP El flujo de corriente o estado lógico de la salida Q es TRUE (verdadero) cuando se detecta un flanco ascendente (OFF a ON) en el estado lógico de CLK (en FUP) o en el flujo de corriente de CLK (en KOP). En KOP, la instrucción P_TRIG no se puede disponer ni al comienzo ni al final de un segmento. En FUP, la instrucción P_TRIG puede disponerse en cualquier posición de la rama, excepto al final.
- N_TRIG
(KOP/FUP) El flujo de corriente o estado lógico de la salida Q es TRUE (verdadero) cuando se detecta un flanco descendente (ON a OFF) en el estado lógico de CLK (en FUP) o en el flujo de corriente de CLK (en KOP). En KOP, la instrucción N_TRIG no se puede disponer ni al comienzo ni al final de un segmento. En FUP, la instrucción P_TRIG puede disponerse en cualquier posición de la rama, excepto al final.

Todas las instrucciones de detección de flancos utilizan una marca (M_BIT) para almacenar el estado anterior de la señal de entrada que se está vigilando. Un flanco se detecta comparando el estado de la entrada con el estado de la marca. Si los estados indican un cambio de la entrada en el sentido deseado, se notifica un flanco activando la salida (TRUE). De lo contrario, se desactivará la salida (FALSE).

Nota

Las instrucciones de detección de flancos evalúan los valores de la entrada y de la marca cada vez que se ejecutan, incluyendo la primera ejecución. Los estados iniciales de la entrada y de la marca deben considerarse al diseñar el programa, con objeto de permitir o impedir la detección de flancos en el primer ciclo.

Puesto que la marca debe conservarse desde una ejecución hasta la siguiente, es preciso utilizar un bit unívoco para cada instrucción de detección de flancos. Este bit no se puede utilizar en ninguna otra ubicación del programa. También se debe evitar la memoria temporal y la memoria que pueda ser modificada por otras funciones de sistema, p. ej. una actualización de E/S. Utilice sólo el área de marcas (M), DB global o memoria estática (en un DB instancia) para las asignaciones de memoria de M_BIT.

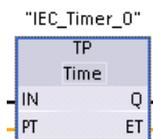
6.1.2 Temporizadores

Las instrucciones con temporizadores se utilizan para crear retardos programados:

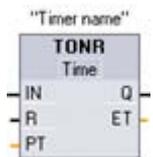
- TP: El temporizador Impulso genera un impulso con una duración predeterminada.
- TON: La salida Q del temporizador de retardo al conectar se activa al cabo de un tiempo de retardo predeterminado.
- TOF: La salida Q del temporizador de retardo al desconectar se desactiva al cabo de un tiempo de retardo predeterminado.
- TONR: La salida Q del acumulador de tiempo se activa al cabo de un tiempo de retardo predeterminado. El tiempo transcurrido se acumula a lo largo de varios periodos de temporización hasta que la entrada R inicializa el tiempo transcurrido.
- RT: Inicializa un temporizador borrando los datos de tiempo almacenados en el bloque de datos instancia del temporizador indicado.

Todos los temporizadores utilizan una estructura almacenada en un bloque de datos para mantener los datos. El bloque de datos se asigna al colocar la instrucción de temporización en el editor.

Al colocar instrucciones de temporización en un bloque de función es posible seleccionar la opción de bloque de datos multiinstancia. Los nombres de estructura de los temporizadores pueden diferir en las distintas estructuras, pero los datos de los temporizadores se encuentran en un bloque de datos individual y no requieren un bloque de datos propio para cada temporizador. Esto reduce el tiempo de procesamiento y la memoria de datos necesaria para gestionar los temporizadores. No hay interacción entre las estructuras de datos de los temporizadores en el bloque de datos multiinstancia compartido.



Los temporizadores TP, TON y TOF tienen los mismos parámetros de entrada y salida.



El temporizador TONR dispone adicionalmente de la entrada de reset R. Cree un "Nombre de temporizador" propio para designar el bloque de datos temporizador y describir el objetivo de este temporizador en el proceso.

"Nombre de temporizador"
 ---[RT]---

La instrucción RT inicializa el tiempo del temporizador indicado.

Parámetro	Tipo de datos	Descripción
IN	Bool	Habilitar entrada del temporizador
R	Bool	Poner a cero el tiempo transcurrido de TONR
PT	Bool	Entrada que indica el tiempo predeterminado
Q	Bool	Salida del temporizador
ET	Time	Salida que indica el tiempo transcurrido
Bloque de datos temporizador	DB	Indica qué temporizador debe inicializarse con la instrucción RT

El parámetro IN inicia y detiene los temporizadores:

- Un cambio de 0 a 1 del parámetro IN inicia los temporizadores TP, TON y TONR.
- Un cambio de 1 a 0 del parámetro IN inicia el temporizador TOF.

La tabla siguiente muestra el efecto de los cambios de valores en los parámetros PT e IN.

Temporizador	Cambios de los parámetros PT e IN
TP	<ul style="list-style-type: none"> • Un cambio de PT no tiene efecto alguno durante el funcionamiento del temporizador. • Un cambio de IN no tiene efecto alguno durante el funcionamiento del temporizador.
TON	<ul style="list-style-type: none"> • Un cambio de PT no tiene efecto alguno durante el funcionamiento del temporizador. • Si IN cambia a FALSE durante el funcionamiento del temporizador, éste se inicializará y se detendrá.
TOF	<ul style="list-style-type: none"> • Un cambio de PT no tiene efecto alguno durante el funcionamiento del temporizador. • Si IN cambia a TRUE durante el funcionamiento del temporizador, éste se inicializará y se detendrá.
TONR	<ul style="list-style-type: none"> • Un cambio de PT no tiene efecto alguno durante el funcionamiento del temporizador, pero sí cuando reanuda el contaje. • Si IN cambia a FALSE durante el funcionamiento del temporizador, éste se detendrá pero no se inicializará. Si IN vuelve a cambiar a TRUE, el temporizador comenzará a contar desde el valor de tiempo acumulado.

Valores TIME

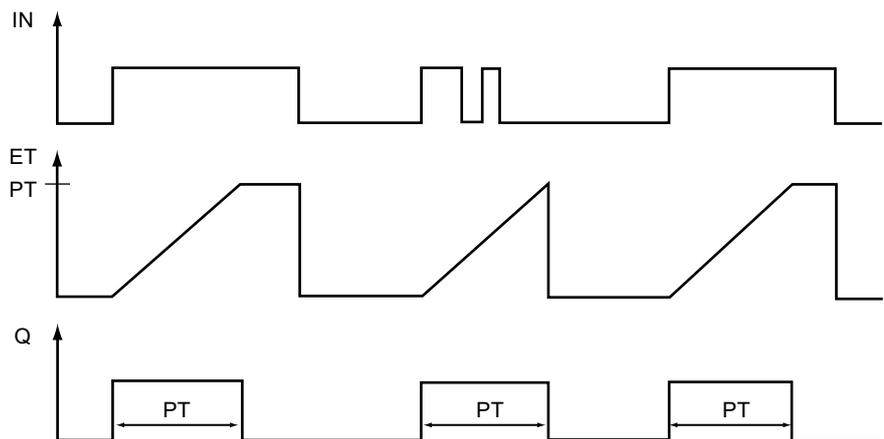
Los valores PT (tiempo predeterminado) y ET (tiempo transcurrido) se almacenan en la memoria como enteros dobles con signo que representan milisegundos. Los datos TIME utilizan el identificador T# y pueden introducirse como unidad de tiempo simple "T#200ms" o como unidades de tiempo compuestas "T#2s_200ms".

Tipo de datos	Tamaño	Rangos válidos
TIME	32 bits Almacenado como	T#-24d_20h_31m_23s_648ms hasta T#24d_20h_31m_23s_647ms -2.147.483.648 ms hasta +2.147.483.647 ms

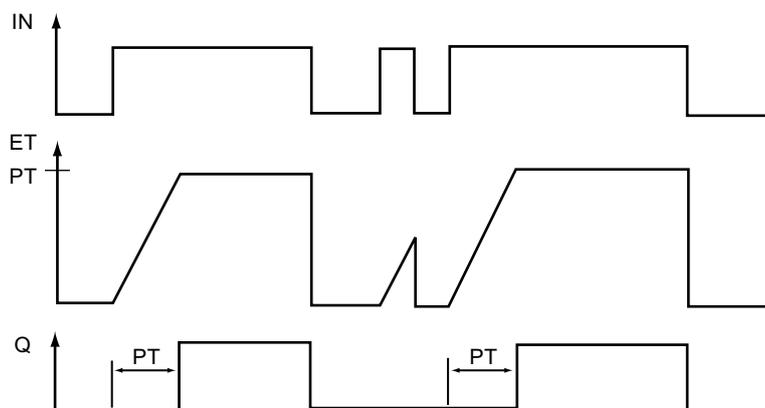
Nota

El rango negativo del tipo de datos TIME indicado arriba no puede utilizarse con las instrucciones de temporización. Los valores PT (tiempo predeterminado) negativos se ponen a cero cuando se ejecuta la instrucción de temporización. ET (tiempo transcurrido) es siempre un valor positivo.

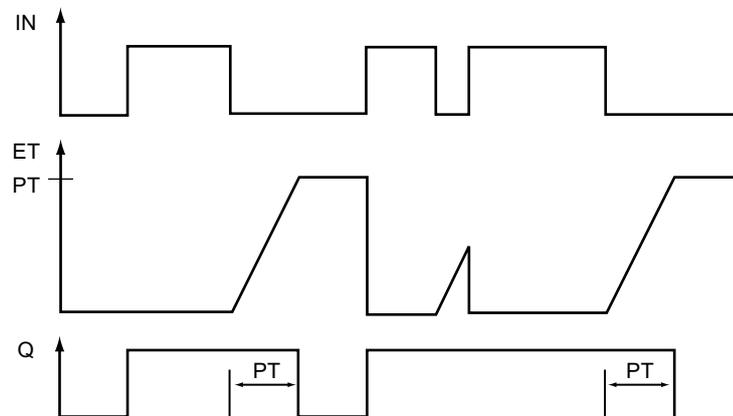
TP:
Cronograma de impulsos



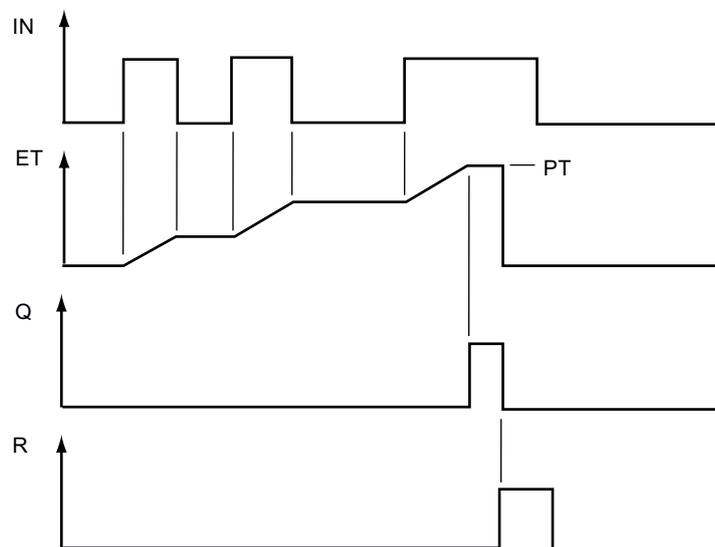
TON:
Cronograma de retardo al conectar



TOF:
Cronograma
de retardo al
desconectar



TONR:
Cronograma
del
acumulador
de tiempo



6.1.3 Contadores

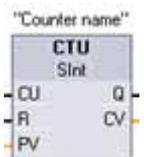
6.1.3.1 Contadores

Las instrucciones con contadores se utilizan para contar eventos del programa internos y eventos del proceso externos:

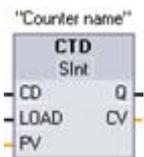
- CTU es un contador ascendente.
- CTD es un contador descendente.
- CTUD es un contador ascendente/descendente.

Todo contador utiliza una estructura almacenada en un bloque de datos para conservar sus datos. El bloque de datos se asigna al colocar la instrucción de conteo en el editor. Estas instrucciones utilizan contadores por software cuya frecuencia de conteo máxima está limitada por la frecuencia de ejecución del OB en el que están contenidas. El OB en el que se depositan las instrucciones debe ejecutarse con suficiente frecuencia para detectar todas las transiciones de las entradas CU o CD. Para operaciones de conteo rápido, consulte la instrucción CTRL_HSC.

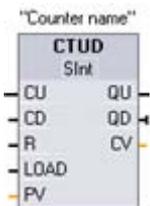
Al colocar instrucciones con contadores en un bloque de función es posible seleccionar la opción de bloque de datos multiinstancia. Los nombres de estructura de los contadores pueden diferir en las distintas estructuras, pero los datos de los contadores se encuentran en un bloque de datos individual y no requieren un bloque de datos propio para cada contador. Esto reduce el tiempo de procesamiento y la memoria de datos necesaria para los contadores. No hay interacción entre las estructuras de datos de los contadores en el bloque de datos multiinstancia compartido.



Seleccione el tipo de datos del valor de conteo en la lista desplegable debajo del nombre del cuadro.



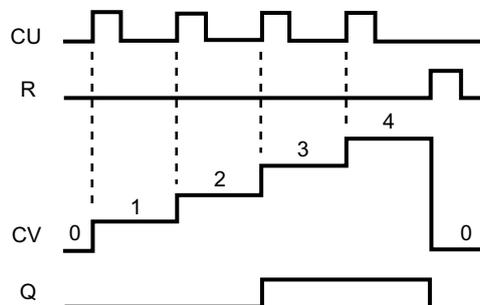
Cree un "Nombre de contador" propio para designar el bloque de datos contador y describir el objetivo de este contador en el proceso.



Parámetro	Tipo de datos	Descripción
CU, CD	Bool	Contaje ascendente o descendente, en incrementos de uno
R (CTU, CTUD)	Bool	Poner a cero el valor del contador
LOAD (CTD, CTUD)	Bool	Control de carga del valor predeterminado
PV	SInt, Int, DInt, USInt, UInt, UDInt	Valor de contaje predeterminado
Q, QU	Bool	Es verdadero si $CV \geq PV$
QD	Bool	Es verdadero si $CV \leq 0$
CV	SInt, Int, DInt, USInt, UInt, UDInt	Valor de contaje actual

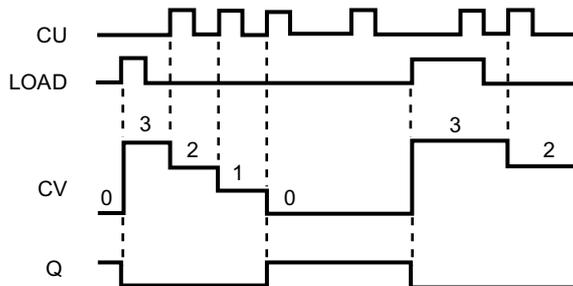
El rango numérico de valores de contaje depende del tipo de datos seleccionado. Si el valor de contaje es un entero sin signo, es posible contar hacia atrás hasta cero o hacia delante hasta el límite del rango. Si el valor de contaje es un entero con signo, es posible contar hacia atrás hasta el límite de entero negativo y contar hacia delante hasta el límite de entero positivo.

CTU: CTU se incrementa en 1 cuando el valor del parámetro CU cambia de 0 a 1. Si el valor del parámetro CV (valor de contaje actual) es mayor o igual al valor del parámetro PV (valor de contaje predeterminado), el parámetro de salida del contador Q = 1. Si el valor del parámetro de reset R cambia de 0 a 1, el valor de contaje actual se pone a 0. La figura siguiente muestra un cronograma de la instrucción CTU con un valor de contaje de entero sin signo (donde $PV = 3$).

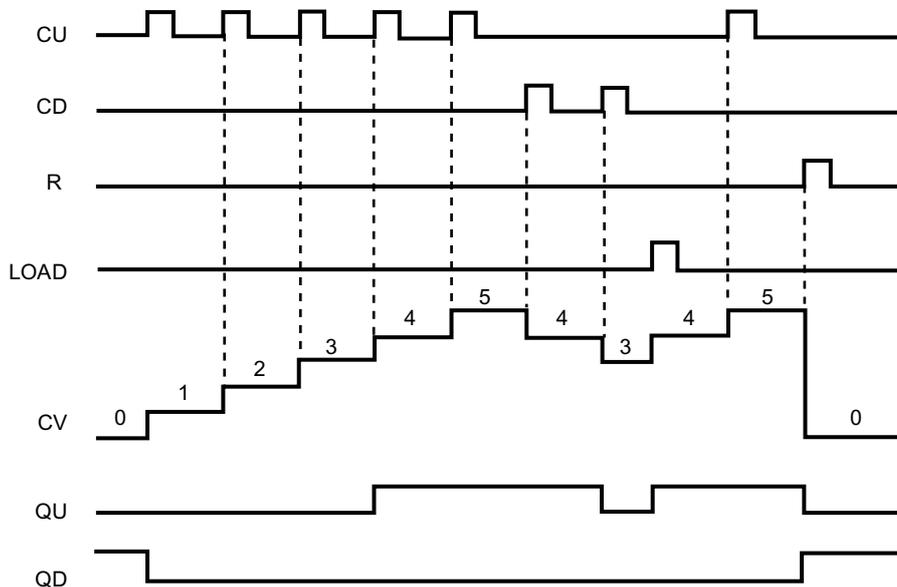


6.1 Instrucciones básicas

CTD: CTD se decrementa en 1 cuando el valor del parámetro CD cambia de 0 a 1. Si el valor del parámetro CV (valor de conteaje actual) es menor o igual a 0, el parámetro de salida del contador Q = 1. Si el valor del parámetro LOAD cambia de 0 a 1, el valor del parámetro PV (valor predeterminado) se carga en el contador como nuevo CV (valor de conteaje actual). La figura siguiente muestra un cronograma de la instrucción CTD con un valor de conteaje de entero sin signo (donde PV = 3).



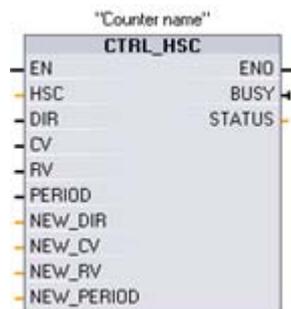
CTUD: CTUD se incrementa o decrementa en 1 cuando el estado lógico de las entradas de conteaje ascendente (CU) o descendente (CD) cambia de 0 a 1. Si el valor del parámetro CV (valor de conteaje actual) es mayor o igual al valor del parámetro PV (valor predeterminado), el parámetro de salida del contador QU = 1. Si el valor del parámetro CV es menor o igual a cero, el parámetro de salida del contador QD = 1. Si el valor del parámetro LOAD cambia de 0 a 1, el valor del parámetro PV (valor predeterminado) se carga en el contador como nuevo CV (valor de conteaje actual). Si el valor del parámetro de reset R cambia de 0 a 1, el valor de conteaje actual se pone a 0. La figura siguiente muestra un cronograma de la instrucción CTUD con un valor de conteaje de entero sin signo (donde PV = 4).



6.1.3.2 Instrucción CTRL_HSC

La instrucción CTRL_HSC permite controlar los contadores rápidos utilizados para contar eventos que ocurren más rápidamente que la frecuencia de ejecución del OB. La frecuencia de conteo de las instrucciones CTU, CTD y CTUD está limitada por la frecuencia de ejecución del OB en el que están contenidas. Encontrará más información sobre las frecuencias de entrada de reloj máximas del HSC en los datos técnicos (Página 325) de la CPU.

Una aplicación típica de los contadores rápidos es el conteo de impulsos generados por un encoder rotativo de control de movimiento.



Toda instrucción CTRL_HSC utiliza una estructura almacenada en un bloque de datos para conservar los datos. El bloque de datos se asigna al disponer la instrucción CTRL_HSC en el editor.

Cree un "Nombre de contador" propio para designar el bloque de datos contador y describir el objetivo de este contador en el proceso.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
HSC	IN	HW_HSC	Identificador del HSC
DIR	IN	Bool	1 = solicitar nuevo sentido de conteo
CV	IN	Bool	1 = solicitar activación del nuevo valor del contador
RV	IN	Bool	1 = solicitar activación del nuevo valor de referencia
PERIOD	IN	Bool	1 = solicitar activación del nuevo periodo (sólo para el modo de medición de frecuencia)
NEW_DIR	IN	Int	Nuevo sentido: 1= hacia delante -1= hacia atrás
NEW_CV	IN	Dint	Nuevo valor del contador
NEW_RV	IN	Dint	Nuevo valor de referencia
NEW_PERIOD	IN	Int	Nuevo periodo en segundos: 0,01, 0,1 ó 1 (sólo para el modo de medición de frecuencia)
BUSY	OUT	Bool	Función ocupada
STATUS	OUT	Word	Código de condición de ejecución

Los contadores rápidos deben configurarse en la configuración de dispositivos PLC del proyecto antes de poder utilizarlos en el programa. Al configurar los dispositivos HSC se seleccionan el modo de contaje, los conectores de E/S, la asignación de alarmas y el funcionamiento como contador rápido o dispositivo para medir la frecuencia de pulsos. Los contadores rápidos pueden operarse con o sin control del programa.

Numerosos parámetros de configuración de los contadores rápidos se ajustan sólo en la "Configuración de dispositivos" del proyecto. Algunos parámetros de los contadores rápidos se inicializan en la "Configuración de dispositivos" del proyecto, pero se pueden modificar luego bajo el control del programa.

Los parámetros de la instrucción CTRL_HSC permiten controlar el programa del proceso de contaje:

- Ajustar el sentido de contaje al valor NEW_DIR
- Ajustar el valor de contaje actual al valor NEW_CV
- Ajustar el valor de referencia al valor NEW_RV
- Ajustar el periodo (para el modo de medición de frecuencia) al valor NEW_PERIOD

Si las siguientes marcas booleanas están puestas a 1 cuando se ejecuta la instrucción CTRL_HSC, el valor NEW_XXX correspondiente se carga en el contador. Las peticiones múltiples (varias marcas se activan simultáneamente) se procesan en una sola ejecución de la instrucción CTRL_HSC.

- DIR = 1 es una petición para cargar un valor NEW_DIR, 0 = ningún cambio
- CV = 1 es una petición para cargar un valor NEW_CV, 0 = ningún cambio
- RV = 1 es una petición para cargar un valor NEW_RV, 0 = ningún cambio
- PERIOD = 1 es una petición para cargar un valor NEW_PERIOD, 0 = ningún cambio

Generalmente, la instrucción CTRL_HSC se inserta en un OB de alarma de proceso que se ejecuta cuando se dispara el evento de alarma de proceso del contador. Por ejemplo, si un evento CV=RV dispara la alarma del contador, un OB de alarma de proceso ejecuta la instrucción CTRL_HSC y puede cambiar el valor de referencia cargando un valor NEW_RV.

El valor de contaje actual no está disponible en los parámetros de CTRL_HSC. La dirección de la memoria imagen de proceso que almacena el valor de contaje actual se asigna al configurar el hardware del contador rápido. La lógica del programa puede utilizarse para leer directamente el valor de contaje. El valor devuelto al programa será el contaje correcto en el instante en el que se ha leído el contador. El contador continuará contando eventos rápidos. Por tanto, el valor de contaje real puede cambiar antes de que el programa finalice un proceso utilizando un valor de contaje antiguo.

Detalles de los parámetros de CTRL_HSC:

- Si no se solicita la actualización de un parámetro, se ignoran los valores de entrada correspondientes.
- El parámetro DIR es válido sólo si el sentido de contaje se ajusta a "Programa de usuario (control interno de sentido)". La forma de utilizar este parámetro se determina en la configuración de dispositivos HSC.
- En un HSC S7-1200 en la CPU o en la Signal Board, el parámetro BUSY siempre está puesto a 0.

Códigos de condición: En caso de error, ENO se pone a 0 y la salida STATUS contiene un código de condición.

Valor de STATUS (W#16#...)	Descripción
0	Sin error
80A1	El identificador HSC no direcciona un HSC
80B1	Valor de NEW_DIR no permitido
80B2	Valor de NEW_CV no permitido
80B3	Valor de NEW_RV no permitido
80B4	Valor de NEW_PERIOD no permitido

6.1.3.3 Funcionamiento del contador rápido

Un contador rápido (HSC) puede utilizarse como entrada para un encoder rotativo incremental. El encoder rotativo ofrece un número determinado de valores de contaje por revolución, así como un impulso de reset que ocurre una vez por revolución. El o los relojes y el impulso de reset del encoder suministran las entradas para el contador rápido.

El primero de los valores predeterminados se carga en el HSC y las salidas se activan durante el periodo en el que el contaje actual es menor que el valor predeterminado. El HSC pone a disposición una alarma cuando el contaje actual es igual al valor predeterminado, al ocurrir un reset y también al producirse un cambio de sentido.

Cuando el contaje actual es igual al valor predeterminado y se presenta un evento de alarma, se carga un nuevo valor predeterminado y se activa el siguiente estado para las salidas. Cuando ocurre el evento de alarma de reset, se activan el primer valor predeterminado y los primeros estados de las salidas y se repite el ciclo.

Puesto que las alarmas ocurren con una frecuencia mucho menor que la frecuencia de contaje del HSC, es posible implementar un control preciso de las operaciones rápidas con un impacto relativamente bajo en el ciclo de la CPU. El método de asociar alarmas permite cargar cada valor predeterminado nuevo en un subprograma por separado, lo que simplifica el control del estado. (Alternativamente, todos los eventos de alarma se pueden ejecutar en una solo subprograma.)

Seleccionar las funciones del HSC

Todos los HSCs funcionan de la misma manera en el mismo modo de operación del contador. Hay cuatro tipos básicos de HSCs, a saber:

- Contador de fase simple con control interno del sentido de contaje
- Contador de fase simple con control externo del sentido de contaje
- Contador de dos fases con 2 entradas de reloj
- Contador A/B

Todo tipo de HSC puede utilizarse con o sin entrada de reset. Cuando se activa la entrada de reset (con algunas restricciones, v. la tabla siguiente), el valor actual se borra y se mantiene borrado hasta que se desactive la entrada de reset.

- Función de frecuencia: Algunos modos del HSC (Modo de contaje) permiten configurarlo de manera que notifique la frecuencia en vez del contaje actual de impulsos. Hay tres periodos de medición de frecuencia disponibles: 0,01, 0,1 ó 1,0 segundos.

El periodo de medición de frecuencia determina cada cuánto calcula y notifica el HSC un nuevo valor de frecuencia. La frecuencia notificada es un valor promedio determinado por el número total de contajes en el último periodo de medición. Si la frecuencia cambia rápidamente, el valor notificado será el valor medio entre la frecuencia más alta y más baja registrada durante el periodo de medición. La frecuencia se indica siempre en hertzios (impulsos por segundo), independientemente del ajuste del periodo de medición de frecuencia.

- Modos y entradas de contador: La tabla siguiente muestra las entradas utilizadas para las funciones de reloj, control de sentido y reset asociadas al HSC.

Una misma entrada no se puede utilizar para dos funciones diferentes. Sin embargo, cualquier entrada que no se esté utilizando en el modo actual del HSC se puede usar para otro fin. Por ejemplo, si el HSC1 está en un modo que utiliza entradas integradas, pero que no usa el reset externo (I0.3), la entrada I0.3 puede utilizarse para alarmas de flanco o para el HSC2.

Descripción		Asignación de entradas predeterminada			Función	
HSC	HSC1	Integrado o Signal Board o supervisión PTO 0 ¹	I0.0 I4.0 PTO 0 Impulso	I0.1 I4.1 PTO 0 Sentido	I0.3 I4.3 -	
	HSC:	Integrado o Signal Board o supervisión PTO 1 ¹	I0.2 I4.2 PTO 1 Impulso	I0.3 I4.3 PTO 1 Sentido	I0.1 I4.1 -	
	HSC3 ²	Integrado	I0.4	I0.5	I0.7	
	HSC4 ³	Integrado	I0.6	I0.7	I0.5	
	HSC5 ⁴	Integrado o Signal Board	I1.0 I4.0	I1.1 I4.1	I1.2 I4.3	
HSC6 ⁴	Integrado o Signal Board	I1.3 I4.2	I1.4 I4.3	I1.5 I4.1		
Modo	Contador de fase simple con control interno del sentido de contaje		Reloj	-	-	Contaje o frecuencia
					Reset	Contaje
	Contador de fase simple con control externo del sentido de contaje		Reloj	Sentido	-	Contaje o frecuencia
					Reset	Contaje

Descripción		Asignación de entradas predeterminada			Función
Contador de dos fases con 2 entradas de reloj		Reloj adelante	Reloj atrás	-	Contaje o frecuencia
				Reset	Contaje
Contador en cuadratura (fases A/B)		Fase A	Fase B	-	Contaje o frecuencia
				Fase Z	Contaje
Supervisión de trenes de impulsos (PTO) ¹		Reloj	Sentido	-	Contaje

¹ La supervisión de trenes de impulsos utiliza siempre las funciones de reloj y sentido. Si la salida PTO correspondiente está configurada sólo para impulsos, la salida de sentido se debería ajustar generalmente para el contaje adelante.

² HSC3 con una entrada de reset no es posible para la CPU 1211C, que soporta sólo 6 entradas integradas.

³ HSC4 no es posible para la CPU 1211C, que soporta sólo 6 entradas integradas.

⁴ HSC5 y HSC6 sólo son soportados por la CPU 1211C y CPU 1212C si está instalada una Signal Board.

Acceso al valor actual del HSC

La CPU almacena el valor actual de cada HSC en una dirección de entrada (I). La tabla siguiente muestra las direcciones predeterminadas asignadas al valor actual de cada HSC. La dirección I del valor actual se puede cambiar modificando las propiedades de la CPU en la "Configuración de dispositivos".

Contador rápido	Tipo de datos	Dirección predeterminada
HSC1	Dint	ID1000
HSC2	Dint	ID1004
HSC3	Dint	ID1008
HSC4	Dint	ID1012
HSC5	Dint	ID1016
HSC6	Dint	ID1020

Las E/S digitales asignadas a dispositivos HSC no se pueden forzar permanentemente

Las E/S digitales utilizadas por los contadores rápidos se asignan durante la configuración de dispositivos. Si se asignan direcciones de E/S digitales a estos dispositivos, los valores de las direcciones de E/S asignadas no podrán ser modificados por la función de forzado permanente de la tabla de observación.

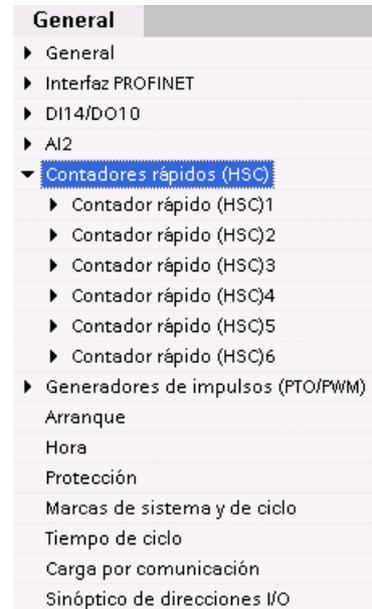
6.1.3.4 Configuración del HSC

La CPU permite configurar hasta 6 contadores rápidos. Las "Propiedades" de la CPU se editan para configurar los parámetros de cada HSC.

Los parámetros de los contadores rápidos se configuran editando las "Propiedades" de la CPU.

Tras habilitar el HSC se deben configurar los demás parámetros, tales como la función del contador, los valores iniciales, las opciones de reset y los eventos de alarma.

Una vez configurado el HSC, se utiliza la instrucción CTRL_HSC en el programa de usuario para controlar el funcionamiento del HSC.



Activar

Activar este contador rápido

Modo: Contaje

Fase servicio: Monofásica

Origen señal: Entrada de CPU integrada

Sentido de conteaje dado por: Programa de usuario (control i)

Sentido de conteaje inicial: Incrementar contador

Período de medición de frecuencia: +/- sec

Restablecer valores

Valor de conteaje inicial: 0

Valor de referencia inicial: 0

Opciones de reset

Este HSC debe utilizar una entrada externa de reset. Al resetear se borrará el valor actual.

Restablecer nivel de señal: Alto

Generar alarma para evento si el valor de conteaje es igual al valor de referencia.:

Nombre del evento: Valor de conteaje igual a va

Alarma de proceso: --

Generar alarma para evento de reset externo.:

Nombre del evento:

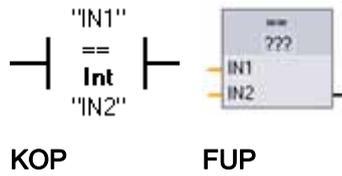
Alarma de proceso: --

Activar alarma para evento de cambio de sentido.:

Nombre del evento:

Alarma de proceso: --

6.1.4 Comparación



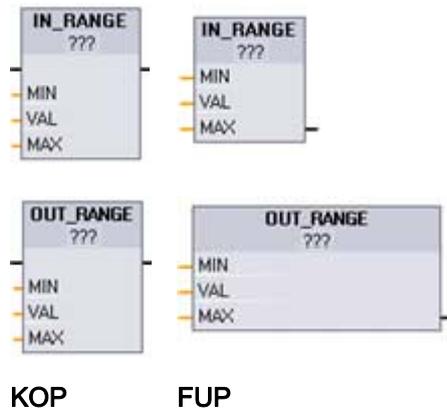
Las instrucciones de comparación se utilizan para comparar dos valores de un mismo tipo de datos. Si la comparación de contactos KOP es TRUE (verdadera), se activa el contacto. Si la comparación de cuadros FUP es TRUE (verdadera), la salida del cuadro es TRUE.

Tras hacer clic en la instrucción en el editor de programación, es posible seleccionar el tipo de comparación y el tipo de datos en las listas desplegables respectivas.

Tipo de relación	La comparación se cumple si:
==	IN1 es igual a IN2
<>	IN1 es diferente de IN2
>=	IN1 es mayor o igual a IN2
<=	IN1 es menor o igual a IN2
>	IN1 es mayor que IN2
<	IN1 es menor que IN2

Parámetro	Tipo de datos	Descripción
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, String, Char, Time, DTL, constante	Valores que deben compararse

Valor dentro del rango y Valor fuera del rango



Las instrucciones IN_RANGE (Valor dentro del rango) y OUT_RANGE (Valor fuera del rango) permiten comprobar si un valor de entrada está dentro o fuera de un rango de valores especificado. Si la comparación es TRUE (verdadera), la salida del cuadro es TRUE.

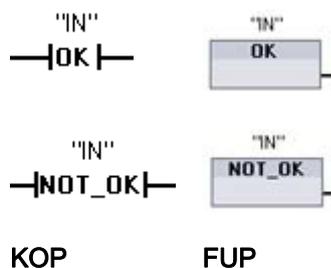
Los parámetros de entrada MIN, VAL y MAX deben tener un mismo tipo de datos.

Tras hacer clic en la instrucción en el editor de programación, el tipo de datos se puede seleccionar en las listas desplegables.

Tipo de relación	La comparación es TRUE (verdadera) si:
IN_RANGE	$MIN \leq VAL \leq MAX$
OUT_RANGE	$VAL < MIN$ o $VAL > MAX$

Parámetro	Tipo de datos	Descripción
MIN, VAL, MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, constante	Entradas de comparación

Instrucciones "Comprobar validez" y "Comprobar invalidez"



Las instrucciones OK (Comprobar validez) y NOT_OK (Comprobar invalidez) permiten comprobar si un valor de referencia de entrada es un número real válido según la especificación IEEE 754. Si el contacto KOP es TRUE (verdadero), se activa el contacto y conduce corriente. Si el cuadro FUP es TRUE (verdadero), la salida del cuadro es TRUE.

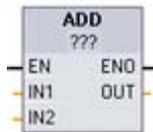
Un valor Real o LReal no es válido si es +/- INF (infinito), NaN (no es un número) o si es un valor desnormalizado. Un valor desnormalizado es un número muy próximo a cero. La CPU sustituye un valor desnormalizado por cero en los cálculos.

Instrucción	La comprobación del número Real es TRUE (verdadera) si:
OK	El valor de entrada es un número Real válido
NOT_OK	El valor de entrada no es un número Real válido

Parámetro	Tipo de datos	Descripción
IN	Real, LReal	Datos de entrada

6.1.5 Funciones matemáticas

Instrucciones "Sumar", "Restar", "Multiplicar" y "Dividir"



Las instrucciones matemáticas con cuadros se utilizan para programar las operaciones matemáticas básicas:

- ADD: Sumar ($IN1 + IN2 = OUT$)
- SUB: Restar ($IN1 - IN2 = OUT$)
- MUL: Multiplicar ($IN1 * IN2 = OUT$)
- DIV: Dividir ($IN1 / IN2 = OUT$)

Una operación de división de enteros trunca la parte fraccionaria del cociente y produce un valor de salida entero.

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Nota

Los parámetros IN1, IN2 y OUT de las instrucciones matemáticas básicas deben tener un mismo tipo de datos.

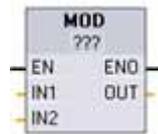
Parámetro	Tipo de datos	Descripción
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, constante	Entradas de la operación matemática
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Salida de la operación matemática

Si está habilitada (EN = 1), la instrucción matemática realiza la operación indicada en los valores de entrada (IN1 e IN2) y almacena el resultado en la dirección de memoria que indica el parámetro de salida (OUT). Una vez finalizada correctamente la operación, la instrucción pone ENO a 1.

Estado de ENO	Descripción
1	Sin error
0	El resultado de la operación matemática quedaría fuera del rango numérico válido del tipo de datos seleccionado. Se devuelve la parte menos significativa del resultado que quepa en el tamaño de destino.
0	División por 0 (IN2 = 0): El resultado es indefinido y se devuelve cero.
0	Real/LReal: Si uno de los valores de entrada es NaN (no es un número), se devuelve NaN.
0	ADD Real/LReal: Si ambos valores IN son INF con signos diferentes, la operación no está permitida y se devuelve NaN.
0	SUB Real/LReal: Si ambos valores IN son INF con signos iguales, la operación no está permitida y se devuelve NaN.

Estado de ENO	Descripción
0	MUL Real/LReal: Si un valor IN es cero y el otro es INF, la operación no está permitida y se devuelve NaN.
0	DIV Real/LReal: Si ambos valores IN son cero o INF, la operación no está permitida y se devuelve NaN.

6.1.5.1 Instrucción "Obtener resto de división"



La instrucción MOD (Obtener resto de división) se utiliza para la operación matemática IN1 modulo IN2. La operación $IN1 \text{ MOD } IN2 = IN1 - (IN1 / IN2) = \text{parámetro OUT}$.

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

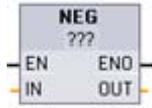
Nota

Los parámetros IN1, IN2 y OUT deben tener un mismo tipo de datos.

Parámetro	Tipo de datos	Descripción
IN1 e IN2	Int, DInt, USInt, UInt, UDInt, constante	Entradas modulo
OUT	Int, DInt, USInt, UInt, UDInt	Salida modulo

Estado de ENO	Descripción
1	Sin error
0	Valor IN2 = 0, el valor cero se asigna a OUT

Instrucción "Generar complemento a dos"



La instrucción NEG (Generar complemento a dos) permite invertir el signo aritmético del valor del parámetro IN y almacenar el resultado en el parámetro OUT.

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

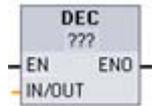
Nota

Los parámetros IN y OUT deben tener un mismo tipo de datos.

Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DInt, Real, LReal, constante	Entrada de la operación matemática
OUT	SInt, Int, DInt, Real, LReal	Salida de la operación matemática

Estado de ENO	Descripción
1	Sin error
0	El resultado está fuera del rango numérico válido del tipo de datos seleccionado. Ejemplo de SInt: NEG (-128) arroja el resultado +128 que excede el límite máximo del tipo de datos.

Instrucciones "Incrementar" y "Decrementar"



Las Instrucciones INC (Incrementar) y DEC (Decrementar) se utilizan para:

- Incrementar un valor de número entero con o sin signo

INC (Incrementar): Valor del parámetro IN/OUT + 1 = valor del parámetro IN/OUT

- Decrementar un valor de número entero con o sin signo

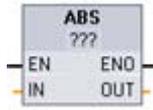
DEC (Decrementar): Valor del parámetro IN/OUT - 1 = valor del parámetro IN/OUT

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Parámetro	Tipo de datos	Descripción
IN/OUT	SInt, Int, DInt, USInt, UInt, UDInt	Entrada/salida de la operación matemática

Estado de ENO	Descripción
1	Sin error
0	El resultado está fuera del rango numérico válido del tipo de datos seleccionado. Ejemplo de SInt: INC (127) arroja el resultado -128 que excede el límite máximo del tipo de datos.

Instrucción "Calcular valor absoluto"



La instrucción ABS (Calcular valor absoluto) permite calcular el valor absoluto de un entero con signo o número real indicado en el parámetro IN y almacenar el resultado en el parámetro OUT.

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

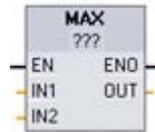
Nota

Los parámetros IN y OUT deben tener un mismo tipo de datos.

Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DInt, Real, LReal	Entrada de la operación matemática
OUT	SInt, Int, DInt, Real, LReal	Salida de la operación matemática

Estado de ENO	Descripción
1	Sin error
0	El resultado de la operación aritmética está fuera del rango numérico válido del tipo de datos seleccionado. Ejemplo de SInt: ABS (-128) arroja el resultado +128 que excede el límite máximo del tipo de datos.

Instrucciones "Determinar mínimo" y "Determinar máximo"



Las instrucciones MIN (Determinar mínimo) y MAX (Determinar máximo) se utilizan del siguiente modo:

- MIN compara el valor de dos parámetros IN1 e IN2 y asigna el valor mínimo (menor) al parámetro OUT.
- MAX compara el valor de dos parámetros IN1 e IN2 y asigna el valor máximo (mayor) al parámetro OUT.

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Nota

Los parámetros IN1, IN2 y OUT deben tener un mismo tipo de datos.

Parámetro	Tipo de datos	Descripción
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Real, constante	Entradas de la operación matemática
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real	Salida de la operación matemática

Estado de ENO	Descripción
1	Sin error
0	Sólo para el tipo de datos Real: <ul style="list-style-type: none"> • Una o ambas entradas no son un número Real (NaN). • La salida OUT resultante es +/- INF (infinito).

Instrucción "Ajustar valor límite"

La instrucción "Ajustar valor límite" permite comprobar si el valor del parámetro IN está dentro del rango de valores especificado por los parámetros MIN y MAX. El valor OUT se conecta al valor MIN o MAX si el valor IN está fuera de este rango.



- Si el valor del parámetro IN está dentro del rango indicado, el valor de IN se almacena en el parámetro OUT.
- Si el valor del parámetro IN está fuera del rango indicado, el valor OUT es entonces el valor del parámetro MIN (si el valor IN es menor que el valor MIN) o del parámetro MAX (si el valor IN es mayor que el valor MAX).

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Nota

Los parámetros MIN, IN, MAX y OUT deben tener un mismo tipo de datos.

Parámetro	Tipo de datos	Descripción
MIN, IN y MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real, constante	Entradas de la operación matemática
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real	Salida de la operación matemática

Estado de ENO	Descripción
1	Sin error
0	Real: Si uno o varios de los valores de MIN, IN y MAX es NaN (no es un número), se devuelve NaN.
0	Si MIN es mayor que MAX, el valor IN se asigna a OUT.

Instrucciones matemáticas en coma flotante

Las instrucciones en coma flotante sirven para programar operaciones matemáticas utilizando los tipos de datos Real o LReal:

- SQR: Calcular cuadrado ($IN^2 = OUT$)
- SQRT: Calcular raíz cuadrada ($\sqrt{IN} = OUT$)
- LN: Calcular logaritmo natural ($LN(IN) = OUT$)
- EXP: Calcular valor exponencial ($e^{IN} = OUT$), donde la base e = 2.71828182845904523536
- SIN: Calcular valor de seno ($\text{seno}(IN \text{ radianes}) = OUT$)
- COS: Calcular valor de coseno ($\text{coseno}(IN \text{ radianes}) = OUT$)
- TAN: Calcular valor de tangente ($\text{tangente}(IN \text{ radianes}) = OUT$)
- ASIN: Calcular valor de arcoseno ($\text{arcoseno}(IN) = OUT \text{ radianes}$), donde $\text{seno}(OUT \text{ radianes}) = IN$
- ACOS: Calcular valor de arcocoseno ($\text{arcocoseno}(IN) = OUT \text{ radianes}$), donde $\text{coseno}(OUT \text{ radianes}) = IN$
- ATAN: Calcular valor de arcotangente ($\text{arcotangente}(IN) = OUT \text{ radianes}$), donde $\text{tangente}(OUT \text{ radianes}) = IN$
- FRAC: Determinar decimales (parte fraccionaria del número en coma flotante $IN = OUT$)
- EXPT: Elevar a potencia ($IN1^{IN2} = OUT$)



Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable. Los parámetros IN1 y OUT de la instrucción EXPT son siempre números reales. Es posible seleccionar el tipo de datos del parámetro del exponente IN2.



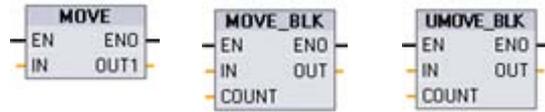
Parámetro	Tipo de datos	Descripción
IN, IN1	Real, LReal, constante	Entradas
IN2	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, constante	Entrada de la instrucción EXPT
OUT	Real, LReal	Salidas

Estado de ENO	Instrucción	Condición	Resultado (OUT)
1	Todas	Sin error	Resultado válido
0	SQR	El resultado excede el rango Real/LReal válido	+INF
		IN es +/- NaN (no es un número)	+NaN

Estado de ENO	Instrucción	Condición	Resultado (OUT)
	SQRT	IN es negativo	-NaN
		IN es +/- INF (infinito) o +/- NaN	+/- INF o +/- NaN
	LN	IN es 0,0, negativo, -INF o -NaN	-NaN
		IN es +INF o +NaN	+INF o +NaN
	EXP	El resultado excede el rango Real/LReal válido	+INF
		IN es +/- NaN	+/- NaN
	SIN, COS, TAN	IN es +/- INF o +/- NaN	+/- INF o +/- NaN
	ASIN, ACOS	IN está fuera del rango válido de -1,0 a +1,0	+NaN
		IN es +/- NaN	+/- NaN
	ATAN	IN es +/- NaN	+/- NaN
	FRAC	IN es +/- INF o +/- NaN	+NaN
	EXPT	IN1 es +INF e IN2 no es -INF	+INF
		IN1 es negativo o -INF	+NaN si IN2 es Real/LReal, -INF en caso contrario
		IN1 o IN2 es +/- NaN	+NaN
		IN1 es 0,0 e IN2 es Real/LReal (sólo)	+NaN

6.1.6 Desplazamiento

Instrucciones "Copiar valor" y "Copiar área"



Las instrucciones de desplazamiento permiten copiar elementos de datos a otra dirección de memoria y convertir un tipo de datos en otro. El proceso de desplazamiento no modifica los datos de origen.

- MOVE: Copia un elemento de datos almacenado en una dirección indicada a una dirección diferente
- MOVE_BLK: Desplazamiento interruptible que copia un área de elementos de datos a otra dirección
- UMOVE_BLK: Desplazamiento no interruptible que copia un área de elementos de datos a otra dirección

MOVE		
Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, Array, Struct, DTL, Time	Dirección de origen
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, Array, Struct, DTL, Time	Dirección de destino

MOVE_BLK, UMOVE_BLK		
Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, Byte, Word, DWord	Dirección de origen inicial
COUNT	UInt	Número de elementos de datos que deben copiarse
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, Byte, Word, DWord	Dirección de destino inicial

Nota

Reglas para las operaciones de copia de datos

- Para copiar el tipo de datos Bool, utilice las instrucciones SET_BF, RESET_BF, R, S o Bobina de relé, salida (KOP)
- Para copiar un solo tipo de datos simple, utilice MOVE
- Para copiar una matriz de un tipo de datos simple, utilice MOVE_BLK o UMOVE_BLK
- Para copiar una estructura, utilice MOVE
- Para copiar una cadena, utilice S_CONV
- Para copiar un solo carácter en una cadena, utilice MOVE
- Las instrucciones MOVE_BLK y UMOVE_BLK no pueden utilizarse para copiar matrices o estructuras en las áreas de memoria I, Q o M.

La instrucción MOVE copia un elemento de datos individual de la dirección de origen que indica el parámetro IN en la dirección de destino que indica el parámetro OUT.

Las instrucciones MOVE_BLK y UMOVE_BLK tienen un parámetro COUNT adicional. COUNT especifica cuántos elementos de datos se copian. El número de bytes por elemento copiado depende del tipo de datos asignado a los nombres de variables de los parámetros IN y OUT en la tabla de variables PLC.

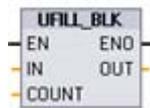
Las instrucciones MOVE_BLK y UMOVE_BLK se diferencian en la forma de procesar las alarmas:

- Los eventos de alarma **se ponen en cola de espera y se procesan** durante la ejecución de MOVE_BLK. Utilice la instrucción MOVE_BLK si los datos contenidos en la dirección de destino del desplazamiento no se utilizan en un OB de alarma. Si se utilizan, los datos de destino no tienen que ser coherentes. Si se interrumpe una instrucción MOVE_BLK, el último elemento de datos desplazado estará completo y será coherente en la dirección de destino. La instrucción MOVE_BLK se reanuda una vez finalizada la ejecución del OB de alarma.
- Los eventos de alarma **se ponen en cola de espera pero no se procesarán** hasta que no finalice la ejecución de UMOVE_BLK. Utilice la instrucción UMOVE_BLK si la operación de desplazamiento debe finalizarse y los datos de destino deben ser coherentes antes de la ejecución de un OB de alarma. Encontrará más información en el apartado Coherencia de datos (Página 96).

ENO siempre es verdadero tras ejecutarse la instrucción MOVE.

Estado de ENO	Condición	Resultado
1	Sin error	Todos los elementos de COUNT se han copiado correctamente
0	El rango de origen (IN) o destino (OUT) excede el área de memoria disponible	Se copian los elementos que quepan. No se copian elementos parciales.

Instrucciones de relleno



Las instrucciones FILL_BLK y UFILL_BLK se utilizan del siguiente modo:

- FILL_BLK: La instrucción "Rellenar área" rellena de forma interruptible un rango de direcciones con copias de un determinado elemento de datos.
- UFILL_BLK: La instrucción "Rellenar área sin interrupciones" rellena de forma no interruptible un rango de direcciones con copias de un determinado elemento de datos.

Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DIntT, USInt, UInt, UDIInt, Real, BYTE, Word, DWord	Dirección de origen de los datos
COUNT	USInt, UInt	Número de elementos de datos que deben copiarse
OUT	SInt, Int, DIntT, USInt, UInt, UDIInt, Real, BYTE, Word, DWord	Dirección de destino de los datos

Nota

Reglas para las operaciones de rellenar área

- Para rellenar con el tipo de datos BOOL, utilice las instrucciones SET_BF, RESET_BF, R, S o Bobina de relé, salida (KOP)
- Para rellenar con un solo tipo de datos simple, utilice MOVE
- Para rellenar una matriz con un tipo de datos simple, utilice FILL_BLK o UFILL_BLK
- Para rellenar un solo carácter en una cadena, utilice MOVE
- Las instrucciones FILL_BLK y UFILL_BLK no pueden utilizarse para rellenar matrices en las áreas de memoria I, Q o M.

Las instrucciones FILL_BLK y UFILL_BLK copian el elemento de datos de origen IN en el destino, cuya dirección inicial se indica en el parámetro OUT. El proceso de copia se repite y un área de direcciones adyacentes se rellena hasta que el número de copias es igual al parámetro COUNT.

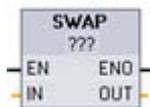
Las instrucciones FILL_BLK y UFILL_BLK se diferencian en la forma de procesar las alarmas:

- Los eventos de alarma **se ponen en cola de espera y se procesan** durante la ejecución de FILL_BLK. Utilice la instrucción FILL_BLK si los datos contenidos en la dirección de destino del desplazamiento no se utilizan en un OB de alarma. Si se utilizan, los datos de destino no tienen que ser coherentes.
- Los eventos de alarma **se ponen en cola de espera pero no se procesarán** hasta que no finalice la ejecución de UFILL_BLK. Utilice la instrucción UFILL_BLK si la operación de desplazamiento debe finalizarse y los datos de destino deben ser coherentes antes de la ejecución de un OB de alarma.

6.1 Instrucciones básicas

Estado de ENO	Condición	Resultado
1	Sin error	El elemento IN se ha copiado correctamente a todos los destinos de COUNT
0	El rango de destino (OUT) excede el área de memoria disponible	Se copian los elementos que quepan. No se copian elementos parciales.

6.1.6.1 Instrucción "Cambiar disposición"



La instrucción SWAP (Cambiar disposición) permite invertir el orden de los bytes de elementos de dos y cuatro bytes. El orden de los bits no se modifica dentro de los distintos bytes. ENO es siempre TRUE (verdadero) tras ejecutarse la instrucción SWAP.

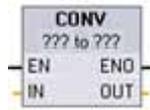
Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Parámetro	Tipo de datos	Descripción
IN	Word, DWord	Bytes de datos ordenados en IN
OUT	Word, DWord	Bytes de datos en orden inverso en OUT

	Ejemplo: Parámetro IN = MB0 antes de ejecutar SWAP				Ejemplo: Parámetro OUT = MB4 tras ejecutar SWAP			
Dirección	MB0	MB1			MB4	MB5		
W#16#1234	12	34			34	12		
WORD	MSB	LSB			MSB	LSB		
Dirección	MB0	MB1	MB2	MB3	MB4	MB5	MB6	MB7
DW#16#12345678	12	34	56	78	78	56	34	12
DWORD	MSB			LSB	MSB			LSB

6.1.7 Convertir

Instrucción "Convertir valor"



La instrucción CONVERT (Convertir valor) permite convertir un elemento de un tipo de datos a otro. Haga clic debajo del nombre del cuadro y seleccione los tipos de datos de IN y OUT en la lista desplegable.

Tras haber seleccionado el tipo de datos que se desea convertir, las conversiones posibles aparecen en la lista desplegable (convertir a). Las conversiones de y a BCD16 están limitadas al tipo de datos Int. Las conversiones de y a BCD32 están limitadas al tipo de datos DInt.

Haga clic debajo del nombre del cuadro y seleccione los tipos de datos en las listas desplegables.

Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Byte, Word, DWord, Real, LReal, Bcd16, Bcd32	Valor de IN
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Byte, Word, DWord, Real, LReal, Bcd16, Bcd32	Valor de IN convertido al nuevo tipo de datos

Estado de ENO	Descripción	Resultado de OUT
1	Sin error	Resultado válido
0	IN es +/- INF o +/- NaN	+/- INF o +/- NaN
0	El resultado excede el rango válido del tipo de datos de OUT	OUT se ajusta a los bytes menos significativos de IN

Instrucciones "Redondear número" y "Truncar a entero doble"



ROUND convierte un número real en un entero. La fracción del número real se redondea al número entero más cercano (IEEE - redondear al número más cercano). Si el número Real se encuentra exactamente entre dos enteros (p. ej. 10,5), el número Real se redondeará al entero par. Ejemplo, ROUND (10,5) = 10 ó ROUND (11,5) = 12.

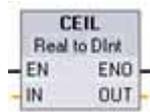


TRUNC convierte un número real en un entero. La parte fraccionaria del número real se trunca a cero (IEEE - redondear hacia cero).

Parámetro	Tipo de datos	Descripción
IN	Real, LReal	Número en coma flotante en la entrada
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Salida redondeada o truncada

Estado de ENO	Descripción	Resultado de OUT
1	Sin error	Resultado válido
0	IN es +/- INF o +/- NaN	+/- INF o +/- NaN

Instrucciones "Crear el siguiente número entero superior a partir del número en coma flotante" y "Crear el siguiente número entero inferior a partir del número en coma flotante"



La instrucción CEIL (Crear el siguiente número entero superior a partir del número en coma flotante) convierte un número real en el siguiente entero mayor o igual a ese número real (IEEE - redondear hacia el infinito positivo).



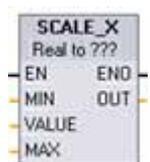
La instrucción FLOOR (Crear el siguiente número entero inferior a partir del número en coma flotante) convierte un número real en el siguiente entero menor o igual a ese número real (IEEE - redondear hacia el infinito negativo).

Parámetro	Tipo de datos	Descripción
IN	Real, LReal	Número en coma flotante en la entrada
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Salida convertida

Estado de ENO	Descripción	Resultado de OUT
1	Sin error	Resultado válido
0	IN es +/- INF o +/- NaN	+/- INF o +/- NaN

6.1.7.1 Instrucciones "Escalar" y "Normalizar"

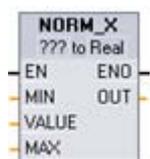
Instrucciones "Escalar" y "Normalizar"



La instrucción SCALE_X (Escalar) escala el parámetro VALUE real normalizado (donde $0,0 \leq \text{VALUE} \leq 1,0$) al tipo de datos y rango de valores especificados por los parámetros MIN y MAX:

$$\text{OUT} = \text{VALUE} (\text{MAX} - \text{MIN}) + \text{MIN}$$

Para SCALE_X, los parámetros MIN, MAX y OUT deben tener un mismo tipo de datos.



La instrucción NORM_X (Normalizar) normaliza el parámetro VALUE dentro del rango de valores especificado por los parámetros MIN y MAX:

$$\text{OUT} = (\text{VALUE} - \text{MIN}) / (\text{MAX} - \text{MIN}), \text{ donde } (0,0 \leq \text{OUT} \leq 1,0)$$

Para NORM_X, los parámetros MIN, VALUE y MAX deben tener un mismo tipo de datos.

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Parámetro	Tipo de datos	Descripción
MIN	SInt, Int, DInt, USInt, UInt, UDIInt, Real	Entrada que indica el valor mínimo del rango
VALUE	SCALE_X: Real NORM_X: SInt, Int, DInt, USInt, UInt, UDIInt, Real	Valor de entrada que se debe escalar o normalizar
MAX	SInt, Int, DInt, USInt, UInt, UDIInt, Real	Entrada que indica el valor máximo del rango
OUT	SCALE_X: SInt, Int, DInt, USInt, UInt, UDIInt, Real NORM_X: Real	Valor de salida escalado o normalizado

Nota

El parámetro VALUE de SCALE_X debe estar limitado a (0,0 <= VALUE <= 1,0)

Si el parámetro VALUE es menor que 0,0 o mayor que 1,0:

- La operación de escala lineal puede producir valores OUT menores que el valor del parámetro MIN o mayores que el del parámetro MAX en el caso de valores OUT comprendidos en el rango de valores del tipo de datos de OUT. La ejecución de SCALE_X pone ENO = TRUE en estos casos.
- Es posible generar números escalados no comprendidos en el rango del tipo de datos de OUT. En estos casos, el parámetro OUT se ajusta a un valor intermedio igual a la parte menos significativa del número real escalado antes de la conversión final al tipo de datos de OUT. La ejecución de SCALE_X pone ENO = FALSE en este caso.

El parámetro VALUE de NORM_X debe estar limitado a (MIN <= VALUE <= MAX)

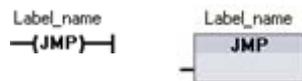
Si el parámetro VALUE es menor que MIN o mayor que MAX, la operación de escala lineal puede producir valores OUT normalizados menores que 0,0 o mayores que 1,0. La ejecución de NORM_X pone ENO = TRUE en este caso.

Estado de ENO	Condición	Resultado de OUT
1	Sin error	Resultado válido
0	El resultado excede el rango válido del tipo de datos de OUT	Resultado intermedio: La parte menos significativa del número real antes de la conversión final al tipo de datos de OUT.
0	Parámetros MAX <= MIN	SCALE_X: La parte menos significativa del número Real VALUE que debe rellenar el tamaño de OUT. NORM_X: VALUE en el tipo de datos VALUE ampliado para rellenar una palabra doble.
0	Parámetro VALUE = +/- INF o +/- NaN	VALUE se escribe en OUT

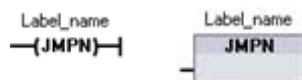
6.1.8 Control del programa

Instrucciones de salto y etiquetas

Las operaciones de control del programa permiten un control de la secuencia de ejecución en función de determinadas condiciones:



JMP: Si fluye corriente a una bobina JMP (KOP) o si se cumple el cuadro JMP (FUP), la ejecución del programa continúa con la primera instrucción que le sigue a la etiqueta indicada.



JMPN: Si no fluye corriente a una bobina JMP (KOP) o si el cuadro JMP no se cumple (FUP), la ejecución del programa continúa con la primera instrucción que le sigue a la etiqueta indicada.



Etiqueta: Etiqueta de destino de una instrucción de salto JMP o JMPN.

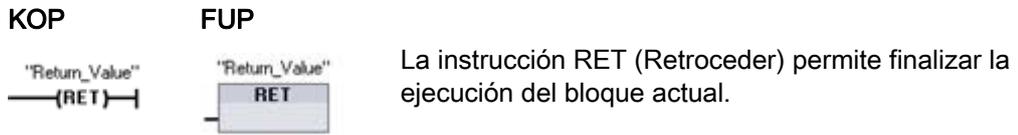
KOP

FUP

Parámetro	Tipo de datos	Descripción
Label_name	Identificador de la etiqueta	Identificador de las instrucciones de salto y etiqueta correspondiente al destino de salto

Los nombres de las etiquetas se crean tecleando directamente en la instrucción LABEL. Los nombres disponibles para el campo del nombre de JMP y JMPN se pueden seleccionar mediante el símbolo de ayuda del parámetro. El nombre de la etiqueta también puede teclearse directamente en la instrucción JMP o JMPN.

Instrucción de control de ejecución "Retroceder" (RET)



Parámetro	Tipo de datos	Descripción
Return_Value	Bool	El parámetro "Return_value" de la instrucción RET se asigna a la salida ENO del cuadro de llamada de bloque en el bloque que efectúa la llamada.

La instrucción RET opcional sirve para finalizar la ejecución del bloque actual. Sólo si fluye corriente a la bobina RET (LAD) o si se cumple la entrada del cuadro RET (FUP), la ejecución del programa del bloque actual finalizará en ese punto y las instrucciones posteriores a la instrucción RET no se ejecutarán. Si el bloque actual es un OB, se ignora el parámetro "Return_Value". Si el bloque actual es un FB o FC, el valor del parámetro "Return_Value" se devuelve a la rutina que efectúa la llamada como valor ENO del cuadro llamado.

No es necesario introducir manualmente la instrucción RET como última instrucción de un bloque. Esto se efectúa automáticamente. Un solo bloque puede comprender varias instrucciones RET.

Procedimiento para utilizar la instrucción RET en un bloque lógico FC (ejemplo):

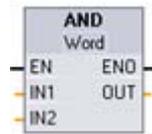
1. Cree un proyecto nuevo y agregue una FC:
2. Edite la FC:
 - Agregue instrucciones del árbol de instrucciones.
 - Agregue una instrucción RET, incluyendo uno de los siguientes valores para el parámetro "Return_Value":
TRUE, FALSE o una posición de memoria que indique el valor de retorno requerido.
 - Agregue más instrucciones.
3. Llame la FC desde el bloque MAIN [OB1].

La entrada EN del cuadro FC del bloque lógico MAIN se debe cumplir para comenzar la ejecución de la FC.

El valor que indica la instrucción RET en la FC se encontrará en la salida ENO del cuadro FC del bloque lógico tras ejecutarse la FC para la que fluye corriente a la instrucción RET.

6.1.9 Operaciones lógicas

Instrucciones Y, O y O-exclusiva



Y: Combinación Y lógica de tipos de datos BYTE, WORD y DWORD

O: Combinación O lógica de tipos de datos BYTE, WORD y DWORD

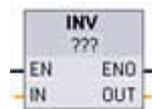
XOR: Combinación O-exclusiva lógica de tipos de datos BYTE, WORD y DWORD

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Parámetro	Tipo de datos	Descripción
IN1, IN2	Byte, Word, DWord	Entradas lógicas
OUT	Byte, Word, DWord	Salida lógica

La selección del tipo de datos ajusta los parámetros IN1, IN2 y OUT a un mismo tipo de datos. Los valores de bit correspondientes de IN1 e IN2 se combinan para producir un resultado lógico binario en el parámetro OUT. ENO es siempre TRUE (verdadero) tras ejecutarse estas instrucciones.

Instrucción "Complemento a uno"



La instrucción INV permite obtener el complemento a uno binario del parámetro IN. El complemento a uno se forma invirtiendo los valores de los distintos bits del parámetro IN (es decir, cambiando todo 0 a 1 y todo 1 a 0). ENO es siempre TRUE (verdadero) tras ejecutarse esta instrucción.

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Byte, Word, DWord	Elemento que debe invertirse
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Byte, Word, DWord	Salida invertida

Instrucciones "Codificar" y "Decodificar"



ENCO codifica un patrón de bits a un número binario.

DECO descodifica un número binario a un patrón de bits.

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Parámetro	Tipo de datos	Descripción
IN	ENCO: Byte, Word, DWord DECO: UInt	ENCO: Patrón de bits que debe codificarse DECO: Valor que debe descodificarse
OUT	ENCO: Int DECO: Byte, Word, DWord	ENCO: Valor codificado DECO: Patrón de bits descodificado

La instrucción ENCO (Codificar) convierte el parámetro IN al número binario correspondiente al bit activado menos significativo del parámetro IN y deposita el resultado en el parámetro OUT. Si el parámetro IN es 0000 0001 ó 0000 0000, el valor 0 se deposita en OUT. Si el valor del parámetro IN es 0000 0000, ENO adopta el estado lógico FALSE (falso).

La instrucción DECO (Descodificar) descodifica un número binario del parámetro IN, poniendo a "1" el bit correspondiente en el parámetro OUT (todos los demás bits se ponen a 0). ENO es siempre TRUE (verdadero) tras ejecutarse la instrucción DECO.

Seleccionando el tipo de datos Byte, Word o DWord para el parámetro OUT de la instrucción DECO se restringe el rango utilizable del parámetro IN. Si el valor del parámetro IN excede el rango utilizable, se ejecuta una operación modulo para extraer los bits menos significativos que se muestran abajo.

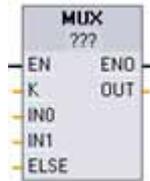
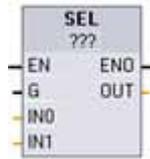
Rango del parámetro IN de DECO:

- 3 bits (valores 0-7) IN se utilizan para activar 1 bit en un byte OUT
- 4 bits (valores 0-15) IN se utilizan para activar 1 bit en una palabra OUT
- 5 bits (valores 0-31) IN se utilizan para activar 1 bit en una palabra doble OUT

Valor IN de DECO		Valor OUT de DECO (descodificar un bit)
		Byte OUT (8 bits):
Mín. IN	0	00000001
Máx. IN	7	10000000
		Word OUT (16 bits):
Mín. IN	0	0000000000000001
Máx. IN	15	1000000000000000
		DWord OUT: (32 bits):
Mín. IN	0	00000000000000000000000000000001
Máx. IN	31	10000000000000000000000000000000

Estado de ENO	Condición	Resultado (OUT)
1	Sin error	Número de bit válido
0	IN es cero	OUT se pone a cero

Instrucciones "Seleccionar" (SEL) y "Multiplexar" (MUX)



- La instrucción SEL (Seleccionar) asigna uno de dos valores de entrada al parámetro OUT, dependiendo del valor del parámetro G.
- La instrucción MUX (Multiplexar) asigna uno de varios valores de entrada al parámetro OUT, dependiendo del valor del parámetro K. Si el parámetro K excede el rango válido, el valor del parámetro ELSE se asigna al parámetro OUT.

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

SEL	Tipo de datos	Descripción
G	Bool	Interruptor selector: <ul style="list-style-type: none"> • FALSE para IN0 • TRUE para IN1
IN0, IN1	SInt, Int, DInt, USInt, UInt, UDInt, Real, Byte, Word, DWord, Time, Char	Entradas
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, Byte, Word, DWord, Time, Char	Salida

MUX	Tipo de datos	Descripción
K	UInt	Valor selector: <ul style="list-style-type: none"> • 0 para IN0 • 1 para IN1 • ...
IN0, IN1,	SInt, Int, DInt, USInt, UInt, UDInt, Real, Byte, Word, DWord, Time, Char	Entradas
ELSE	SInt, Int, DInt, USInt, UInt, UDInt, Real, Byte, Word, DWord, Time, Char	Entrada del valor sustitutivo (opcional)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, Byte, Word, DWord, Time, Char	Salida

Las variables de entrada y salida deben tener un mismo tipo de datos.

- La instrucción SEL selecciona siempre entre dos valores IN.
- La instrucción MUX tiene dos parámetros IN cuando se inserta por primera vez en el editor de programación. No obstante, se puede ampliar agregando más parámetros IN.

Utilice los métodos siguientes para agregar y quitar parámetros de entrada de la instrucción MUX:

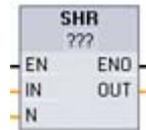
- Para agregar una entrada, + haga clic con el botón derecho del ratón en el conector de entrada de un parámetro IN existente y seleccione el comando "Insertar entrada".
- Para quitar una entrada, haga clic con el botón derecho del ratón en el conector de entrada de un parámetro IN existente (si hay más de las dos entradas originales) y seleccione el comando "Borrar".

Códigos de condición: ENO es siempre TRUE (verdadero) tras ejecutarse la instrucción SEL.

Estado de ENO (MUX)	Condición de MUX	Resultado OUT de MUX
1	Sin error	El valor IN seleccionado se asigna a OUT
0	K es mayor o igual al número de parámetros IN	No se ha indicado ELSE: OUT no se modifica Se ha indicado ELSE: el valor ELSE se asigna a OUT

6.1.10 Instrucciones de desplazamiento y rotación

Instrucciones de desplazamiento



Las instrucciones de desplazamiento permiten desplazar el patrón de bits del parámetro IN. El resultado se deposita en el parámetro OUT. El parámetro N indica el número de bits desplazados:

- SHR: Desplazar patrón de bits hacia la derecha
- SHL: Desplazar patrón de bits hacia la izquierda

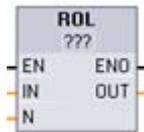
Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Parámetro	Tipo de datos	Descripción
IN	Byte, Word, DWord	Patrón de bits que debe desplazarse
N	UInt	Número de bits que deben desplazarse
OUT	Byte, Word, DWord	Patrón de bits después del desplazamiento

- Si N=0, no se realiza ningún desplazamiento y el valor de IN se asigna a OUT.
- Los ceros se desplazan a los bits que quedan vacíos tras el desplazamiento.
- Si el número de posiciones que deben desplazarse (N) excede el número de bits en el valor de destino (8 para Byte, 16 para Word, 32 para DWord), todos los valores de bit originales se desplazarán hacia fuera y se reemplazarán por ceros (cero se asigna a OUT).
- ENO es siempre TRUE (verdadero) en las operaciones de desplazamiento.

Ejemplo de SHL para datos en formato Word: Desplazar ceros desde el lado izquierdo			
IN	1110 0010 1010 1101	Valor de OUT antes del primer desplazamiento:	1110 0010 1010 1101
		Después del primer desplazamiento a la izquierda:	1100 0101 0101 1010
		Después del segundo desplazamiento a la izquierda:	1000 1010 1011 0100
		Después del tercer desplazamiento a la izquierda:	0001 0101 0110 1000

Instrucciones de rotación



Las instrucciones de rotación permiten rotar el patrón de bits del parámetro IN. El resultado se deposita en el parámetro OUT. El parámetro N define el número de bits rotados.

- ROR: Rotar patrón de bits hacia la derecha
- ROL: Rotar patrón de bits hacia la izquierda

Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en la lista desplegable.

Parámetro	Tipo de datos	Descripción
IN	Byte, Word, DWord	Patrón de bits que debe rotarse
N	UInt	Número de bits que deben rotarse
OUT	Byte, Word, DWord	Patrón de bits después de la rotación

- Si N=0, no se realiza ninguna rotación y el valor de IN se asigna a OUT.
- Los bits rotados hacia fuera en un lado del valor de destino se rotan hacia el otro lado, por lo que no se pierden los valores de bit originales.
- Si el número de bits que deben rotarse (N) excede el número de bits en el valor de destino (8 para Byte, 16 para Word, 32 para DWord), la rotación se efectúa de todas maneras.
- ENO es siempre TRUE (verdadero) tras ejecutarse las instrucciones de rotación.

Ejemplo de ROR para datos en formato WORD: Rotar bits hacia fuera a la derecha y hacia dentro a la izquierda			
IN	0100 0000 0000 0001	Valor de OUT antes de la primera rotación:	0100 0000 0000 0001
		Después de la primera rotación a la derecha:	1010 0000 0000 0000
		Después de la segunda rotación a la derecha:	0101 0000 0000 0000

6.2 Instrucciones avanzadas

6.2.1 Parámetros de error comunes de las instrucciones avanzadas

En las descripciones de las instrucciones avanzadas se describen los errores en tiempo de ejecución que pueden ocurrir en relación con las distintas instrucciones. Además de estos errores, también son posibles los errores comunes indicados a continuación. Cuando se ejecuta un bloque lógico y ocurre uno de los errores comunes, la CPU pasará a estado operativo STOP a menos que las instrucciones GetError o GetErrorID se hayan utilizado en ese bloque lógico para crear una reacción programada al error.

Valor del código de condición (W#16#...)	Descripción
8022	Área demasiado pequeña para la entrada
8023	Área demasiado pequeña para la salida
8024	Área de entrada no válida
8025	Área de salida no válida
8028	Asignación del bit de entrada no válida
8029	Asignación del bit de salida no válida
8030	El área de salida es un DB de sólo lectura
803A	El DB no existe

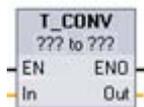
6.2.2 Instrucciones de reloj y calendario

Instrucciones de fecha y hora

Las instrucciones de fecha y hora se utilizan para programar cálculos de calendario y hora.

- T_CONV convierte el tipo de datos de un valor de hora: (Time a DInt) o (DInt a Time)
- T_ADD suma valores Time y DTL: (Time + Time = Time) o (DTL + Time = DTL)
- T_SUB resta valores Time y DTL: (Time - Time = Time) o (DTL - Time = DTL)
- T_DIFF provee la diferencia entre dos valores DTL como valor Time: DTL - DTL = Time

Tipo de datos	Tamaño (bits)	Rangos válidos
Time	32 Almacenado como	T#-24d_20h_31m_23s_648ms hasta T#24d_20h_31m_23s_647ms -2.147.483.648 ms hasta +2.147.483.647 ms
Estructura de datos DTL		
Año: UInt	16	1970 a 2554
Mes: UInt	8	1 a 12
Día: UInt	8	1 a 31
Día de la semana: UInt	8	1=domingo a 7=sábado
Hora: UInt	8	0 a 23
Minuto: UInt	8	0 a 59
Segundo: UInt	8	0 a 59
Nanosegundos: UInt	32	0 a 999.999.999



T_CONV (Convertir tiempos) convierte un tipo de datos Time a un tipo de datos DInt, o viceversa.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN	IN	DInt, Time	Entrada que indica el valor Time o Dint
OUT	OUT	DInt, Time	Valor Dint o Time convertido

Seleccione los tipos de datos de IN y OUT en las listas desplegables debajo del nombre de la instrucción.



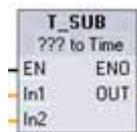
T_ADD (Sumar tiempos) suma el valor de la entrada IN1 (tipos de datos DTL o Time) al valor Time de la entrada IN2. El valor DTL o Time resultante se deposita en el parámetro OUT.

Son posibles dos operaciones con estos tipos de datos, a saber:

- Time + Time = Time
- DTL + Time = DTL

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN1	IN	DTL, Time	Valor DTL o Time
IN2	IN	Time	Valor Time que debe sumarse
OUT	OUT	DTL, Time	Suma de DTL o Time

Seleccione el tipo de datos de IN1 en la lista desplegable debajo del nombre de la instrucción. La selección del tipo de datos de IN1 ajusta también el tipo de datos del parámetro OUT.



T_SUB (Restar tiempos) resta el valor Time de IN2 del valor de IN1 (valor DTL o Time). La diferencia se deposita como tipo de datos DTL o Time en el parámetro OUT.

Son posibles dos operaciones con estos tipos de datos, a saber:

- Time - Time = Time
- DTL - Time = DTL

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN1	IN	DTL, Time	Valor DTL o Time
IN2	IN	Time	Valor Time que debe restarse
OUT	OUT	DTL, Time	Diferencia de DTL o Time

Seleccione el tipo de datos de IN1 en la lista desplegable debajo del nombre de la instrucción. La selección del tipo de datos de IN1 ajusta también el tipo de datos del parámetro OUT.



T_DIFF (Diferencia de hora) resta el valor DTL de IN2 del valor DTL de IN1. La diferencia se deposita como tipo de datos Time en el parámetro OUT.

- DTL - DTL = Time

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN1	IN	DTL	Valor DTL
IN2	IN	DTL	Valor DTL que debe restarse
OUT	OUT	Time	Diferencia de hora

Códigos de condición: ENO = 1 significa que no ha ocurrido ningún error. ENO = 0 y parámetro OUT = 0 errores:

- Valor DTL no válido
- Valor Time no válido

Instrucciones de reloj

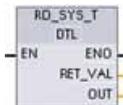
Las instrucciones de reloj sirven para ajustar y leer el reloj del sistema PLC. El tipo de datos DTL pone a disposición valores de fecha y hora.

Estructura de DTL	Tamaño	Rangos válidos
Año: UInt	16 bits	1970 a 2554
Mes: UInt	8 bits	1 a 12
Día: UInt	8 bits	1 a 31
Día de la semana: UInt	8 bits	1=domingo a 7=sábado
Hora: UInt	8 bits	0 a 23
Minuto: UInt	8 bits	0 a 59
Segundo: UInt	8 bits	0 a 59
Nanosegundos: UInt	32 bits	0 a 999.999.999



La instrucción WR_SYS_T (Escribir hora del sistema) ajusta la hora del reloj del PLC a un valor DTL en el parámetro IN. Este valor de hora no incluye la diferencia con respecto a la hora local ni tampoco al horario de verano.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN	IN	DTL	Hora que debe ajustarse en el reloj del sistema PLC
RET_VAL	OUT	Int	Código de condición de ejecución



La instrucción RD_SYS_T (Leer hora del sistema) lee la hora del sistema actual del PLC. Este valor de hora no incluye la diferencia con respecto a la hora local ni tampoco al horario de verano.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
RET_VAL	OUT	Int	Código de condición de ejecución
OUT	OUT	DTL	Hora del sistema PLC actual



RD_LOC_T (Leer hora local) lee la hora local actual del PLC como tipo de datos DTL.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
RET_VAL	OUT	Int	Código de condición de ejecución
OUT	OUT	DTL	Hora local

- La hora local se calcula utilizando la diferencia con respecto a la zona horaria y al horario de verano que se ajusta en la "Configuración de dispositivos" del reloj del PLC.
- La configuración de la zona horaria es un offset de la hora del sistema UTC (Coordinated Universal Time u Hora Universal Coordinada).
- La configuración del horario de verano especifica el mes, semana, día y hora de comienzo del horario de verano.
- La configuración del horario estándar especifica el mes, semana, día y hora de comienzo del horario estándar.
- La diferencia con respecto a la zona horaria se aplica siempre al valor de hora del sistema. La diferencia con respecto al horario de verano se aplica únicamente si el horario de verano está en vigor.

Códigos de condición: ENO = 1 significa que no ha ocurrido ningún error. ENO = 0 significa que ha ocurrido un error de ejecución. El código de condición se indica en la salida RET_VAL.

RET_VAL (W#16#...)	Descripción
0000	Sin error
8080	Hora local no disponible
8081	Valor de año no válido
8082	Valor de mes no válido
8083	Valor de día no válido
8084	Valor de hora no válido
8085	Valor de minuto no válido
8086	Valor de segundo no válido
8087	Valor de nanosegundo no válido
80B0	Ha fallado el reloj de tiempo real

6.2.3 Instrucciones con cadenas y caracteres

6.2.3.1 Sinopsis del tipo de datos String

Tipo de datos STRING

Los datos String se almacenan como encabezado de 2 bytes seguido de 254 bytes de caracteres en código ASCII. Un encabezado String contiene dos longitudes. El primer byte contiene la longitud máxima que se indica entre corchetes cuando se inicializa una cadena o 254 (ajuste predeterminado). El segundo byte del encabezado es la longitud actual, es decir, el número de caracteres válidos de la cadena. La longitud actual debe ser menor o igual a la longitud máxima. El número de bytes almacenados que ocupa el formato String es 2 bytes mayor que la longitud máxima.

Inicialización de los datos String

Los datos de entrada y salida String deben inicializarse como cadenas válidas en la memoria antes de ejecutar cualquier instrucción con cadenas.

Datos String válidos

Una cadena válida tiene una longitud máxima que debe ser mayor que cero pero menor que 255. La longitud actual debe ser menor o igual a la longitud máxima.

Las cadenas no pueden asignarse a áreas de memoria I ni Q.

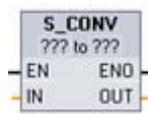
Encontrará más información en: Formato del tipo de datos String (Página 65)

6.2.3.2 Instrucciones de conversión de cadenas

Conversión de cadenas en valores y de valores en cadenas

Las siguientes instrucciones permiten convertir cadenas de caracteres numéricos en valores numéricos, y viceversa:

- S_CONV convierte una cadena numérica en un valor numérico, o viceversa.
- STRG_VAL convierte una cadena numérica en un valor numérico con opciones de formato.
- VAL_STRG convierte un valor numérico en una cadena numérica con opciones de formato.



S_CONV (Convertir cadena de caracteres) convierte una cadena de caracteres en un valor correspondiente, o viceversa. La instrucción S_CONV no tiene opciones de formato de salida. Gracias a ello, la instrucción S_CONV es más simple pero menos flexible que las instrucciones STRG_VAL y VAL_STRG.

Seleccione los tipos de datos de los parámetros en las listas desplegables.

S_CONV (Convertir cadena en valor)

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN	IN	String	Entrada que contiene la cadena de caracteres
OUT	OUT	String, SInt, Int, DInt, USInt, UInt, UDInt, Real	Valor numérico resultante

La conversión del parámetro IN que contiene la cadena comienza en el primer carácter y continúa hasta el final de la cadena, o bien hasta que se encuentra el primer carácter que no sea "0" a "9", "+", "-" o ".". El resultado se deposita en la ubicación que indica el parámetro OUT. Si el valor numérico resultante no está comprendido en el rango del tipo de datos de OUT, el parámetro OUT se pone a 0 y ENO adopta el estado lógico FALSE (falso). De lo contrario, el parámetro OUT contendrá un resultado válido y ENO adoptará el estado lógico TRUE (verdadero).

Reglas de formato de las cadenas de entrada:

- Si se utiliza un punto decimal en la cadena IN, es preciso utilizar el carácter ".".
- Las comas "," utilizadas como separadores de miles a la izquierda del punto decimal están permitidas, aunque se ignoran.
- Los espacios iniciales se ignoran.
- Sólo se soporta la representación en coma fija. Los caracteres "e" y "E" no se reconocen como notación exponencial.

S_CONV (Convertir valor en cadena)

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN	IN	String, SInt, Int, DInt, USInt, UInt, UDInt, Real	Entrada que contiene el valor numérico
OUT	OUT	String	Cadena de caracteres resultante

Un valor entero, entero sin signo o en coma flotante de la entrada IN se convierte en una cadena de caracteres correspondiente que se deposita en OUT. El parámetro OUT debe referenciar una cadena válida antes de ejecutarse la conversión. Una cadena válida consta de una longitud de cadena máxima en el primer byte, la longitud de cadena actual en el segundo byte y los caracteres de la cadena actuales en los siguientes bytes. La cadena convertida sustituye los caracteres de la cadena OUT comenzando en el primer carácter y ajusta el byte de longitud actual de la cadena OUT. El byte de longitud máxima de la cadena OUT no se modifica.

El número de caracteres que se sustituyen depende del tipo de datos del parámetro IN y del valor numérico. El número de caracteres sustituidos no debe exceder la longitud de la cadena depositada en el parámetro OUT. La longitud máxima (primer byte) de la cadena OUT debe ser mayor o igual al número máximo esperado de caracteres convertidos.

La tabla siguiente muestra las longitudes de cadena máximas posibles requeridas para los distintos tipos de datos soportados.

Tipo de datos IN	Número máximo de caracteres convertidos en la cadena OUT	Ejemplo	Longitud total de la cadena incluyendo los bytes de longitud máxima y actual
USInt	3	255	5
SInt	4	-128	6
UInt	5	65535	7
Int	6	-32768	8
UDInt	10	4294967295	12
DInt	11	-2147483648	13

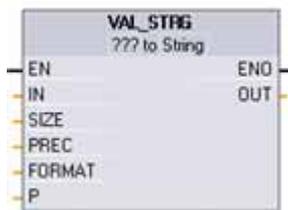
Reglas de formato de las cadenas de salida:

- Los valores que se escriben en el parámetro OUT aparecen sin signo "+" inicial.
- Se utiliza la representación en coma fija (no la notación exponencial).
- El carácter de punto "." se utiliza para representar el punto decimal si el parámetro IN tiene el tipo de datos Real.

Reglas para la conversión STRG_VAL:

- Si el carácter de punto "." se utiliza como punto decimal, las comas "," a la izquierda del punto decimal se interpretan como separadores de miles. Las comas están permitidas, aunque se ignoran.
- Si el carácter de coma "," se utiliza como punto decimal, los puntos "." a la izquierda del punto decimal se interpretan como separadores de miles. Estos puntos están permitidos, aunque se ignoran.
- Los espacios iniciales se ignoran.

Instrucción VAL_STRG



VAL_STRG (Convertir valor en cadena) convierte un valor entero, entero sin signo o en coma flotante en la cadena de caracteres correspondiente. El valor que indica el parámetro IN se convierte a una cadena referenciada por el parámetro OUT. El parámetro OUT debe ser una cadena válida antes de ejecutar la conversión.

La cadena convertida sustituye los caracteres en la cadena OUT, comenzando en el conteo de offset de carácter P hasta el número de caracteres que indica el parámetro SIZE. El número de caracteres de SIZE no debe exceder la longitud de la cadena depositada en el parámetro OUT, contando a partir de la posición de carácter P. Esta instrucción permite incorporar caracteres numéricos a una cadena de texto. Por ejemplo, la cifra "120" puede incorporarse a la cadena "Presión bomba = 120 psi".

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN	IN	SInt, Int, DInt, USInt, UInt, UDInt, Real	Valor que debe convertirse
SIZE	IN	USInt	Número de caracteres que deben escribirse en la cadena OUT
PREC	IN	USInt	Precisión o tamaño de la parte fraccionaria. Esto no incluye el punto decimal.
FORMAT	IN	Word	Opciones de formato de salida
P	IN_OUT	UInt	IN: Índice al primer carácter de la cadena OUT que debe convertirse (primer carácter = 1) OUT: Índice al siguiente carácter de la cadena OUT tras la sustitución
OUT	OUT	String	Cadena convertida

El parámetro PREC indica la precisión o el número de dígitos de la parte fraccionaria de la cadena. Si el parámetro IN es un número entero, PREC indica la posición del punto decimal. Por ejemplo, si el valor es 123 y PREC = 1, el resultado es "12.3". La precisión máxima soportada para el tipo de datos REAL es 7 dígitos.

Si el parámetro P es mayor que el tamaño actual de la cadena OUT, se agregan espacios hasta la posición P y el resultado se añade al final de la cadena. La conversión finaliza si se alcanza la longitud máxima de la cadena OUT.

Parámetro FORMAT de VAL_STRG

El parámetro FORMAT de la instrucción VAL_STRG se define a continuación. Los bits no utilizados deben ponerse a cero.

Bit 16								Bit 8	Bit 7										Bit 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	s	f	r		

- | | |
|------------------------------|------------------------------|
| s = Carácter de signo | 1= usar los signos "+" y "-" |
| | 0 = usar sólo el signo "-" |
| f = Formato de notación | 1= Notación exponencial |
| | 0 = Notación en coma fija |
| r = Formato de punto decimal | 1 = "," (coma) |
| | 0 = "." (punto) |

FORMAT (WORD)	Carácter de signo	Formato de notación	Representación del punto decimal
W#16#0000	Sólo "-"	Coma fija	" . "
W#16#0001			" " "
W#16#0002		Exponencial	" . "
W#16#0003			" " "
W#16#0004	"+" y "-"	Coma fija	" . "
W#16#0005			" " "
W#16#0006		Exponencial	" . "
W#16#0007			" " "
W#16#0008 a W#16#FFFF	Valores no válidos		

Reglas de formato de la cadena del parámetro OUT:

- Si el tamaño de la cadena convertida es menor que el especificado, se insertan espacios en blanco iniciales en el extremo izquierdo de la cadena.
- Si el bit de signo del parámetro FORMAT es FALSE (falso), los valores de los tipos de datos USINT y SINT se escriben en el búfer de salida sin el signo "+" inicial. El "-" signo se utiliza en caso necesario.
<espacios iniciales><dígitos sin ceros iniciales>'.'<dígitos PREC>
- Si el bit de signo es TRUE (verdadero), los valores de los tipos de datos USINT y SINT se escriben en el búfer de salida siempre con un carácter de signo inicial.
<espacios iniciales><signo><dígitos sin ceros iniciales>'.'<dígitos PREC>
- Si el parámetro FORMAT está ajustado a notación exponencial, los valores del tipo de datos REAL se escriben en el búfer de salida de la siguiente manera:
<espacios iniciales><signo><dígito> '.' <dígitos PREC>'E' <signo><dígitos sin cero inicial>
- Si el parámetro FORMAT está ajustado a notación en coma fija, los valores de los tipos de datos INT, USINT y REAL se escriben en el búfer de salida de la siguiente manera:
<espacios iniciales><signo><dígitos sin ceros iniciales>'.'<dígitos PREC>
- Los ceros a la izquierda del punto decimal (con excepción del dígito adyacente a éste) se suprimen.
- Los valores a la derecha del punto decimal se redondean para que se correspondan con el número de dígitos a la derecha del punto decimal que indica el parámetro PREC.
- La cadena de salida debe ser como mínimo tres bytes más grande que el número de dígitos a la derecha del punto decimal.
- Los valores se justifican a la derecha en la cadena de salida.

Condiciones que notifica ENO

Si ocurre un error durante la operación de conversión, se devolverán los siguientes resultados:

- ENO se pone a 0.
- OUT se pone a 0 o como se muestra en los ejemplos de conversión de una cadena en un valor.
- OUT no se modifica o como se muestra en los ejemplos cuando OUT es una cadena.

Estado de ENO	Descripción
1	Sin error
0	Parámetro no permitido o no válido; p. ej. acceso a un DB que no existe
0	Cadena no permitida; la longitud máxima es 0 ó 255
0	Cadena no permitida; la longitud actual excede la longitud máxima
0	El valor numérico convertido es demasiado grande para el tipo de datos de OUT indicado
0	El tamaño máximo de la cadena del parámetro OUT debe ser lo suficientemente grande para aceptar el número de caracteres que indica el parámetro SIZE, comenzando en el parámetro P de posición de carácter.
0	Valor de P no permitido; P=0 o P es mayor que la longitud actual de la cadena
0	El parámetro SIZE debe ser mayor que el parámetro PREC

Ejemplos de S_CONV (Convertir cadena en valor)

Cadena IN	Tipo de datos de OUT	Valor de OUT	ENO
"123"	Int/DInt	123	TRUE
"-00456"	Int/DInt	-456	TRUE
"123.45"	Int/DInt	123	TRUE
"+2345"	Int/DInt	2345	TRUE
"00123AB"	Int/DInt	123	TRUE
"123"	Real	123.0	TRUE
"123.45"	Real	123.45	TRUE
"1.23e-4"	Real	1.23	TRUE
"1.23E-4"	Real	1.23	TRUE
"12,345.67"	Real	12345.67	TRUE
"3.4e39"	Real	3.4	TRUE
"-3.4e39"	Real	-3.4	TRUE
"1.17549e-38"	Real	1.17549	TRUE
"12345"	SInt	0	FALSE
"A123"	N/A	0	FALSE
""	N/A	0	FALSE
"++123"	N/A	0	FALSE
"+-123"	N/A	0	FALSE

Ejemplos de S_CONV (Convertir valor en cadena)

Tipo de datos	Valor de IN	Cadena OUT	ENO
UInt	123	"123"	TRUE
UInt	0	"0"	TRUE
UDInt	12345678	"12345678"	TRUE
Real	-INF	"INF"	FALSE
Real	+INF	"INF"	FALSE
Real	NaN	"NaN"	FALSE

Ejemplos de conversión STRG_VAL

Cadena IN	FORMAT (W#16#...)	Tipo de datos de OUT	Valor de OUT	ENO
"123"	0000	Int/DInt	123	TRUE
"-00456"	0000	Int/DInt	-456	TRUE
"123.45"	0000	Int/DInt	123	TRUE
" +2345"	0000	Int/DInt	2345	TRUE
"00123AB"	0000	Int/DInt	123	TRUE
"123"	0000	Real	123.0	TRUE
"-00456"	0001	Real	-456.0	TRUE
" +00456"	0001	Real	456.0	TRUE
"123.45"	0000	Real	123.45	TRUE
"123.45"	0001	Real	12345.0	TRUE
"123,45"	0000	Real	12345.0	TRUE
"123,45"	0001	Real	123.45	TRUE
".00123AB"	0001	Real	123.0	TRUE
"1.23e-4"	0000	Real	1.23	TRUE
"1.23E-4"	0000	Real	1.23	TRUE
"1.23E-4"	0002	Real	1.23E-4	TRUE
"12,345.67"	0000	Real	12345.67	TRUE
"12,345.67"	0001	Real	12.345	TRUE
"3.4e39"	0002	Real	+INF	TRUE
"-3.4e39"	0002	Real	-INF	TRUE
"1.1754943e-38" (o menor)	0002	Real	0.0	TRUE
"12345"	N/A	SInt	0	FALSE
"A123"	N/A	N/A	0	FALSE
""	N/A	N/A	0	FALSE
"++123"	N/A	N/A	0	FALSE
"+-123"	N/A	N/A	0	FALSE

Ejemplos de conversión VAL_STRG

Los ejemplos se basan en una cadena OUT que se inicializa de la manera siguiente:

"Current Temp = xxxxxxxxxxx C"

El carácter "x" representa caracteres de espacio asignados al valor convertido.

Tipo de datos	Valor de IN	P	SIZE	FORMAT (W#16#....)	PREC	Cadena OUT	ENO
UInt	123	16	10	0000	0	Current Temp = xxxxxxxx123 C	TRUE
UInt	0	16	10	0000	2	Current Temp = xxxxxx0.00 C	TRUE
UDInt	12345678	16	10	0000	3	Current Temp = x12345.678 C	TRUE
UDInt	12345678	16	10	0001	3	Current Temp = x12345,678 C	TRUE
Int	123	16	10	0004	0	Current Temp = xxxxxx+123 C	TRUE
Int	-123	16	10	0004	0	Current Temp = xxxxxx-123 C	TRUE
Real	-0.00123	16	10	0004	4	Current Temp = xxx-0.0012 C	TRUE
Real	-0.00123	16	10	0006	4	Current Temp = -1.2300E-3 C	TRUE
Real	-INF	16	10	N/A	4	Current Temp = xxxxxx-INF C	FALSE
Real	+INF	16	10	N/A	4	Current Temp = xxxxxx+INF C	FALSE
Real	NaN	16	10	N/A	4	Current Temp = xxxxxxxNaN C	FALSE
UDInt	12345678	16	6	N/A	3	Current Temp = xxxxxxxxxxx C	FALSE

6.2.3.3 Instrucciones con cadenas

El programa de control puede utilizar las siguientes instrucciones con cadenas y caracteres para crear avisos para displays de operador e históricos del proceso.

Errores comunes de todas las operaciones String

Las operaciones String que se ejecutan con las condiciones String no permitidas o no válidas indicadas a continuación arrojan como resultado ENO = 0 y una salida string null. Las condiciones de error que ocurren para una instrucción en particular se indican debajo de la descripción de la operación.

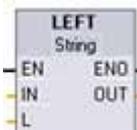
ENO	Condición	OUT
0	La longitud actual de IN1 excede la longitud máxima de IN1 o la longitud actual de IN2 excede la longitud máxima de IN2 (cadena no válida)	La longitud actual se pone a 0
	La longitud máxima de IN1, IN2 u OUT excede el rango de memoria asignado	
	La longitud máxima de IN1, IN2 u OUT es 0 ó 255 (longitud no permitida)	



LEN: Determinar la longitud de una cadena de caracteres



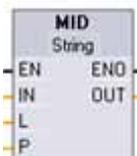
CONCAT: Concatenar dos cadenas



LEFT: Leer los caracteres izquierdos de una cadena



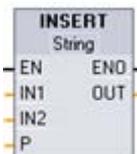
RIGHT: Leer los caracteres derechos de una cadena



MID: Leer los caracteres centrales de una cadena



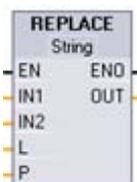
FIND: Buscar caracteres en la cadena



INSERT: Insertar caracteres en la cadena



DELETE: Borrar caracteres de la cadena



REPLACE: Reemplazar caracteres de una cadena

Instrucción LEN

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN	IN	String	Cadena de entrada
OUT	OUT	UInt	Número de caracteres válidos de la cadena IN

LEN (Determinar la longitud de una cadena de caracteres) indica la longitud actual de la cadena IN en la salida OUT. Una cadena vacía tiene una longitud cero. La tabla siguiente muestra los códigos de condición de la instrucción.

ENO	Condición	OUT
1	No hay condición de cadena no válida	Longitud de cadena válida

Instrucción CONCAT

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN1	IN	String	Cadena de entrada 1
IN2	IN	String	Cadena de entrada 2
OUT	OUT	String	Cadena combinada (cadena 1 + cadena 2)

CONCAT (Concatenar dos cadenas) combina los parámetros String de IN1 e IN2 para formar una cadena que se deposita en OUT. Después de la concatenación, la cadena IN1 es la parte izquierda y, la cadena IN2, la parte derecha de la cadena combinada. La tabla siguiente muestra los códigos de condición de la instrucción.

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	La cadena resultante tras la concatenación excede la longitud máxima de la cadena OUT	Los caracteres de la cadena resultante se copian hasta alcanzarse la longitud máxima de OUT

Instrucción LEFT

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN	IN	String	Cadena de entrada
L	IN	Int	Longitud de la subcadena que debe crearse utilizando los caracteres L en el extremo izquierdo de la cadena IN
OUT	OUT	String	Cadena de salida

LEFT (Leer los caracteres izquierdos de una cadena) crea una subcadena formada por los primeros caracteres L del parámetro de cadena IN.

- Si L es mayor que la longitud actual de la cadena IN, OUT devuelve la cadena IN completa.
- Si la entrada contiene una cadena vacía, OUT devuelve una cadena vacía.

La tabla siguiente muestra los códigos de condición de la instrucción.

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	L es menor o igual a 0	La longitud actual se pone a 0
	La longitud (L) de la subcadena que debe copiarse excede la longitud máxima de la cadena OUT	Los caracteres se copian hasta alcanzarse la longitud máxima de OUT

Instrucción RIGHT

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN	IN	String	Cadena de entrada
L	IN	Int	Longitud de la subcadena que debe crearse utilizando los caracteres L en el extremo derecho de la cadena IN
OUT	OUT	String	Cadena de salida

RIGHT (Leer los caracteres derechos de una cadena) provee los últimos caracteres L de una cadena.

- Si L es mayor que la longitud actual de la cadena IN, el parámetro OUT devuelve la cadena IN completa.
- Si la entrada contiene una cadena vacía, OUT devuelve una cadena vacía.

La tabla siguiente muestra los códigos de condición de la instrucción.

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	L es menor o igual a 0	La longitud actual se pone a 0
	La longitud (L) de la subcadena que debe copiarse excede la longitud máxima de la cadena OUT	Los caracteres se copian hasta alcanzarse la longitud máxima de OUT

Instrucción MID

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN	IN	String	Cadena de entrada
L	IN	Int	Longitud de la subcadena que debe crearse utilizando caracteres L de la cadena IN a partir de la posición de carácter P
P	IN	Int	Posición del primer carácter de la subcadena que debe copiarse: P= 1 si debe ser el carácter inicial de la cadena IN
OUT	OUT	String	Cadena de salida

MID (Leer los caracteres centrales de una cadena) provee la parte central de una cadena. La subcadena central tiene una longitud de L caracteres y comienza en la posición de carácter P (inclusive).

Si la suma de L y P excede la longitud actual del parámetro String de IN, se devuelve una subcadena que comienza en la posición de carácter P y que continúa hasta el final de la cadena IN. La tabla siguiente muestra los códigos de condición de la instrucción.

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	L o P es menor o igual a 0	La longitud actual se pone a 0
	P excede la longitud máxima de IN	
	La longitud (L) de la subcadena que debe copiarse excede la longitud máxima de la cadena OUT	Los caracteres se copian, comenzando en la posición P, hasta alcanzarse la longitud máxima de OUT

Instrucción DELETE

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN	IN	String	Cadena de entrada
L	IN	Int	Número de caracteres que deben borrarse
P	IN	Int	Posición del primer carácter que debe borrarse: El primer carácter de la cadena IN tiene el número de posición 1
OUT	OUT	String	Cadena de salida

DELETE (Borrar caracteres de la cadena) borra L caracteres de la cadena IN. El borrado de caracteres comienza en la posición de carácter P (inclusive) y la subcadena restante se deposita en el parámetro OUT.

- Si L es igual a cero, OUT devuelve la cadena de entrada.
- Si la suma de L y P es mayor que la longitud de la cadena de entrada, la cadena se borra hasta el final.

La tabla siguiente muestra los códigos de condición de la instrucción.

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	P excede la longitud actual de IN	IN se copia en OUT sin borrarse caracteres
	L es menor que 0 o P es menor o igual a 0	La longitud actual se pone a 0
	La cadena resultante tras borrar los caracteres excede la longitud máxima de la cadena OUT	Los caracteres de la cadena resultante se copian hasta alcanzarse la longitud máxima de OUT

INSERT

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN1	IN	String	Cadena de entrada 1
IN2	IN	String	Cadena de entrada 2
P	IN	Int	Última posición de carácter en la cadena IN1 antes del punto de inserción de la cadena IN2. El primer carácter de la cadena IN1 tiene el número de posición 1.
OUT	OUT	String	Cadena resultante

INSERT (Insertar caracteres en la cadena) inserta la cadena IN2 en la cadena IN1. La inserción comienza tras el carácter de la posición P. La tabla siguiente muestra los códigos de condición de la instrucción.

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	P excede la longitud de IN1	IN2 se concatena con IN1 inmediatamente después del último carácter de IN1
	P es menor o igual a 0	La longitud actual se pone a 0
	La cadena resultante tras la inserción excede la longitud máxima de la cadena OUT	Los caracteres de la cadena resultante se copian hasta alcanzarse la longitud máxima de OUT

REPLACE

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN1	IN	String	Cadena de entrada
IN2	IN	String	Cadena de caracteres de reemplazo
L	IN	Int	Número de caracteres que deben reemplazarse
P	IN	Int	Posición del primer carácter que debe reemplazarse
OUT	OUT	String	Cadena resultante

REPLACE (Reemplazar caracteres de una cadena) reemplaza caracteres L de la cadena IN1. La sustitución comienza en la posición de carácter P (inclusive) de la cadena IN1. Los caracteres de reemplazo provienen de la cadena IN2.

- Si el parámetro L es igual a cero, la cadena IN2 se inserta en la posición P de la cadena IN1 sin que se borre ningún carácter de la cadena IN1.
- Si P es igual a uno, los primeros caracteres L de la cadena IN1 se reemplazan por caracteres de la cadena IN2.

La tabla siguiente muestra los códigos de condición de la instrucción.

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	P excede la longitud de IN1	IN2 se concatena con IN1 inmediatamente después del último carácter de IN1
	P cabe en IN1, pero menos de L caracteres permanecen en IN1	IN2 reemplaza los caracteres finales de IN1 comenzando en la posición P
	L es menor que 0 o P es menor o igual a 0	La longitud actual se pone a 0
	La cadena resultante tras la sustitución excede la longitud máxima de la cadena OUT	Los caracteres de la cadena resultante se copian hasta alcanzarse la longitud máxima de OUT

FIND

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
IN1	IN	String	Buscar en esta cadena
IN2	IN	String	Buscar esta cadena
OUT	OUT	Int	Posición de carácter de la cadena IN1, primer resultado de búsqueda

FIND (Buscar caracteres en la cadena) busca la posición de carácter de la subcadena o el carácter que indica IN2 en la cadena IN1. La búsqueda comienza en el lado izquierdo. La posición de carácter del primer resultado encontrado en la cadena IN2 se devuelve en OUT. Si la cadena IN2 no se encuentra en la cadena IN1, se devuelve cero. La tabla siguiente muestra los códigos de condición de la instrucción.

ENO	Condición	OUT
1	No se han detectado errores	Posición de carácter válida
0	IN2 es mayor que IN1	La posición de carácter se pone a 0

6.2.4 Instrucciones de control del programa

6.2.4.1 Instrucción "Reiniciar la vigilancia del tiempo de ciclo"



La instrucción RE_TRIGR (Reiniciar la vigilancia del tiempo de ciclo) sirve para prolongar el tiempo máximo permitido antes de que el temporizador de vigilancia del ciclo genere un error.

La instrucción RE_TRIGR se utiliza para reiniciar el temporizador de vigilancia del ciclo durante un único ciclo. De esta manera, el tiempo de ciclo máximo se prolonga un periodo de tiempo de ciclo máximo desde la última ejecución de la función RE_TRIGR.

La CPU restringe el uso de la instrucción RE_TRIGR al ciclo del programa, p. ej. al OB1 y las funciones llamadas desde el ciclo del programa. Esto significa que se inicializa el temporizador de vigilancia y ENO = EN si RE_TRIGR se llama desde cualquier OB de la lista de OBs de ciclo.

ENO = FALSE y el temporizador de vigilancia no se inicializa si RE_TRIGR se ejecuta desde un OB de arranque, de alarma o de error.

Ajustar el tiempo de ciclo máximo del PLC

El tiempo de ciclo máximo puede ajustarse en la configuración de dispositivos PLC en "Tiempo de ciclo".

Vigilancia del tiempo de ciclo	Valor mínimo	Valor máximo	Valor predeterminado
Tiempo de ciclo máximo	1 ms	6000 ms	150 ms

Timeout de vigilancia

Si el tiempo de ciclo máximo finaliza antes de haberse completado el ciclo, se generará un error. Si el OB 80 (bloque lógico de tratamiento de errores) se incluye en el programa de usuario, el PLC lo ejecuta. En el OB 80 es posible agregar lógica para crear una reacción especial. Si no se incluye el OB 80, se ignora la primera condición de timeout.

Si ocurre un segundo timeout de tiempo de ciclo máximo en ese mismo ciclo del programa (valor del tiempo de ciclo máximo multiplicado por 2), se disparará un error y el PLC cambiará a estado operativo STOP.

En el estado operativo STOP se detiene la ejecución del programa mientras continúan las comunicaciones y los diagnósticos del sistema PLC.

6.2.4.2 Instrucción "Parar ciclo del PLC"



La instrucción STP (Parar ciclo del PLC) cambia el PLC a estado operativo STOP. Cuando el PLC está en estado operativo STOP, se detienen la ejecución del programa de usuario y las actualizaciones físicas desde la memoria imagen de proceso.

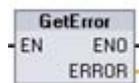
Encontrará más información en: Configurar las salidas en una transición de RUN a STOP (Página 54)

Si EN = TRUE (verdadero), el PLC pasa a estado operativo STOP, se detiene la ejecución del programa y el estado de ENO carece de importancia. De lo contrario, EN = ENO = 0.

6.2.4.3 Instrucciones GET_ERROR

Las instrucciones GET_ERROR proporcionan información acerca de errores de ejecución de bloques de programa. Si se ha insertado una instrucción GetError o GetErrorID en el bloque lógico, los errores del programa podrán tratarse en el bloque de programa.

GET_ERROR



GET_ERROR indica que ha ocurrido un error de ejecución de un bloque y rellena una estructura de datos de error predefinida con información detallada acerca del error.

Parámetro	Tipo de datos	Descripción
ERROR	ErrorStruct	Estructura de datos del error: Es posible cambiar el nombre de la estructura pero no sus elementos.

Elemento de datos ErrorStruct	Tipo de datos	Descripción
ERROR_ID	Word	Identificador del error
FLAGS	Byte	Siempre puesto a 0.
REACTION	Byte	Reacción al error: <ul style="list-style-type: none"> • 0 = Ignorar; no se ha escrito nada (error de escritura) • 1 = Sustituir: se ha utilizado 0 para el valor de entrada (error de lectura) • 2 = Omitir la instrucción
BLOCK_TYPE	Byte	Tipo de bloque en el que ha ocurrido el error: <ul style="list-style-type: none"> • 1 = OB • 2 = FC • 3 = FB
PAD_0	Byte	Byte de relleno interno para la alineación; será 0
CODE_BLOCK_NUMBER	UInt	Número de bloque en el que ha ocurrido el error

Elemento de datos ErrorStruct	Tipo de datos	Descripción
ADDRESS	UDInt	Posición de memoria interna de la instrucción en la que ha ocurrido el error
MODE	Byte	Mapeo interno de cómo se interpretarán los campos restantes que debe utilizar STEP 7 Basic
PAD_1	Byte	Byte de relleno interno para la alineación; no se utiliza, será 0
OPERAND_NUMBER	UInt	Número de operando de la instrucción interna
POINTER_NUMBER_LOCATION	UInt	(A) Ubicación del puntero de la instrucción interna
SLOT_NUMBER_SCOPE	UInt	(B) Ubicación de almacenamiento en la memoria interna
AREA	Byte	(C) Área de memoria referenciada cuando ocurrió el error: <ul style="list-style-type: none"> • L: 16#40 – 4E, 86, 87, 8E, 8F, C0 – CE • I: 16#81 • Q: 16#82 • M: 16#83 • DB: 16#84, 85, 8A, 8B
PAD_2	Byte	Byte de relleno interno para la alineación; no se utiliza, será 0
DB_NUMBER	UInt	(D) DB referenciado cuando ocurrió el error de DB, 0 en los demás casos
OFFSET	UDInt	(E) Offset de bit referenciado cuando ocurrió el error (ejemplo: 12 = byte 1, bit 4)

GET_ERR_ID



GET_ERR_ID indica que ha ocurrido un error de ejecución de bloque de programa y notifica la ID (identificación) del error.

Parámetro	Tipo de datos	Descripción
ID	Word	Valores de identificador del error del elemento ERROR_ID de ErrorStruct

ERROR_ID hexadecimal	ERROR_ID decimal	Error de ejecución de bloque de programa
2503	9475	Error de puntero no inicializado
2522	9506	Error de lectura de operando fuera de rango
2523	9507	Error de escritura de operando fuera de rango
2524	9508	Error de lectura de área no válida
2525	9509	Error de escritura de área no válida
2528	9512	Error de lectura de alineación de datos (alineación de bit incorrecta)
2529	9513	Error de escritura de alineación de datos (alineación de bit incorrecta)

ERROR_ID hexadecimal	ERROR_ID decimal	Error de ejecución de bloque de programa
2530	9520	DB protegido contra escritura
253A	9530	El DB global no existe
253C	9532	Versión incorrecta o la FC no existe
253D	9533	La instrucción no existe
253E	9534	Versión incorrecta o el FB no existe
253F	9535	La instrucción no existe
2575	9589	Error de profundidad de anidamiento del programa
2576	9590	Error de asignación de datos locales
2942	10562	La entrada física no existe
2943	10563	La salida física no existe

Funcionamiento

De forma predeterminada, la CPU reacciona a un error de ejecución de bloque registrando un error en el búfer de diagnóstico y cambiando a estado operativo STOP. No obstante, si se insertan una o más instrucciones GET_ERROR o GET_ERR_ID en un bloque lógico, éste puede tratar los errores en el bloque. En este caso, la CPU no cambia a estado operativo STOP y tampoco registra un error en el búfer de diagnóstico. En su lugar, la información de error se deposita en la salida de la instrucción GET_ERROR o GET_ERR_ID. Es posible leer la información de error detallada con la instrucción GET_ERROR, o bien sólo el identificador del error con la instrucción GET_ERR_ID. Normalmente, el primer error es el más importante; los errores siguientes son sólo consecuencias del primer error.

La primera ejecución de una instrucción GET_ERROR o GET_ERR_ID en un bloque devuelve el primer error detectado durante la ejecución del bloque. Este error puede haber ocurrido en cualquier punto entre el inicio del bloque y la ejecución de la instrucción GET_ERROR o GET_ERR_ID. Las ejecuciones posteriores de GET_ERROR o GET_ERR_ID devuelven el primer error desde la ejecución anterior de GET_ERROR o GET_ERR_ID. El historial de errores no se almacena y la ejecución de cualquiera de estas instrucciones rearmará el sistema PLC de manera que pueda detectar el error siguiente.

El tipo de datos ErrorStruct que utiliza la instrucción GET_ERROR puede agregarse en el editor del bloque de datos y en la interfaz del bloque, de manera que la lógica del programa pueda acceder a estos valores. Seleccione ErrorStruct en la lista desplegable de tipos de datos para agregar esta estructura. Es posible crear varios ErrorStructs utilizando nombres unívocos. No es posible cambiar el nombre de los elementos de un ErrorStruct.

Condición de error indicada por ENO

Si EN = TRUE y se ejecuta GET_ERROR o GET_ERR_ID, entonces:

- ENO = TRUE indica que ha ocurrido un error de ejecución del bloque lógico y que hay un error de datos
- ENO = FALSE indica que no ha ocurrido ningún error de ejecución del bloque lógico

Es posible conectar la lógica del programa de reacción a errores a ENO que se activa cuando ocurre un error. Si existe un error, sus datos se almacenan en el parámetro de salida, donde el programa puede acceder a ellos.

GET_ERROR y GET_ERR_ID pueden utilizarse para enviar información de error desde el bloque que se está ejecutando (bloque llamado) a un bloque invocante. Coloque la instrucción en el último segmento del bloque de programa llamado para notificar el estado de ejecución final del bloque llamado.

6.2.5 Instrucciones de comunicación

6.2.5.1 Comunicación Ethernet abierta

Comunicación Ethernet abierta con conexión/desconexión automática (TSEND_C y TRCV_C)

Nota

El procesamiento de las instrucciones TSEND_C y TRCV_C puede tardar un tiempo indeterminado. Para garantizar que estas instrucciones se procesen en cada ciclo, es preciso llamarlas siempre desde el ciclo del programa principal, p. ej. desde un OB de ciclo o un bloque lógico llamado desde el ciclo del programa. **No** llame estas instrucciones desde un OB de alarma de proceso, OB de alarma de retardo, OB de alarma cíclica, OB de error de tiempo ni OB de arranque.

Encontrará más información sobre cómo transferir datos con estas instrucciones en el apartado Coherencia de datos (Página 96).

Descripción de TSEND_C

TSEND_C establece una conexión TCP o ISO on TCP con un interlocutor, envía datos y puede deshacer la conexión. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente. TSEND_C combina las funciones de TCON, TDISCON y TSEND.

El tamaño mínimo de los datos que pueden transmitirse con la instrucción TSEND_C es un byte.

Nota

El ajuste predeterminado del parámetro LEN (LEN = 0) utiliza el parámetro DATA para determinar la longitud de los datos que se están transmitiendo. Asegúrese de que los datos (DATA) transmitidos por la instrucción TSEND_C tienen el mismo tamaño que el parámetro DATA de la instrucción TRCV_C.

Las funciones siguientes describen la operación de la instrucción TSEND_C:

- Para establecer una conexión, ejecute TSEND_C con CONT = 1.
- Una vez establecida correctamente la conexión, TSEND_C activa el parámetro DONE durante un ciclo.
- Para deshacer la conexión, ejecute TSEND_C con CONT = 0. La conexión se interrumpirá inmediatamente. Esto afecta también la estación receptora. La conexión se cierra allí y pueden perderse los datos del búfer de recepción.
- Para enviar datos a través de una conexión establecida, ejecute TSEND_C cuando se produzca un flanco ascendente en REQ. Tras una operación de envío correcta, TSEND_C activa el parámetro DONE durante un ciclo.
- Para establecer una conexión y enviar datos, ejecute TSEND_C con CONT = 1 y REQ = 1. Tras una operación de envío correcta, TSEND_C activa el parámetro DONE durante un ciclo.

Descripción de TRCV_C

TRCV_C establece una conexión TCP o ISO on TCP con una CPU interlocutora, recibe datos y puede deshacer la conexión. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente. La instrucción TRCV_C combina las funciones de las instrucciones TCON, TDISCON y TRCV.

El tamaño mínimo de los datos que pueden recibirse con la instrucción TRCV_C es un byte. La instrucción TRCV_C no soporta la transmisión de datos booleanos ni de matrices booleanas.

Nota

El ajuste predeterminado del parámetro LEN (LEN = 0) utiliza el parámetro DATA para determinar la longitud de los datos que se están transmitiendo. Asegúrese de que los datos (DATA) transmitidos por la instrucción TSEND_C tienen el mismo tamaño que el parámetro DATA de la instrucción TRCV_C.

Las funciones siguientes describen la operación de la instrucción TRCV_C:

- Para establecer una conexión, ejecute TRCV_C con el parámetro CONT = 1.
- Para recibir datos, ejecute TRCV_C con el parámetro EN_R = 1. TRCV_C recibe los datos continuamente si los parámetros EN_R = 1 y CONT = 1.
- Para deshacer la conexión, ejecute TRCV_C con el parámetro CONT = 0. La conexión se deshace inmediatamente y pueden perderse datos.

Modos de recepción

TRCV_C utiliza los mismos modos de recepción que la instrucción TRCV. La tabla siguiente muestra cómo se introducen los datos en el área de recepción.

Variante de protocolo	Entrada de datos en el área de recepción	Parámetro "connection_type"
TCP	Recepción de datos con la longitud especificada	B#16#11
ISO on TCP	Controlado por protocolo	B#16#12

Nota

Debido al procesamiento asíncrono de TSEND_C, es preciso conservar la coherencia de los datos en el área de emisión hasta que el parámetro DONE o ERROR adopta el valor TRUE.

Si el parámetro DONE de la instrucción TSEND_C tiene el estado TRUE, significa que los datos se han enviado correctamente. Sin embargo, no significa que la CPU interlocutora haya leído realmente el búfer de recepción.

Debido al procesamiento asíncrono de TRCV_C, los datos en el área de recepción sólo son coherentes si el parámetro DONE = 1.

La tabla siguiente muestra la relación entre los parámetros BUSY, DONE y ERROR.

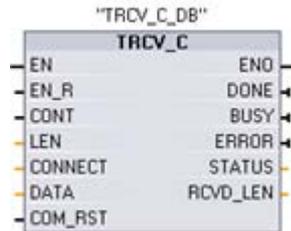
BUSY	DONE	ERROR	Descripción
TRUE	Irrelevante	Irrelevante	La tarea se está procesando.
FALSE	TRUE	FALSE	La tarea se ha ejecutado correctamente.
FALSE	FALSE	TRUE	La tarea se ha finalizado con un error. La causa del error se indica en el parámetro STATUS.
FALSE	FALSE	FALSE	No se ha asignado ninguna tarea nueva.

Parámetros de TSEND_C



Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	INPUT	Bool	El parámetro de control REQ inicia la tarea de transmisión con la conexión descrita en CONNECT cuando se detecta un flanco ascendente.
CONT	INPUT	Bool	<ul style="list-style-type: none"> 0: desconectar 1: establecer y mantener la conexión
LEN	INPUT	Int	Número máximo de bytes que deben enviarse. (El ajuste predeterminado es 0, es decir, el parámetro DATA determina la longitud de los datos por enviar).
CONNECT	IN_OUT	Parám TCON	Puntero a la descripción de la conexión
DATA	IN_OUT	Variante	Área de emisión; contiene la dirección y la longitud de los datos por enviar.
COM_RST	IN_OUT	Bool	<ul style="list-style-type: none"> 1: Reinicio completo del bloque de función; se deshace la conexión existente.
DONE	OUTPUT	Bool	<ul style="list-style-type: none"> 0: Tarea no iniciada aún o en proceso. 1: Tarea finalizada sin errores.
BUSY	OUTPUT	Bool	<ul style="list-style-type: none"> 0: Tarea finalizada. 1: Tarea no finalizada aún. No se puede iniciar una tarea nueva.
ERROR	OUTPUT	Bool	<ul style="list-style-type: none"> 1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.
STATUS	OUTPUT	Word	Información de error

Parámetros de TRCV_C



Parámetro	Tipo de parámetro	Tipo de datos	Descripción
EN_R	IN	Bool	Parámetro de control habilitado para recibir: Si EN_R = 1, TRCV_C está listo para recibir. La tarea de recepción se procesa.
CONT	IN	Bool	Parámetro de control CONT: <ul style="list-style-type: none"> 0: desconectar 1: establecer y mantener la conexión
LEN	IN	Int	Longitud del área de recepción en bytes. (El ajuste predeterminado es 0, es decir, el parámetro DATA determina la longitud de los datos por enviar).
CONNECT	IN_OUT	Parám TCON	Puntero a la descripción de la conexión
DATA	IN_OUT	Variante	El área de recepción contiene la dirección inicial y la longitud máxima de los datos recibidos.
COM_RST	IN_OUT	Bool	<ul style="list-style-type: none"> 1: Reinicio completo del bloque de función; se deshace la conexión existente.
DONE	OUT	Bool	<ul style="list-style-type: none"> 0: Tarea no iniciada aún o en proceso. 1: Tarea finalizada sin errores.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0: Tarea finalizada. 1: Tarea no finalizada aún. No se puede iniciar una tarea nueva.
ERROR	OUT	Bool	<ul style="list-style-type: none"> 1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.
STATUS	OUT	Word	Información de error
RCVD_LEN	OUT	Int	Cantidad de datos (en bytes) recibida realmente

Parámetros ERROR y STATUS

ERROR	STATUS (W#16#...)	Descripción
0	0000	Tarea ejecutada sin errores
0	7000	No se está procesando ninguna tarea
0	7001	Iniciar procesamiento de la tarea, estableciendo la conexión, esperando al interlocutor
0	7002	Enviando o recibiendo datos
0	7003	Deshaciendo la conexión
0	7004	Conexión establecida y vigilada, no se está procesando ninguna tarea
1	8085	El parámetro LEN es mayor que el valor máximo admisible
1	8086	El parámetro CONNECT está fuera del rango admisible
1	8087	Se ha alcanzado el número máximo de conexiones; no es posible establecer más conexiones
1	8088	El parámetro LEN excede el área de memoria especificada en DATA; el área de memoria para la recepción es demasiado pequeña
1	8089	El parámetro CONNECT no apunta a un bloque de datos.
1	8091	Se ha excedido la profundidad de anidamiento máxima
1	809A	El parámetro CONNECT apunta a un campo que no concuerda con la longitud de la descripción de la conexión.
1	809B	La "local_device_id" indicada en la descripción de la conexión no concuerda con la CPU.
1	80A1	Error de comunicación: <ul style="list-style-type: none"> La conexión indicada no se ha establecido todavía La conexión indicada se está deshaciendo; no es posible transferir a través de esta conexión La interfaz se está reiniciando
1	80A3	Se está intentando deshacer una conexión no existente
1	80A4	La dirección IP del interlocutor remoto de la conexión no es válida. Por ejemplo, la dirección IP del interlocutor remoto es igual a la dirección IP del interlocutor local.
1	80A7	Error de comunicación: TDISCON se ha llamado antes de finalizar TCON (TDISCON debe deshacer primero por completo la conexión referenciada por la ID)
1	80B2	El parámetro CONNECT apunta a un bloque de datos generado con la palabra clave UNLINKED
1	80B3	Parámetros incoherentes: <ul style="list-style-type: none"> Error en la descripción de la conexión El puerto local (parámetro local_tsap_id) ya existe en una descripción de conexión diferente La ID indicada en la descripción de la conexión no concuerda con la ID especificada como parámetro

ERROR	STATUS (W#16#...)	Descripción
1	80B4	Si se utiliza ISO on TCP (connection_type = B#16#12) para establecer una conexión pasiva, el código de condición 80B4 advierte de que el TSAP introducido no cumple uno de los siguientes requisitos para la dirección: <ul style="list-style-type: none"> • Si la longitud del TSAP local es 2 y el valor de la ID TSAP es E0 o E1 (hexadecimal) para el primer byte, el segundo byte deberá ser 00 ó 01. • Si la longitud del TSAP local es 3 o superior y el valor de la ID TSAP es E0 o E1 (hexadecimal) para el primer byte, el segundo byte deberá ser 00 ó 01 y todos los demás bytes deberán ser caracteres ASCII válidos. • Si la longitud del TSAP local es 3 o superior y el valor del primer byte de la ID TSAP no es E0 ni E1 (hexadecimal), todos los demás bytes de la ID TSAP deberán ser caracteres ASCII válidos. Los caracteres ASCII válidos son valores de byte comprendidos entre 20 y 7E (hexadecimal).
1	80C3	Se están utilizando todos los recursos de conexión.
1	80C4	Error de comunicación temporal: <ul style="list-style-type: none"> • La conexión no se puede establecer en estos momentos • La interfaz está recibiendo nuevos parámetros • La instrucción TDISCON está deshaciendo la conexión configurada
1	8722	Parámetro CONNECT: Área de origen no válida: el área no existe en el DB
1	873A	Parámetro CONNECT: Imposible acceder a la descripción de la conexión (p. ej. porque el DB no está disponible)
1	877F	Parámetro CONNECT: Error interno, p. ej. referencia no válida a ANY

Comunicación Ethernet abierta con control de conexión/desconexión

Nota

El procesamiento de las instrucciones TCON, TDISCON, TSEND y TRCV puede tardar un tiempo indeterminado. Para garantizar que estas instrucciones se procesen en cada ciclo, es preciso llamarlas siempre desde el ciclo del programa principal, p. ej. desde un OB de ciclo o un bloque lógico llamado desde el ciclo del programa. **No** llame estas instrucciones desde un OB de alarma de proceso, OB de alarma de retardo, OB de alarma cíclica, OB de error de tiempo ni OB de arranque.

Comunicación Ethernet mediante los protocolos TCP e ISO on TCP

Estas instrucciones del programa controlan el proceso de comunicación:

- TCON sirve para establecer una conexión.
- TSEND y TRCV permiten enviar y recibir datos.
- TDISCON deshace la conexión.

El tamaño mínimo de los datos que pueden transmitirse o recibirse con las instrucciones TSEND y TRCV es un byte. La instrucción TRCV no soporta la transmisión de datos booleanos ni de matrices booleanas. Encontrará más información sobre cómo transferir datos con estas instrucciones en el apartado Coherencia de datos (Página 96).

Nota

El ajuste predeterminado del parámetro LEN (LEN = 0) utiliza el parámetro DATA para determinar la longitud de los datos que se están transmitiendo. Asegúrese de que los datos (DATA) transmitidos por la instrucción TSEND tienen el mismo tamaño que el parámetro DATA de la instrucción TRCV.

Ambos interlocutores ejecutan la instrucción TCON para configurar y establecer la conexión. El punto final activo y el punto final pasivo de la comunicación se especifican mediante parámetros. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.

Si la conexión se deshace p. ej. debido a una interrupción de la línea o por el interlocutor remoto, el interlocutor activo intenta restablecer la conexión configurada. No es necesario volver a ejecutar TCON.

Una conexión existente se deshace y la conexión configurada se elimina cuando se ejecuta la instrucción TDISCON o cuando la CPU pasa a estado operativo STOP. Para configurar y restablecer la conexión es preciso ejecutar TCON de nuevo.

Descripción de la función

TCON, TDISCON, TSEND y TRCV funcionan de forma asíncrona, con lo que el procesamiento de la tarea abarca varias ejecuciones de la instrucción.

Por ejemplo, una tarea para configurar y establecer una conexión se inicia ejecutando un instrucción TCON con el parámetro REQ = 1. Después se utilizan ejecuciones adicionales de TCON para vigilar la tarea y comprobar si ha finalizado con el parámetro DONE.

La tabla siguiente muestra la relación entre los parámetros BUSY, DONE y ERROR. Utilice la tabla para determinar el estado actual de la tarea.

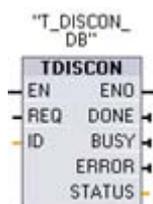
BUSY	DONE	ERROR	Descripción
TRUE	Irrelevant e	Irrelevant e	La tarea se está procesando.
FALSE	TRUE	FALSE	La tarea se ha ejecutado correctamente.
FALSE	FALSE	TRUE	La tarea se ha finalizado con un error. La causa del error se indica en el parámetro STATUS.
FALSE	FALSE	FALSE	No se ha asignado ninguna tarea nueva.

TCON



Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	El parámetro de control REQUEST inicia la tarea para establecer la conexión que indica ID. La tarea comienza cuando se produce un flanco ascendente.
ID	IN	CONN_OUC (Word)	Referencia a la conexión que debe establecerse con el interlocutor remoto, o bien entre el programa de usuario y la capa de comunicación del sistema operativo. La ID debe ser idéntica al parámetro ID asociado en la descripción de la conexión local. Rango de valores: W#16#0001 a W#16#0FFF
CONNECT	IN_OUT	Parám TCON	Puntero a la descripción de la conexión
DONE	OUT	Bool	Parámetro de estado DONE: <ul style="list-style-type: none"> • 0: Tarea no iniciada aún o en proceso • 1: Tarea ejecutada sin errores
BUSY	OUT	Bool	BUSY = 1: Tarea no finalizada aún BUSY = 0: Tarea finalizada
ERROR	OUT	Bool	Parámetro de estado ERROR: ERROR = 1: Ha ocurrido un error al procesar la tarea. STATUS proporciona información detallada sobre el tipo de error.
STATUS	OUT	Word	Parámetro de estado STATUS: Información de error

TDISCON



TCP e ISO on TCP: TDISCON deshace una conexión entre la CPU y un interlocutor.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	El parámetro de control REQUEST inicia la tarea para establecer la conexión que indica ID. La tarea comienza cuando se produce un flanco ascendente.
ID	IN	CONN_OUC (Word)	Referencia a la conexión que debe deshacerse con el interlocutor remoto, o bien entre el programa de usuario y el nivel de comunicación del sistema operativo. La ID debe ser idéntica al parámetro ID asociado en la descripción de la conexión local. Rango de valores: W#16#0001 a W#16#0FFF
DONE	OUT	Bool	Parámetro de estado DONE: <ul style="list-style-type: none"> • 0: Tarea no iniciada aún o en proceso • 1: Tarea ejecutada sin errores
BUSY	OUT	Bool	BUSY = 1: Tarea no finalizada aún BUSY = 0: Tarea finalizada
ERROR	OUT	Bool	ERROR = 1: Ha ocurrido un error durante el procesamiento.
STATUS	OUT	Word	Código de error

TSEND



Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	El parámetro de control REQUEST inicia la tarea de transmisión cuando se produce un flanco ascendente. Los datos se transfieren desde el área que indican los parámetros DATA y LEN.
ID	IN	CONN_OUC (Word)	Referencia a la conexión asociada. La ID debe ser idéntica al parámetro ID asociado en la descripción de la conexión local. Rango de valores: W#16#0001 a W#16#0FFF
LEN	IN	Int	Número máximo de bytes que deben enviarse con la tarea
DATA	IN_OUT	Variante	Puntero al área de los datos que deben enviarse: Área de emisión; contiene la dirección y longitud. La dirección hace referencia a: <ul style="list-style-type: none"> • La memoria imagen de proceso de las entradas • La memoria imagen de proceso de las salidas • Una marca • Un bloque de datos
DONE	OUT	Bool	Parámetro de estado DONE: <ul style="list-style-type: none"> • 0: Tarea no iniciada aún o en proceso. • 1: Tarea finalizada sin errores.
BUSY	OUT	Bool	<ul style="list-style-type: none"> • BUSY = 1: La tarea no ha finalizado aún. No se puede iniciar una tarea nueva. • BUSY = 0: Tarea finalizada.
ERROR	OUT	Bool	Parámetro de estado ERROR: ERROR = 1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error
STATUS	OUT	Word	Parámetro de estado STATUS: Información de error

TRCV



Parámetro	Tipo de parámetro	Tipo de datos	Descripción
EN_R	IN	Bool	Parámetro de control habilitado para recibir: Si EN_R = 1, TRCV está listo para recibir. La tarea de recepción se está procesando.
ID	IN	CONN_OUC (Word)	Referencia a la conexión asociada. La ID debe ser idéntica al parámetro ID asociado en la descripción de la conexión local. Rango de valores: W#16#0001 a W#16#0FFF
LEN	IN	Int	Longitud del área de recepción en bytes (el ajuste predeterminado es 0, es decir, el parámetro DATA determina la longitud de los datos por recibir).
DATA	IN_OUT	Variante	Puntero a los datos recibidos: Área de recepción que contiene la dirección y longitud. La dirección hace referencia a: <ul style="list-style-type: none"> • La memoria imagen de proceso de las entradas • La memoria imagen de proceso de las salidas • Una marca • Un bloque de datos
NDR	OUT	Bool	Parámetro de estado NDR: <ul style="list-style-type: none"> • NDR = 0: Tarea no iniciada aún o en proceso. • NDR = 1: Tarea finalizada correctamente.
BUSY	OUT	Bool	<ul style="list-style-type: none"> • BUSY = 1: La tarea no ha finalizado aún. No se puede iniciar una tarea nueva. • BUSY = 0: Tarea finalizada.
ERROR	OUT	Bool	ERROR=1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.
STATUS	OUT	Word	Información de error
RCVD_LEN	OUT	Int	Cantidad de datos (en bytes) recibida realmente

Área de recepción

La instrucción TRCV escribe los datos recibidos en un área de recepción especificada por las dos variables siguientes:

- Puntero al comienzo del área
- Longitud del área

Nota

El ajuste predeterminado del parámetro LEN (LEN = 0) utiliza el parámetro DATA para determinar la longitud de los datos que se están transmitiendo. Asegúrese de que los datos (DATA) transmitidos por la instrucción TSEND tienen el mismo tamaño que el parámetro DATA de la instrucción TRCV.

La tabla siguiente muestra cómo TRCV introduce los datos recibidos en el área de recepción.

Variante de protocolo	Entrada de datos en el área de recepción	Parámetro "connection_type"
TCP	Recepción de datos con la longitud especificada	B#16#11
ISO on TCP	Controlado por protocolo	B#16#12

Tan pronto como se hayan recibido todos los datos de la tarea, TRCV los transferirá al área de recepción y pondrá NDR a 1.

Códigos de condición de TCON

ERROR	STATUS (W#16#...)	Significado
0	0000	La conexión se ha establecido correctamente
0	7000	No se está procesando ninguna tarea
0	7001	Iniciar el procesamiento de la tarea, estableciendo la conexión
0	7002	Llamada intermedia (REQ es irrelevante), estableciendo la conexión
1	8086	El parámetro ID está fuera del rango admisible.
1	8087	Se ha alcanzado el número máximo de conexiones; no es posible establecer más conexiones
1	809B	La "local_device_id" indicada en la descripción de la conexión no concuerda con la CPU.
1	80A1	El usuario ya está utilizando el puerto o conexión
1	80A2	El sistema está utilizando el puerto local o remoto
1	80A3	Se está intentando restablecer una conexión existente
1	80A4	La dirección IP del punto final remoto de la conexión no es válida; puede ser que concuerde con la dirección IP local
1	80A7	Error de comunicación: TDISCON se ha ejecutado antes de finalizar TCON. TDISCON debe deshacer primero por completo la conexión referenciada por la ID.

ERROR	STATUS (W#16#...)	Significado
1	80B3	Parametrización incoherente: Error agrupado para los códigos de error W#16#80A0 hasta W#16#80A2, W#16#80A4, W#16#80B4 hasta W#16#80B9
1	80B4	Si se utiliza ISO on TCP (connection_type = B#16#12) para establecer una conexión pasiva, el código de condición 80B4 advierte de que el TSAP introducido no cumple uno de los siguientes requisitos para la dirección: <ul style="list-style-type: none"> • Si la longitud del TSAP local es 2 y el valor de la ID TSAP es E0 o E1 (hexadecimal) para el primer byte, el segundo byte deberá ser 00 ó 01. • Si la longitud del TSAP local es 3 o superior y el valor de la ID TSAP es E0 o E1 (hexadecimal) para el primer byte, el segundo byte deberá ser 00 ó 01 y todos los demás bytes deberán ser caracteres ASCII válidos. • Si la longitud del TSAP local es 3 o superior y el valor del primer byte de la ID TSAP no es E0 ni E1 (hexadecimal), todos los demás bytes de la ID TSAP deberán ser caracteres ASCII válidos. Los caracteres ASCII válidos son valores de byte comprendidos entre 20 y 7E (hexadecimal).
1	80B5	Error en el parámetro active_est
1	80B6	Error de parametrización en el parámetro connection_type
1	80B7	Error en uno de los parámetros siguientes: block_length, local_tsap_id_len, rem_subnet_id_len, rem_staddr_len, rem_tsap_id_len, next_staddr_len
1	80B8	Los parámetros de la descripción de la conexión local y el parámetro ID son diferentes
1	80C3	Se están utilizando todos los recursos de conexión.
1	80C4	Error de comunicación temporal: <ul style="list-style-type: none"> • La conexión no se puede establecer en estos momentos. • La interfaz está recibiendo nuevos parámetros. • La instrucción TDISCON está deshaciendo la conexión configurada.

Códigos de condición de TDISCON

ERROR	STATUS (W#16#...)	Significado
0	0000	La conexión se ha deshecho correctamente
0	7000	No se está procesando ninguna tarea
0	7001	Inicio del procesamiento de la tarea, deshaciendo la conexión
0	7002	Llamada intermedia (REQ es irrelevante), deshaciendo la conexión
1	8086	El parámetro ID está fuera del rango de direcciones admisible.
1	80A3	Se está intentando deshacer una conexión no existente
1	80C4	Error de comunicación temporal: La interfaz está recibiendo nuevos parámetros o la conexión se está estableciendo.

Códigos de condición de TSEND

ERROR	STATUS (W#16#...)	Significado
0	0000	Tarea de transmisión finalizada sin errores
0	7000	No se está procesando ninguna tarea
0	7001	Inicio del procesamiento de la tarea, enviando los datos: Durante este procesamiento, el sistema operativo accede a los datos del área de emisión DATA.
0	7002	Llamada intermedia (REQ es irrelevante), procesando la tarea: El sistema operativo accede a los datos del área de emisión DATA durante este procesamiento.
1	8085	El parámetro LEN es mayor que el valor máximo admisible.
1	8086	El parámetro ID está fuera del rango de direcciones admisible
1	8088	El parámetro LEN excede el área de memoria especificada en DATA
1	80A1	Error de comunicación: <ul style="list-style-type: none"> La conexión indicada no se ha establecido todavía La conexión indicada se está deshaciendo. No es posible transmitir datos a través de esta conexión. La interfaz se está reiniciando.
1	80C3	Falta interna de recursos: Ya se está procesando un bloque con esta ID en una clase de prioridad diferente.
1	80C4	Error de comunicación temporal: <ul style="list-style-type: none"> La conexión con el interlocutor no se puede establecer en estos momentos. La interfaz está recibiendo nuevos parámetros o la conexión se está estableciendo.

Códigos de condición de TRCV

ERROR	STATUS (W#16#...)	Significado
0	0000	Se han aceptado datos nuevos: La longitud actual de los datos recibidos se muestra en RCVD_LEN.
0	7000	El bloque no está listo para la recepción
0	7001	El bloque está listo para la recepción, se ha activado la tarea de recepción.
0	7002	Llamada intermedia, se está procesando la tarea de recepción: Durante este proceso se escriben datos en el área de recepción. Por este motivo puede ocurrir un error y ocasionar datos incoherentes en el área de recepción.
1	8085	El parámetro LEN excede el valor máximo admisible o el valor del parámetro LEN o DATA fue modificado desde la primera llamada.
1	8086	El parámetro ID está fuera del rango de direcciones admisible
1	8088	El área de recepción es demasiado pequeña: El valor de LEN excede el área de recepción especificada en DATA.

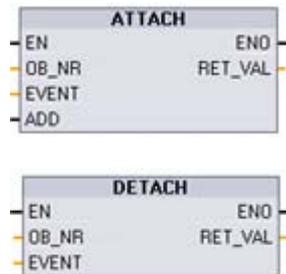
ERROR	STATUS (W#16#...)	Significado
1	80A1	Error de comunicación: <ul style="list-style-type: none">• La conexión indicada no se ha establecido todavía• La conexión indicada se está deshaciendo. La tarea de recepción no es posible a través de esta conexión.• La interfaz está recibiendo nuevos parámetros.
1	80C3	Falta interna de recursos: Ya se está procesando un bloque con esta ID en una clase de prioridad diferente.
1	80C4	Error de comunicación temporal: <ul style="list-style-type: none">• La conexión con el interlocutor no se puede establecer en estos momentos.• La interfaz está recibiendo nuevos ajustes de parámetros o la conexión se está estableciendo.

6.2.5.2 Instrucciones de comunicación punto a punto

El capítulo Comunicación punto a punto (PtP) (Página 269) ofrece información detallada sobre las instrucciones PtP y los módulos de comunicación.

6.2.6 Instrucciones con alarmas

6.2.6.1 Instrucciones ATTACH y DETACH



Las instrucciones ATTACH y DETACH sirven para activar y desactivar subprogramas controlados por eventos de alarma.

- ATTACH habilita la ejecución de un OB de alarma para un evento de alarma de proceso.
- DETACH inhibe la ejecución de un OB de alarma para un evento de alarma de proceso.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
OB_NR	IN	Int	Identificador del bloque de organización: Seleccione uno de los OBs de alarma de proceso que fueron creados utilizando la función "Agregar nuevo bloque". Haga doble clic en el campo de parámetro. A continuación, haga clic en el símbolo de ayuda para ver los OBs disponibles.
EVENT	IN	DWord	Identificador del evento: Seleccione uno de los eventos de alarma de proceso habilitados en la configuración de dispositivos PLC para las entradas digitales o los contadores rápidos. Haga doble clic en el campo de parámetro. A continuación, haga clic en el símbolo de ayuda para ver los eventos disponibles.
ADD (sólo ATTACH)	IN	Bool	ADD = 0 (ajuste predeterminado): Este evento reemplaza a todas las asignaciones de eventos anteriores de este OB. ADD = 1: Este evento se agrega a las asignaciones de eventos anteriores de este OB.
RET_VAL	OUT	Int	Código de condición de ejecución

Eventos de alarma de proceso

La CPU soporta los siguientes eventos de alarma de proceso:

- Eventos de flanco ascendente (todas las entradas digitales integradas en la CPU, más las entradas digitales de la Signal Board)
 - Un flanco ascendente ocurre cuando la entrada digital cambia de OFF a ON como reacción a una modificación de la señal de un aparato de campo conectado a la entrada.
- Eventos de flanco descendente (todas las entradas digitales integradas en la CPU y toda entrada de la Signal Board)
 - Un flanco descendente ocurre cuando la entrada digital cambia de ON a OFF.
- Eventos en los que el valor actual del contador rápido (HSC) = valor de referencia (CV = RV) (HSC 1 a 6)
 - Una alarma CV = RV de un HSC se genera cuando el contejo actual cambia de un valor adyacente al valor que concuerda exactamente con un valor de referencia establecido previamente.
- Eventos de cambio de sentido del HSC (HSC 1 a 6)
 - Un evento de cambio de sentido ocurre cuando se detecta que el sentido de contejo del HSC cambia de ascendente a descendente, o viceversa.
- Eventos de reset externo del HSC (HSC 1 a 6)
 - Algunos modos del HSC permiten asignar una entrada digital como reset externo para poner a cero el valor de contejo del HSC. Un evento de reset externo del HSC ocurre cuando esta entrada cambia de OFF a ON.

Habilitar eventos de alarma de proceso en la configuración de dispositivos

Las alarmas de proceso deben habilitarse durante la configuración de dispositivos. La casilla de habilitación de eventos se debe activar en la configuración de eventos para un canal de entrada digital o un HSC si este evento debe asignarse durante la configuración o en runtime.

Opciones de las casillas de verificación en la configuración de dispositivos PLC:

- Entrada digital
 - Habilitar detección de flancos ascendentes
 - Habilitar detección de flancos descendentes
- Contador rápido (HSC)
 - Habilitar este contador rápido para su uso
 - Generar alarma si el valor del contador es igual al valor de referencia
 - Generar alarma en caso de evento de reset externo
 - Generar alarma en caso de evento de cambio de sentido

Agregar OBs de alarma de proceso nuevos al programa de usuario

De forma predeterminada, ningún OB está asignado a un evento cuando éste se habilita por primera vez. Esto se indica mediante la etiqueta "<no conectado>" correspondiente a "Alarma de proceso:" en la configuración de dispositivos. Sólo los OBs de alarma de proceso pueden asignarse a un evento de alarma de proceso. Todos los OBs de alarma de proceso existentes aparecen en la lista desplegable "Alarma de proceso:". Si no se lista ningún OB, es preciso crear un OB de "Alarma de proceso" como se indica a continuación. En la rama "Bloques de programa" del árbol del proyecto:

1. Haga doble clic en "Agregar nuevo bloque", seleccione "Bloque de organización (OB)" y elija "Alarma de proceso".
2. Opcionalmente, puede cambiar el nombre del OB, elegir el lenguaje de programación (KOP o FUP) y seleccionar el número de bloque (cambiar a modo manual y seleccionar un número de bloque diferente del propuesto).
3. Edite el OB y agregue la reacción programada que debe ejecutarse cuando ocurra el evento. Es posible llamar FCs y FBs desde este OB, hasta una profundidad de anidamiento de cuatro niveles.

Parámetro OB_NR

Todos los nombres de OB de alarma de proceso existentes aparecen en la lista desplegable "Alarma de proceso:" de la configuración de dispositivos y en la lista desplegable OB_NR del parámetro ATTACH/DETACH.

Parámetro EVENT

Si está habilitado un evento de alarma de proceso, se le asigna un nombre predeterminado y unívoco. Este nombre se puede cambiar editando el campo de edición "Nombre de evento:", pero debe ser un nombre unívoco. Los nombres de eventos se convierten en nombres de variables en la tabla de variables "Constantes" y aparecen en la lista desplegable del parámetro EVENT en los cuadros de las instrucciones ATTACH y DETACH. El valor de la variable es un número interno utilizado para identificar el evento.

Funcionamiento general

Todo evento de hardware puede asignarse a un OB de alarma de proceso que se pondrá en cola de espera para ser ejecutado cuando ocurra el evento de alarma de proceso. El OB puede asignarse al evento durante la configuración o en runtime.

Es posible asignar o cancelar la asignación de un OB a un evento habilitado al realizar la configuración. Para asignar un OB a un evento durante la configuración, abra la lista desplegable "Alarma de proceso:" (haciendo clic en el botón de flecha abajo en el lado derecho) y seleccione un OB en la lista de OBs de alarma de proceso disponibles. Seleccione el nombre de OB deseado en esta lista, o bien elija "<no conectado>" para cancelar la asignación.

También es posible asignar o deshacer la asignación de un evento de alarma de proceso habilitado en runtime. Utilice las instrucciones ATTACH o DETACH en runtime (varias veces en caso necesario) para asignar o cancelar la asignación de un evento de alarma de proceso al OB respectivo. Si ningún OB está asignado actualmente (debido a que se ha seleccionado "<no conectado>" en la configuración de dispositivos o como resultado de la ejecución de la instrucción DETACH), se ignorará el evento de alarma de proceso habilitado.

Instrucción DETACH

La instrucción DETACH permite deshacer la asignación de uno o todos los eventos de un OB en particular. Si se especifica un EVENT, se cancelará la asignación sólo de este evento al OB_NR indicado. Todos los demás eventos asignados actualmente a este OB_NR permanecerán asignados. Si no se especifica ningún EVENT, se deshace la asignación de todos los eventos asignados actualmente al OB_NR.

Códigos de condición

RET_VAL (W#16#....)	Estado de ENO	Descripción
0000	1	Sin error
0001	0	No hay ninguna asignación que pueda deshacerse (sólo DETACH)
8090	0	El OB no existe
8091	0	Tipo de OB incorrecto
8093	0	El evento no existe

6.2.6.2 Instrucciones para iniciar y cancelar alarmas de retardo

El procesamiento de alarmas de retardo se puede iniciar y cancelar con las instrucciones SRT_DINT y CAN_DINT, respectivamente. Toda alarma de retardo es un evento único que ocurre al cabo del tiempo de retardo indicado. Si el evento de retardo se cancela antes de que transcurra el tiempo de retardo, no ocurrirá la alarma.



SRT_DINT inicia una alarma de retardo que llama un OB (bloque de organización) una vez transcurrido el tiempo de retardo especificado en el parámetro DTIME.



CAN_DINT cancela una alarma de retardo ya iniciada. El OB de alarma de retardo no se ejecuta en este caso.

Parámetros de SRT_DINT

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
OB_NR	IN	Int	Bloque de organización (OB) que debe ejecutarse tras un tiempo de retardo: Seleccione uno de los OBs de alarma de retardo creados utilizando la función "Agregar nuevo bloque" del árbol del proyecto. Haga doble clic en el campo de parámetro. A continuación, haga clic en el símbolo de ayuda para ver los OBs disponibles.
DTIME	IN	Time	Tiempo de retardo (1 hasta 60000 ms) Para crear tiempos de retardo más prolongados es posible utilizar p. ej. un contador dentro de un OB de alarma de retardo.
SIGN	IN	Word	El S7-1200 no lo utiliza. Se acepta cualquier valor.
RET_VAL	OUT	Int	Código de condición de ejecución

Parámetros de CAN_DINT

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
OB_NR	IN	Int	Identificador del OB de alarma de retardo. Es posible utilizar un número de OB o un nombre simbólico.
RET_VAL	OUT	Int	Código de condición de ejecución

Funcionamiento

La instrucción SRT_DINT especifica un tiempo de retardo, inicia el temporizador de retardo interno y asigna un OB de alarma de retardo al evento de timeout de retardo. Una vez transcurrido el tiempo de retardo especificado, se genera una alarma que dispara la ejecución del OB de alarma de retardo asociado. Una alarma de retardo iniciada puede cancelarse antes de que transcurra el tiempo de retardo especificado, ejecutando para ello la instrucción CAN_DINT. No puede haber más de cuatro eventos de alarma de retardo y de alarma cíclica activos.

Agregar OBs de alarma de retardo al proyecto

Los OBs de alarma de retardo son los únicos que pueden asignarse a las instrucciones SRT_DINT y CAN_DINT. Un proyecto nuevo no contiene aún OBs de alarma de retardo. Es preciso agregar OBs de alarma de retardo al proyecto. Para crear un OB de alarma de retardo, proceda del siguiente modo:

1. Haga doble clic en "Agregar nuevo bloque" en la rama "Bloques de programa" del árbol del proyecto, seleccione "Bloque de organización (OB)" y elija "Alarma de retardo".
2. Es posible cambiar el nombre del OB, así como seleccionar el lenguaje de programación y el número de bloque. Cambie al modo de numeración manual si desea asignar un número de bloque diferente del propuesto automáticamente.
3. Edite el OB de alarma de retardo y cree la reacción programada que debe ejecutarse cuando ocurra el evento de timeout de retardo. Es posible llamar FCs y FBs desde el OB de alarma de retardo, hasta una profundidad de anidamiento de cuatro niveles.
4. Los nuevos nombres de los OBs de alarma de retardo estarán disponibles al editar el parámetro OB_NR de las instrucciones SRT_DINT y CAN_DINT.

Códigos de condición

RET_VAL (W#16#...)	Descripción
0000	No ha ocurrido ningún error
8090	Parámetro OB_NR incorrecto
8091	Parámetro DTIME incorrecto
80A0	La alarma de retardo no se ha iniciado

6.2.6.3 Instrucciones para inhibir y habilitar el procesamiento de alarmas

Las instrucciones DIS_AIRT y EN_AIRT sirven para inhibir y habilitar el procesamiento de alarmas.



DIS_AIRT retarda el procesamiento de eventos de alarma nuevos. DIS_AIRT puede ejecutarse más de una vez en un OB. El sistema operativo cuenta las ejecuciones de DIS_AIRT. Cada ejecución permanece activa hasta que es cancelada específicamente por una instrucción EN_AIRT o hasta que se termina de procesar el OB actual.

Una vez habilitadas nuevamente, se procesan las alarmas que ocurrieron cuando estaba activa la instrucción DIS_AIRT, o bien se procesan tan pronto como se haya ejecutado el OB actual.



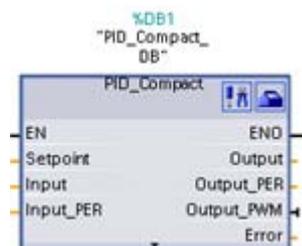
EN_AIRT habilita el procesamiento de eventos de alarma inhibidos anteriormente con la instrucción DIS_AIRT. Toda ejecución de DIS_AIRT debe ser cancelada por una ejecución de EN_AIRT. Por ejemplo, si el procesamiento de alarmas se ha inhibido cinco veces con cinco ejecuciones de DIS_AIRT, éstas deben cancelarse con cinco ejecuciones de EN_AIRT.

Las ejecuciones de EN_AIRT deben ocurrir en un mismo OB, o bien en una FC o FB llamado desde el mismo OB, antes de poder habilitar las alarmas nuevamente para este OB.

El parámetro RET_VAL indica el número de veces que se ha inhibido el procesamiento de alarmas. Éste es el número de ejecuciones de DIS_AIRT en la cola de espera. El procesamiento de alarmas sólo se puede habilitar nuevamente cuando el parámetro RET_VAL = 0.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
RET_VAL	OUT	Int	Número de retardos = número de ejecuciones de DIS_AIRT en la cola de espera.

6.2.7 Control PID



La instrucción "PID_Compact" ofrece un regulador PID con función de optimización automática para los modos automático y manual.

Encontrará más información acerca de la instrucción PID_Compact en la Ayuda en pantalla del TIA Portal.

6.2.8 Instrucciones de Motion Control

Las instrucciones de Motion Control utilizan un bloque de datos tecnológico asociado y el PTO (tren de impulsos) de la CPU para controlar el movimiento de un eje. Encontrará más información acerca de las instrucciones de Motion Control en la Ayuda en pantalla de STEP 7 Basic.

ATENCIÓN

La frecuencia de pulsos máxima de los generadores de impulsos de salida es 100 KHz para las salidas digitales de la CPU y 20 KHz para las de la Signal Board. Sin embargo, STEP 7 Basic no alerta cuando se configura un eje que, a una velocidad o frecuencia máximas, excede esta limitación de hardware. Ello podría ocasionar problemas en la aplicación. Por tanto, asegúrese de que no se exceda la frecuencia de pulsos máxima del hardware.



MC_Power habilita e inhibe un eje de control de movimiento.



MC_Reset resetea todos los errores de control de movimiento. Se acusan todos los errores de control de movimiento que pueden acusarse.



MC_Home establece la relación entre el programa de control del eje y el sistema de posicionamiento mecánico del eje.



MC_Halt cancela todos los procesos de movimiento y detiene el movimiento del eje. La posición de parada no está definida.



MC_MoveJog ejecuta el modo jog para fines de test y arranque.



MC_MoveAbsolute inicia el movimiento hacia una posición absoluta. La tarea finaliza cuando se alcanza la posición de destino.

MC_MoveRelative inicia un movimiento de posicionamiento relativo a la posición inicial.

MC_MoveVelocity hace que el eje se mueva a la velocidad indicada.

Nota

Los trenes de impulsos no pueden ser utilizados por otras instrucciones del programa de usuario

Si las salidas de la CPU o Signal Board se configuran como generadores de impulsos (para su utilización con la PWM o instrucciones de Motion Control básicas), las direcciones de las salidas correspondientes (Q0.0, Q0.1, Q4.0 y Q4.1) se eliminarán de la memoria Q y no podrán utilizarse para ningún otro fin en el programa de usuario. Si el programa de usuario escribe un valor en una salida utilizada como generador de impulsos, la CPU no escribirá ese valor en la salida física.

6.2.9 Instrucción "Impulso"

6.2.9.1 Instrucción CTRL_PWM

La instrucción CTRL_PWM (Controlar modulación del ancho de pulso) ofrece un tiempo de ciclo fijo con un ciclo de trabajo variable. La salida PWM se ejecuta continuamente tras haberse iniciado a la frecuencia indicada (tiempo de ciclo).

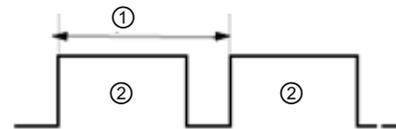
La duración de impulso varía según sea necesario para obtener el control deseado.

La duración de impulso puede expresarse en centésimos del tiempo de ciclo (0 – 100), milésimos (0 – 1000), diezmilésimos (0 – 10000) o formato analógico S7. La duración de impulso puede variar entre 0 (sin impulso, siempre off) y escala completa (sin impulso, siempre on).

Puesto que la salida PWM puede variar entre 0 y escala completa, ofrece una salida digital que, en numerosos aspectos, es similar a una salida analógica. Por ejemplo, la salida PWM puede utilizarse para controlar la velocidad de un motor (desde "paro" hasta "a toda velocidad") o la posición de una válvula (desde "cerrada" hasta "totalmente abierta").

Hay dos generadores de impulsos disponibles para controlar las salidas de impulsos rápidos: PWM y PTO (tren de impulsos). Las instrucciones de Motion Control utilizan PTO. Cada generador de impulsos puede asignarse a PWM o PTO, pero no a ambos simultáneamente.

Los dos generadores de impulsos se mapean en las salidas digitales específicas como muestra la tabla siguiente. Es posible utilizar las salidas integradas de la CPU o las salidas opcionales de la Signal Board. La tabla siguiente muestra los números de las salidas (suponiendo que se utiliza la configuración predeterminada de las salidas). Si se ha modificado la numeración de las salidas, sus números serán los que se hayan asignado. Independientemente de ello, PTO1/PWM1 utiliza las dos primeras salidas digitales, en tanto que PTO2/PWM2 utiliza las dos salidas digitales siguientes, bien sea en la CPU o en la Signal Board acoplada. Tenga en cuenta que PWM sólo requiere una salida, mientras que PTO puede utilizar opcionalmente dos salidas por canal. Si una salida no se requiere para una función de impulsos, estará disponible para otros usos.



- ① Tiempo de ciclo
- ② Duración de impulso

Descripción	Asignación de salidas predeterminada		
		Impulso	Sentido
PTO 1	Integrada en la CPU	Q0.0	Q0.1
	Signal Board	Q4.0	Q4.1
PWM 1	Integrada en la CPU	Q0.0	--
	Signal Board	Q4.0	--
PTO 2	Integrada en la CPU	Q0.2	Q0.3
	Signal Board	Q4.2	Q4.3
PWM 2	Integrada en la CPU	Q0.2	--
	Signal Board	Q4.2	--

Configurar un canal de impulsos para PWM

Para preparar la modulación de ancho de pulsos (PWM), configure primero un canal de impulsos en la configuración de dispositivos, seleccionando la CPU, luego el generador de impulsos (PTO/PWM) y elija PWM1 o PWM2. Habilite el generador de impulsos (casilla de verificación). Si está habilitado un generador de impulsos, se le asigna un nombre predeterminado y unívoco. Este nombre puede cambiarse editando el campo de edición "Nombre:", pero debe ser un nombre unívoco. Los nombres de los generadores de impulsos habilitados se convierten en variables en la tabla de variables "Constantes" y están disponibles para ser utilizados como parámetro PWM de la instrucción CTRL_PWM.

ATENCIÓN
La frecuencia de pulsos máxima de los generadores de impulsos de salida es 100 KHz para las salidas digitales de la CPU y 20 KHz para las de la Signal Board. Sin embargo, STEP 7 Basic no alerta cuando se configura un eje que, a una velocidad o frecuencia máximas, excede esta limitación de hardware. Ello podría ocasionar problemas en la aplicación. Por tanto, asegúrese de que no se exceda la frecuencia de pulsos máxima del hardware.

Es posible cambiar el nombre del generador de impulsos, agregar un comentario y asignar parámetros como se indica a continuación:

- Generador de impulsos utilizado: PWM o PTO (seleccione PWM)
- Fuente de salida: salidas integradas en la CPU o Signal Board
- Base de tiempo: milisegundos o microsegundos
- Formato de la duración de impulso:
 - Centésimos (0 a 100)
 - Milésimos (0 a 1000)
 - Diezmilésimos (0 a 10000)
 - Formato analógico S7 (0 a 27648)
- Tiempo de ciclo: Introduzca el valor del tiempo de ciclo. Este valor sólo se puede modificar en la "Configuración de dispositivos".
- Duración de impulso inicial: Introduzca la duración de impulso inicial. El valor de la duración de impulso puede modificarse en runtime.

Direcciones de salida



Dirección inicial: Introduzca la dirección de palabra Q en la que desea depositar el valor de la duración de impulso. La dirección predeterminada es QW1000 para PWM1 y QW1002 para PWM2. El valor de esta dirección controla el ancho del impulso y se inicializa al valor de la "Duración de impulso inicial:" indicado arriba cada vez que la CPU pasa de STOP a RUN. Este valor de palabra Q puede cambiarse en runtime para modificar la duración de impulso.

Parámetro	Tipo de parámetro	Tipo de datos	Valor inicial	Descripción
PWM	IN	Word	0	Identificador PWM: Los nombres de los generadores de impulsos habilitados se convierten en variables en la tabla de variables "Constantes" y están disponibles para ser utilizados como parámetro PWM.
ENABLE	IN	Bool		1= iniciar generador de impulsos 0 = parar generador de impulsos
BUSY	OUT	Bool	0	Función ocupada
STATUS	OUT	Word	0	Código de condición de ejecución

Funcionamiento

La instrucción CTRL_PWM utiliza un bloque de datos (DB) para almacenar la información de parámetros. Cuando se inserta una instrucción CTRL_PWM en el editor de programación, se asigna un DB. El usuario no modifica por separado los parámetros del bloque de datos, sino que la instrucción CTRL_PWM los controla.

Determine el generador de impulsos habilitado que desea utilizar, utilizando su nombre de variable para el parámetro PWM.

Cuando la entrada EN es TRUE (verdadera), la instrucción PWM_CTRL inicia o detiene el PWM identificado, según el valor de la entrada ENABLE. El valor de la dirección de salida de palabra Q asociada indica la duración de impulso.

Puesto que el S7-1200 procesa la petición cuando se ejecuta la instrucción CTRL_PWM, el parámetro BUSY siempre notificará FALSE (falso) en las CPUs S7-1200.

Si se detecta un error, ENO se pone a FALSE y el parámetro STATUS contiene un código de condición.

La duración de impulso se pone al valor inicial ajustado en la configuración de dispositivos cuando el PLC cambia por primera vez al estado operativo RUN. Los valores se escriben en la dirección de palabra Q especificada en la configuración de dispositivos ("Direcciones de salida" / "Dirección inicial:") de la forma requerida para cambiar la duración de impulso. Para escribir la duración de impulso deseada en la palabra Q apropiada se utiliza una instrucción de desplazamiento, conversión, matemática o un cuadro PID. El valor de palabra Q debe estar comprendido en el rango válido (porcentaje, milésimos, diezmilésimos o formato analógico S7).

Códigos de condición

Valor de STATUS	Descripción
0	Sin error
80A1	El identificador PWM no direcciona un PWM válido

Las E/S digitales asignadas a PWM y PTO no se pueden forzar permanentemente

Las E/S digitales utilizadas por los dispositivos con modulación del ancho de pulso (PWM) y tren de impulsos (PTO) se asignan durante la configuración de dispositivos. Si se asignan direcciones de E/S digitales a estos dispositivos, los valores de las direcciones de E/S asignadas no podrán ser modificados por la función de forzado permanente de la tabla de observación.

Los trenes de impulsos no pueden ser utilizados por otras instrucciones del programa de usuario

Si las salidas de la CPU o Signal Board se configuran como generadores de impulsos (para su utilización con la PWM o instrucciones de Motion Control básicas), las direcciones de las salidas correspondientes (Q0.0, Q0.1, Q4.0 y Q4.1) se eliminarán de la memoria Q y no podrán utilizarse para ningún otro fin en el programa de usuario. Si el programa de usuario escribe un valor en una salida utilizada como generador de impulsos, la CPU no escribirá ese valor en la salida física.

6.3 Instrucciones de la librería global

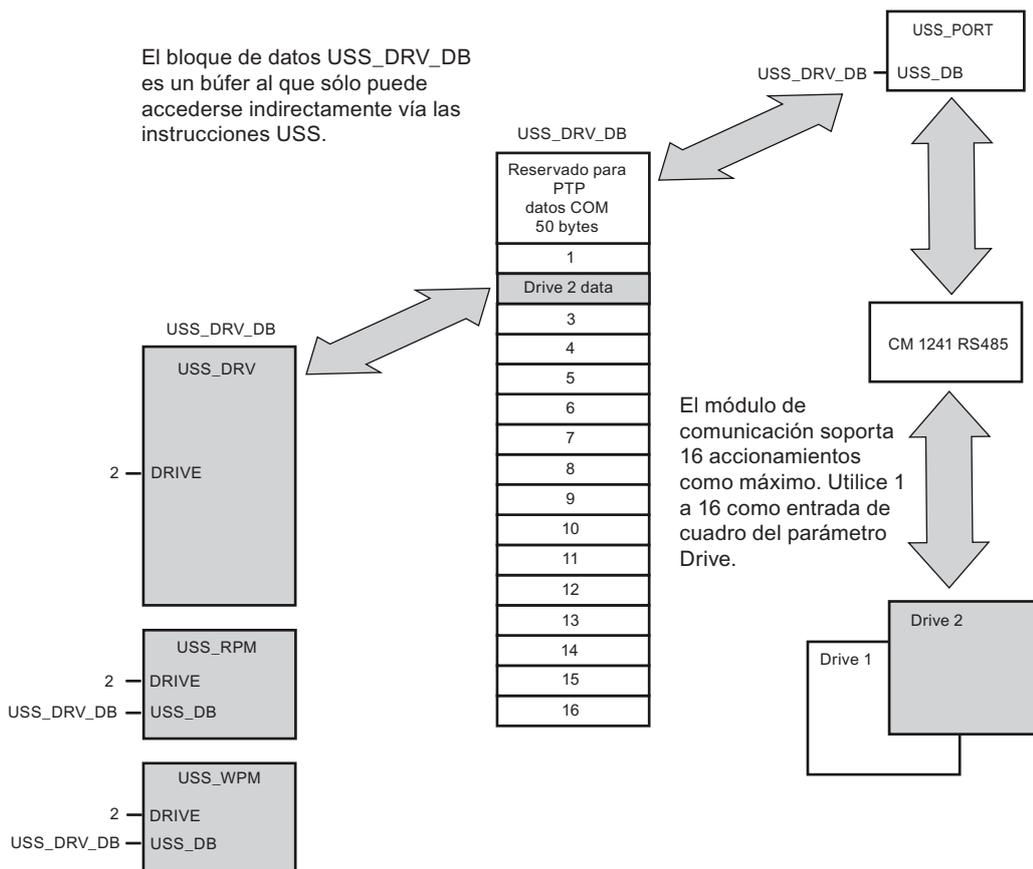
6.3.1 USS

La librería del protocolo USS permite controlar accionamientos Siemens que soportan el protocolo USS. Estas instrucciones incluyen funciones diseñadas especialmente para utilizar el protocolo USS para la comunicación con el accionamiento. El módulo CM 1241 RS485 se comunica con los accionamientos a través de puertos RS485. El accionamiento físico y los parámetros de lectura/escritura pueden controlarse con la librería USS.

6.3.1.1 Requisitos para utilizar el protocolo USS

La librería contiene 1 FB y 3 FCs que soportan el protocolo USS. Todo módulo de comunicación CM 1241 RS485 soporta como máximo 16 accionamientos.

Un solo bloque de datos instancia ofrece almacenamiento temporal y búferes para todos los accionamientos de la red USS conectados con cada módulo de comunicación PtP que se instale. Las funciones USS de estos accionamientos comparten la información en este bloque de datos.



Todos los accionamientos (máx. 16) conectados a un solo CM 1241 RS485 pertenecen a una misma red USS. Todos los accionamientos conectados a diferentes CM 1241 RS485 pertenecen a diferentes redes USS. Puesto que el S7-1200 soporta tres dispositivos CM 1241 RS485 como máximo, pueden existir tres redes USS como máximo, cada una de ellas con 16 accionamientos como máximo, con lo que se soporta un número total de 48 accionamientos USS.

Toda red USS se gestiona utilizando un bloque de datos unívoco (para tres redes USS que utilicen tres dispositivos CM 1241 RS485 se requieren tres bloques de datos). Todas las instrucciones asociadas a una red USS deben compartir dicho bloque de datos. Esto incluye todas las instrucciones USS_DRV, USS_PORT, USS_RPM y USS_WPM utilizadas para controlar todos los accionamientos de una red USS.

La instrucción USS_DRV es un bloque de función (FB). Cuando la instrucción USS_DRV se coloca en el editor, es preciso indicar qué DB debe asignarse a ese FB en el diálogo "Opciones de llamada". Si se trata de la primera instrucción USS_DRV de este programa para esta red USS, es posible aceptar la asignación de DB predeterminada (o cambiar el nombre a discreción). Entonces se crea el nuevo DB. No obstante, si no es la primera instrucción USS_DRV de este canal, es preciso utilizar la lista desplegable del diálogo "Opciones de llamada" para seleccionar el DB que fue asignado previamente a esta red USS.

Las instrucciones USS_PORT, USS_RPM y USS_WPM son funciones (FCs). A estas FCs no se les asigna ningún DB cuando se colocan en el editor. En cambio, es preciso asignar el DB adecuado a la entrada "USS_DB" de estas instrucciones (haciendo doble clic en el campo del parámetro y luego clic en el símbolo de ayuda para ver los DBs disponibles).

La función USS_PORT gestiona la comunicación real entre la CPU y los accionamientos vía el módulo de comunicación PtP. Cada llamada a esta función gestiona una comunicación con un accionamiento. El programa debe llamar esta función lo suficientemente rápido para impedir un timeout de comunicación por parte de los módulos. Esta función puede llamarse desde el OB principal o desde cualquier OB de alarma.

El bloque de función USS_DRV permite al programa acceder a un accionamiento determinado en la red USS. Sus entradas y salidas representan el estado y los controles del accionamiento. Si la red comprende 16 accionamientos, el programa debe tener como mínimo 16 llamadas de USS_DRV, es decir, una para cada accionamiento. Estos bloques deben llamarse a la velocidad necesaria para controlar las funciones del accionamiento.

El bloque de función USS_DRV sólo se puede llamar desde el OB principal.

 PRECAUCIÓN

USS_DRV, USS_RPM y USS_WPM deben llamarse desde el OB principal. USS_PORT puede llamarse desde un OB cualquiera (generalmente desde un OB de alarma de retardo).
--

Si no se impide la interrupción de USS_PORT podrían producirse errores inesperados.

Las funciones USS_RPM y USS_WPM leen y escriben los parámetros operativos del accionamiento remoto. Estos parámetros controlan el funcionamiento interno del accionamiento. Estos parámetros se definen en el manual del accionamiento. El programa puede contener un número cualquiera de estas funciones. No obstante, sólo una petición de lectura o escritura puede estar activa en un accionamiento en un momento determinado. Las funciones USS_RPM y USS_WPM sólo pueden llamarse desde el OB principal.

Calcular el tiempo necesario para la comunicación con el accionamiento

La comunicación con el accionamiento es asíncrona al ciclo del S7-1200. Por lo general, pueden transcurrir varios ciclos del S7-1200 antes de que finalice una transacción de comunicación con un accionamiento.

El intervalo de USS_PORT es el tiempo necesario para una transacción con un accionamiento. La tabla siguiente muestra el intervalo de USS_PORT mínimo para cada velocidad de transferencia. Si la función USS_PORT se llama más frecuentemente que el intervalo de USS_PORT, no se incrementará el número de transacciones. El intervalo de timeout del accionamiento es el tiempo disponible para una transacción si, debido a errores de comunicación, se requieren 3 intentos para finalizar la transacción. De forma predeterminada, la librería del protocolo USS realiza automáticamente 2 reintentos por transacción.

Velocidad de transferencia	Intervalo mínimo calculado para la llamada de USS_PORT (milisegundos)	Intervalo de timeout por accionamiento (milisegundos)
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

6.3.1.2 Instrucción USS_DRV

La instrucción USS_DRV intercambia datos con el accionamiento creando peticiones e interpretando las respuestas del accionamiento. Para cada accionamiento debe utilizarse un bloque de función propio. No obstante, todas las funciones USS asociadas con una red USS y el módulo de comunicación PtP deben utilizar el mismo bloque de datos instancia. Es preciso crear el nombre del DB cuando se inserta la primera instrucción USS_DRV. Este DB creado al insertar la instrucción por primera vez se reutiliza posteriormente.

Cuando la instrucción USS_DRV se ejecuta por primera vez, el accionamiento que indica la dirección USS (parámetro DRIVE) se inicializa en el DB instancia. Después de esta inicialización, las ejecuciones siguientes de USS_PORT pueden iniciar la comunicación con el accionamiento en este número de accionamiento.

Si se modifica el número del accionamiento, el PLC debe cambiar de STOP a RUN con objeto de inicializar el DB instancia. Los parámetros de entrada se configuran en el búfer de mensajes USS TX y las salidas se leen de un búfer de respuesta válido "anterior" (si existe). Durante la ejecución de USS_DRV no se transmiten datos. Los accionamientos se comunican cuando se ejecuta USS_PORT. USS_DRV configura únicamente los mensajes que deben enviarse e interpreta los datos que puedan haberse recibido de una petición anterior.

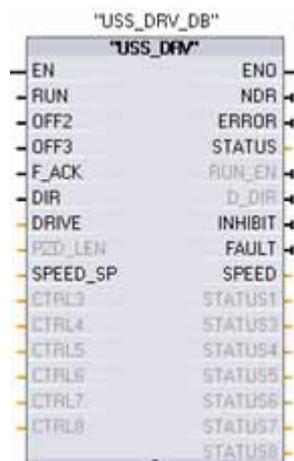
El sentido de rotación del accionamiento se puede controlar utilizando la entrada DIR (BOOL) o el signo (positivo o negativo) con la entrada SPEED_SP (REAL). La tabla siguiente indica cómo interactúan estas entradas para determinar el sentido del accionamiento, suponiendo que el motor está cableado para la rotación adelante.

SPEED_SP	DIR	Sentido del accionamiento
Valor > 0	0	Atrás
Valor > 0	1	Adelante
Valor < 0	0	Adelante
Valor < 0	1	Atrás

KOP (vista predeterminada)



KOP (vista ampliada)



Haga clic en el lado inferior del cuadro para ampliarlo y ver todos los parámetros.

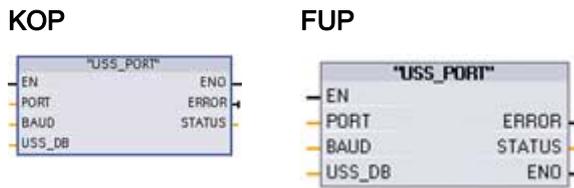
Los parámetros que aparecen atenuados son opcionales, por lo que no es necesario asignarlos.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
RUN	IN	Bool	Bit de arranque del accionamiento: Si es TRUE (verdadera), esta entrada habilita el accionamiento para que funcione a la velocidad predeterminada.
OFF2	IN	Bool	Bit de parada eléctrica: Si es FALSE (falso), este bit hace que el accionamiento marche en inercia hasta parar sin frenar.
OFF3	IN	Bool	Bit de parada rápida – Si es FALSE (falso), este bit ocasiona una parada rápida frenando el accionamiento, en vez de que marche en inercia hasta parar.
F_ACK	IN	Bool	Bit de acuse de fallo – Este bit se activa para resetear el bit de fallo en un accionamiento. Este bit se activa tras haberse solucionado el fallo para indicar al accionamiento que no tiene que seguir notificando el fallo anterior.
DIR	IN	Bool	Control de sentido del accionamiento – Este bit se activa para indicar que el sentido es hacia delante (SPEED_SP positiva).
DRIVE	IN	USInt	Dirección del accionamiento: Esta entrada es la dirección del accionamiento USS. El rango válido está comprendido entre el accionamiento 1 y el 16.
PZD_LEN	IN	USInt	Longitud de palabra – Este el número de palabras de los datos PZD. Los valores válidos son 2, 4, 6 u 8 palabras. El ajuste predeterminado es 2.
SPEED_SP	IN	Real	Consigna de velocidad – Esta es la velocidad del accionamiento expresada como porcentaje de la frecuencia configurada. Un valor positivo indica el sentido hacia delante (si DIR es TRUE).
CTRL3	IN	UInt	Palabra de control 3 – Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento. Es un parámetro opcional.
CTRL4	IN	UInt	Palabra de control 4 – Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento. Es un parámetro opcional.
CTRL5	IN	UInt	Palabra de control 5 – Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento. Es un parámetro opcional.
CTRL6	IN	UInt	Palabra de control 6 – Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento.
CTRL7	IN	UInt	Palabra de control 7 – Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento. Es un parámetro opcional.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
CTRL8	IN	UInt	Palabra de control 8 – Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento. Es un parámetro opcional.
NDR	OUT	Bool	Nuevos datos listos – Si es TRUE (verdadero), el bit indica que las salidas contienen datos de una petición de comunicación nueva.
ERROR	OUT	Bool	Ha ocurrido un error – Si es TRUE (verdadero), indica que ha ocurrido un error y la salida STATUS es válida. Todas las demás salidas se ponen a cero cuando ocurre un error. Los errores de comunicación se notifican sólo en las salidas ERROR y STATUS de la instrucción USS_PORT.
STATUS	OUT	UInt	Valor de estado de la petición. Indica el resultado del ciclo. Esta no es una palabra de estado devuelta del accionamiento.
RUN_EN	OUT	Bool	Funcionamiento habilitado – Este bit indica si está funcionando el accionamiento.
D_DIR	OUT	Bool	Sentido del accionamiento – Este bit indica si el accionamiento está funcionando hacia delante.
INHIBIT	OUT	Bool	Accionamiento inhibido – Este bit indica el estado del bit de inhibición del accionamiento.
FAULT	OUT	Bool	Fallo del accionamiento – Este bit indica que el accionamiento ha registrado un fallo. El usuario debe corregir el problema y activar el bit F_ACK para borrar este bit si está activado.
SPEED	OUT	REAL	Velocidad actual del accionamiento (valor escalado de la palabra de estado 2 del accionamiento) – Valor de velocidad del accionamiento expresado como porcentaje de la velocidad configurada.
STATUS1	OUT	UInt	Palabra de estado 1 del accionamiento – Este valor contiene bits de estado fijos de un accionamiento.
STATUS3	OUT	UInt	Palabra de estado 3 del accionamiento – Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS4	OUT	UInt	Palabra de estado 4 del accionamiento – Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS5	OUT	UInt	Palabra de estado 5 del accionamiento – Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS6	OUT	UInt	Palabra de estado 6 del accionamiento – Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS7	OUT	UInt	Palabra de estado 7 del accionamiento – Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS8	OUT	UInt	Palabra de estado 8 del accionamiento – Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.

6.3.1.3 Instrucción USS_PORT

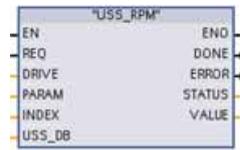
La instrucción USS_PORT gestiona la comunicación en la red USS. Generalmente, el programa contiene sólo una función USS_PORT por cada módulo de comunicación PtP. Cada llamada de esta función gestiona una transferencia hacia o desde un accionamiento. El programa debe ejecutar la función USS_PORT con suficiente frecuencia para impedir timeouts del accionamiento. Todas las funciones USS asociadas a una red USS y a un módulo de comunicación PtP deben utilizar el mismo bloque de datos instancia. USS_PORT se llama generalmente desde un OB de alarma de retardo para impedir timeouts del accionamiento y para que las actualizaciones de datos USS más recientes estén disponibles para las llamadas de USS_DRV.



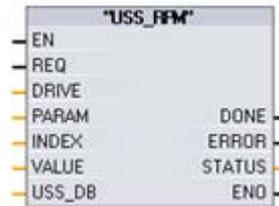
Parámetro	Tipo de parámetro	Tipo de datos	Descripción
PORT	IN	Puerto	Módulo de comunicación PtP. Identificador: Constante que puede referenciarse en la ficha "Constantes" de la tabla de variables predeterminadas.
BAUD	IN	Dint	Velocidad de transferencia que debe usarse para la comunicación USS.
USS_DB	IN	Dint	Referencia al DB instancia que se crea e inicializa cuando se inserta una instrucción USS_DRV en el programa.
ERROR	OUT	Bool	Si es TRUE (verdadero), este parámetro indica que ha ocurrido un error y la salida STATUS es válida.
STATUS	OUT	UInt	Valor de estado de la petición. Indica el resultado del ciclo o inicialización. Encontrará más información acerca de algunos códigos de estado en la variable "USS_Extended_Error".

6.3.1.4 Instrucción USS_RPM

KOP



FUP



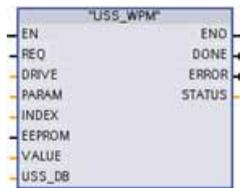
La instrucción USS_RPM lee un parámetro del accionamiento. Todas las funciones USS asociadas a una red USS y a un módulo de comunicación PtP deben utilizar el mismo bloque de datos. USS_RPM debe llamarse desde el OB principal.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	Enviar petición: Si es TRUE (verdadero), indica que se desea una nueva petición de lectura. Esto se ignora si la petición para este parámetro ya está pendiente.
DRIVE	IN	USInt	Dirección del accionamiento: Esta entrada es la dirección del accionamiento USS. El rango válido está comprendido entre el accionamiento 1 y el 16.
PARAM	IN	UInt	Número de parámetro: Esta entrada designa el parámetro del accionamiento que se escribe. El rango de este parámetro está comprendido entre 0 y 2047. Para más información sobre cómo acceder a los parámetros que excedan este rango, consulte el manual del accionamiento.
INDEX	IN	UInt	Índice de parámetro: Esta entrada designa el índice de parámetro del accionamiento que se escribirá. Valor de 16 bits en el que el byte menos significativo es el valor de índice real en un rango de 0 a 255. El accionamiento también puede utilizar el byte más significativo. Este byte es específico del accionamiento. Para más información, consulte el manual del accionamiento.
USS_DB	IN	Variante	Referencia al DB instancia que se crea e inicializa cuando se inserta una instrucción USS_DRV en el programa.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Valor del parámetro que se ha leído y que es válido sólo si el bit DONE es TRUE (verdadero).
DONE	OUT	Bool	Listo: Si es TRUE (verdadero), indica que la salida VALUE contiene el valor del parámetro de lectura solicitado anteriormente. Este bit se activa cuando USS_DRV detecta los datos de respuesta de lectura del accionamiento. Este bit se desactiva en los casos siguientes: <ul style="list-style-type: none"> • Los datos de respuesta se solicitan mediante otra consulta USS_RPM ○ • En la segunda de las dos llamadas siguientes de USS_DRV

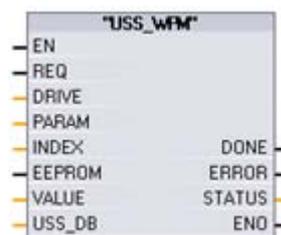
Parámetro	Tipo de parámetro	Tipo de datos	Descripción
ERROR	OUT	Bool	Ha ocurrido un error – Si es TRUE (verdadero), indica que ha ocurrido un error y la salida STATUS es válida. Todas las demás salidas se ponen a cero cuando ocurre un error. Los errores de comunicación se notifican sólo en las salidas ERROR y STATUS de la instrucción USS_PORT.
STATUS	OUT	UInt	Este es el valor de estado de la petición. Indica el resultado de la petición de lectura. Encontrará más información acerca de algunos códigos de estado en la variable "USS_Extended_Error".

6.3.1.5 Instrucción USS_WPM

KOP



FUP



La instrucción USS_WPM modifica un parámetro en el accionamiento. Todas las funciones USS asociadas a una red USS y a un módulo de comunicación PtP deben utilizar el mismo bloque de datos. USS_WPM debe llamarse desde el OB principal.

Nota

Operaciones de escritura en EEPROM

No utilice excesivamente la operación de escritura permanente en EEPROM. Minimice el número de operaciones de escritura en EEPROM para prolongar la vida útil de la EEPROM.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	Enviar petición: Si es TRUE (verdadero), indica que se desea una nueva petición de escritura. Esto se ignora si la petición para este parámetro ya está pendiente.
DRIVE	IN	USInt	Dirección del accionamiento: Esta entrada es la dirección del accionamiento USS. El rango válido está comprendido entre el accionamiento 1 y el 16.
PARAM	IN	UInt	Número de parámetro: Esta entrada designa el parámetro del accionamiento que se escribe. El rango de este parámetro está comprendido entre 0 y 2047. Para más información sobre cómo acceder a los parámetros que excedan este rango, consulte el manual del accionamiento.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
INDEX	IN	UInt	Índice de parámetro: Esta entrada designa el índice de parámetro del accionamiento que se escribirá. Valor de 16 bits en el que el byte menos significativo es el valor de índice real en un rango de 0 a 255. El accionamiento también puede utilizar el byte más significativo. Este byte es específico del accionamiento. Para más información, consulte el manual del accionamiento.
EEPROM	IN	Bool	Guardar en la EEPROM del accionamiento: Si es TRUE (verdadero), las operaciones de escritura en el parámetro del accionamiento se guardarán en la EEPROM de éste. Si es FALSE (falso), la operación de escritura será temporal por lo que no se conservará tras desconectar y volver a conectar la alimentación del accionamiento.
VALUE	IN	Word, Int, UInt, DWord, DInt, UInt, Real	Valor del parámetro en el que se debe escribir. Debe ser válido en la transición de REQ.
USS_DB	IN	Variante	Referencia al DB instancia que se crea e inicializa cuando se inserta una instrucción USS_DRV en el programa.
DONE	OUT	Bool	Listo: Si es TRUE (verdadero), indica que la entrada VALUE se ha escrito en el accionamiento. Este bit se activa cuando USS_DRV detecta los datos de respuesta de escritura del accionamiento. Este bit se desactiva en los casos siguientes: La confirmación del accionamiento de que ha finalizado la operación de escritura se solicita mediante otra consulta USS_WPM, o bien en la segunda de las dos llamadas siguientes de USS_DRV.
ERROR	OUT	Bool	Ha ocurrido un error: Si es TRUE (verdadero), indica que ha ocurrido un error y la salida STATUS es válida. Todas las demás salidas se ponen a cero cuando ocurre un error. Los errores de comunicación se notifican sólo en las salidas ERROR y STATUS de la instrucción USS_PORT.
STATUS	OUT	UInt	Este es el valor de estado de la petición. Indica el resultado de la petición de escritura. Encontrará más información acerca de algunos códigos de estado en la variable "USS_Extended_Error".

6.3.1.6 Códigos de estado USS

La salida STATUS de las funciones USS devuelve los códigos de estado de estas funciones.

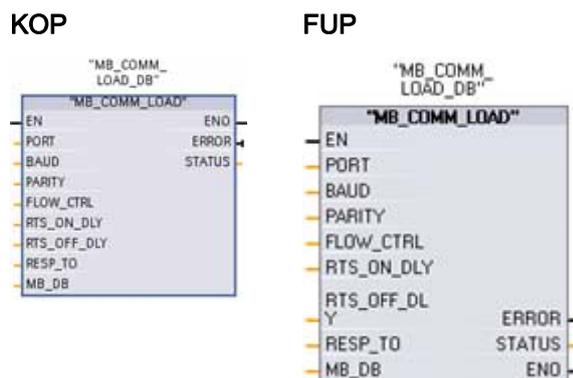
Valor de STATUS (W#16#...)	Descripción
0000	Sin error
8180	La longitud de la respuesta del accionamiento no concuerda con los caracteres recibidos del accionamiento. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
8181	El parámetro VALUE no es un tipo de datos WORD, REAL o DWORD
8182	El usuario ha introducido un parámetro en formato WORD y se ha recibido una respuesta en formato DWORD o REAL del accionamiento
8183	El usuario ha introducido un parámetro en formato DWORD o REAL y se ha recibido una respuesta en formato WORD del accionamiento
8184	El telegrama de respuesta del accionamiento tiene una suma de verificación incorrecta. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
8185	Dirección del accionamiento no válida (rango de direcciones válidas: 1-16)
8186	Consigna de velocidad fuera del rango válido (rango de SP de velocidad válido: -200% a 200%)
8187	Un número de accionamiento incorrecto ha respondido a la petición enviada. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
8188	Se ha indicado una longitud de palabra PZD no permitida (rango válido = 2, 4, 6 u 8 palabras)
8189	Se ha indicado una velocidad de transferencia no permitida
818A	Otra petición para este accionamiento está utilizando el canal de petición de parámetros
818B	El accionamiento no ha respondido a las peticiones ni reintentos. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
818C	El accionamiento ha devuelto un error avanzado relativo a la petición de parámetros. La descripción de los errores avanzados aparece a continuación de esta tabla.
818D	El accionamiento ha devuelto un error de acceso no permitido respecto a la petición de parámetros. Para más información sobre la limitación de acceso a los parámetros, consulte el manual del accionamiento.
818E	El accionamiento no se ha inicializado: Este código de error se devuelve a USS_RPM o USS_WPM si USS_DRV no se ha llamado por lo menos una vez para este accionamiento. De esta manera se impide que la inicialización de USS_DRV en el primer ciclo sobrescriba una petición pendiente de lectura o escritura de los parámetros, puesto que inicializa el accionamiento como entrada nueva. Para corregir este error, llame USS_DRV para este número de accionamiento.
80Ax-80Fx	Errores específicos que devuelven los FBs de comunicación PtP (punto a punto) llamados por la librería USS: La librería USS no modifica estos códigos de error que se definen en las descripciones de la instrucción PtP.

Códigos de error avanzados de los accionamientos USS

Los accionamientos USS soportan accesos de lectura y escritura a los parámetros internos de los accionamientos. Esta función permite controlar y configurar el accionamiento de forma remota. Las operaciones de acceso a los parámetros del accionamiento pueden fallar debido a errores tales como valores fuera de rango o peticiones no permitidas del modo actual del accionamiento. El accionamiento genera un código de error cuyo valor se devuelve en la variable "USS_Extended_Error" del DB instancia de USS_DRV. Este valor del código de error es válido únicamente para la última ejecución de una instrucción USS_RPM o USS_WPM. El código de error del accionamiento se deposita en la variable "USS_Extended_Error" si el código STATUS tiene el valor hexadecimal 818C. El valor del código de error de "USS_Extended_Error" depende del modelo de accionamiento. Los códigos de error avanzados para las operaciones de lectura y escritura de parámetros se describen en el manual del accionamiento.

6.3.2 MODBUS

6.3.2.1 MB_COMM_LOAD



La instrucción MB_COMM_LOAD configura un puerto del módulo de comunicación punto a punto (PtP) CM 1241 RS485 o CM 1241 RS232 para la comunicación vía el protocolo Modbus RTU.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
PORT	IN	UInt	Identificador del puerto de comunicación: Tras haber instalado el módulo CM en la configuración de dispositivos, el identificador de puerto aparece en la lista desplegable disponible en la conexión PORT del cuadro. Esta constante también se puede referenciar en la ficha "Constantes" de la tabla de variables predeterminadas.
BAUD	IN	UDInt	Selección de la velocidad de transferencia: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200 Los demás valores no son válidos.
PARITY	IN	UInt	Selección de paridad: <ul style="list-style-type: none"> • 0 – Ninguna • 1 – Impar • 2 – Par

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
FLOW_CTRL	IN	UInt	Selección del control de flujo: <ul style="list-style-type: none"> • 0 – (ajuste predeterminado) Sin control de flujo • 1 – Control de flujo por hardware con RTS siempre ON (no es aplicable a los puertos RS485) • 2 - Control de flujo por hardware con RTS conmutado
RTS_ON_DLY	IN	UInt	Selección de retardo RTS ON: <ul style="list-style-type: none"> • 0 – (ajuste predeterminado) Sin retardo desde RTS ON hasta que se transmite el primer carácter del mensaje • 1 a 65535 – Retardo en milisegundos desde RTS ON hasta que se transmite el primer carácter del mensaje (no es aplicable a los puertos RS-485). Los retardos RTS se aplican siempre independientemente de la selección de FLOW_CTRL.
RTS_OFF_DLY	IN	UInt	Selección de retardo RTS OFF: <ul style="list-style-type: none"> • 0 – (ajuste predeterminado) Sin retardo desde el último carácter transmitido hasta que se desactiva RTS • 1 a 65535 – Retardo en milisegundos desde el último carácter transmitido hasta que se desactiva RTS (no es aplicable a los puertos RS-485). Los retardos RTS se aplican siempre independientemente de la selección de FLOW_CTRL.
RESP_TO	IN	UInt	Timeout de respuesta: Tiempo en milisegundos permitido por el MB_MASTER para la respuesta del esclavo. Si el esclavo no responde en este tiempo, MB_MASTER repetirá la petición o la finalizará con un error si se ha enviado el número de reintentos indicado. 5 ms a 65535 ms (valor predeterminado = 1000ms).
MB_DB	IN	Variante	Referencia al bloque de datos instancia que utilizan las instrucciones MB_MASTER o MB_SLAVE. Una vez insertadas las instrucciones MB_SLAVE o MB_MASTER en el programa, el identificador de DB aparece en la lista desplegable disponible en la conexión MB_DB del cuadro.
ERROR	OUT	Bool	Error: <ul style="list-style-type: none"> • 0 – No se ha detectado ningún error • 1 – Indica que se ha detectado un error y el código de error depositado en el parámetro STATUS es válido
STATUS	OUT	Word	Código de error de configuración del puerto

MB_COMM_LOAD se ejecuta para configurar un puerto para el protocolo Modbus RTU. Una vez configurado el puerto, la comunicación se realiza en el Modbus ejecutando las instrucciones MB_SLAVE o MB_MASTER.

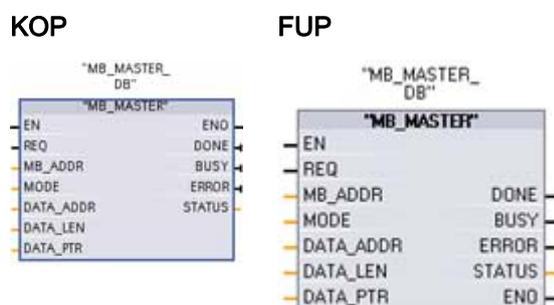
MB_COMM_LOAD se debería llamar una vez para inicializar el puerto. Sólo es necesario volver a llamar MB_COMM_LOAD si debe cambiar uno de los parámetros de comunicación. Es posible llamar MB_COMM_LOAD desde un OB de arranque y ejecutar esta instrucción una vez, o bien utilizar la marca de sistema del primer ciclo con el fin de iniciar la llamada para ejecutarla una vez.

Es preciso utilizar una instancia de MB_COMM_LOAD para configurar cada uno de los puertos de todo módulo de comunicación utilizado para la comunicación Modbus. Un bloque de datos instancia MB_COMM_LOAD unívoco se debe asignar a cada puerto utilizado. La CPU S7-1200 está limitada a 3 módulos de comunicación.

Un bloque de datos instancia se asigna cuando las instrucciones MB_MASTER o MB_SLAVE se insertan en el programa. Este bloque de datos instancia se referencia cuando se especifica el parámetro MB_DB de la instrucción MB_COMM_LOAD.

Valor de STATUS (W#16#...)	Descripción
0000	Sin error
8180	ID de puerto no válida
8181	Velocidad de transferencia no válida
8182	Paridad no válida
8183	Valor de control de flujo no válido
8184	Valor de timeout de respuesta no válido
8185	Puntero MB_DB incorrecto al DB instancia para un MB_MASTER o MB_SLAVE

6.3.2.2 MB_MASTER



La instrucción MB_MASTER permite al programa comunicarse como maestro Modbus utilizando un puerto del módulo de comunicación punto a punto (PtP) CM 1241 RS485 o CM 1241 RS232. Es posible acceder a los datos de uno o más esclavos Modbus.

Un bloque de datos instancia se asigna cuando la instrucción MB_MASTER se inserta en el programa. El nombre de este bloque de datos instancia MB_MASTER se utiliza cuando se especifica el parámetro MB_DB de la instrucción MB_COMM_LOAD.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	Entrada que indica la petición: <ul style="list-style-type: none"> • 0 – Ninguna petición • 1 – Petición de transmitir datos a uno o varios esclavos Modbus
MB_ADR	IN	USInt	Dirección de estación Modbus RTU: Rango de direcciones válidas: 0 a 247 El valor 0 está reservado para enviar un mensaje Broadcast a todos los esclavos Modbus. Los códigos de función Modbus 05, 06, 15 y 16 son los únicos que se soportan para el Broadcast.
MODE	IN	USInt	Selección de modo: Determina el tipo de petición: lectura, escritura o diagnóstico Los detalles se indican en la tabla de funciones Modbus que aparece más abajo.
DATA_ADDR	IN	UDInt	Dirección inicial en el esclavo: Determina la dirección inicial de los datos a los que debe accederse en el esclavo Modbus. Las direcciones válidas se indican en la tabla de funciones Modbus que aparece más abajo.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
DATA_LEN	IN	UInt	Longitud de datos: Determina el número de bits o palabras a las que debe accederse en esta petición. Las longitudes válidas se indican en la tabla de funciones Modbus que aparece más abajo.
DATA_PTR	IN	Variante	Puntero a los datos: Apunta a la dirección del DB en la CPU de los datos que se están escribiendo o leyendo. El tipo de DB no puede tener el atributo "Sólo con direccionamiento simbólico". Tenga en cuenta la indicación relativa a DATA_PTR que aparece más abajo.
NDR	OUT	Bool	Nuevos datos listos: <ul style="list-style-type: none"> • 0 – Transacción no finalizada • 1 – Indica que la instrucción MB_MASTER ha finalizado la transacción solicitada con el o los esclavos Modbus
BUSY	OUT	Bool	Ocupado: <ul style="list-style-type: none"> • 0 – No hay ninguna transacción MB_MASTER en curso • 1 – Transacción MB_MASTER en curso
ERROR	OUT	Bool	Error: <ul style="list-style-type: none"> • 0 – No se ha detectado ningún error • 1 – Indica que se ha detectado un error y el código de error depositado en el parámetro STATUS es válido
STATUS	OUT	Word	Código de condición de ejecución

Reglas de comunicación del maestro Modbus

- MB_COMM_LOAD debe ejecutarse para configurar un puerto antes de que la instrucción MB_MASTER pueda comunicarse con ese puerto.
- Si un puerto debe utilizarse para iniciar peticiones de maestro Modbus, MB_SLAVE no podrá utilizar este puerto. Una o más instancias de ejecución de MB_MASTER pueden utilizarse en este puerto.
- Las instrucciones Modbus no utilizan eventos de alarma de comunicación para controlar el proceso de comunicación. El programa debe consultar la instrucción MB_MASTER para transmitir y recibir condiciones completas.
- Si el programa opera un maestro Modbus y utiliza MB_MASTER para enviar una petición a un esclavo, MB_MASTER se deberá seguir ejecutando hasta que se devuelva la respuesta del esclavo.
- Todas las ejecuciones de MB_MASTER para un determinado puerto deben llamarse desde un mismo OB (o clase de prioridad de OB).

Parámetro REQ

Si el valor de REQ es FALSE (falso), no hay ninguna petición.

Si el valor de REQ es TRUE (verdadero), hay una petición de transmitir datos a uno o varios esclavos Modbus.

Esta entrada debe recibir suministro de un contacto disparado por un flanco ascendente en la primera llamada de la ejecución de MB_MASTER. El impulso disparado por flancos llama una vez la petición de transmisión. Todas las entradas se capturan y no se modifican durante una petición y respuesta disparadas por esta entrada.

MB_MASTER inicia internamente una máquina de estados para asegurarse de que ninguna otra MB_MASTER pueda lanzar una petición hasta que no haya finalizado esta petición.

Además, si la misma instancia de la llamada del FB MB_MASTER se ejecuta nuevamente con la entrada REQ = TRUE (verdadero) antes de que se finalice la petición, no se realizarán más transmisiones. No obstante, en cuanto finalice la petición, se lanzará otra petición si MB_MASTER se ejecuta con la entrada REQ puesta a TRUE.

Los parámetros DATA_ADDR y MODE seleccionan el tipo de función Modbus

DATA_ADDR (dirección Modbus inicial en el esclavo): Determina la dirección inicial de los datos a los que debe accederse en el esclavo Modbus.

MB_MASTER utiliza la entrada MODE en vez de una entrada de código de función. La combinación de MODE y el rango de direcciones Modbus determinan el código de función utilizado en el mensaje Modbus real. La tabla siguiente muestra la correlación entre el parámetro MODE de MBUS_MASTER, el código de función Modbus y el rango de direcciones Modbus.

Funciones Modbus de MB_MASTER				
	Parámetro de dirección DATA_ADDR Modbus	Tipo de dirección	Parámetro de longitud de datos DATA_LEN Modbus	Función Modbus
Modo 0				
Leer	00001 a 09999	Bits de salida	1 a 2000	01H
	10001 - 19999	Bits de entrada	1 a 2000	02H
	30001 - 39999	Registros de entrada	1 a 125	04H
	40001 a 49999 400001 a 465536 (ampliado)	Registros de retención	1 a 125	03H
Modo 1				
Escribir	00001 a 09999	Bits de salida	1 (bit individual)	05H
	40001 a 49999 400001 a 465536 (ampliado)	Registros de retención	1 (palabra individual)	06H
	00001 a 09999	Bits de salida	2 a 1968	15H
	40001 a 49999 400001 a 465536 (ampliado)	Registros de retención	2 a 123	16H

Funciones Modbus de MB_MASTER				
Modo 2				
Algunos esclavos Modbus no soportan la escritura en bits o palabras individuales con las funciones Modbus 05H y 06H. En estos casos, el modo 2 se utiliza para forzar la escritura en bits o palabras individuales utilizando las funciones Modbus 15H y 16H.				
Escribir	00001 a 09999	Bits de salida	1 a 1968	15H
	40001 a 49999	Registros de retención	1 a 123	16H
	400001 a 465536 (ampliado)			
Modo 11				
<ul style="list-style-type: none"> • Lee una palabra contadora de eventos del esclavo Modbus referenciado como entrada a MB_ADDR • En un esclavo Modbus S7-1200 Siemens, este contador se incrementa cada vez que el esclavo recibe una petición de lectura o escritura (no Broadcast) válida de un maestro Modbus. • El valor devuelto se almacena en la dirección de palabra especificada como entrada a DATA_PTR. • Para este modo no se requiere un DATA_LEN válido. 				
Modo 80				
<ul style="list-style-type: none"> • Verifica el estado de comunicación del esclavo Modbus referenciado como entrada a MB_ADDR • El ajuste del bit de salida NDR de la instrucción MB_MASTER indica que el esclavo Modbus direccionado ha respondido con datos de respuesta apropiados. • No se devuelven datos al programa. • Para este modo no se requiere un DATA_LEN válido. 				
Modo 81				
<ul style="list-style-type: none"> • Inicializa el contador de eventos (devuelto por el modo 11) en el esclavo Modbus referenciado como entrada a MB_ADDR • El ajuste del bit de salida NDR de la instrucción MB_MASTER indica que el esclavo Modbus direccionado ha respondido con datos de respuesta apropiados. • No se devuelven datos al programa. • Para este modo no se requiere un DATA_LEN válido. 				

Parámetro DATA_PTR

El parámetro DATA_PTR apunta a la dirección de origen local o de destino (la dirección de la CPU S7-1200) de los datos que se están escribiendo o leyendo, respectivamente. Si la instrucción MB_MASTER se utiliza para crear un maestro Modbus, es preciso crear un bloque de datos global que permita almacenar las operaciones de lectura y escritura en los esclavos Modbus.

Nota

El parámetro DATA_PTR debe referenciar un tipo de bloque de datos global que haya sido creado sin el atributo "Sólo con direccionamiento simbólico".

La casilla de verificación "Sólo con direccionamiento simbólico" se debe desactivar al agregar un bloque de datos nuevo para crear un tipo de DB global clásico.

Estructuras del bloque de datos para el parámetro DATA_PTR

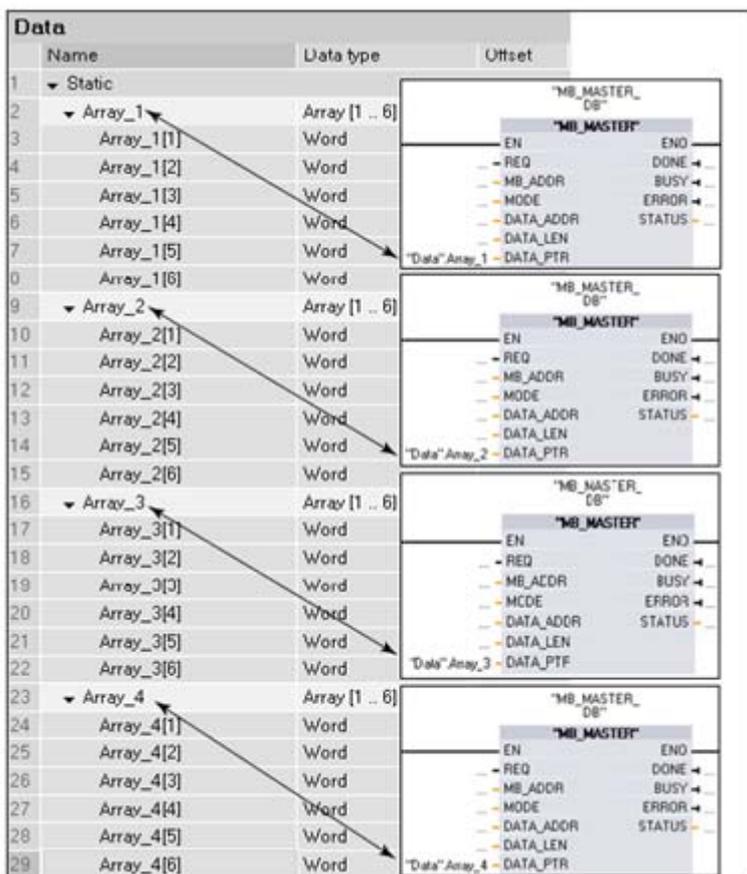
- Estos tipos de datos son válidos para la **lectura de palabras** de las direcciones Modbus 30001 a 39999, 40001 a 49999 y 400001 a 465536, así como para la **escritura de palabras** en las direcciones Modbus 40001 a 49999 y 400001 a 465536.
 - Matriz estándar de tipos de datos WORD, UINT o INT como se indica más abajo.
 - Estructura WORD, UINT o INT con nombres, en la que todo elemento tiene un nombre unívoco y un tipo de datos de 16 bits.
 - Estructura compleja con nombres, en la que todo elemento tiene un nombre unívoco y un tipo de datos de 16 ó 32 bits.
- Para la lectura y escritura de bits de las direcciones Modbus 00001 a 09999 y 10001 a 19999.
 - Matriz estándar de tipos de datos booleanos.
 - Estructura booleana con nombres que incluye variables booleanas con nombres unívocos.
- Aunque no es imprescindible, se recomienda que cada instrucción MB_MASTER tenga un área propia en un bloque de datos global. El motivo de esta recomendación es que la posibilidad de que se corrompan los datos aumenta si varias instrucciones MB_MASTER están leyendo y escribiendo en la misma área de un bloque de datos global.
- No es necesario que las áreas de datos de DATA_PTR se encuentren en el mismo bloque de datos global. Es posible crear un bloque de datos con varias áreas para lecturas Modbus, uno para escrituras Modbus, o bien uno para cada estación esclava.
- Todas las matrices del ejemplo que aparece abajo se crean como matrices en base 1 [1 ... ##]. Estas matrices se podrían haber creado como matrices en base 0 [0 ... ###] o una mezcla de base 0 y base 1.

Ejemplo de instrucciones MB_MASTER que acceden a bloques de datos globales DATA_PTR

El ejemplo del bloque de datos global que aparece abajo utiliza 4 matrices de 6 palabras con nombres unívocos para almacenar los datos de peticiones Modbus. Aunque las matrices de este ejemplo son de igual tamaño, pueden tener un tamaño cualquiera. Aquí se muestran con un mismo tamaño para simplificar los ejemplos. Cada matriz también se podría reemplazar por una estructura de datos que incluya nombres de variables más descriptivos y tipos de datos mixtos. La descripción del parámetro HR_DB de la instrucción MB_SLAVE instrucción (Página 231) ofrece ejemplos de estructuras de datos alternativas.

Los ejemplos de la instrucción MB_MASTER que aparecen abajo muestran sólo el parámetro DATA_PTR y no los demás parámetros necesarios. El objetivo de estos ejemplos es mostrar cómo la instrucción MB_MASTER utiliza el bloque de datos DATA_PTR.

Las flechas indican cómo las distintas matrices se asocian a diferentes instrucciones MB_MASTER.



El primer elemento de cualquier matriz o estructura es siempre el primer origen o destino de cualquier actividad de lectura o escritura Modbus. Todos los casos descritos abajo se basan en el diagrama de arriba.

Caso 1: Si la primera instrucción MB_MASTER lee 3 palabras de datos de la dirección Modbus 40001 en cualquier esclavo Modbus válido, sucederá lo siguiente:

La palabra de la dirección 40001 se almacena en "Data".Array_1[1].

La palabra de la dirección 40002 se almacena en "Data".Array_1[2].

La palabra de la dirección 40003 se almacena en "Data".Array_1[3].

Caso 2: Si la primera instrucción MB_MASTER lee 4 palabras de datos de la dirección Modbus 40015 en cualquier esclavo Modbus válido, sucederá lo siguiente:

La palabra de la dirección 40015 se almacena en "Data".Array_1[1].

La palabra de la dirección 40016 se almacena en "Data".Array_1[2].

La palabra de la dirección 40017 se almacena en "Data".Array_1[3].

La palabra de la dirección 40018 se almacena en "Data".Array_1[4].

Caso 3: Si la segunda instrucción MB_MASTER lee 2 palabras de datos de la dirección Modbus 30033 en cualquier esclavo Modbus válido, sucederá lo siguiente:

La palabra de la dirección 30033 se almacena en "Data".Array_2[1].

La palabra de la dirección 30034 se almacena en "Data".Array_2[2].

Caso 4: Si la tercera instrucción MB_MASTER escribe 4 palabras de datos en la dirección Modbus 40050 en cualquier esclavo Modbus válido, sucederá lo siguiente:

La palabra de "Data".Array_3[1] se escribe en la dirección Modbus 40050.

La palabra de "Data".Array_3[2] se escribe en la dirección Modbus 40051.

La palabra de "Data".Array_3[3] se escribe en la dirección Modbus 40052.

La palabra de "Data".Array_3[4] se escribe en la dirección Modbus 40053.

Caso 5: Si la tercera instrucción MB_MASTER escribe 3 palabras de datos en la dirección Modbus 40001 en cualquier esclavo Modbus válido, sucederá lo siguiente:

La palabra de "Data".Array_3[1] se escribe en la dirección Modbus 40001.

La palabra de "Data".Array_3[2] se escribe en la dirección Modbus 40002.

La palabra de "Data".Array_3[3] se escribe en la dirección Modbus 40003.

Caso 6: Si la cuarta instrucción MB_MASTER utiliza el modo 11 (consultar contaje de mensajes válidos) de cualquier esclavo Modbus válido, sucederá lo siguiente:

La palabra contadora se almacena en "Data".Array_4[1].

Ejemplo de lectura y escritura de bits utilizando direcciones de palabra como entrada DATA_PTR

Tabla 6- 1 Caso 7: Leer 4 bits de salida comenzando en la dirección Modbus 00001

Valores de entrada de MB_MASTER		Valores del esclavo Modbus	
MB_ADDR	27 (ejemplo de esclavo)	00001	ON
MODE	0 (lectura)	00002	ON
DATA_ADDR	00001 (salidas)	00003	OFF
DATA_LEN	4	00004	ON
DATA_PTR	"Data".Array_4	00005	ON
		00006	OFF
		00007	ON
		00008	OFF

Valores de "Data".Array_4[1] después de la petición Modbus	
Byte MS (más significativo)	Byte LS (menos significativo)
xxxx-1011	xxxx-xxxx
x indica que no se modifican los datos	

Tabla 6- 2 Caso 8: Leer 12 bits de salida comenzando en la dirección Modbus 00003

Valores de entrada de MB_MASTER		Valores del esclavo Modbus			
MB_ADDR	27 (ejemplo de esclavo)	00001	ON	00010	ON
MODE	0 (lectura)	00002	ON	00011	OFF
DATA_ADDR	00003 (salidas)	00003	OFF	00012	OFF
DATA_LEN	12	00004	ON	00013	ON
DATA_PTR	"Data".Array_4	00005	ON	00014	OFF
		00006	OFF	00015	ON
		00007	ON	00016	ON
		00008	ON	00017	OFF
		00009	OFF	00018	ON

Valores de "Data".Array_4[1] después de la petición Modbus	
Byte MS	Byte LS
1011-0110	xxxx-0100-
x indica que no se modifican los datos	

Tabla 6- 3 Caso 9: Escribir 5 bits de salida comenzando en la dirección Modbus 00001

Valores de entrada de MB_MASTER		Salidas del esclavo (antes)		Salidas del esclavo (después)	
MB_ADDR	27 (ejemplo de esclavo)	00001	ON		OFF
MODE	1 (escritura)	00002	ON		ON
DATA_ADDR	00001 (salidas)	00003	OFF		ON
DATA_LEN	5	00004	ON		OFF
DATA_PTR	"Data".Array_4	00005	ON		ON
		00006	OFF		No se modifican
		00007	ON		No se modifican
		00008	ON		No se modifican
		00009	OFF		No se modifican

Valores de "Data".Array_4[1] para la petición de escritura Modbus	
Byte MS	Byte LS
xxx1-0110	xxxxx-xxxx
x indica que los datos no se utilizan en la petición Modbus	

Tabla 6- 4 Caso 10: Leer 22 bits de salida comenzando en la dirección Modbus 00003

Valores de entrada de MB_MASTER		Valores del esclavo Modbus			
MB_ADDR	27 (ejemplo de esclavo)	00001	ON	00014	ON
MODE	0 (lectura)	00002	ON	00015	OFF
DATA_ADDR	00003 (salidas)	00003	OFF	00016	ON
DATA_LEN	22	00004	ON	00017	ON
DATA_PTR	"Data".Array_4	00005	ON	00018	OFF
		00006	OFF	00019	ON
		00007	ON	00020	ON
		00008	ON	00021	OFF
		00009	ON	00022	ON
		00010	OFF	00023	ON
		00011	OFF	00024	OFF
		00012	ON	00025	OFF
		00013	OFF	00026	ON

Valores de "Data".Array_4[1] después de la petición Modbus	
Byte MS	Byte LS
0111-0110	0110-1010

Valores de "Data".Array_4[2] después de la petición Modbus	
Byte MS	Byte LS
xx01-1011	xxxx-xxxx
x indica que no se modifican los datos	

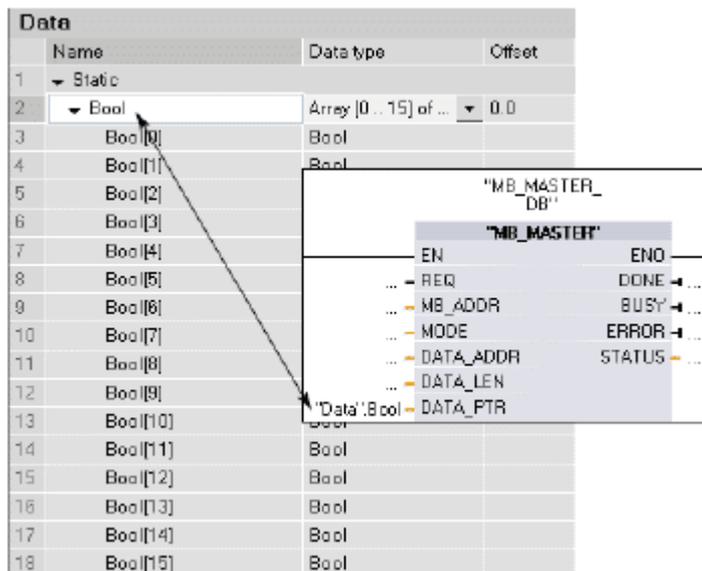
Ejemplo de lectura y escritura de bits utilizando direcciones BOOL como entrada DATA_PTR

Aunque las operaciones de lectura y escritura Modbus en direcciones de bit pueden gestionarse utilizando direcciones de palabra, también es posible configurar las áreas DATA_PTR como tipos de datos booleanos, estructuras o matrices con el fin de ofrecer una correlación uno a uno directa para el primer bit que se lee o escribe utilizando una instrucción MB_MASTER.

Si se utilizan estructuras o matrices booleanas, se recomienda que el tamaño de datos sea un múltiplo de 8 bits (en límites de bytes). Por ejemplo, si se crea una matriz booleana de 10 bits, el software STEP 7 Basic asignará 16 bits (2 bytes) del bloque de datos global para los 10 bits. Dentro del bloque de datos, estos bits se guardarían como byte1 [xxxx xxxx] byte2 [--- --xx], donde x indica las ubicaciones de datos accesibles e – indica las ubicaciones inaccesibles. Aunque se permiten peticiones Modbus con una longitud máxima de 16 bits, los 6 bits superiores se dispondrían en posiciones de memoria del byte 2 no referenciadas y a las que no puede acceder el programa.

Las áreas booleanas pueden crearse como matriz de valores booleanos o estructura de variables booleanas. Ambos métodos funcionan de manera idéntica y se diferencian sólo en la forma como se crean y acceden en el programa.

La siguiente vista del editor de bloques de datos globales muestra una matriz individual de 16 valores booleanos creados en base 0. Esta matriz también se podría haber creado como matriz en base 1. La flecha muestra cómo esta matriz se asocia con una instrucción MB_MASTER.



Los casos 11 y 12 muestran la correlación de las direcciones Modbus con las direcciones de matrices booleanas.

Tabla 6- 5 Caso 11: Escribir 5 bits de salida comenzando en la dirección Modbus 00001

Valores de entrada de MB_MASTER		Salidas del esclavo (antes)		Datos de DATA_PTR	Salidas del esclavo (después)
MB_ADDR	27 (ejemplo de esclavo)	00001	ON	"Data".Bool[0]=FALSE	OFF
MODE	1 (escritura)	00002	ON	"Data".Bool[1]=TRUE	ON
DATA_ADDR	00001 (salidas)	00003	OFF	"Data".Bool[2]=TRUE	ON
DATA_LEN	5	00004	ON	"Data".Bool[3]=FALSE	OFF
DATA_PTR	"Data".Bool	00005	ON	"Data".Bool[4]=FALSE	OFF
		00006	OFF		No se modifican
		00007	ON		No se modifican
		00008	OFF		No se modifican

Tabla 6- 6 Caso 12: Leer 15 bits de salida comenzando en la dirección Modbus 00004

Valores de entrada de MB_MASTER		Valor del esclavo Modbus		Datos de DATA_PTR (después)
MB_ADDR	27 (ejemplo de esclavo)	00001	ON	
MODE	0 (lectura)	00002	ON	
DATA_ADDR	00003 (salidas)	00003	OFF	"Data".Bool[0]=FALSE
DATA_LEN	15	00004	ON	"Data".Bool[1]=TRUE
DATA_PTR	"Data".Bool	00005	ON	"Data".Bool[2]=TRUE
		00006	OFF	"Data".Bool[3]=FALSE
		00007	ON	Data".Bool[4]=TRUE
		00008	ON	Data".Bool[5]=TRUE
		00009	ON	Data".Bool[6]=TRUE
		00010	OFF	Data".Bool[7]=FALSE
		00011	OFF	Data".Bool[8]=FALSE
		00012	ON	Data".Bool[9]=TRUE
		00013	OFF	Data".Bool[10]=FALSE
		00014	ON	Data".Bool[11]=TRUE
		00015	OFF	Data".Bool[12]=FALSE
		00016	ON	Data".Bool[13]=TRUE
		00017	ON	Data".Bool[14]=TRUE
		00018	OFF	
		00019	ON	

Códigos de condición

Valor de STATUS (W#16#....)	Descripción
0000	Sin error
80C8	El timeout de respuesta indicado (consulte RCVTIME o MSGTIME) es 0.
80D1	El receptor ha lanzado una petición de control de flujo para suspender una transmisión activa y no ha habilitado nuevamente la transmisión en el tiempo de espera indicado. Este error también se genera durante el control de flujo por hardware cuando el receptor no confirma CTS en el tiempo de espera indicado.
80D2	La petición de transmisión se ha cancelado porque no se recibe ninguna señal DSR del DCE.
80E0	El mensaje se ha terminado porque el búfer de recepción está lleno.
80E1	El mensaje se ha terminado debido a un error de paridad.
80E2	El mensaje se ha terminado debido a un error de trama.
80E3	El mensaje se ha terminado debido a un error de desbordamiento.
80E4	El mensaje se ha terminado debido a que la longitud especificada excede el tamaño del búfer total.
8180	ID de puerto no válida
8186	Dirección de estación Modbus no válida
8188	Valor de MODE no válido o modo de escritura para leer sólo área de direcciones del esclavo
8189	Valor de dirección de datos no válido
818A	Valor de longitud de datos no válido
818B	<ul style="list-style-type: none"> Puntero no válido al origen/destino de datos local: tamaño incorrecto
818C	Puntero a un tipo de DB de tipo seguro DATA_PTR (debe ser un tipo de DB clásico)
8200	El puerto está ocupado porque está procesando una petición de transmisión

6.3.2.3 MB_SLAVE

La instrucción MB_SLAVE permite al programa comunicarse como esclavo Modbus utilizando un puerto del módulo de comunicación punto a punto (PtP) CM 1241 RS485 o CM 1241 RS232. Un maestro Modbus RTU puede lanzar una petición y el programa responde ejecutando la instrucción MB_SLAVE.

Es preciso asignar un bloque de datos instancia unívoco al insertar la instrucción MB_SLAVE en el programa. El nombre de este bloque de datos instancia MB_SLAVE se utiliza cuando se especifica el parámetro MB_DB de la instrucción MB_COMM_LOAD.

Los códigos de las funciones de comunicación Modbus (1, 2, 4, 5 y 15) pueden leer y escribir bits y palabras directamente en la memoria imagen de proceso de las entradas y salidas del PLC. La tabla siguiente muestra el mapeo de las direcciones Modbus en la memoria imagen de proceso de la CPU.

Funciones Modbus de MB_SLAVE					S7-1200	
Códigos	Función	Área de datos	Rango de direcciones		Área de datos	Dirección de la CPU
01	Leer bits	Salida	1	a	8192	Memoria imagen de proceso de las salidas Q0.0 a Q1023.7
02	Leer bits	Entrada	10001	a	18192	Memoria imagen de proceso de las entradas I0.0 a I1023.7
04	Leer palabras	Entrada	30001	a	30512	Memoria imagen de proceso de las entradas IW0 a IW1022
05	Escribir bit	Salida	1	a	8192	Memoria imagen de proceso de las salidas Q0.0 a Q1023.7
15	Escribir bits	Salida	1	a	8192	Memoria imagen de proceso de las salidas Q0.0 a Q1023.7

Los códigos de las funciones de comunicación Modbus (3, 6, 16) utilizan un bloque de datos de registro de retención Modbus propio y unívoco que debe crearse antes de poder especificar el parámetro MB_HOLD_REG de la instrucción MB_SLAVE. La tabla siguiente muestra el mapeo del registro de retención Modbus en la dirección DB MB_HOLD_REG del PLC.

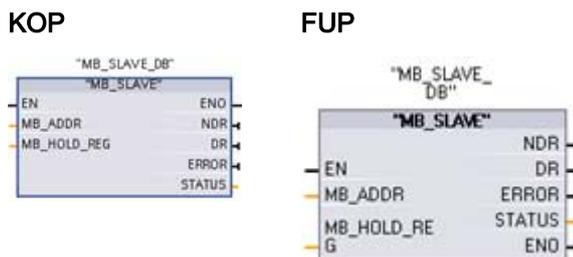
Funciones Modbus de MB_SLAVE				S7-1200	
Códigos	Función	Área de datos	Rango de direcciones	Área de datos DB de la CPU	Dirección DB de la CPU
03	Leer palabras	Registro de retención	40001 a 49999	MB_HOLD_REG	Palabras 1 a 9999
			400001 a 465535		Palabras 1 a 65535
06	Escribir palabra	Registro de retención	40001 a 49999	MB_HOLD_REG	Palabras 1 a 9999
			400001 a 465535		Palabras 1 a 65535
16	Escribir palabras	Registro de retención	40001 a 49999	MB_HOLD_REG	Palabras 1 a 9999
			400001 a 465535		Palabras 1 a 65535

La tabla siguiente muestra las funciones de diagnóstico Modbus soportadas.

Funciones de diagnóstico Modbus de MB_SLAVE en el S7-1200		
Códigos	Subfunción	Descripción
08	0000H	Devolver datos de consulta del test de eco: La instrucción MB_SLAVE responde al maestro Modbus con una palabra de datos que se están recibiendo datos.
08	000AH	Borrar contador de eventos de comunicación: La instrucción MB_SLAVE borra el contador de eventos de comunicación utilizado para la función Modbus 11.
11		Consultar contador de eventos de comunicación: La instrucción MB_SLAVE utiliza un contador de eventos de comunicación interno para registrar el número de peticiones de lectura y escritura Modbus correctas que se envían al esclavo Modbus. El contador no se incrementa con las funciones 8 ni 11, ni tampoco con peticiones Broadcast. Tampoco se incrementa con peticiones que resulten en un error de comunicación (p. ej. errores de paridad o CRC).

La instrucción MB_SLAVE soporta peticiones de escritura Broadcast de cualquier maestro Modbus, mientras que la petición sea para acceder a direcciones válidas.

Independientemente de la validez de una petición, la instrucción MB_SLAVE no responde a un maestro Modbus como resultado de una petición Broadcast.



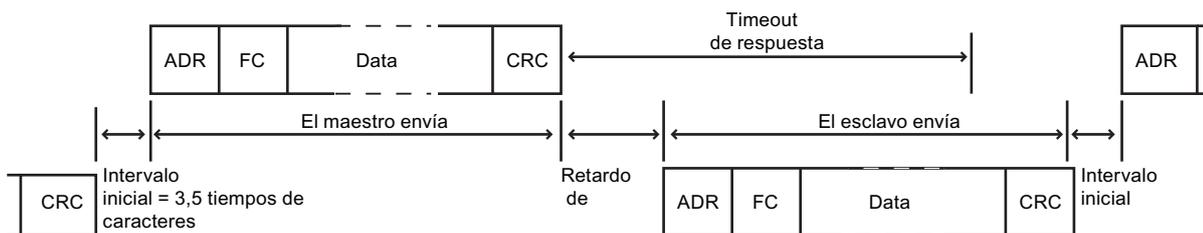
Parámetro	Tipo de parámetro	Tipo de datos	Descripción
MB_ADDR	IN	USINT	Dirección Modbus RTU (1 a 247): Dirección de estación del esclavo Modbus.
MB_HOLD_REG	IN	VARIANT	Puntero al DB del registro de retención Modbus. El DB del registro de retención debe ser un DB global clásico. Consulte la nota relativa a MB_HOLD_REG más abajo.
NDR	OUT	BOOL	Nuevos datos listos: <ul style="list-style-type: none"> 0 – No hay datos nuevos 1 – Indica que el maestro Modbus ha escrito datos nuevos
DR	OUT	BOOL	Lectura de datos: <ul style="list-style-type: none"> 0 – No se han leído datos 1 – Indica que el maestro Modbus ha leído datos
ERROR	OUT	BOOL	Error: <ul style="list-style-type: none"> 0 – No se ha detectado ningún error 1 – Indica que se ha detectado un error y el código de error depositado en el parámetro STATUS es válido.
STATUS	OUT	WORD	Código de error

Reglas de comunicación del esclavo Modbus

- MB_COMM_LOAD debe ejecutarse para configurar un puerto antes de que la instrucción MB_SLAVE pueda comunicarse con ese puerto.
- Si un puerto debe responder como esclavo a un maestro Modbus, MB_MASTER no podrá utilizar este puerto. Sólo se puede utilizar una instancia de MB_SLAVE en un determinado puerto.
- Las instrucciones Modbus no utilizan eventos de alarma de comunicación para controlar el proceso de comunicación. El programa debe controlar el proceso de comunicación consultando la instrucción MB_SLAVE para comprobar si se han finalizado las operaciones de transmisión y recepción.
- La instrucción MB_SLAVE debe ejecutarse periódicamente a una frecuencia que permita responder sin demora a las peticiones entrantes de un maestro Modbus.
- MB_SLAVE se debería llamar en cada ciclo desde un OB de ciclo.

Funcionamiento

MB_SLAVE debe ejecutarse periódicamente para recibir todas las peticiones del maestro Modbus y responder según sea necesario. La frecuencia de ejecución de MB_SLAVE depende del periodo de timeout de respuesta del maestro Modbus. Esto se ilustra en el diagrama siguiente.



El periodo de timeout de respuesta es el tiempo que un maestro Modbus espera hasta el inicio de la respuesta de un esclavo Modbus. Este periodo no está definido en el protocolo Modbus, sino que es un parámetro de todo maestro Modbus. La frecuencia de ejecución (es decir, el tiempo que transcurre entre dos ejecuciones) de MB_SLAVE debe basarse en los parámetros particulares del maestro Modbus. Como mínimo, MB_SLAVE debería ejecutarse dos veces en el periodo de timeout de respuesta del maestro Modbus.

Ejemplos del parámetro MB_HOLD_REG

MB_HOLD_REG es un puntero al bloque de datos del registro de retención Modbus. Este DB se utiliza para retener valores de datos a los que puede acceder un maestro Modbus (con operaciones de lectura o escritura). Es preciso crear el bloque de datos y asignar la estructura del tipo de datos que se debe leer y escribir, antes de poder utilizarlo con la instrucción MB_SLAVE.

Nota

El bloque de datos del registro de retención Modbus debe referenciar un tipo de bloque de datos global que haya sido creado sin el atributo "Sólo con direccionamiento simbólico".

La casilla de verificación "Sólo con direccionamiento simbólico" se debe desactivar al agregar un bloque de datos nuevo para crear un tipo de DB global clásico.

Los registros de retención pueden utilizar las siguientes estructuras de datos DB:

- Matriz estándar de palabras
- Estructura de palabras con nombres
- Estructura compleja con nombres

Los siguientes ejemplos de programación muestran cómo utilizar el parámetro MB_HOLD_REG para gestionar estas estructuras de datos DB.

Ejemplo 1 - Matriz estándar de palabras

Este registro de retención de ejemplo es una matriz de palabras. Las asignaciones de tipos de datos pueden cambiarse a otros tipos en formato de palabra (INT y UINT).

Ventajas:

- Este tipo de estructura del registro de retención puede crearse muy rápida y fácilmente.
- La lógica del programa para acceder a un elemento de datos se simplifica.

•

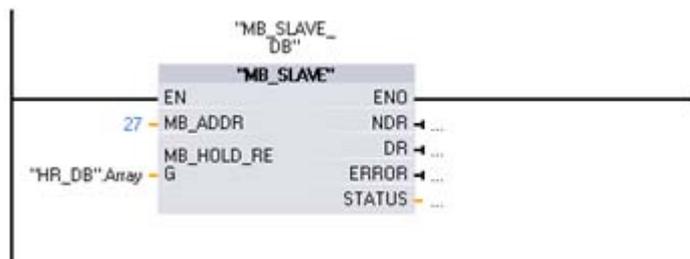
Desventajas:

- Aunque todo elemento de matriz puede referenciarse programáticamente según los nombres simbólicos ("HR_DB"."Array"[1] hasta "HR_DB"."Array"[10]), los nombres no describen la función interna de los datos.
- La matriz sólo puede constar de un tipo de datos. Es posible que sea necesario convertir los tipos en un programa de usuario con control de tipo estricto.

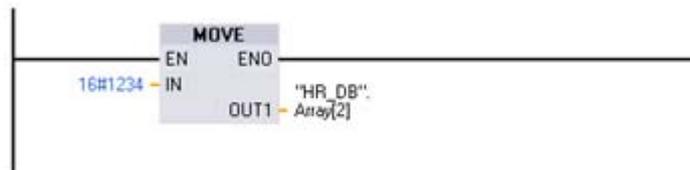
Una matriz de palabras se visualiza de la manera siguiente en el editor de bloques de datos.

HR_DB							
Name	Data type	Offset	Default value	Initial value	Retain	Comment	
1	Static						
2	Array	Array [1..10] of W. 0.0			<input type="checkbox"/>	40001 to 40010	
3	Array[1]		W#16#0000	W#16#0000			
4	Array[2]		W#16#0000	W#16#0000			
5	Array[3]		W#16#0000	W#16#0000			
6	Array[4]		W#16#0000	W#16#0000			
7	Array[5]		W#16#0000	W#16#0000			
8	Array[6]		W#16#0000	W#16#0000			
9	Array[7]		W#16#0000	W#16#0000			
10	Array[8]		W#16#0000	W#16#0000			
11	Array[9]		W#16#0000	W#16#0000			
12	Array[10]		W#16#0000	W#16#0000			

La figura siguiente muestra cómo la matriz se asignaría a la entrada MB_HOLD_REG de una instrucción MB_SLAVE.



A todo elemento de la matriz se puede acceder con su nombre simbólico, como se muestra a continuación. En este ejemplo, un valor nuevo se desplaza al segundo elemento de la matriz que corresponde a la dirección Modbus 40002.



Cada una de las palabras de la matriz, según lo definido en el bloque de datos, suministra a la instrucción MB_SLAVE direcciones del registro de retención Modbus. En esta instancia - puesto que la matriz contiene únicamente 10 elementos - hay sólo 10 direcciones del registro de retención Modbus que puede utilizar esta instrucción MB_SLAVE y a las que puede acceder el maestro Modbus.

La correlación entre los nombres de elementos de la matriz y las direcciones Modbus se indica abajo.

"HR_DB".Array[1]	Dirección Modbus 40001
" HR_DB ". Array[2]	Dirección Modbus 40002
" HR_DB ". Array[3]	Dirección Modbus 40003
...	...
" HR_DB ". Array[9]	Dirección Modbus 40009
" HR_DB ".Array [10]	Dirección Modbus 40010

Ejemplo 2 - Estructura de palabras con nombres

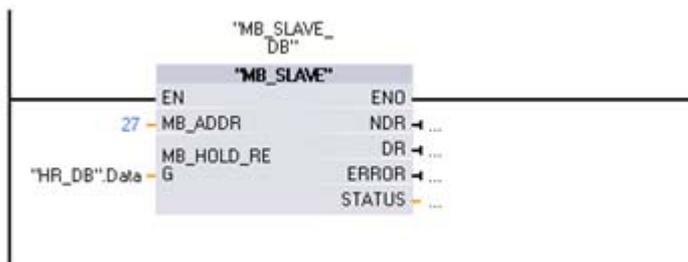
Este registro de retención de ejemplo es una serie de palabras con nombres simbólicos descriptivos.

- Ventajas:**
- Todo elemento de la estructura tiene un nombre descriptivo con un tipo de datos asignado.
- Desventajas:**
- Para crear este tipo de estructura se requiere más tiempo que para una matriz estándar de palabras.
 - Los elementos deben referenciarse adicionalmente de forma simbólica al utilizarlos en el programa de usuario. En tanto que el primer elemento de una matriz simple se referencia como "HR_DB".Array[0], el primer elemento de este tipo se referencia como "HR_DB".Data.Temp_1.

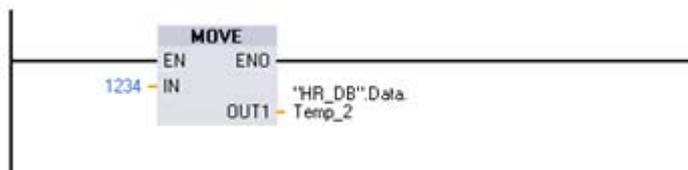
Una estructura de palabras con nombres se visualiza de la manera siguiente en el editor de bloques de datos. Todo elemento tiene un nombre unívoco y puede ser WORD, UINT o INT.

HR_DB						
Name	Data type	Offset	Initial value	Retain	Comment	
1	Static					
2	Data	Struct	0.0	<input type="checkbox"/>		
3	Temp_1	Int	0.0	0	40001	
4	Temp_2	Int	2.0	0	40002	
5	Temp_3	Int	4.0	0	40003	
6	Good_Count	Uint	6.0	0	40004	
7	Bad_Count	Uint	8.0	0	40005	
8	Rework_Count	Uint	10.0	0	40006	
9	Line_Stops	Uint	12.0	0	40007	
10	Avg_Time	Uint	14.0	0	40008	
11	Code_1	Word	16.0	0	40009	
12	Code_2	Word	18.0	0	40010	

La figura siguiente muestra cómo la estructura de datos que aparece arriba se asignaría a la entrada MB_HOLD_REG de una instrucción MB_SLAVE en el programa.



A todo elemento de la matriz se puede acceder con su nombre simbólico, como se muestra a continuación. En este ejemplo, un valor nuevo se desplaza al segundo elemento de la matriz que corresponde a la dirección Modbus 40002.



La correlación entre los nombres de elementos de datos y las direcciones Modbus se muestra abajo.

"HR_DB".Data.Temp_1	Dirección Modbus 40001
"HR_DB".Data.Temp_2	Dirección Modbus 40002
"HR_DB".Data.Temp_3	Dirección Modbus 40003
"HR_DB".Data.Good_Count	Dirección Modbus 40004
"HR_DB".Data.Bad_Count	Dirección Modbus 40005
"HR_DB".Data.Rework_Count	Dirección Modbus 40006
"HR_DB".Data.Line_Stops	Dirección Modbus 40007
"HR_DB".Data.Avg_Time	Dirección Modbus 40008
"HR_DB".Data.Code_1	Dirección Modbus 40009
"HR_DB".Data.Code_2	Dirección Modbus 40010

Ejemplo 3 - Estructura compleja con nombres

Este registro de retención de ejemplo es una serie de tipos de datos mixtos con nombres simbólicos descriptivos.

- Ventajas:**
- Todo elemento de la estructura tiene un nombre descriptivo con un tipo de datos asignado.
 - Permite transferir directamente tipos de datos no basados en palabras.
- Desventajas:**
- Para crear este tipo de estructura se requiere más tiempo que para una matriz estándar de palabras.
 - El maestro Modbus debe configurarse de manera que acepte los datos que recibirá del esclavo Modbus. Como muestra la figura siguiente, Temp_1 es un valor real de 4 bytes. El maestro receptor debe poder reconvertir las 2 palabras recibidas al valor real esperado.
 - Los elementos deben referenciarse adicionalmente de forma simbólica en el programa. En tanto que el primer elemento de una matriz simple se referencia como "HR_DB".Array[0], el primer elemento de este tipo se referencia como "HR_DB".Data.Temp_1.

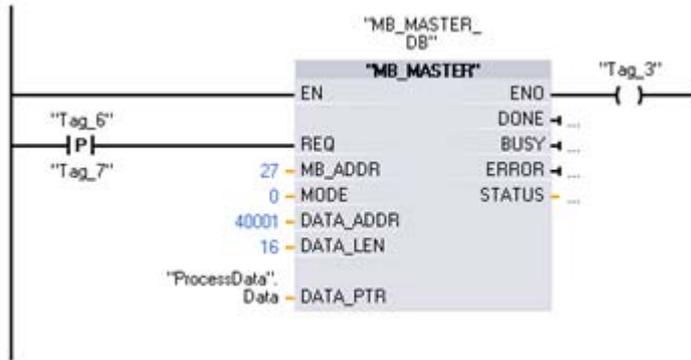
Una estructura compleja con nombres se visualiza de la manera siguiente en el editor de bloques de datos. Todo elemento tiene un nombre unívoco con varios tamaños y tipos de datos.

HR_DB						
Name	Data type	Offset	Initial value	Retain	Comment	
1	Static					
2	Data	Struct	0.0			Modbus addresses 40001 to 40016
3	Temp_1	Real	0.0	0.0		40001 and 40002
4	Temp_2	Real	4.0	0.0		40003 and 40004
5	Good_Count	UDInt	8.0	0		40005 and 40006
6	Bad_Count	UDInt	12.0	0		40007 and 40008
7	Rework_Count	UDInt	16.0	0		40009 and 40010
8	Line_Stops	Int	20.0	0		40011
9	Avg_Time	Int	22.0	0		40012
10	Long_Code	DWord	24.0	0		40013 and 40014
11	Code_1	Word	28.0	0		40015
12	Code_2	Word	30.0	0		40016

La correlación entre los nombres de elementos de datos y las direcciones Modbus se muestra abajo.

"HR_DB".Data.Temp_1	Direcciones Modbus 40001 y 40002
"HR_DB".Data.Temp_2	Direcciones Modbus 40003 y 40004
"HR_DB".Data.Good_Count	Direcciones Modbus 40005 y 40006
"HR_DB".Data.Bad_Count	Direcciones Modbus 40007 y 40008
"HR_DB".Data.Rework_Count	Direcciones Modbus 40009 y 40010
"HR_DB".Data.Line_Stops	Dirección Modbus 400011
"HR_DB".Data.Avg_Time	Dirección Modbus 400012
"HR_DB".Data.Long_Code	Direcciones Modbus 40013 y 40014
"HR_DB".Data.Code_1	Dirección Modbus 40015
"HR_DB".Data.Code_2	Dirección Modbus 40016

Una CPU S7-1200 que actúa de maestro Modbus puede utilizar la instrucción MB_Master y una estructura de datos idéntica para recibir el bloque de datos de la CPU S7-1200 que actúa de esclavo Modbus. Esta instrucción de maestro Modbus copiará las 16 palabras de datos directamente del bloque de datos HR_DB del esclavo en el bloque de datos ProcessData del maestro, como se muestra a continuación.



ProcessData						
Name	Data type	Offset	Initial value	Retain	Comment	
1	Static					
2	Data	0.0		<input type="checkbox"/>	Modbus addresses 40001 to 40016	
3	Temp_1	Real	0.0	0.0	40001 and 40002	
4	Temp_2	Real	4.0	0.0	40003 and 40004	
5	Good_Count	UDInt	8.0	0	40005 and 40006	
6	Bad_Count	UDInt	12.0	0	40007 and 40008	
7	Rework_Count	UDInt	16.0	0	40009 and 40010	
8	Line_Stops	Int	20.0	0	40011	
9	Avg_Time	Int	22.0	0	40012	
10	Long_Code	DWord	24.0	0	40013 and 40014	
11	Code_1	Word	28.0	0	40015	
12	Code_2	Word	30.0	0	40016	

Una serie de ubicaciones del bloque de datos Data_PTR del maestro Modbus pueden utilizarse para transferir estructuras iguales o diferentes desde distintos esclavos Modbus.

Códigos de condición

Valor de STATUS (W#16#...)	Descripción	
80C8	El timeout de respuesta indicado (consulte RCVTIME o MSGTIME) es 0	
80D1	El receptor ha lanzado una petición de control de flujo para suspender una transmisión activa y no ha habilitado nuevamente la transmisión en el tiempo de espera indicado. Este error también se genera durante el control de flujo por hardware cuando el receptor no confirma CTS en el tiempo de espera indicado.	
80D2	La petición de transmisión se ha cancelado porque no se recibe ninguna señal DSR del DCE	
80E0	El mensaje se ha terminado porque el búfer de recepción está lleno	
80E1	El mensaje se ha terminado debido a un error de paridad	
80E2	El mensaje se ha terminado debido a un error de trama	
80E3	El mensaje se ha terminado debido a un error de desbordamiento	
80E4	El mensaje se ha terminado debido a que la longitud especificada excede el tamaño del búfer total	
8180	ID de puerto no válida	
8186	Dirección de estación Modbus no válida	
8187	Puntero no válido a MB_HOLD_REG DB	
818C	Puntero a un tipo de DB de tipo seguro MB_HOLD_REG DB (debe ser un tipo de DB clásico)	
	Código de respuesta enviado al maestro Modbus (B#16#..)	
8380	Sin respuesta	Error CRC
8381	01	Código de función no soportado
8382	Sin respuesta	Error de longitud de datos
8383	02	Error de dirección de datos
8384	03	Error de valor de datos
8385	03	Valor de código de diagnóstico de datos no soportado (código de función 08)

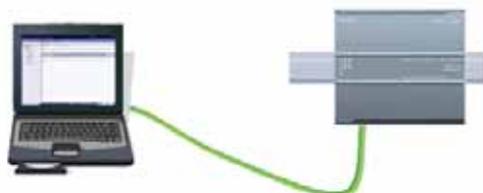
PROFINET

La CPU S7-1200 incorpora un puerto PROFINET que soporta las normas Ethernet y de comunicación basada en TCP/IP. La CPU S7-1200 soporta los siguientes protocolos de aplicación:

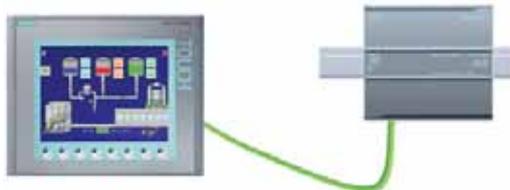
- Transport Control Protocol (TCP)
- ISO on TCP (RFC 1006)

La CPU S7-1200 puede comunicarse con otras CPUs S7-1200, programadoras STEP 7 Basic, dispositivos HMI y dispositivos no Siemens que utilicen protocolos de comunicación TCP estándar. Hay dos formas de comunicación vía PROFINET:

- Conexión directa: La comunicación directa se utiliza para conectar una programadora, dispositivo HMI u otra CPU a una sola CPU.
- Conexión de red: La comunicación de red se utiliza si deben conectarse más de dos dispositivos (p. ej. CPUs, HMIs, programadoras y dispositivos no Siemens).



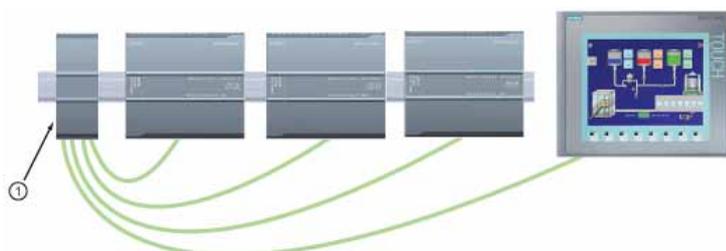
Conexión directa: Programadora conectada a una CPU S7-1200



Conexión directa: HMI conectado a una CPU S7-1200



Conexión directa: Una CPU S7-1200 conectada a otra CPU S7-1200



Conexión de red: Más de dos dispositivos interconectados, utilizando un switch Ethernet CSM1277 ①

Para la conexión directa entre una programadora o un HMI y una CPU no se requiere un switch Ethernet. Un switch Ethernet se requiere para una red que incorpore más de dos CPUs o dispositivos HMI. El switch Ethernet de 4 puertos CSM1277 de Siemens montado en un rack puede utilizarse para conectar las CPUs y los dispositivos HMI. El puerto PROFINET de la CPU S7-1200 no contiene un dispositivo de conmutación Ethernet.

Número máximo de conexiones para el puerto PROFINET

El puerto PROFINET de la CPU soporta las siguientes conexiones simultáneas.

- 3 conexiones para la comunicación entre dispositivos HMI y la CPU
- 1 conexión para la comunicación entre la programadora (PG) y la CPU
- 8 conexiones para la comunicación del programa del S7-1200 utilizando instrucciones del bloque T (TSEND_C, TRCV_C, TCON, TDISCON, TSEN, TRCV)
- 3 conexiones para la comunicación entre una CPU S7-1200 pasiva y una CPU S7 activa
 - La CPU S7 activa utiliza las instrucciones GET y PUT (S7-300 y S7-400) o ETHx_XFER (S7-200).
 - Una conexión S7-1200 activa sólo es posible con las instrucciones del bloque T.

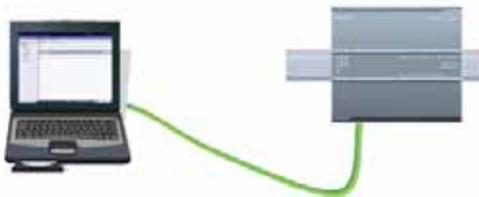
TSAPs o números de puerto restringidos para la comunicación ISO y TCP pasiva

Si la instrucción "TCON" se utiliza para configurar y establecer una conexión pasiva, las siguientes direcciones de puerto están restringidas y no se deben utilizar:

- TSAP ISO (pasivo): 01.00, 01.01, 02.00, 02.01, 03.00, 03.01
- Puerto TCP (pasivo): 5001, 102, 123, 20, 21, 25, 34962, 34963, 34964, 80

7.2 Comunicación con una programadora

Una CPU puede comunicarse con una programadora con STEP 7 Basic en una red.



Al configurar la comunicación entre una CPU y una programadora debe considerarse lo siguiente:

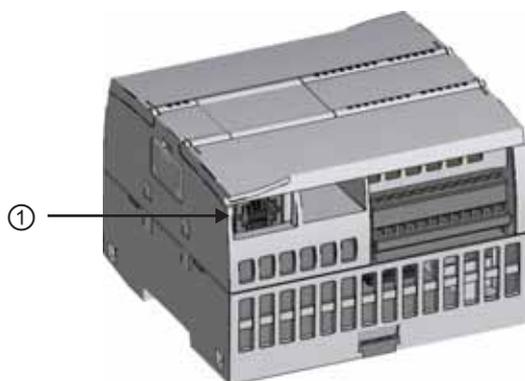
- Configuración/instalación: Es preciso configurar el hardware.
- Para la comunicación entre dos interlocutores no se requiere un switch Ethernet. Un switch Ethernet se requiere sólo si la red comprende más de dos dispositivos.

7.2.1 Establecer la conexión de hardware

Las interfaces PROFINET establecen las conexiones físicas entre una programadora y una CPU. Puesto que la CPU ofrece la función "auto-crossover", es posible utilizar un cable Ethernet estándar o cruzado ("crossover") para la interfaz. Para conectar una programadora directamente a una CPU no se requiere un switch Ethernet.

Para crear la conexión de hardware entre una programadora y una CPU, proceda del siguiente modo:

1. Monte la CPU (Página 29).
2. Conecte el cable Ethernet al puerto PROFINET que se muestra abajo.
3. Conecte el cable Ethernet a la programadora.



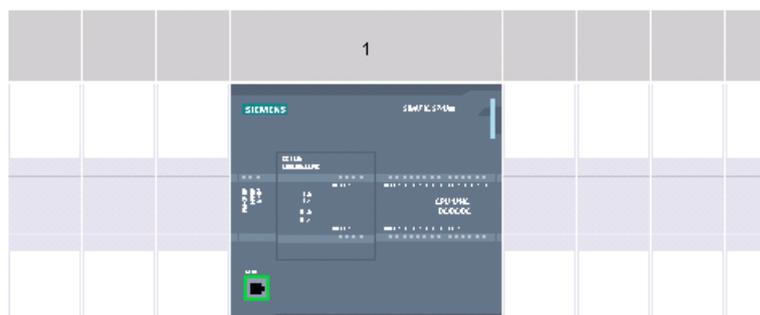
① Puerto PROFINET

Hay una descarga de tracción opcional disponible para reforzar la conexión PROFINET.

7.2.2 Configurar los dispositivos

Si ya se ha creado un proyecto con una CPU, ábralo en el TIA Portal.

En caso contrario, cree un proyecto e inserte una CPU (Página 78) en el rack. En el proyecto que aparece abajo, una CPU se muestra en la "Vista de dispositivos" del TIA Portal.



7.2.3 Asignar direcciones IP (Internet Protocol)

7.2.3.1 Asignar direcciones IP a los dispositivos de programación y red

Si la programadora incorpora una tarjeta adaptadora conectada a la LAN de la instalación (y posiblemente a Internet), la ID de red de la dirección IP y la máscara de subred de la CPU y la tarjeta adaptadora integrada en la programadora deberán ser idénticas. La ID de red es la primera parte de la dirección IP (los tres primeros octetos) (p. ej. **211.154.184.16**) y determina la red IP utilizada. Normalmente, la máscara de subred tiene el valor **255.255.255.0**. No obstante, puesto que el equipo está integrado en una LAN corporativa, la máscara de subred puede tener distintos valores (p. ej. **255.255.254.0**) para configurar subredes unívocas. Al combinar la máscara de subred con la dirección IP del dispositivo en una operación Y matemática se definen los límites de la subred IP.

Nota

En Internet, puesto que las programadoras, dispositivos de red y routers IP se comunican con el mundo entero, es preciso asignar direcciones IP unívocas para evitar conflictos con otros usuarios de la red. Contacte con los especialistas del departamento IT de su empresa, que están familiarizados con la red corporativa, para asignar las direcciones IP.

Si la programadora utiliza una tarjeta adaptadora Ethernet-USB conectada a una red aislada, la ID de red de la dirección IP y la máscara de subred de la CPU y la tarjeta adaptadora Ethernet-USB integrada en la programadora deberán ser exactamente iguales. La ID de red es la primera parte de la dirección IP (los tres primeros octetos) (p. ej. **211.154.184.16**) y determina la red IP utilizada. Normalmente, la máscara de subred tiene el valor **255.255.255.0**. Al combinar la máscara de subred con la dirección IP del dispositivo en una operación Y matemática se definen los límites de la subred IP.

Nota

Una tarjeta adaptadora Ethernet-USB es apropiada si la CPU no debe integrarse en la LAN corporativa. Esta opción es especialmente útil durante la comprobación inicial o los tests de puesta en marcha.

Tarjeta adaptadora de la programadora	Tipo de red	Dirección IP (Internet Protocol)	Máscara de subred
Tarjeta adaptadora integrada	Conectada a la LAN corporativa (y posiblemente a Internet)	La ID de red de la CPU y la tarjeta adaptadora integrada en la programadora deben ser exactamente iguales. La ID de red es la primera parte de la dirección IP (los dos primeros octetos) (p. ej. 211.154.184.16) y determina la red IP utilizada.	La máscara de subred de la CPU y la tarjeta adaptadora integrada deben ser exactamente iguales. Normalmente, la máscara de subred tiene el valor 255.255.255.0 . No obstante, puesto que el equipo está integrado en una LAN corporativa, la máscara de subred puede tener distintos valores (p. ej. 255.255.254.0) para configurar subredes unívocas. Al combinar la máscara de subred con la dirección IP del dispositivo en una operación Y matemática se definen los límites de la subred IP.
Tarjeta adaptadora Ethernet-USB	Conectada a una red aislada	La ID de red de la CPU y la tarjeta adaptadora Ethernet-USB de la programadora deben ser exactamente iguales. La ID de red es la primera parte de la dirección IP (los dos primeros octetos) (p. ej. 211.154.184.16) y determina la red IP utilizada.	La máscara de subred de la CPU y la tarjeta adaptadora Ethernet-USB deben ser exactamente iguales. Normalmente, la máscara de subred tiene el valor 255.255.255.0 . Al combinar la máscara de subred con la dirección IP del dispositivo en una operación Y matemática se definen los límites de la subred IP.

Asignar o comprobar la dirección IP de la programadora utilizando "Mis sitios de red" (en el Escritorio)

La dirección IP de la programadora se puede asignar o comprobar mediante los siguientes comandos de menú:

- (Clic con el botón derecho del ratón en) "Mis sitios de red"
- "Propiedades"
- (Clic con el botón derecho del ratón en) "Conexión de área local"
- "Propiedades"

En el diálogo "Propiedades de conexión de área local", campo "Esta conexión utiliza los siguientes elementos:", desplácese hasta "Protocolo Internet (TCP/IP)". Haga clic en "Protocolo Internet (TCP/IP)" y luego en el botón "Propiedades". Seleccione "Obtener una dirección IP automáticamente (DHCP)" o "Usar la siguiente dirección IP" (para introducir una dirección IP estática).

Nota

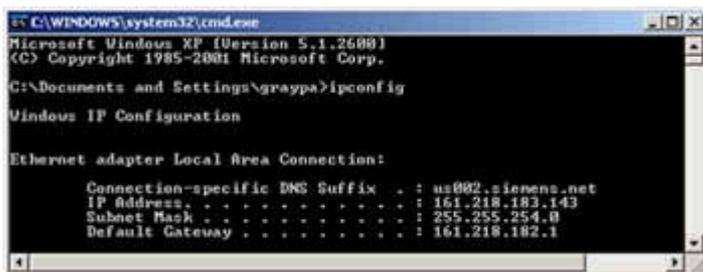
El "Dynamic Host Configuration Protocol" (DHCP o protocolo de configuración dinámica de host) asigna automáticamente una dirección IP a la programadora después del arranque desde el servidor DHCP.

Comprobar la dirección IP de la programadora mediante los comandos "ipconfig" e "ipconfig /all"

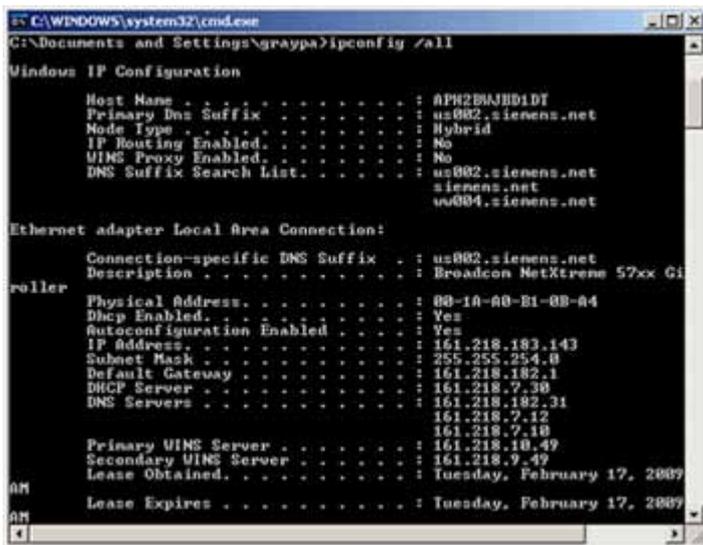
La dirección IP de la programadora y, si es aplicable, la del router IP ("gateway" o pasarela), se puede(n) comprobar a través de los siguientes comandos de menú:

- Botón "Inicio" (en el Escritorio)
- "Ejecutar"

En el campo "Abrir" del diálogo "Ejecutar", introduzca "cmd" y haga clic en el botón "Aceptar". En el diálogo "C:\WINDOWS\system32\cmd.exe" que aparece entonces, introduzca el comando "ipconfig". Un resultado de ejemplo se muestra a continuación:



El comando "ipconfig /all" permite visualizar información adicional. Aquí se indican el tipo de tarjeta adaptadora de la programadora y la dirección Ethernet (MAC):



Asignar una dirección IP a una CPU

Para asignar una dirección IP a una CPU, utilice uno de los métodos siguientes:

- Asignar una dirección IP online
- Configurar una dirección IP en el proyecto

7.2.3.2 Asignar una dirección IP online

Es posible asignar una dirección IP a un dispositivo de red online. Esto es especialmente útil al configurar los dispositivos por primera vez.

Para asignar una dirección IP online, proceda del siguiente modo:

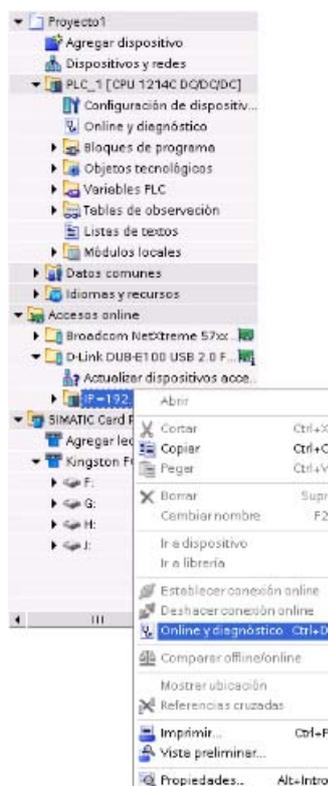
1. En el "Árbol del proyecto", verifique que la CPU no tiene asignada ninguna dirección IP. Utilice para ello los comandos de menú siguientes:

- "Accesos online"
- <Tarjeta adaptadora para la red en la que se encuentra el dispositivo>
- "Actualizar dispositivos accesibles"



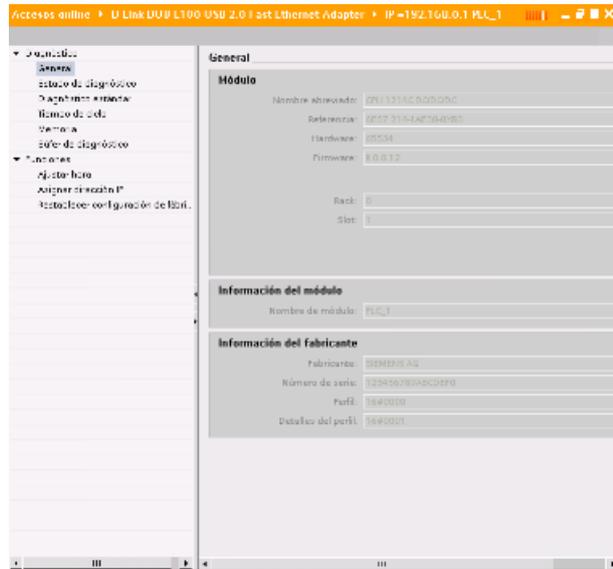
2. Seleccione los siguientes comandos de menú en el "Árbol del proyecto":

- "Accesos online"
- <Tarjeta adaptadora para la red en la que se encuentra el dispositivo>
- "Actualizar dispositivos accesibles"
- <dirección del dispositivo>
- "Online y diagnóstico"

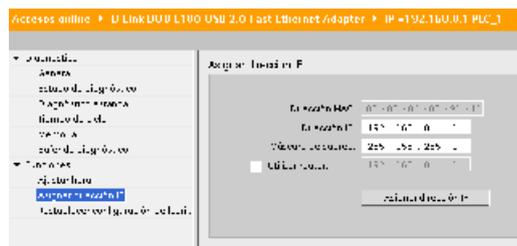


3. Seleccione los siguientes comandos de menú en el diálogo "Online y diagnóstico":

- "Funciones"
- "Asignar dirección IP"



4. Introduzca la nueva dirección IP en el campo "Dirección IP".



5. En el "Árbol del proyecto", verifique que la nueva dirección IP se ha asignado a la CPU. Utilice para ello los comandos de menú siguientes:

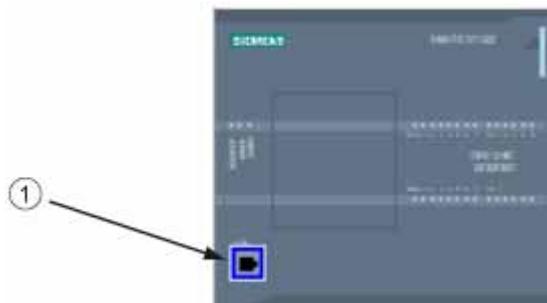
- "Accesos online"
- <Adaptador para la red en la que se encuentra el dispositivo>
- "Actualizar dispositivos accesibles"



7.2.3.3 Configurar una dirección IP en el proyecto

Configurar la interfaz PROFINET

Tras configurar el rack con la CPU (Página 243), es posible configurar los parámetros de la interfaz PROFINET. A este efecto, haga clic en la casilla PROFINET verde en la CPU para seleccionar el puerto PROFINET. La ficha "Propiedades" de la ventana de inspección muestra el puerto PROFINET.



① Puerto PROFINET

Configurar la dirección IP

Dirección Ethernet (MAC): Todo dispositivo de una red PROFINET recibe una dirección MAC (Media Access Control o control de acceso al medio) del fabricante para su identificación. Una dirección MAC consta de seis grupos de dos dígitos hexadecimales, separados por guiones (-) o dos puntos (:), en orden de transmisión (p. ej. 01-23-45-67-89-AB ó 01:23:45:67:89:AB).

Dirección IP: Todo dispositivo debe tener también una dirección IP (Internet Protocol o Protocolo Internet). Esta dirección permite al dispositivo transferir datos a través de una red enrutada y más compleja.

Toda dirección IP se divide en segmentos de ocho bits (octetos) y se expresa en formato decimal separado por puntos (p. ej. 211.154.184.16). La primera parte de la dirección IP se utiliza para la ID de red (¿en qué red se encuentra?) y, la segunda, para la ID del host (unívoca para cada dispositivo de la red). Una dirección IP 192.168.x.y es una designación estándar reconocida como parte de una red privada que no se enruta vía Internet.

Máscara de subred: Una subred es una agrupación lógica de dispositivos de red conectados. Generalmente, los nodos de una subred están próximos físicamente en una red de área local (LAN). Una máscara (denominada "máscara de subred" o "máscara de red") define los límites de una subred IP.

Generalmente, una máscara de subred 255.255.255.0 se adecúa para una red local pequeña. Esto significa que los 3 primeros octetos de todas las direcciones IP de esta red deberían ser iguales. Los diferentes dispositivos de la red se identifican mediante el último octeto (campo de 8 bits). Por ejemplo, es posible asignar la máscara de subred 255.255.255.0 y direcciones IP comprendidas entre 192.168.2.0 y 192.168.2.255 a los dispositivos de una red local pequeña.

La única conexión entre las diferentes subredes se realiza a través de un router. Si se utilizan subredes, es preciso utilizar un router IP.

Router IP: Los routers interconectan las distintas LANs. Si se utiliza un router, un equipo de una LAN puede enviar mensajes a otras redes que, a su vez, pertenezcan a otras LANs. Si el destino de los datos se encuentra fuera de la LAN, el router reenvía los datos a otra red o grupo de redes desde donde pueden transferirse a su destino.

Los routers necesitan direcciones IP para poder transferir y recibir paquetes de datos.



Propiedades de direcciones IP: En la ventana de propiedades, seleccione la entrada de configuración "Dirección Ethernet". El TIA Portal visualiza el diálogo de configuración de direcciones Ethernet, en el que el proyecto de software se asocia a la dirección IP de la CPU que lo recibirá.

Nota

La CPU no tiene una dirección IP preconfigurada. La dirección IP de la CPU se debe asignar manualmente. Si la CPU está conectada a un router de la red, también es preciso introducir la dirección IP del router. Todas las direcciones IP se configuran al cargar el proyecto en el dispositivo.

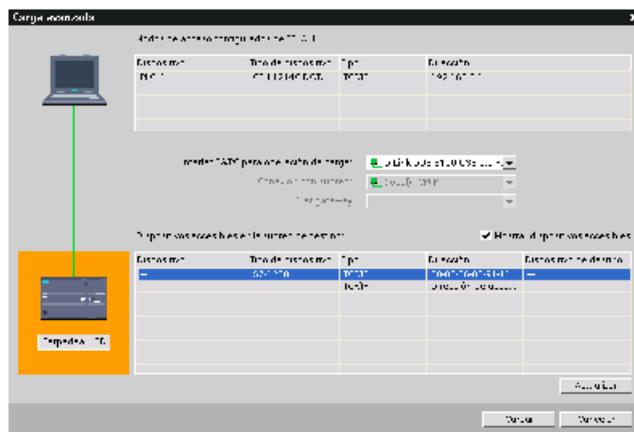
Para más información, consulte el apartado "Asignar direcciones IP a los dispositivos de programación y red".

La tabla siguiente define los parámetros de la dirección IP:

Parámetro	Descripción	
Subred	Nombre de la subred a la que está conectada el dispositivo. Haga clic en el botón "Agregar nueva subred" para crear una subred nueva. El ajuste predeterminado es "no conectado". Hay dos tipos de conexión posibles: <ul style="list-style-type: none"> • El ajuste predeterminado "no conectado" ofrece una conexión local. • Una subred se requiere cuando la red comprende dos o más dispositivos. 	
Protocolo IP	Dirección IP	Dirección IP asignada a la CPU
	Máscara de subred	Máscara de subred asignada
	Utilizar router IP	Haga clic en esta casilla de verificación para indicar el uso de un router IP
	Dirección del router	Dirección IP asignada al router (si es aplicable)

7.2.4 Comprobar la red PROFINET

Tras finalizar la configuración, cargue el proyecto en la CPU. Todas las direcciones IP se configuran al cargar el proyecto en el dispositivo.



Asignar una dirección IP a un dispositivo online

La CPU S7-1200 no tiene dirección IP preconfigurada. La dirección IP de la CPU se debe asignar manualmente.

Para asignar una dirección IP a un dispositivo online, consulte el procedimiento paso a paso descrito en "Asignar una dirección IP online".

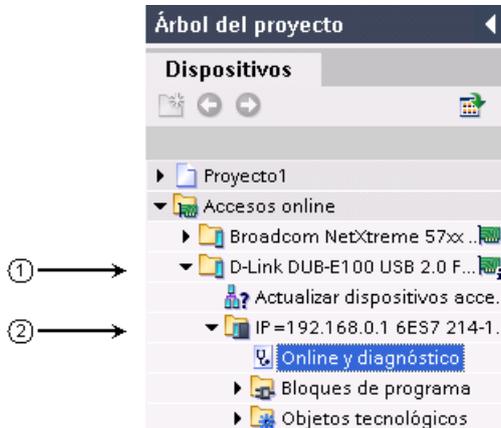
Para asignar una dirección IP en el proyecto, es preciso configurarla en la "Configuración de dispositivos", guardar la configuración y cargarla en el PLC. Encontrará más información al respecto en "Configurar una dirección IP en el proyecto".

Nota

Las direcciones IP que se hayan asignado online pueden cambiarse utilizando el método de configuración de hardware online u offline.

Si las direcciones IP se han asignado en la configuración hardware offline, las direcciones IP asignadas en el proyecto sólo podrán cambiarse utilizando el método de configuración de hardware offline.

Utilice "Accesos online" para visualizar la dirección IP de la CPU conectada como se muestra a continuación.



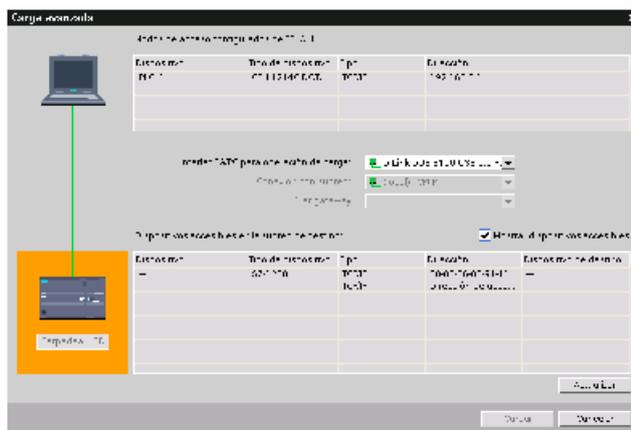
- ① La segunda de dos redes Ethernet de esta programadora
- ② Dirección IP de la única CPU S7-1200 de esta red Ethernet

Nota

Se visualizan todas las redes configuradas de la programadora. Para visualizar la dirección IP de la CPU S7-1200 deseada es preciso seleccionar la red correcta.

Utilizar el diálogo "Carga avanzada" para comprobar los dispositivos de red conectados

La función de la CPU S7-1200 "Cargar en dispositivo" y su diálogo "Carga avanzada" permiten visualizar todos los dispositivos de red accesibles y verificar si se han asignado direcciones IP unívocas a todos ellos. Para visualizar todos los dispositivos accesibles y disponibles con sus respectivas direcciones MAC e IP asignadas, active la casilla de verificación "Mostrar dispositivos accesibles".



Si el dispositivo de red deseado no se encuentra en esta lista, la comunicación con ese dispositivo se habrá interrumpido por algún motivo. En este caso es preciso examinar el dispositivo y la red para buscar errores de hardware y/o configuración.

7.3 Comunicación entre dispositivos HMI y el PLC



La CPU soporta conexiones PROFINET con dispositivos HMI. Los siguientes requisitos deben considerarse al configurar la comunicación entre CPUs y HMIs:

Configuración/instalación:

- El puerto PROFINET de la CPU debe configurarse para poder establecer una conexión con el HMI.
- El HMI se debe instalar y configurar.
- La información de configuración del HMI forma parte del proyecto de la CPU y se puede configurar y cargar desde el proyecto.
- Para la comunicación entre dos interlocutores no se requiere un switch Ethernet. Un switch Ethernet se requiere sólo si la red comprende más de dos dispositivos.

Nota

El switch Ethernet de 4 puertos CSM1277 de Siemens montado en un rack puede utilizarse para conectar las CPUs y los dispositivos HMI. El puerto PROFINET de la CPU no contiene un dispositivo de conmutación Ethernet.

Funciones soportadas:

- El HMI puede leer/escribir datos en la CPU.
- Es posible disparar mensajes, según la información consultada de la CPU.
- Diagnóstico del sistema

Nota

WinCC Basic y STEP 7 Basic son componentes del TIA Portal. Para más información sobre cómo configurar el HMI, consulte la documentación de WinCC Basic.

Pasos necesarios para configurar la comunicación entre un dispositivo HMI y una CPU

Paso	Tarea
1	<p>Establecer la conexión de hardware</p> <p>Una interfaz PROFINET establece la conexión física entre un dispositivo HMI y una CPU. Puesto que la función "auto-crossover" está integrada en la CPU, es posible utilizar un cable Ethernet estándar o cruzado ("crossover") para la interfaz. Para conectar un HMI a una CPU no se requiere un switch Ethernet.</p> <p>Encontrará más información en "Comunicación con una programadora: Establecer la conexión de hardware" (Página 243).</p>
2	<p>Configurar los dispositivos</p> <p>Encontrará más información en "Comunicación con una programadora: Configurar los dispositivos" (Página 243).</p>
3	<p>Configurar las conexiones de red lógicas entre un dispositivo HMI y una CPU</p> <p>Encontrará más información en "Comunicación entre el HMI y el PLC: Configurar las conexiones de red lógicas entre un dispositivo HMI y una CPU" (Página 255).</p>
4	<p>Configurar una dirección IP en el proyecto</p> <p>Utilice el mismo proceso de configuración. No obstante, es preciso configurar direcciones IP para el HMI y la CPU.</p> <p>Encontrará más información en "Comunicación con una programadora: Configurar una dirección IP en el proyecto" (Página 249).</p>
5	<p>Comprobar la red PROFINET</p> <p>La configuración debe cargarse en cada una de las CPUs.</p> <p>Encontrará más información en "Comunicación con una programadora: Comprobar la red PROFINET" (Página 251).</p>

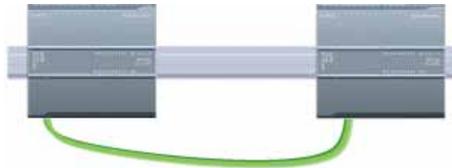
7.3.1 Configurar las conexiones de red lógicas entre un dispositivo HMI y una CPU

Tras configurar el rack con la CPU podrá configurar las conexiones de red.

En el portal "Dispositivos y redes", utilice la "Vista de red" para crear las conexiones de red entre los dispositivos del proyecto. Para crear la conexión Ethernet, seleccione la casilla (Ethernet) verde en la CPU. Arrastre una línea hasta la casilla Ethernet del dispositivo HMI. Suelte el botón del rotón para crear la conexión Ethernet.

Acción	Resultado
<p>Seleccione "Vista de red" para visualizar los dispositivos que deben conectarse.</p>	
<p>Seleccione el puerto de uno de los dispositivos y arrastre la conexión hasta el puerto del otro dispositivo.</p>	
<p>Suelte el botón del ratón para crear la conexión de red.</p>	

7.4 Comunicación entre PLCs



Una CPU puede comunicarse con otra CPU utilizando las instrucciones TSEND_C y TRCV_C.

Considere lo siguiente al configurar la comunicación entre dos CPUs:

- Configuración/instalación: Es preciso configurar el hardware.
- Funciones soportadas: Leer/escribir datos en una CPU interlocutora
- Para la comunicación entre dos interlocutores no se requiere un switch Ethernet. Un switch Ethernet se requiere sólo si la red comprende más de dos dispositivos.

Pasos necesarios para configurar la comunicación entre dos CPUs

Paso	Tarea
1	<p>Establecer la conexión de hardware</p> <p>Una interfaz PROFINET establece la conexión física entre dos CPUs. Puesto que la función "auto-crossover" está integrada en la CPU, es posible utilizar un cable Ethernet estándar o cruzado ("crossover") para la interfaz. Para conectar dos CPUs no se requiere un switch Ethernet.</p> <p>Encontrará más información en "Comunicación con una programadora: Establecer la conexión de hardware".</p>
2	<p>Configurar los dispositivos</p> <p>Es preciso configurar dos proyectos. Cada uno de ellos debe contener una CPU.</p> <p>Encontrará más información en "Comunicación con una programadora: Configurar los dispositivos".</p>
3	<p>Configurar las conexiones de red lógicas entre dos CPUs</p> <p>Encontrará más información en el apartado "Configurar la comunicación entre dos CPUs: Configurar las conexiones de red lógicas entre dos CPUs" (Página 257).</p>
4	<p>Configurar una dirección IP en el proyecto</p> <p>Utilice el mismo proceso de configuración. No obstante, es preciso configurar direcciones IP para dos CPUs (p. ej. PLC_1 y PLC_2).</p> <p>Encontrará más información en "Comunicación con una programadora: Configurar una dirección IP en el proyecto".</p>
5	<p>Configurar los parámetros de transmisión y recepción</p> <p>Las instrucciones TSEND_C y TRCV_C deben configurarse en ambas CPUs para habilitar la comunicación entre ellas.</p> <p>Encontrará más información en el apartado "Configurar la comunicación entre dos CPUs: Configurar los parámetros de transmisión y recepción" (Página 257).</p>
6	<p>Comprobar la red PROFINET</p> <p>La configuración debe cargarse en cada una de las CPUs.</p> <p>Encontrará más información en "Configurar la comunicación entre una programadora y una CPU: Comprobar la red PROFINET".</p>

7.4.1 Configurar las conexiones de red lógicas entre dos CPUs

Tras configurar el rack con la CPU podrá configurar las conexiones de red.

En el portal "Dispositivos y redes", utilice la "Vista de red" para crear las conexiones de red entre los dispositivos del proyecto. Para crear la conexión PROFINET, seleccione la casilla (PROFINET) verde en el primer PLC. Arrastre una línea hasta la casilla PROFINET del segundo PLC. Suelte el botón del rotón para crear la conexión PROFINET.

Acción	Resultado
<p>Seleccione "Vista de red" para visualizar los dispositivos que deben conectarse.</p>	
<p>Seleccione el puerto de uno de los dispositivos y arrastre la conexión hasta el puerto del otro dispositivo.</p>	
<p>Suelte el botón del ratón para crear la conexión de red.</p>	

7.4.2 Configurar los parámetros de transmisión y recepción

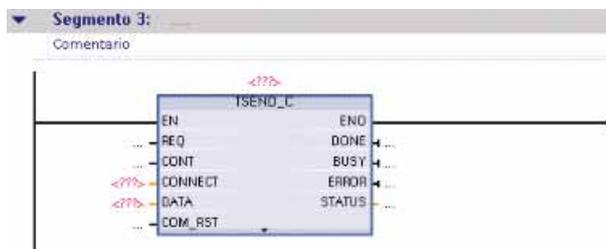
Las comunicaciones vía el bloque de transferencia (bloque T) sirven para establecer conexiones entre dos CPUs. Para que las CPUs puedan intervenir en la comunicación PROFINET es preciso configurar parámetros para transmitir y recibir mensajes. Estos parámetros determinan cómo deben funcionar las comunicaciones al transmitir o recibir mensajes a/de un dispositivo de destino.

7.4.2.1 Configurar los parámetros de transmisión de la instrucción TSEND_C

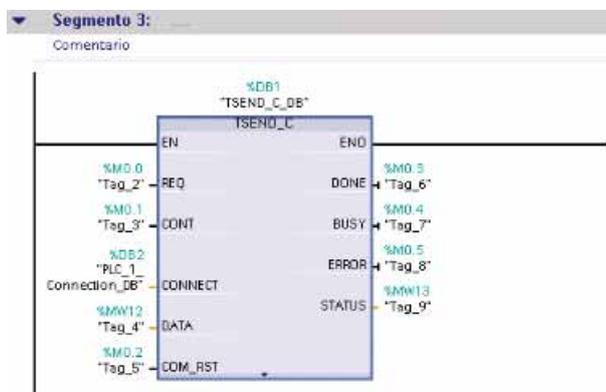
Instrucción TSEND_C

La instrucción TSEND_C (Página 178) crea una conexión con un interlocutor. La conexión se configura, establece y vigila automáticamente hasta que la instrucción ordene que sea desconectada. La instrucción TSEND_C combina las funciones de las instrucciones TCON, TDISCON y TSEND.

En la "Configuración de dispositivos" de STEP 7 Basic es posible configurar cómo la instrucción TSEND_C debe transmitir los datos. Para comenzar, inserte la instrucción en el programa desde la carpeta "Comunicación" en las "Instrucciones avanzadas". La instrucción se visualizará junto con el diálogo "Opciones de llamada" en el que se asigna un DB para almacenar los parámetros de la instrucción TSEND_C.



Como muestra la figura siguiente, es posible asignar posiciones de memoria a las entradas y salidas en la memoria de variables.



Configurar los parámetros generales

Los parámetros de comunicación se configuran en el diálogo "Propiedades" de la instrucción TSEND_C. Este diálogo aparece en el lado inferior de la página cuando se ha seleccionado alguna parte de la instrucción TSEND_C.

Configurar los parámetros de conexión

Toda CPU incorpora un puerto PROFINET que soporta la comunicación PROFINET estándar. Los protocolos Ethernet soportados se describen en los dos tipos de conexión siguientes:

Protocolo	Nombre del protocolo	Uso
RFC 1006	ISO on TCP	Fragmentación y reensamblado de mensajes
TCP	Transport Control Protocol	Transporte de tramas

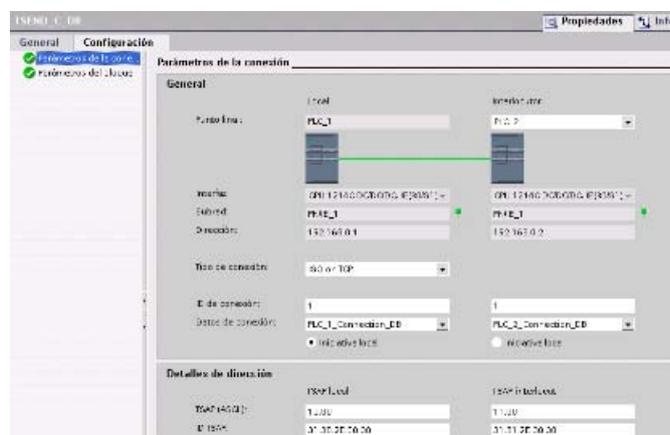
ISO on TCP (RFC 1006)

ISO on TCP es un mecanismo que permite portar aplicaciones ISO a la red TCP/IP. Este protocolo tiene las características siguientes:

- Protocolo de comunicación eficiente vinculado estrechamente al hardware
- Adecuado para cantidades de datos medianas y grandes (hasta 8192 bytes)
- A diferencia de TCP, los mensajes tienen un indicador de fin y están orientados a los mensajes.
- Apto para routing; puede utilizarse en WAN
- Las longitudes de datos dinámicas son posibles.
- Es necesario programar la gestión de datos debido a la interfaz de programación SEND / RECEIVE.

Puesto que utiliza Transport Service Access Points (TSAPs), el protocolo TCP permite establecer varias conexiones con una sola dirección IP (hasta 64K conexiones). Gracias a RFC 1006, los TSAPs identifican unívocamente estas conexiones de puntos finales de comunicación a una dirección IP.

En el área "Detalles de dirección" del diálogo "Parámetros de la conexión" se definen los TSAPs que deben utilizarse. El TSAP de una conexión en la CPU se introduce en el campo "TSAP local". El TSAP asignado a la conexión en la CPU interlocutora se introduce en el campo "TSAP del interlocutor".



Parámetro	Definición
General	
Punto final: Interlocutor	Nombre asignado a la CPU interlocutora (receptora)
Interfaz	Nombre asignado a las interfaces
Subred	Nombre asignado a las subredes
Dirección	Direcciones IP asignadas
Tipo de conexión	Tipo de protocolo Ethernet
ID de conexión	Número de ID
Datos de conexión	Ubicación de almacenamiento de datos de las CPUs local e interlocutora
Establecimiento de conexión activo	Botón de opción para seleccionar la CPU local o interlocutora como conexión activa
Detalles de dirección	
TSAP ¹ (ASCII)	TSAPs de las CPUs local e interlocutora en formato ASCII
ID TSAP	TSAPs de las CPUs local e interlocutora en formato hexadecimal

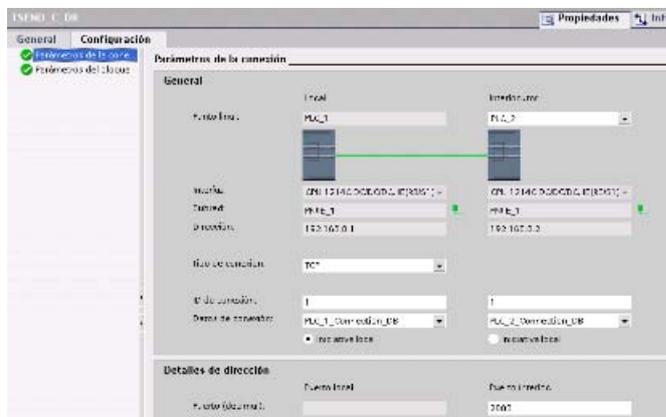
¹ Al configurar una conexión con una CPU S7-1200 para ISO on TCP, utilice sólo caracteres ASCII en la extensión TSAP para los interlocutores pasivos.

Transport Control Protocol (TCP)

TCP es un protocolo estándar descrito por RFC 793: Transmission Control Protocol. El objetivo principal de TCP es ofrecer un servicio de conexión seguro y fiable entre pares de procesos. Este protocolo tiene las características siguientes:

- Protocolo de comunicación eficiente puesto que está vinculado estrechamente al hardware
- Adecuado para cantidades de datos medianas y grandes (hasta 8192 bytes)
- Ofrece numerosas prestaciones más a las aplicaciones, en particular:
 - Recuperación de errores
 - Control de flujo
 - Fiabilidad
- Protocolo orientado a la conexión
- Puede utilizarse muy flexiblemente con sistemas de terceros que soporten únicamente TCP
- Apto para routing
- Son aplicables sólo las longitudes de datos estáticas.
- Los mensajes se acusan.
- Las aplicaciones se direccionan usando números de puerto.

- La mayoría de los protocolos de aplicación (p. ej. TELNET y FTP) utilizan TCP.
- Es necesario programar la gestión de datos debido a la interfaz de programación SEND / RECEIVE.



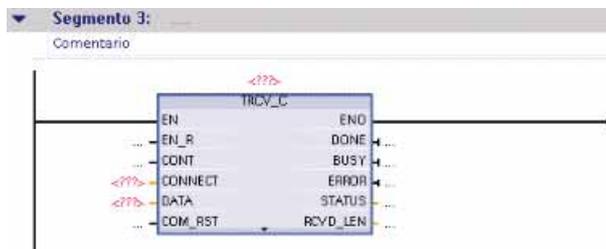
Parámetro	Definición
General	
Punto final: Interlocutor	Nombre asignado a la CPU interlocutora (receptora)
Interfaz	Nombre asignado a las interfaces
Subred	Nombre asignado a las subredes
Dirección	Direcciones IP asignadas
Tipo de conexión	Tipo de protocolo Ethernet
ID de conexión	Número de ID
Datos de conexión	Ubicación de almacenamiento de datos de las CPUs local e interlocutora
Establecimiento de conexión activo	Botón de opción para seleccionar la CPU local o interlocutora como conexión activa
Detalles de dirección	
Puerto (decimal)	Puerto de la CPU interlocutora en formato decimal

7.4.2.2 Configurar los parámetros de recepción de la instrucción TRCV_C

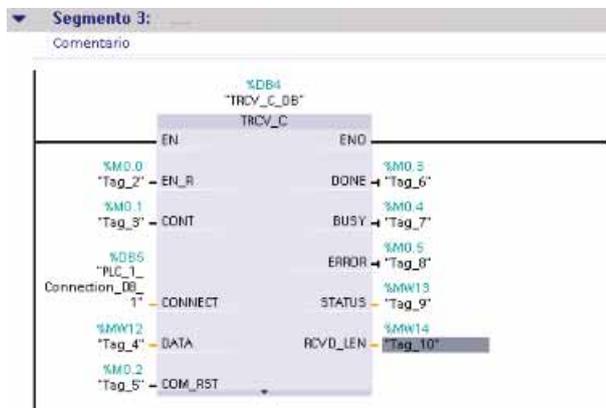
Instrucción TRCV_C

La instrucción TRCV_C (Página 178) crea una conexión con un interlocutor. La conexión se configura, establece y vigila automáticamente hasta que la instrucción ordene que sea desconectada. La instrucción TRCV_C combina las funciones de las instrucciones TCON, TDISCON y TRCV.

Desde la configuración de la CPU en STEP 7 Basic es posible configurar cómo la instrucción TRCV_C debe recibir los datos. Para comenzar, inserte la instrucción en el programa desde la carpeta "Comunicación" en las "Instrucciones avanzadas". La instrucción se visualizará junto con el diálogo "Opciones de llamada" en el que se asigna un DB para almacenar los parámetros de la instrucción TRCV_C.



Como muestra la figura siguiente, es posible asignar posiciones de memoria a las entradas y salidas en la memoria de variables.



Configurar los parámetros generales

Los parámetros de comunicación se configuran en el diálogo "Propiedades" de la instrucción TRCV_C. Este diálogo aparece en el lado inferior de la página cuando se ha seleccionado alguna parte de la instrucción TRCV_C.

Configurar los parámetros de conexión

Toda CPU incorpora un puerto PROFINET que soporta la comunicación PROFINET estándar. Los protocolos Ethernet soportados se describen en los dos tipos de conexión siguientes:

Protocolo	Nombre del protocolo	Uso
RFC 1006	ISO on TCP	Fragmentación y reensamblado de mensajes
TCP	Transport Control Protocol	Transporte de tramas

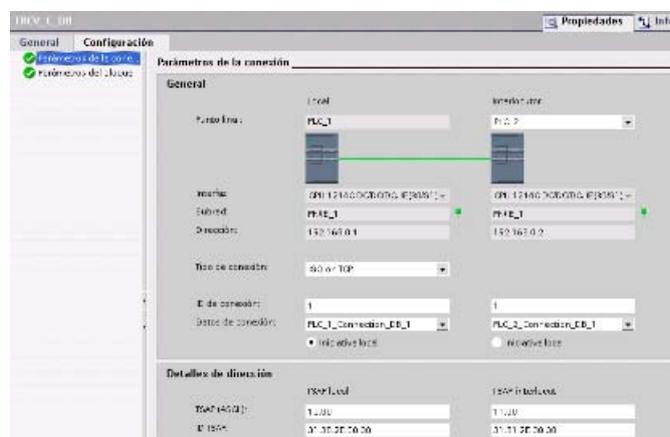
ISO on TCP (RFC 1006)

ISO on TCP es un mecanismo que permite portar aplicaciones ISO a la red TCP/IP. Este protocolo tiene las características siguientes:

- Protocolo de comunicación eficiente vinculado estrechamente al hardware
- Adecuado para cantidades de datos medianas y grandes (hasta 8192 bytes)
- A diferencia de TCP, los mensajes tienen un indicador de fin y están orientados a los mensajes.
- Apto para routing; puede utilizarse en WAN
- Las longitudes de datos dinámicas son posibles.
- Es necesario programar la gestión de datos debido a la interfaz de programación SEND / RECEIVE.

Puesto que utiliza Transport Service Access Points (TSAPs), el protocolo TCP permite establecer varias conexiones con una sola dirección IP (hasta 64K conexiones). Gracias a RFC 1006, los TSAPs identifican unívocamente estas conexiones de puntos finales de comunicación a una dirección IP.

En el área "Detalles de dirección" del diálogo "Parámetros de la conexión" se definen los TSAPs que deben utilizarse. El TSAP de una conexión en la CPU se introduce en el campo "TSAP local". El TSAP asignado a la conexión en la CPU interlocutora se introduce en el campo "TSAP del interlocutor".



Parámetro	Definición
General	
Punto final: Interlocutor	Nombre asignado a la CPU interlocutora (receptora)
Interfaz	Nombre asignado a las interfaces
Subred	Nombre asignado a las subredes
Dirección	Direcciones IP asignadas
Tipo de conexión	Tipo de protocolo Ethernet
ID de conexión	Número de ID
Datos de conexión	Ubicación de almacenamiento de datos de las CPUs local e interlocutora
Establecimiento de conexión activo	Botón de opción para seleccionar la CPU local o interlocutora como conexión activa
Detalles de dirección	
TSAP ¹ (ASCII)	TSAPs de las CPUs local e interlocutora en formato ASCII
ID TSAP	TSAPs de las CPUs local e interlocutora en formato hexadecimal

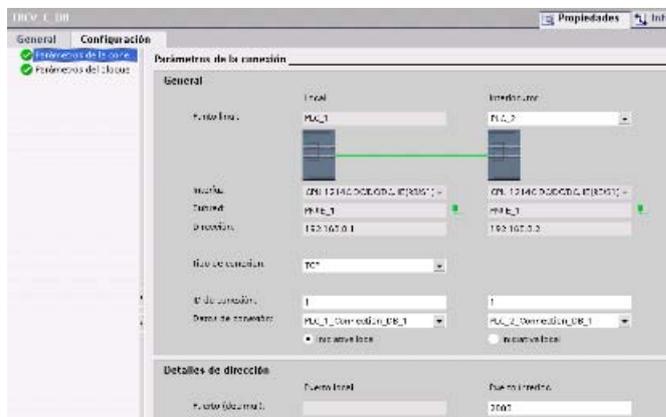
¹ Al configurar una conexión con una CPU S7-1200 para ISO on TCP, utilice sólo caracteres ASCII en la extensión TSAP para los interlocutores pasivos.

Transport Control Protocol (TCP)

TCP es un protocolo estándar descrito por RFC 793: Transmission Control Protocol. El objetivo principal de TCP es ofrecer un servicio de conexión seguro y fiable entre pares de procesos. Este protocolo tiene las características siguientes:

- Protocolo de comunicación eficiente puesto que está vinculado estrechamente al hardware
- Adecuado para cantidades de datos medianas y grandes (hasta 8192 bytes)
- Ofrece numerosas prestaciones más a las aplicaciones, en particular:
 - Recuperación de errores
 - Control de flujo
 - Fiabilidad
- Protocolo orientado a la conexión
- Puede utilizarse muy flexiblemente con sistemas de terceros que soporten únicamente TCP
- Apto para routing
- Son aplicables sólo las longitudes de datos estáticas.
- Los mensajes se acusan.
- Las aplicaciones se direccionan usando números de puerto.

- La mayoría de los protocolos de aplicación (p. ej. TELNET y FTP) utilizan TCP.
- Es necesario programar la gestión de datos debido a la interfaz de programación SEND / RECEIVE.



Parámetro	Definición
General	
Punto final: Interlocutor	Nombre asignado a la CPU interlocutora (receptora)
Interfaz	Nombre asignado a las interfaces
Subred	Nombre asignado a las subredes
Dirección	Direcciones IP asignadas
Tipo de conexión	Tipo de protocolo Ethernet
ID de conexión	Número de ID
Datos de conexión	Ubicación de almacenamiento de datos de las CPUs local e interlocutora
Establecimiento de conexión activo	Botón de opción para seleccionar la CPU local o interlocutora como conexión activa
Detalles de dirección	
Puerto (decimal)	Puerto de la CPU local en formato decimal

7.5 Información de referencia

7.5.1 Localizar la dirección Ethernet (MAC) en la CPU

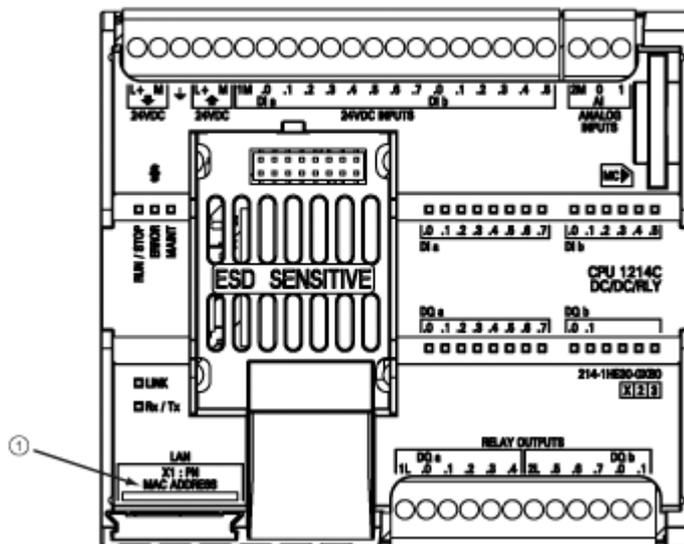
En las redes PROFINET, una dirección MAC (Media Access Control o control de acceso al medio) es un identificador que el fabricante asigna a las tarjetas adaptadoras para su identificación. Generalmente, una dirección MAC codifica el número de identificación registrado del fabricante.

El formato estándar (IEEE 802.3) permite imprimir direcciones MAC de forma amigable y consta de seis grupos de dos dígitos hexadecimales, separados por guiones (-) o dos puntos (:), en orden de transmisión (p. ej. 01-23-45-67-89-ab ó 01:23:45:67:89:ab).

Nota

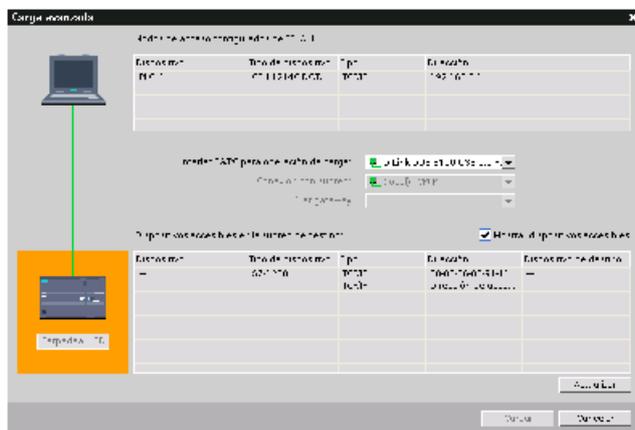
Toda CPU se suministra de fábrica con una dirección MAC unívoca y permanente. La dirección MAC de la CPU no se puede cambiar.

La dirección MAC está impresa en la esquina inferior izquierda en el frente de la CPU. Para ver la dirección MAC es necesario abrir las tapas inferiores del bloque de terminales.



① Dirección MAC

Inicialmente, la CPU no tiene dirección IP, sino sólo una dirección MAC ajustada de fábrica. Para la comunicación PROFINET es necesario que todos los dispositivos tengan asignada una dirección IP unívoca.



La función de la CPU "Cargar en dispositivo" y el diálogo "Carga avanzada en dispositivo" permiten visualizar todos los dispositivos de red accesibles y asegurar que se han asignado direcciones IP unívocas a todos ellos. Este diálogo muestra todos los dispositivos accesibles y disponibles con sus respectivas direcciones MAC e IP asignadas. Las direcciones MAC son especialmente importantes para identificar dispositivos que no disponen de la dirección IP unívoca necesaria.

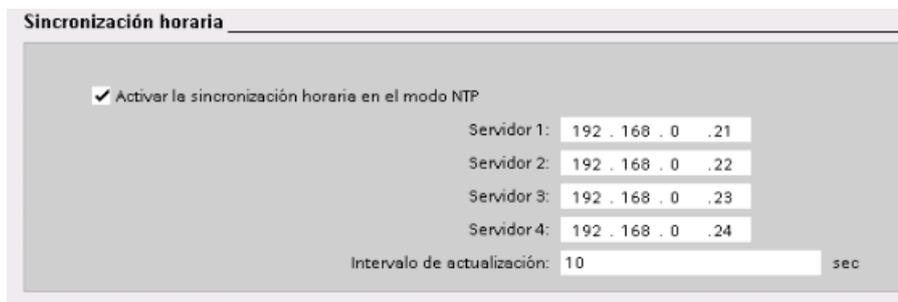
7.5.2 Configurar la sincronización del Network Time Protocol (NTP)

El Network Time Protocol (NTP) es un protocolo ampliamente utilizado para sincronizar los relojes de los sistemas informáticos con los servidores de hora de Internet. Ofrece una precisión típica de menos de un milisegundo en LANs y de hasta pocos milisegundos en WANs. Las configuraciones NTP típicas utilizan varios servidores redundantes y distintas rutas de red para alcanzar una alta precisión y fiabilidad.

La subred NTP funciona con una jerarquía de niveles en la que un número - denominado "stratum" - se asigna a cada nivel. Los servidores stratum 1 (primarios) del nivel más inferior se sincronizan directamente con los servicios de hora nacionales. Los servidores stratum 2 (secundarios) del nivel inmediatamente superior se sincronizan con los servidores stratum 1, etc.

Parámetros de sincronización horaria

En la ventana de propiedades, seleccione la entrada de configuración "Sincronización horaria". El TIA Portal visualiza el diálogo de configuración de la sincronización horaria:



Nota

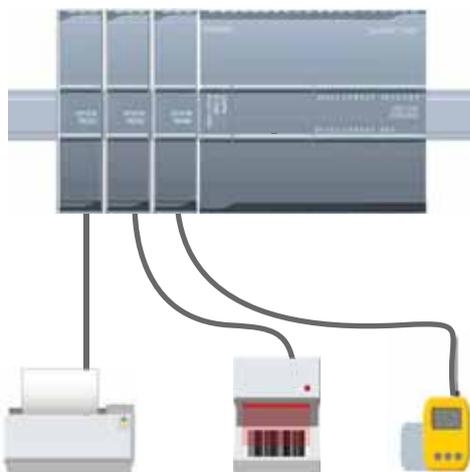
Todas las direcciones IP se configuran al cargar el proyecto en el dispositivo.

La tabla siguiente define los parámetros de la sincronización horaria:

Parámetro	Definición
Activar la sincronización horaria vía servidores NTP (Network Time Protocol)	Haga clic en la casilla de verificación para activar la sincronización horaria vía servidores NTP.
Servidor 1	Dirección IP asignada al servidor de hora de red 1
Servidor 2	Dirección IP asignada al servidor de hora de red 2
Servidor 3	Dirección IP asignada al servidor de hora de red 3
Servidor 4	Dirección IP asignada al servidor de hora de red 4
Intervalo de sincronización horaria	Valor del intervalo (seg)

Comunicación punto a punto (PtP)

La CPU soporta el protocolo punto a punto (PtP) para la comunicación serie basada en caracteres, en la que la aplicación de usuario define e implementa íntegramente el protocolo seleccionado. PtP ofrece una libertad y flexibilidad máximas, pero requiere una implementación exhaustiva en el programa de usuario.



PtP ofrece numerosas posibilidades, a saber:

- Posibilidad de enviar directamente información a un dispositivo externo, p. ej. una impresora
- Posibilidad de recibir información de otros dispositivos, p. ej. lectores de código de barras, lectores RFID, cámaras o sistemas de visión de terceros y muchos dispositivos más
- Posibilidad de intercambiar información (enviar y recibir datos) con otros dispositivos tales como equipos GPS, cámaras o sistemas de visión de terceros, módems radio y muchos más

La comunicación PtP es una comunicación serie que utiliza UARTs estándar para soportar distintas velocidades de transferencia y opciones de paridad. El módulo de comunicación (CM) RS232 o RS485 ofrece la interfaz eléctrica para realizar la comunicación PtP.

STEP 7 Basic ofrece librerías de instrucciones que pueden utilizarse para programar la aplicación. Estas librerías incluyen funciones PtP para los protocolos siguientes:

- Protocolo de accionamientos USS
- Protocolo maestro Modbus RTU
- Protocolo esclavo Modbus RTU

8.2 Utilizar los módulos de comunicación RS232 y RS485

Dos módulos de comunicación (CMs) ofrecen la interfaz para la comunicación PtP: CM 1241 RS485 (Página 357) y CM 1241 RS232 (Página 358). Es posible conectar como máximo tres CMs (de cualquier tipo). El CM se monta a la izquierda de la CPU o de otro CM. Encontrará información detallada acerca del montaje y desmontaje de módulos en el capítulo "Montaje" (Página 33).

Los módulos de comunicación RS232 y RS485 tienen las características siguientes:

- Puerto aislado galvánicamente
- Soporte de protocolos punto a punto
- Configuración y programación mediante instrucciones avanzadas y funciones de librería
- Visualización de la actividad de transmisión y recepción mediante LEDs
- LED de diagnóstico
- Alimentación eléctrica suministrada por la CPU. No necesita conexión a una fuente de alimentación externa.

Para más información, consulte los datos técnicos de los módulos de comunicación (Página 357).

8.3 Configurar los puertos de comunicación

Los módulos de comunicación pueden configurarse de dos maneras:

- Utilizando la configuración de dispositivos en STEP 7 Basic para configurar los parámetros de puerto (velocidad de transferencia y paridad), así como los de transmisión y recepción. Los ajustes de la configuración de dispositivos se guardan permanentemente en la CPU. Estos ajustes se aplican tras desconectar y conectar la alimentación y una transición de RUN a STOP.
- Utilice las instrucciones PORT_CFG, SEND_CFG y RCV_CFG para configurar los parámetros. La configuración del puerto ajustada mediante las instrucciones es válida mientras la CPU esté en modo RUN. La configuración del puerto se inicializa con los valores predeterminados de la configuración del dispositivo tras cambiar a STOP o desconectar y volver a conectar la alimentación.

Tras configurar los dispositivos de hardware (Página 77) es preciso ajustar los parámetros de las interfaces de comunicación seleccionando uno de los CMs del rack.

Los parámetros del CM seleccionado se visualizan en la ficha "Propiedades" de la ventana de inspección. Seleccione "Configuración del puerto" para editar los parámetros siguientes:

- Velocidad de transferencia
- Paridad
- Bits de parada
- Control de flujo (sólo RS232)
- Tiempo de espera

A excepción del control de flujo, los parámetros de configuración del puerto son iguales, indistintamente de si se desea configurar un módulo de comunicación RS232 o RS485. Los valores de los parámetros pueden diferir.

También es posible configurar el puerto (o modificar una configuración existente) desde el programa de usuario con la instrucción PORT_CFG (Página 284).



Nota

Los valores de parámetros ajustados mediante la instrucción PORT_CFG en el programa de usuario prevalecen sobre la configuración definida en STEP 7 Basic. El S7-1200 no conserva los parámetros ajustados mediante la instrucción PORT_CFG si se desconecta la alimentación.

Velocidad de transferencia: El valor predeterminado para la velocidad de transferencia es 9,6 Kbits/s. Los ajustes posibles son:

300 baudios	2,4 Kbits	19,2 Kbits	76,8 Kbits
600 baudios	4,8 Kbits	28,4 Kbits	115,2 Kbits
1,2 Kbits	9,6 Kbits	57,6 Kbits	

Paridad: El valor predeterminado para la paridad es "sin paridad". Los ajustes posibles son:

- Sin paridad
- Paridad par
- Paridad impar
- Paridad Mark (poner bit de paridad a 1)
- Paridad Space (poner bit de paridad a 0)

Bits de parada: Es posible ajustar uno o dos bits de parada. El ajuste predeterminado es uno.

Control de flujo: Para el módulo de comunicación RS232 puede seleccionarse el control de flujo por hardware o software de la manera descrita en el apartado "Gestionar el control de flujo (Página 272)". Si se selecciona el control de flujo por hardware, es posible indicar si la señal RTS debe estar siempre ON o si debe conmutarse. Si se selecciona el control de flujo por software, es posible definir los caracteres ASCII para los caracteres XON y XOFF.

El módulo de comunicación RS485 no soporta el control de flujo.

Tiempo de espera: El tiempo de espera especifica el periodo que el módulo de comunicación espera hasta recibir CTS tras confirmar RTS, o bien hasta recibir un XON tras recibir un XOFF, dependiendo del tipo de control de flujo. Si el tiempo de espera transcurre antes de que el módulo de comunicación reciba un CTS o XON esperado, el módulo de comunicación cancelará la operación de transmisión y devolverá un error al programa de usuario. El tiempo de espera se indica en milisegundos. El rango válido está comprendido entre 0 y 65535 milisegundos.

8.4 Gestionar el control de flujo

El control de flujo es un mecanismo que permite regular el intercambio de datos entre un emisor y un receptor para evitar pérdidas de datos. El control de flujo garantiza que un emisor no envíe más información de la que el receptor es capaz de procesar. El control de flujo puede realizarse por hardware o software. El CM RS232 soporta el control de flujo tanto por hardware como por software. El CM RS485 no soporta el control de flujo. El tipo de control de flujo se especifica al configurar el puerto (Página 271) o con la instrucción `PORT_CFG`.

El control de flujo por hardware funciona a través de las señales de comunicación RTS (Request To Send o petición de transmitir) y CTS (Clear To Send o listo para transmitir). En el caso del CM RS232, la señal RTS se emite desde el pin 7 y la señal CTS se recibe por el pin 8. El CM 1241 es un DTE (Data Terminal Equipment o equipo terminal de datos) que confirma RTS como salida y monitoriza CTS como entrada.

Control de flujo por hardware: RTS conmutado

Si se habilita el control de flujo por hardware con RTS conmutado en un CM RS232, el módulo activa la señal RTS para enviar datos. El módulo vigila la señal CTS para determinar si el receptor puede aceptar datos. Estando activa la señal CTS, el módulo puede transmitir datos mientras que la señal CTS permanezca activa. Si se desactiva la señal CTS, la transmisión debe detenerse.

La transmisión se reanuda cuando se reactiva la señal CTS. Si la señal CTS no se vuelve a activar dentro del tiempo de espera configurado, el módulo cancelará la transmisión y devolverá un error al programa de usuario. El tiempo de espera se especifica en la configuración del puerto (Página 271).

El control de flujo con RTS conmutado es útil para los dispositivos que requieren una señal de que la transmisión está activa. Un ejemplo sería un módem radio que utiliza RTS como señal "Key" para energizar el transmisor de radio. El control de flujo con RTS conmutado no funciona con módems telefónicos estándar. Seleccione la opción "RTS siempre on" para los módems telefónicos.

Control de flujo por hardware: RTS siempre ON

Si se utiliza la opción "RTS siempre on", el CM 1241 activará RTS de forma predeterminada. Un dispositivo (p. ej. un módem telefónico) vigila la señal RTS del CM y la utiliza como CTS (Clear To Send o listo para transmitir). El módem transmitirá al CM sólo si RTS está activo, es decir, cuando el módem telefónico detecte un CTS activo. Si RTS no está activo, el módem telefónico no transmitirá al CM.

Para permitir que el módem envíe datos al CM en cualquier momento, configure el control de flujo por hardware con "RTS siempre ON". De esta manera, el CM activa la señal RTS permanentemente. El CM no desactivará RTS incluso si el módulo no puede aceptar caracteres. El emisor debe garantizar que no se desborde el búfer de recepción del CM.

Uso de las señales DTR (Data Terminal Block Ready) y DSR (Data Set Ready)

El CM activa DTR para cualquier tipo de control de flujo por hardware. El módulo transmite datos sólo cuando se activa la señal DSR. El estado de DSR se evalúa únicamente al comienzo de la transmisión. Si DSR se desactiva tras haberse iniciado la transmisión, ésta no se detendrá.

Control de flujo por software

El control de flujo por software utiliza caracteres especiales en los mensajes para proporcionar el control de flujo. Se trata de caracteres ASCII que representan XON y XOFF.

XOFF indica que una transmisión debe detenerse. XON indica que una transmisión puede reanudarse.

Cuando el emisor recibe un carácter XOFF del receptor, deja de transmitir datos. La transmisión se reanuda cuando el emisor recibe un carácter XON. Si no recibe un carácter XON dentro del tiempo de espera indicado en la configuración del puerto (Página 271), el CM cancelará la transmisión y devolverá un error al programa de usuario.

Para el control de flujo por software se requiere la comunicación dúplex, puesto que el receptor debe poder enviar XOFF al emisor durante una transmisión. El control de flujo por software sólo es posible en los mensajes que contengan únicamente caracteres ASCII. Los protocolos binarios no pueden utilizar el control de flujo por software.

8.5 Configurar los parámetros de transmisión y recepción

Para que el PLC pueda intervenir en la comunicación PtP es preciso configurar parámetros para transmitir y recibir mensajes. Estos parámetros determinan cómo deben funcionar las comunicaciones al transmitir o recibir mensajes a/de un dispositivo de destino.

Configurar los parámetros de transmisión



Durante la configuración del CM se ajusta cómo una interfaz de comunicación debe transmitir datos, definiendo la propiedad "Configuración de la transferencia de mensajes" para el CM seleccionado.

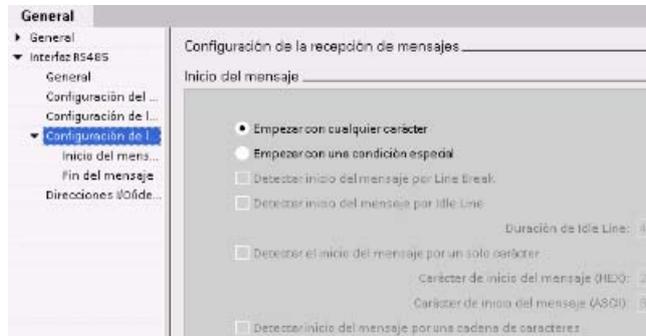
Los parámetros de transmisión de mensajes también se pueden configurar o modificar dinámicamente desde el programa de usuario utilizando la instrucción SEND_CFG (Página 286).

Nota

Los valores de parámetros ajustados mediante la instrucción SEND_CFG en el programa de usuario prevalecen sobre la configuración del puerto. La CPU no conserva los parámetros ajustados mediante la instrucción SEND_CFG si se desconecta la alimentación.

Parámetro	Definición
Retardo RTS ON	Determina el tiempo que debe esperarse tras activar RTS antes de iniciar la transmisión. El rango está comprendido entre 0 y 65535 ms (el valor predeterminado es 0). Este parámetro sólo es válido si en la configuración del puerto (Página 271) se ha definido el control de flujo por hardware. CTS se evalúa una vez transcurrido el retardo RTS ON. Este parámetro es aplicable únicamente a los módulos RS232.
Retardo RTS OFF	Determina el tiempo que debe esperarse antes de desactivar RTS tras finalizar la transmisión. El rango está comprendido entre 0 y 65535 ms (el valor predeterminado es 0). Este parámetro sólo es válido si en la configuración del puerto (Página 271) se ha definido el control de flujo por hardware. Este parámetro es aplicable únicamente a los módulos RS232.
Enviar pausa al inicio del mensaje Número de bit times en una pausa	Determina que, al inicio de cada mensaje, se enviará una pausa una vez transcurrido el tiempo de retardo RTS ON (si se ha configurado) y si CTS está activo. Es preciso indicar cuántos tiempos de bit ("bit times") constituyen una pausa cuando la línea se mantiene en una condición "Space". El ajuste predeterminado es 12 y el valor máximo es 65535, hasta un límite de ocho segundos.
Enviar Idle Line tras la pausa Idle Line tras pausa	Determina que se enviará una "idle line" tras una pausa al inicio del mensaje. El parámetro "Idle Line tras pausa" indica cuántos tiempos de bit constituyen una "idle line" cuando la línea se mantiene en una condición "Mark". El ajuste predeterminado es 12 y el valor máximo es 65535, hasta un límite de ocho segundos.

Configurar los parámetros de recepción



En la configuración de dispositivos se determina cómo debe recibir datos una interfaz de comunicación, así como detectar el comienzo y fin de un mensaje. Estos parámetros se definen en la configuración de la recepción de mensajes para el CM seleccionado.

Los parámetros de recepción de mensajes también se pueden configurar o modificar dinámicamente desde el programa de usuario utilizando la instrucción RCV_CFG (Página 288).

Nota

Los valores de parámetros ajustados mediante la instrucción RCV_CFG en el programa de usuario prevalecen sobre la configuración del puerto. La CPU no conserva los parámetros ajustados mediante la instrucción RCV_CFG si se desconecta la alimentación.

Encontrará más información en la descripción de la instrucción RCV_CFG.

Parámetros de inicio del mensaje

Es posible determinar cómo el módulo de comunicación debe detectar el inicio de un mensaje. Los caracteres iniciales y los caracteres que contiene el mensaje se depositan en el búfer de recepción hasta que se cumpla una condición final configurada.

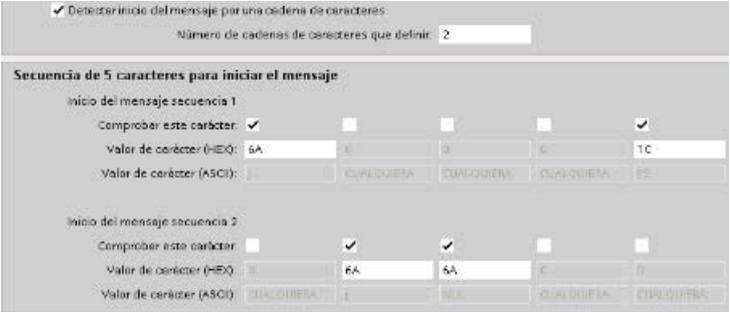
Es posible definir varias condiciones de inicio. Todas las condiciones de inicio deberán cumplirse antes de que se considere iniciado el mensaje. Por ejemplo, si se configura un tiempo de "idle line" y un carácter de inicio específico, el CM determinará primero si se cumple el requisito de tiempo de "idle line" y buscará luego el carácter de inicio indicado. Si se recibe algún otro carácter (que no sea el carácter de inicio indicado), el CM reiniciará la búsqueda del inicio del mensaje buscando nuevamente el tiempo de "idle line".

El orden de comprobación de las condiciones de inicio es el siguiente:

- Idle Line
- Line Break
- Caracteres o secuencias de caracteres

Si al comprobar varias condiciones de inicio no se cumple una de ellas, el CM reiniciará la comprobación con la primera condición requerida.

Parámetro	Definición
Carácter de inicio	La condición "Carácter de inicio" indica que, en cuanto se reciba correctamente un carácter en particular, se iniciará el mensaje. Este carácter será el primer carácter del mensaje. Todo carácter que se reciba antes de este carácter específico se descartará.
Empezar con cualquier carácter	La condición "Carácter cualquiera" indica que, en cuanto se reciba correctamente un carácter, se iniciará el mensaje. Este carácter será el primer carácter del mensaje.
Line Break	La condición "Line Break" especifica que la recepción de un mensaje debe comenzar cuando se reciba un carácter de salto de línea.
Idle Line	La condición "Idle Line" especifica que la recepción de un mensaje debe comenzar una vez que la línea de recepción haya estado inactiva o en reposo durante el número de tiempos de bit indicado. Cuando se cumpla esta condición, se iniciará el mensaje.
Condición especial: Detectar el inicio del mensaje por un solo carácter	Determina que un carácter en particular indique el principio de un mensaje. El ajuste predeterminado STX.

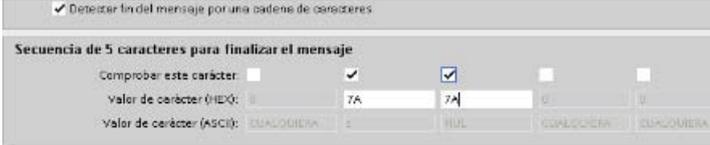
Parámetro	Definición
<p>Condición especial: Detectar inicio del mensaje por una cadena de caracteres</p>	<p>Determina que una secuencia de caracteres en particular indica el inicio de un mensaje. Es posible especificar cinco caracteres como máximo para una secuencia. Para cada posición de carácter se indica bien sea un carácter hexadecimal específico, o bien que el carácter se ignore en la comparación de secuencias.</p> <p>Las secuencias entrantes se comparan con las condiciones de inicio configuradas hasta que se cumple una condición de inicio. Una vez que cumplida la secuencia de inicio, se iniciará la evaluación de las condiciones de fin.</p> <p>Es posible configurar como máximo cinco secuencias de caracteres específicas que se pueden habilitar o inhibir según sea necesario. La condición de inicio se cumple cuando se presenta una de las secuencias de caracteres configuradas.</p>
<p>Ejemplo de configuración</p>	 <p>En esta configuración, la condición de inicio se cumple cuando se presenta uno de los patrones siguientes:</p> <ul style="list-style-type: none"> • Cuando se recibe una secuencia de cinco caracteres en la que el primer carácter es 0x6A y, el quinto, 0x1C. Los caracteres de las posiciones 2, 3 y 4 pueden ser un carácter cualquiera en esta configuración. Tras recibirse el quinto carácter comienza la evaluación de las condiciones de fin. • Cuando se reciben dos caracteres 0x6A consecutivos precedidos de un carácter cualquiera. En este caso, la evaluación de las condiciones de fin comienza tras recibirse el segundo 0x6A (3 caracteres). El carácter que precede el primer 0x6A se incluye en la condición de inicio. <p>Secuencias de ejemplo que cumplirían esta condición de inicio:</p> <ul style="list-style-type: none"> • <carácter cualquiera> 6A 6A • 6A 12 14 18 1C • 6A 44 A5 D2 1C

Parámetros de fin del mensaje

También es posible definir cómo la interfaz de comunicación debe detectar el fin de un mensaje. Es posible configurar varias condiciones de fin del mensaje. Si se cumple alguna de las condiciones configuradas, finalizará el mensaje.

Es posible especificar simultáneamente varias condiciones de fin. El mensaje finalizará cuando se cumpla alguna de las condiciones de fin. Por ejemplo, es posible especificar una condición de fin con un timeout de fin de mensaje de 300 milisegundos, un timeout entre caracteres de 40 tiempos de bit y una longitud máxima de 50 bytes. El mensaje terminará si la recepción tarda más de 300 milisegundos o si la distancia temporal entre dos caracteres es superior a 40 tiempos de bit o si se reciben 50 bytes.

Parámetro	Definición
Detectar fin del mensaje por tiempo de mensaje excedido	El mensaje finaliza cuando ha transcurrido el tiempo de espera configurado para el fin del mensaje. El periodo de timeout del mensaje comienza cuando se recibe el primer carácter que cumpla los criterios de inicio del mensaje. El valor predeterminado es 200 ms. El rango válido está comprendido entre 0 y 65535 ms.
Detectar fin del mensaje por tiempo de respuesta excedido	El mensaje finaliza cuando ha transcurrido el tiempo de espera configurado para una respuesta antes de que se reciba una secuencia de inicio válida. El periodo de timeout de respuesta comienza cuando finaliza la transmisión. El timeout de respuesta predeterminado es 200 ms. El rango válido está comprendido entre 0 y 65535 ms. Es preciso configurar una condición de fin adicional para indicar el fin real del mensaje.
Detectar fin del mensaje por tiempo excedido entre caracteres	El mensaje finaliza cuando ha transcurrido el timeout máximo configurado entre caracteres consecutivos de un mensaje. El valor predeterminado del tiempo excedido entre caracteres es 12 tiempos de bit y el valor máximo es 65535 tiempos de bit, hasta un límite máximo de ocho segundos.
Detectar fin del mensaje por longitud máxima	El mensaje finaliza cuando se ha recibido el número de caracteres máximo configurado. El ajuste predeterminado es 0 bytes y el valor máximo es 1024 bytes.
Leer longitud del mensaje en el mensaje	El mensaje en sí especifica la longitud del mensaje. El mensaje finaliza cuando se recibe un mensaje con la longitud especificada. El método para especificar e interpretar la longitud del mensaje se describe más adelante.
Detectar fin del mensaje por un carácter	El mensaje finaliza cuando se recibe un carácter especificado.

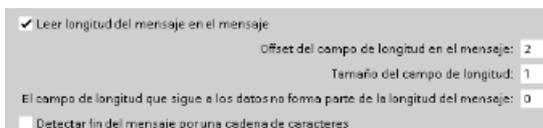
Parámetro	Definición
Detectar fin del mensaje por una cadena de caracteres	<p>El mensaje finaliza cuando se recibe una secuencia de caracteres especificada. Es posible especificar una secuencia de cinco caracteres como máximo. Para cada posición de carácter se indica bien sea un carácter hexadecimal específico, o bien que el carácter se ignore en la comparación de secuencias.</p> <p>Los caracteres iniciales ignorados no forman parte de la condición de fin. Los caracteres finales ignorados forman parte de la condición de fin.</p>
Ejemplo de configuración	 <p>En este caso, la condición de fin se cumple cuando se reciben dos caracteres 0x7A consecutivos, seguidos de dos caracteres cualquiera. El carácter que precede el patrón 0x7A 0x7A no forma parte de la secuencia de caracteres final. Los dos caracteres que le siguen al patrón 0x7A 0x7A se requieren para terminar la secuencia de caracteres final. Aunque los valores de las posiciones de carácter 4 y 5 son irrelevantes, deben recibirse para que se cumpla la condición de fin.</p>

Indicación de la longitud del mensaje dentro del mensaje

Si se selecciona la condición especial en la que la longitud del mensaje se incluye en el mensaje, es preciso indicar tres parámetros que definen la información acerca de la longitud del mensaje.

La estructura real del mensaje varía según el protocolo utilizado. Los tres parámetros son los siguientes:

- n: Posición de carácter (en base 1) dentro del mensaje que inicia el indicador de longitud
- Tamaño de longitud: Número de bytes (uno, dos o cuatro) del indicador de longitud
- Longitud m: Número de caracteres posteriores al indicador de longitud que no se incluyen en el conteo de longitud



Estos campos aparecen en la configuración de la recepción de mensajes de las propiedades del dispositivo.

Ejemplo 1: Considerar un mensaje estructurado según el protocolo siguiente:

STX	Len (n)	Caracteres 3 a 14 contados según la longitud											
		ADR		PKE		INDEX		PWD		STW		HSW	
1	2	3	4	5	6	7	8	9	10	11	12	13	14
STX	0x0C	xx	xxxx		xxxx		xxxx		xxxx		xxxx		xx

Configure los parámetros de longitud de recepción de este mensaje como se indica a continuación:

- n = 2 (la longitud del mensaje comienza con el byte 2.)
- Tamaño de longitud = 1 (la longitud del mensaje se define en un byte.)
- Longitud m = 0 (no hay caracteres adicionales posteriores al indicador de longitud no incluidos en el conteo de longitud. Doce caracteres le siguen al indicador de longitud.)

En este ejemplo, los caracteres 3 a 14 (inclusive) son los caracteres que cuenta Len (n).

Ejemplo 2: Considerar otro mensaje estructurado según el protocolo siguiente:

SD1	Len (n)	Len (n)	SD2	Caracteres 5 a 10 contados según la longitud						FCS	ED
				DA	SA	FA	Unidad de datos=3 bytes				
1	2	3	4	5	6	7	8	9	10	11	12
xx	0x06	0x06	xx	xx	xx	xx	xx	xx	xx	xx	xx

Configure los parámetros de longitud de recepción de este mensaje como se indica a continuación:

- n = 3 (la longitud del mensaje comienza con el byte 3.)
- Tamaño de longitud = 1 (la longitud del mensaje se define en un byte.)
- Longitud m = 3 (tres caracteres posteriores al indicador de longitud no se cuentan en la longitud. En el protocolo de este ejemplo, los caracteres SD2, FCS y ED no se incluyen en el conteo de longitud. Los seis caracteres restantes se incluyen en el conteo de longitud. Por tanto, el número total de caracteres posteriores al indicador de longitud es nueve.)

En este ejemplo, los caracteres 5 a 10 (inclusive) son los caracteres que cuenta Len (n).

8.6 Programar la comunicación PtP

STEP 7 Basic ofrece instrucciones avanzadas que permiten al programa de usuario establecer comunicaciones punto a punto utilizando un protocolo diseñado y definido en el programa de usuario. Estas instrucciones se dividen en dos categorías, a saber:

- Instrucciones de configuración
- Instrucciones de comunicación

Instrucciones de configuración

Para que el programa de usuario pueda intervenir en la comunicación PtP, es preciso configurar el puerto de comunicación y los parámetros para enviar y recibir datos.

El puerto y los mensajes pueden configurarse para todo módulo de comunicación en la configuración de dispositivos o mediante estas instrucciones en el programa de usuario:

- PORT_CFG
- SEND_CFG
- RCV_CFG

Instrucciones de comunicación

Las instrucciones de comunicación PtP permiten al programa de usuario enviar y recibir mensajes a/de los módulos de comunicación. Encontrará más información sobre cómo transferir datos con estas instrucciones en el apartado Coherencia de datos (Página 96).

Todas las funciones PtP funcionan de forma asíncrona. El programa de usuario puede utilizar una arquitectura de sondeo para determinar el estado de las transmisiones y recepciones. SEND_PTP y RCV_PTP pueden ejecutarse simultáneamente. Según sea necesario, los módulos de comunicación depositan los mensajes de transmisión y recepción en un búfer cuyo tamaño máximo es 1024 bytes.

Los módulos de comunicación envían y reciben mensajes a/de los dispositivos punto a punto reales. El protocolo de mensajes se encuentra en un búfer que se recibe de o se envía a un puerto de comunicación específico.

- SEND_PTP
- RCV_PTP

Se dispone de instrucciones adicionales que permiten inicializar el búfer de recepción, así como consultar y activar determinadas señales RS232.

- RCV_RST
- SGN_GET
- SGN_SET

8.6.1 Arquitectura de sondeo

Las instrucciones punto a punto del S7-1200 deben llamarse cíclicamente/periódicamente para comprobar si se han recibido mensajes. Un sondeo de la transmisión notifica al programa de usuario el fin de la transmisión.

Arquitectura de sondeo: maestro

La secuencia típica de un maestro es la siguiente:

1. Una instrucción SEND_PTP inicia una transmisión al módulo de comunicación.
2. La instrucción SEND_PTP se ejecuta en los ciclos posteriores para determinar el progreso de la transmisión.
3. Cuando la instrucción SEND_PTP indica que ha finalizado la transmisión, el programa de usuario se prepara para recibir la respuesta.
4. La instrucción RCV_PTP se ejecuta repetidamente para comprobar si hay una respuesta. Una vez que el CM haya recibido un mensaje de respuesta, la instrucción RCV_PTP copiará la respuesta en la CPU e indicará que se han recibido datos nuevos.
5. El programa de usuario puede procesar la respuesta.
6. Vaya al paso 1 y repita el ciclo.

Arquitectura de sondeo: esclavo

La secuencia típica de un esclavo es la siguiente:

1. El programa de usuario debería ejecutar la instrucción RCV_PTP en cada ciclo.
2. Una vez que el CM haya recibido una petición, la instrucción RCV_PTP indicará que los nuevos datos están listos y la petición se copiará en la CPU.
3. El programa de usuario debería procesar la petición y generar una respuesta.
4. Utilice una instrucción SEND_PTP para enviar la respuesta al maestro.
5. Ejecute repetidamente SEND_PTP para asegurarse de que se realiza la transmisión.
6. Vaya al paso 1 y repita el ciclo.

El esclavo debe llamar RCV_PTP con la suficiente frecuencia para recibir una transmisión del maestro antes de que se produzca un timeout del maestro a la espera de una respuesta. Para realizar esta tarea, el programa de usuario puede llamar RCV_PTP desde un OB de ciclo, previendo un tiempo de ciclo suficiente para recibir una transmisión del maestro antes de que transcurra el periodo de timeout. Si el tiempo de ciclo del OB se ajusta para que pueda ejecutarse dos veces dentro del periodo de timeout del maestro, el programa de usuario debería recibir todas las transmisiones sin pérdidas.

8.7 Instrucciones de comunicación punto a punto

8.7.1 Parámetros comunes de las instrucciones de comunicación punto a punto

Reacción de los LEDs del módulo de comunicación

El módulo de comunicación (CM) dispone de tres indicadores LED, a saber:

- LED de diagnóstico: este LED parpadea en color rojo hasta ser direccionado por la CPU. Tras arrancar la CPU, comprobará los módulos y direccionará el módulo CM. El LED de diagnóstico comienza a parpadear en color verde. Esto significa que la CPU ha direccionado el CM, pero que aún no ha cargado la configuración en él. La configuración se carga en el módulo cuando el programa se carga en la CPU. Una vez cargado el programa en la CPU, el LED de diagnóstico del módulo de comunicación debería encenderse en color verde.
- LED de transmisión: este LED se encuentra encima del LED de recepción. El LED de transmisión se enciende cuando se están enviando datos por el puerto de comunicación.
- LED de recepción: este LED se enciende cuando se están recibiendo datos por el puerto de comunicación.

Resolución de tiempos de bit

Numerosos parámetros se definen en un número de tiempos de bit a la velocidad de transferencia configurada. Si el parámetro se define en tiempos de bit, puede ser independiente de la velocidad de transferencia. Todos los parámetros definidos en unidades de tiempos de bit pueden especificarse hasta un número máximo de 65535. No obstante, el tiempo máximo que puede medir el S7-1200 es 8 segundos.

Parámetro de entrada REQ

Numerosas de las instrucciones de comunicación punto a punto (PtP) utilizan una entrada REQ que inicia la operación en una transición de "low" a "high". El estado lógico de la entrada REQ debe ser "high" (TRUE) durante una ejecución de la instrucción, aunque puede permanecer TRUE durante un tiempo cualquiera. La instrucción no iniciará ninguna operación diferente hasta que no sea llamada con la entrada REQ puesta a FALSE, de manera que pueda inicializar el histórico de la entrada REQ. Esto es necesario para que la instrucción pueda detectar la transición de "low" a "high" para iniciar la siguiente operación.

Cuando se inserta una instrucción PtP, es preciso identificar el DB instancia. Utilice un DB unívoco para cada tipo de instrucción PtP. Todas las instrucciones SEND_PTP para un puerto determinado deben tener un mismo DB instancia, pero SEND_PTP y RCV_PTP deben tener DBs instancia diferentes. Esto garantiza que cada instrucción procese correctamente las entradas tales como REQ.

Parámetro de entrada PORT

En la lista desplegable (asociada a la entrada PORT), seleccione el identificador de puerto del CM que debe operar esta instancia de la instrucción. Este número equivale al "identificador de hardware" en la información de configuración del CM.

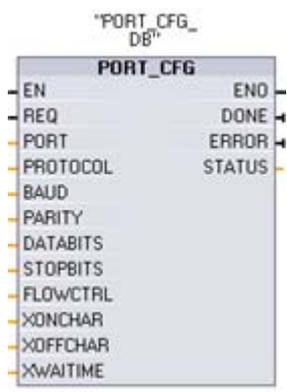
Parámetros de salida NDR, DONE, ERROR y STATUS

- La salida DONE indica que la operación solicitada se ha finalizado sin errores. Esta salida se activa durante un ciclo.
- La salida NDR (New Data Ready) indica que la acción solicitada se ha finalizado sin errores y que se han recibido datos nuevos. Esta salida se activa durante un ciclo.
- La salida ERROR indica que la acción solicitada ha finalizado con un error. Esta salida se activa durante un ciclo.
- La salida STATUS se utiliza para notificar errores o resultados de estado intermedios.
 - Si se activa el bit DONE o NDR, STATUS se pondrá a 0 o a un código de información.
 - Si se activa el bit ERROR, STATUS devolverá un código de error.
 - Si no se activa ninguno de los bits mencionados, la instrucción devolverá resultados que describen el estado actual de la función (p. ej. el estado "ocupado").

Códigos de condición comunes

STATUS (W#16#....)	Descripción
0000	Sin error
8x3A	Puntero no permitido en el parámetro x
8070	Se está utilizando toda la memoria de instancia interna
8080	Número de puerto no permitido
8081	Timeout, error del módulo u otro error interno
8082	Ha fallado la parametrización porque se está parametrizando en segundo plano
8083	Desbordamiento del búfer: El CM ha devuelto un mensaje recibido con una longitud superior a la que permite el parámetro de longitud.
8090	Longitud de mensaje incorrecta, submódulo incorrecto o mensaje no permitido
8091	Versión incorrecta en el mensaje de parametrización
8092	Longitud de registro incorrecta en el mensaje de parametrización

8.7.2 Instrucción PORT_CFG



La instrucción PORT_CFG (Configuración de puerto) permite cambiar los parámetros de puerto (p. ej. la velocidad de transferencia) desde el programa.

La configuración estática inicial del puerto puede ajustarse en las propiedades de la configuración de dispositivos. Como alternativa, es posible utilizar los valores predeterminados. La instrucción PORT_CFG puede ejecutarse en el programa con el fin de modificar la configuración. Los cambios de la configuración de PORT_CFG no se almacenan de forma permanente en la CPU. Los parámetros ajustados en la configuración de dispositivos se restablecen cuando la CPU cambia de RUN a STOP y tras desconectar y volver a conectar la alimentación.

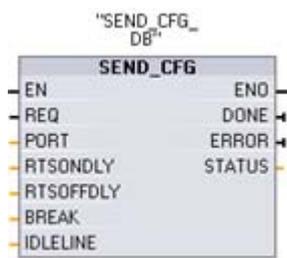
Encontrará más información en Configurar los puertos de comunicación (Página 271) y Gestionar el control de flujo (Página 272).

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de la configuración cuando se detecta un flanco ascendente en esta entrada.
PORT	IN	PORT	Identificador del puerto de comunicación: Esta dirección lógica es una constante que puede referenciarse en la ficha "Constantes" de la tabla de variables predeterminadas.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
PROTOCOL	IN	UInt	0 - Protocolo de comunicación punto a punto 1..n - Definición futura para protocolos específicos
BAUD	IN	UInt	Velocidad de transferencia del puerto: 1 - 300 baudios 2 - 600 baudios 3 - 1200 baudios 4 - 2400 baudios 5 - 4800 baudios 6 - 9600 baudios 7 - 19200 baudios 8 - 38400 baudios 9 - 57600 baudios 10 - 76800 baudios 11 - 115200 baudios
PARITY	IN	UInt	Paridad del puerto: 1 - Sin paridad 2 - Paridad par 3 - Paridad impar 4- Paridad Mark 5 - Paridad Space
DATABITS	IN	UInt	Bits por carácter: 1 - 8 bits de datos 2 - 7 bits de datos
STOPBITS	IN	UInt	Bits de parada: 1 - 1 bit de parada 2 - 2 bits de parada
FLOWCTRL	IN	UInt	Control de flujo: 1 - Sin control de flujo 2 - XON/XOFF 3 - RTS de hardware siempre ON 4 - RTS de hardware conmutado
XONCHAR	IN	Char	Determina el carácter que se utiliza como carácter XON. Generalmente, es un carácter DC1 (11H). Este parámetro se evalúa sólo si está habilitado el control de flujo.
XOFFCHAR	IN	Char	Determina el carácter que se utiliza como carácter XOFF. Generalmente, es un carácter DC3 (13H). Este parámetro se evalúa sólo si está habilitado el control de flujo.
XWAITIME	IN	UInt	Determina cuánto tiempo se debe esperar un carácter XON tras recibir un carácter XOFF o cuánto tiempo se debe esperar la señal CTS tras habilitar RTC (0 a 65535 ms). Este parámetro se evalúa sólo si está habilitado el control de flujo.
DONE	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución

STATUS (W#16#....)	Descripción
80A0	El protocolo indicado no existe.
80A1	La velocidad de transferencia indicada no existe.
80A2	La opción de paridad indicada no existe.
80A3	El número de bits de datos indicado no existe.
80A4	El número de bits de parada indicado no existe.
80A5	El tipo de control de flujo indicado no existe.
80A6	El tiempo de espera es 0 y el control de flujo está habilitado
80A7	XON y XOFF son valores no válidos

8.7.3 Instrucción SEND_CFG



La instrucción SEND_CFG (Enviar configuración) permite configurar dinámicamente los parámetros de transmisión serie de un puerto de comunicación punto a punto. Todos los mensajes en cola de espera en un módulo de comunicación (CM) se rechazarán una vez ejecutada la instrucción SEND_CFG.

La configuración estática inicial del puerto puede ajustarse en las propiedades de la configuración de dispositivos. Como alternativa, es posible utilizar los valores predeterminados. La instrucción SEND_CFG puede ejecutarse en el programa con el fin de modificar la configuración. Los cambios de configuración con SEND_CFG no se almacenan de forma permanente en el PLC. Los parámetros ajustados en la configuración de dispositivos se restablecen cuando la CPU cambia de RUN a STOP y tras desconectar y volver a conectar la alimentación. Consulte Configurar los parámetros de transmisión y recepción (Página 274).

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de configuración cuando se produce un flanco ascendente en esta entrada.
PORT	IN	PORT	Identificador del puerto de comunicación: Esta dirección lógica es una constante que puede referenciarse en la ficha "Constantes" de la tabla de variables predeterminadas.
RTSONDLY	IN	UInt	Número de milisegundos que debe esperarse tras la habilitación de RTS antes de cualquier transmisión de datos Tx. Este parámetro sólo es válido si está habilitado el control de flujo por hardware. 0 - 65535 ms. 0 inhibe la función.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
RTSOFFDLY	IN	UInt	Número de milisegundos que se debe esperar después de la transmisión de datos Tx antes de que se inhiba RTS: Este parámetro sólo es válido si está habilitado el control de flujo por hardware. 0 - 65535 ms. 0 inhibe la función.
BREAK	IN	UInt	Este parámetro indica que se enviará una pausa tras el inicio de cada mensaje durante el número de tiempos de bit indicado. El valor máximo posible es 65535 tiempos de bit. 0 inhibe la función. 8 segundos máx.
IDLELINE	IN	UInt	Este parámetro indica que la línea permanecerá inactiva durante el número de tiempos de bit indicado hasta el inicio de cada mensaje. El valor máximo posible es 65535 tiempos de bit. 0 inhibe la función. 8 segundos máx.
DONE	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución

STATUS (W#16#....)	Descripción
80B0	No se permite configurar una alarma de transmisión
80B1	El tiempo de pausa excede el valor permitido (2500 tiempos de bit)
80B2	El tiempo de inactividad excede el valor permitido (2500 tiempos de bit)

8.7.4 Instrucción RCV_CFG



La instrucción RCV_CFG (Recibir configuración) permite configurar dinámicamente los parámetros de recepción serie de un puerto de comunicación punto a punto. Esta instrucción configura las condiciones que indican el inicio y fin de un mensaje recibido. Todos los mensajes en cola de espera en un CM se rechazarán cuando se ejecute RCV_CFG.

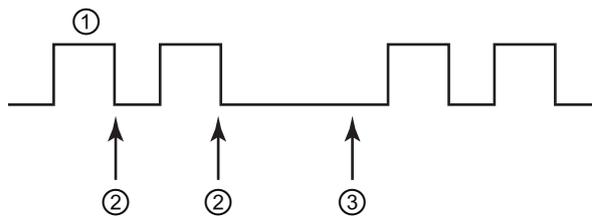
La configuración estática inicial del puerto del CM puede ajustarse en las propiedades de la configuración de dispositivos. Como alternativa, es posible utilizar los valores predeterminados. La instrucción RCV_CFG puede ejecutarse en el programa con el fin de modificar la configuración. Los cambios de configuración con RCV_CFG no se almacenan de forma permanente en el PLC. Los parámetros ajustados en la configuración de dispositivos se restablecen cuando la CPU cambia de RUN a STOP y tras desconectar y volver a conectar la alimentación. Encontrará más información en Configurar los parámetros de recepción (Página 274).

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de configuración cuando se produce un flanco ascendente en esta entrada.
PORT	IN	PORT	Identificador del puerto de comunicación: Esta dirección lógica es una constante que puede referenciarse en la ficha "Constantes" de la tabla de variables predeterminadas.
CONDITIONS	IN	CONDITIONS	La estructura de datos CONDITIONS indica las condiciones de inicio y fin del mensaje. Estas se describen más abajo.
DONE	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución

Condiciones de inicio de la instrucción RCV_PTP

La instrucción RCV_PTP utiliza la configuración indicada por la instrucción RCV_CFG para determinar el inicio y fin de los mensajes de comunicación punto a punto. Las condiciones de inicio determinan el inicio de un mensaje. Una o más condiciones de inicio pueden determinar el inicio de un mensaje. Si se especifica más de una condición de inicio, todas las condiciones deberán cumplirse antes de iniciar el mensaje. Condiciones de inicio posibles:

- "Carácter de inicio" indica que, en cuanto se reciba correctamente un carácter en particular, se iniciará el mensaje. Este carácter será el primer carácter del mensaje. Todo carácter que se reciba antes de este carácter específico se descartará.
- "Carácter cualquiera" indica que, en cuanto se reciba correctamente un carácter cualquiera, se iniciará el mensaje. Este carácter será el primer carácter del mensaje.
- "Line Break" especifica que la recepción de un mensaje debe comenzar cuando se reciba un carácter de salto de línea.
- "Idle Line" especifica que la recepción de un mensaje debe comenzar una vez que la línea de recepción haya estado inactiva o en reposo durante el número de tiempos de bit indicado. Cuando se cumpla esta condición, se iniciará el mensaje.



- ① Caracteres
- ② Reinicia el temporizador de línea inactiva
- ③ La línea inactiva se detecta y se inicia la recepción del mensaje

- **Secuencias variables:** Es posible crear condiciones de inicio basadas en un número variable de secuencias de caracteres (4 como máximo) que comprenden un número variable de caracteres (5 como máximo). Para toda posición de carácter en toda secuencia puede seleccionarse un carácter específico, o bien un carácter comodín, con lo que cualquier carácter cumplirá la condición. Esta condición de inicio puede utilizarse cuando diferentes secuencias de caracteres indican el inicio de un mensaje.

Considere el siguiente mensaje recibido en código hexadecimal: "68 10 aa 68 bb 10 aa 16" y las secuencias de inicio configuradas que muestra la tabla siguiente. Las secuencias de inicio se comienzan a evaluar al recibirse correctamente el primer carácter 68H. Tras recibirse correctamente el cuarto carácter (el segundo 68H), se cumple la condición de inicio 1. Una vez cumplidas las condiciones de inicio, se iniciará la evaluación de las condiciones de fin.

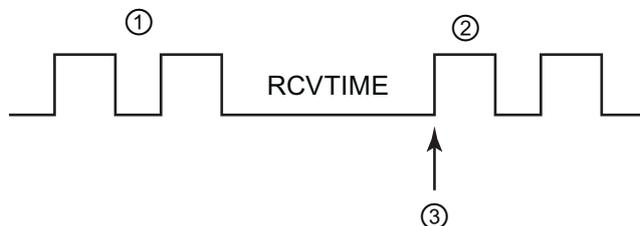
El procesamiento de la secuencia de inicio puede cancelarse debido a distintos errores de paridad, trama o tiempo excedido entre caracteres. Debido a estos errores, no se recibe el mensaje puesto que no se ha cumplido la condición de inicio.

Condición de inicio	Primer carácter	Primer carácter +1	Primer carácter +2	Primer carácter +3	Primer carácter +4
1	68H	xx	xx	68H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

Condiciones de fin de la instrucción RCV_PTP

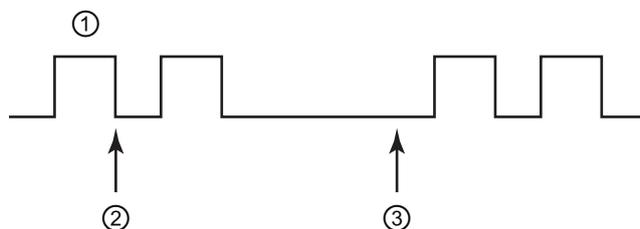
Las condiciones de fin configuradas determinan el fin de un mensaje. La primera aparición de una o más condiciones de fin configuradas determina el fin de un mensaje. Condiciones de fin posibles:

- "Tiempo de respuesta excedido" especifica que un carácter de la respuesta debe recibirse correctamente dentro del tiempo indicado por RCVTIME. El temporizador comenzará a contar en cuanto la transmisión finalice correctamente y el módulo inicie la recepción. Si no se recibe ningún carácter durante el periodo que indica RCVTIME, se devuelve un error a la instrucción RCV_PTP correspondiente. El timeout de respuesta no define una condición de fin específica. Sólo determina que un carácter debe recibirse correctamente dentro del tiempo indicado. Es preciso utilizar una condición de fin distinta para definir la condición de fin de los mensajes de respuesta.



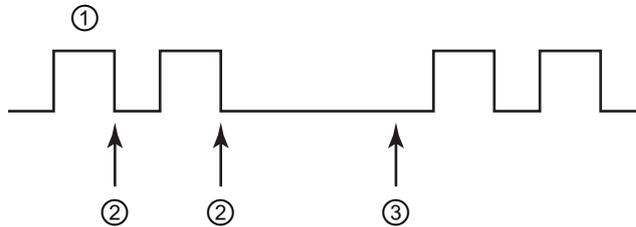
- ① Caracteres transmitidos
- ② Caracteres recibidos
- ③ El primer carácter debe haberse recibido correctamente hasta este instante

- "Tiempo de mensaje excedido" especifica que un mensaje debe recibirse dentro del tiempo indicado por MSGTIME. El temporizador comenzará a contar tan pronto como se cumpla la condición de inicio indicada.



- ① Caracteres recibidos
- ② Se cumple la condición de inicio del mensaje: el temporizador de mensajes arranca
- ③ El temporizador de mensajes se detiene y finaliza el mensaje

- El tiempo entre caracteres es el periodo medido desde el final de un carácter (el último bit de parada) hasta el final del carácter siguiente. Si el tiempo entre dos caracteres cualquiera excede el número de tiempos de bit configurado, se terminará el mensaje.



- ① Caracteres recibidos
- ② Inicializa el temporizador entre caracteres.
- ③ El temporizador entre caracteres se detiene y finaliza el mensaje con errores

- Longitud máxima: La recepción se detiene una vez que se haya recibido el número de caracteres indicado. Esta condición sirve para impedir un error de desbordamiento del búfer de mensajes.

Si esta condición de fin se combina con condiciones de fin por tiempo excedido (timeout) y ocurre una condición de timeout, los caracteres válidos recibidos hasta entonces estarán disponibles, aunque no se haya alcanzado la longitud máxima. Esto permite soportar protocolos de longitud variable si sólo se conoce la longitud máxima.

- Condición combinada "N + tamaño de longitud + longitud M". Esta condición de fin puede utilizarse para procesar un mensaje de tamaño variable que contenga un campo de longitud.
 - "N" indica la posición (número de caracteres desde el comienzo del mensaje) donde comienza el campo de longitud. (en base 1)
 - El "Tamaño de longitud" indica el tamaño del campo de longitud. Los valores válidos son 1, 2 ó 4 bytes.
 - La "Longitud M" indica el número de caracteres de fin (posteriores al campo de longitud) que no se incluyen en la longitud del mensaje. Este valor puede utilizarse para indicar la longitud de un campo de suma de verificación cuyo tamaño no se incluye en el campo de longitud.
 - Un ejemplo podría ser un formato de mensaje compuesto por un carácter de inicio, un carácter de dirección, un campo de longitud de un byte, los datos del mensaje, caracteres de suma de verificación y un carácter de fin. Las entradas identificadas con "Len" se corresponden con el parámetro N. El valor de N sería 3, indicando que el byte de longitud se posiciona en el 3er. byte desde el comienzo del mensaje. El valor de "Tamaño de longitud" sería 1, indicando que el valor de la longitud del mensaje está contenido en 1 byte. Los campos de suma de verificación y de los caracteres finales se corresponden con el parámetro "Longitud M". El valor de "Longitud M" sería 3, indicando el número de bytes de los campos de suma de verificación y caracteres.

Car. inicial (1)	Dirección (2)	Len (N) (3)	Mensaje ... (x)		Suma verif. y car. de fin Longitud M x+1 x+2 x+3		
xx	xx	xx	xx	xx	xx	xx	xx

- Caracteres variables: Esta condición de fin puede utilizarse para finalizar la recepción según diferentes secuencias de caracteres. Las secuencias pueden comprender un número variable de caracteres (5 como máximo). Para toda posición de carácter en toda secuencia puede seleccionarse un carácter específico, o bien un carácter comodín, con lo que cualquier carácter cumplirá la condición. Los caracteres iniciales configurados para ser ignorados no deben formar parte del mensaje. Los caracteres finales ignorados deben formar parte del mensaje.

Estructura de los tipos de datos del parámetro CONDITIONS, 1ª parte (condiciones de inicio)

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
STARTCOND	IN	UInt	Indica la condición de inicio: <ul style="list-style-type: none"> • 01H - Carácter inicial • 02H - Cualquier carácter • 04H - Line Break • 08H - Idle Line • 10H - Secuencia 1 • 20H - Secuencia 2 • 40H - Secuencia 3 • 80H - Secuencia 4
IDLETIME	IN	UInt	Número de tiempos de bit necesario para el timeout por "idle line". Se utiliza únicamente para una condición de "idle line". 0 a 65535
STARTCHAR	IN	Byte	Carácter de inicio utilizado con la condición de carácter de inicio.
STRSEQ1CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 1: Estos son los bits de habilitación de cada carácter de la secuencia de inicio. <ul style="list-style-type: none"> • 01H - Carácter 1 • 02H - Carácter 2 • 04H - Carácter 3 • 08H - Carácter 4 • 10H - Carácter 5 Si se inhibe el bit asociado a un carácter, cualquier carácter será válido en esta posición de la secuencia.
STRSEQ1	IN	Char[5]	Caracteres de inicio de la secuencia 1 (5 caracteres)
STRSEQ2CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 2
STRSEQ2	IN	Char[5]	Caracteres de inicio de la secuencia 2 (5 caracteres)
STRSEQ3CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 3
STRSEQ3	IN	Char[5]	Caracteres de inicio de la secuencia 3 (5 caracteres)
STRSEQ4CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 4
STRSEQ4	IN	Char[5]	Caracteres de inicio de la secuencia 4 (5 caracteres)

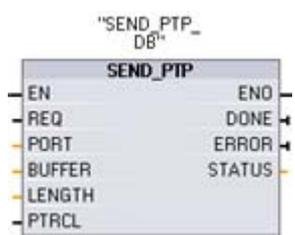
Estructura de los tipos de datos del parámetro CONDITIONS, 2ª parte (condiciones de fin)

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
ENDCOND	IN	UInt	Este parámetro define la condición de fin del mensaje: <ul style="list-style-type: none"> • 01H -Tiempo de respuesta • 02H -Tiempo del mensaje • 04H - Tiempo excedido entre caracteres • 08H - Longitud máxima • 10H - N + LEN + M • 20H - Secuencia
MAXLEN	IN	UInt	Longitud máxima del mensaje: Se utiliza sólo si está seleccionada la condición de fin "Longitud máxima". 0 a 1023 bytes
N	IN	UInt	Posición de byte dentro del campo de longitud en el mensaje. Se utiliza sólo con la condición de fin N + LEN + M. 1 a 1023 bytes
LENGTHSIZE	IN	UInt	Tamaño del campo de byte (1, 2 ó 4 bytes). Se utiliza sólo con la condición de fin N + LEN + M.
LENGTHM	IN	UInt	Indica el número de caracteres posteriores al campo de longitud que no se incluyen en el valor del campo de longitud. Se utiliza sólo con la condición de fin N + LEN + M. 0 a 255 bytes
RCVTIME	IN	UInt	Indica cuánto tiempo debe esperarse hasta que se reciba el primer carácter. La recepción se terminará con un error si no se recibe correctamente ningún carácter dentro del tiempo indicado. Se utiliza sólo con la condición de fin "Tiempo de respuesta". 0 a 65535 tiempos de bit, 8 segundos máx. Este parámetro no se evalúa realmente como condición de fin, puesto que sólo evalúa las condiciones de inicio. Es preciso seleccionar una condición de fin distinta.
MSGTIME	IN	UInt	Indica cuánto tiempo debe esperarse hasta la recepción completa del mensaje una vez recibido el primer carácter. Este parámetro se utiliza sólo si está seleccionada la condición de fin "Tiempo de mensaje excedido". 0 - 65535 milisegundos
CHARGAP	IN	UInt	Indica el número de tiempos de bit entre caracteres. Si el número de tiempos de bit entre caracteres excede el valor especificado, se cumplirá la condición de fin. Se utiliza sólo con la condición de fin "Tiempo excedido entre caracteres". 0 a 65535 milisegundos
ENDSEQ1CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 1: Estos son los bits de habilitación de cada carácter de la secuencia de fin. El carácter 1 es el bit 0, el carácter 2 es el bit 1, ..., el carácter 5 es el bit 4. Si se inhibe el bit asociado a un carácter, cualquier carácter será válido en esta posición de la secuencia.
ENDSEQ1	IN	Char[5]	Caracteres de inicio de la secuencia 1 (5 caracteres)

Códigos de condición

STATUS (W#16#...)	Descripción
80C0	Se ha seleccionado una condición de inicio no permitida
80C1	Se ha seleccionado una condición de fin no permitida o no se ha seleccionado ninguna condición de fin
80C2	Hay una alarma de recepción habilitada, pero esto no es posible
80C3	La condición de fin "Longitud máxima" está habilitada y la longitud máxima es 0 ó > 1024
80C4	La longitud calculada está habilitada y N es >= 1023
80C5	La longitud calculada está habilitada y la longitud no es 1, 2 ó 4
80C6	La longitud calculada está habilitada y el valor M es > 255
80C7	La longitud calculada está habilitada y es > 1024
80C8	El timeout de respuesta está habilitado y es igual a cero
80C9	El tiempo excedido entre caracteres está habilitado y es igual a cero ó > 2500
80CA	El timeout por "idle line" está habilitado y es igual a cero ó > 2500
80CB	La secuencia de fin está habilitada, pero todos los caracteres son irrelevantes
80CC	La secuencia de inicio (una cualquiera de 4) está habilitada, pero todos los caracteres son irrelevantes

8.7.5 Instrucción SEND_PTP



La instrucción SEND_PTP (Enviar datos punto a punto) inicia la transferencia de datos. SEND_PTP transfiere el búfer indicado al CM. El programa de la CPU se sigue ejecutando mientras el CM envía los datos a la velocidad de transferencia indicada. Sólo una operación de transmisión puede estar pendiente en un momento dado. El CM devuelve un error si una segunda instrucción SEND_PTP se ejecuta mientras el CM está transmitiendo un mensaje.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	Activa la transmisión solicitada cuando se produce un flanco ascendente en esta entrada de habilitación de la transmisión. Esto inicia la transferencia del contenido del búfer al módulo de comunicación punto a punto (CM).
PORT	IN	PORT	Identificador del puerto de comunicación: Esta dirección lógica es una constante que puede referenciarse en la ficha "Constantes" de la tabla de variables predeterminadas.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
BUFFER	IN	Variante	Este parámetro apunta a la dirección inicial del búfer de transmisión. No se soportan los datos booleanos ni las matrices booleanas.
LENGTH	IN	UInt	Longitud de la trama transmitida de bytes Al transmitir una estructura compleja, utilice siempre la longitud 0.
PTCL	IN	Bool	Este parámetro selecciona el búfer para la comunicación punto a punto normal o los protocolos suministrados por Siemens que están implementados en el CM acoplado. FALSE = operaciones punto a punto controladas por el programa de usuario. (única opción válida)
DONE	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución

El estado lógico de las salidas DONE y ERROR es FALSE durante la operación de transmisión. Una vez finalizada la operación de transmisión, la salida DONE o ERROR se pone a TRUE durante un ciclo para mostrar el estado de la transmisión. Mientras que DONE o ERROR sea TRUE, la salida STATUS será válida.

La instrucción devuelve el estado 16#7001 si el módulo de comunicación (CM) acepta los datos transmitidos. Las ejecuciones posteriores de SEND_PTP devuelven 16#7002 si el CM aún está ocupado transmitiendo datos. Una vez finalizada la operación de transmisión, el CM devuelve el estado de transmisión; 16#0000 si no han ocurrido errores. Las ejecuciones posteriores de SEND_PTP con REQ puesto a "low" devuelven el estado 16#7000 (no ocupado).

Relación de los valores de salida respecto a REQ:

Para esto se supone que la instrucción se llama periódicamente para comprobar el estado del proceso de transmisión. En el diagrama siguiente se supone que la instrucción se llama en cada ciclo (representado por los valores de STATUS).

REQ							
DONE							
ERROR							
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H

8.7 Instrucciones de comunicación punto a punto

El diagrama siguiente muestra cómo los parámetros DONE y STATUS son válidos sólo durante un ciclo si la línea REQ se impulsa (durante un ciclo) para iniciar la operación de transmisión.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H	7000H

El diagrama siguiente muestra la relación de los parámetros DONE, ERROR y STATUS cuando se presenta un error.

REQ								
DONE								
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	80D1H	7000H	7000H

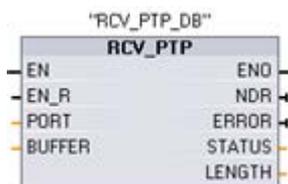
STATUS (W#16#...)	Descripción
80D0	Nueva petición estando activo el transmisor
80D1	Transmisión cancelada debido a que no se ha detectado ningún CTS durante el tiempo de espera
80D2	Transmisión cancelada debido a que no se ha recibido DSR del dispositivo DCE
80D3	Transmisión cancelada debido a desbordamiento de la cola de espera (transmisión de más de 1024 bytes)
7000	No ocupado
7001	Ocupado aceptando una petición (primera llamada)
7002	Ocupado en un sondeo (n-ésima llamada)

Interacción de los parámetros LENGTH y DATA para PTP_SEND

El tamaño mínimo de los datos que pueden transmitirse con la instrucción PTP_SEND es un byte. El parámetro DATA determina el tamaño de los datos por transmitir. Para el parámetro DATA no pueden utilizarse BOOL ni matrices de BOOL.

Parámetro LENGTH	Parámetro DATA	Descripción
LENGTH = 0	No utilizado	Los datos completos se envían según se define en el parámetro DATA. No es necesario especificar el número de bytes transmitidos si LENGTH = 0.
LENGTH > 0	Tipo de datos simple	El valor de LENGTH debe contener el conteo de bytes de este tipo de datos. De lo contrario, no se transferirá nada y se devolverá el error 8088H.
	Estructura	El valor de LENGTH puede contener un conteo de bytes menor que la longitud de bytes completa de la estructura. En este caso, se transferirán sólo los primeros bytes de LENGTH.
	Matriz	El valor de LENGTH puede contener un conteo de bytes menor que la longitud de bytes completa de la matriz. En este caso, se transferirán sólo los elementos de matriz que quepan por completo en los bytes de LENGTH. El valor de LENGTH debe ser un múltiplo del conteo de bytes del elemento de datos. De lo contrario, STATUS = 8088H, ERROR = 1 y la transmisión no se efectuará.
	String	Se transfiere la disposición de memoria completa del formato de cadena. El valor de LENGTH debe incluir bytes para la longitud máxima, la longitud real y los caracteres de cadena. Para el tipo de datos STRING, todas las longitudes y caracteres tienen un tamaño de un byte. Si se utiliza una cadena como parámetro real en el parámetro DATA, el valor de LENGTH también deberá incluir dos bytes para los dos campos de longitud.

8.7.6 Instrucción RCV_PTP



La instrucción RCV_PTP (Recepción punto a punto) comprueba si el CM ha recibido mensajes. Si hay un mensaje disponible, se transferirá del CM a la CPU. Un error devuelve el valor de STATUS apropiado.

El valor de STATUS es válido si NDR o ERROR es TRUE. El valor de STATUS indica el motivo por el que ha finalizado la recepción en el CM. Generalmente, es un valor positivo que indica que la recepción se ha realizado correctamente y que el proceso ha finalizado de forma normal. Si el valor de STATUS es negativo (es decir, si está activado el bit más significativo del valor hexadecimal), significa que la recepción se ha finalizado debido a un error p. ej. de paridad, trama o desbordamiento.

Todo módulo de comunicación punto a punto puede almacenar como máximo 1 KB en un búfer. Esto podría ser un mensaje grande o varios mensajes pequeños.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
EN_R	IN	Bool	Si esta entrada es TRUE, comprobar si el módulo CM ha recibido mensajes. Si un mensaje se ha recibido correctamente, se transferirá del módulo a la CPU. Si EN_R es FALSE, se comprobará si el CM ha recibido mensajes y se activará la salida STATUS, pero el mensaje no se transferirá a la CPU.
PORT	IN	PORT	Identificador del puerto de comunicación: Esta dirección lógica es una constante que puede referenciarse en la ficha "Constantes" de la tabla de variables predeterminadas.
BUFFER	IN	Variante	Este parámetro apunta a la dirección inicial del búfer de recepción. Este búfer debería ser lo suficientemente grande para recibir el mensaje de longitud máxima. No se soportan los datos booleanos ni las matrices booleanas.
NDR	OUT	Bool	TRUE durante un ciclo cuando se dispone de datos nuevos y la operación ha finalizado sin errores.
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la operación con un error
STATUS	OUT	Word	Código de condición de ejecución
LENGTH	OUT	UInt	Longitud del mensaje devuelto (en bytes)

STATUS (W#16#...)	Descripción
0000	No hay ningún búfer disponible
80E0	Mensaje terminado debido a que el búfer de recepción está lleno
80E1	Mensaje terminado debido a un error de paridad
80E2	Mensaje terminado debido a un error de trama
80E3	Mensaje terminado debido a un error de desbordamiento
80E4	Mensaje terminado porque la longitud calculada excede el tamaño del búfer
0094	Mensaje terminado porque se ha recibido la longitud máxima de caracteres
0095	Mensaje terminado debido a timeout del mensaje
0096	Mensaje terminado porque se ha excedido el tiempo entre caracteres
0097	Mensaje terminado debido a timeout de respuesta
0098	Mensaje terminado porque se ha cumplido la condición de longitud "N+LEN+M"
0099	Mensaje terminado porque se ha cumplido la condición de secuencia de fin

8.7.7 Instrucción RCV_RST



La instrucción RCV_RST (Inicializar receptor) borra el búfer de recepción en el CM.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	Activa la inicialización del receptor cuando se produce un flanco ascendente en esta entrada de habilitación
PORT	IN	PORT	Identificador del puerto de comunicación: Este puerto debe determinarse utilizando la dirección lógica del módulo.
DONE	OUT	Bool	Si es TRUE durante un ciclo, indica que la última petición se ha finalizado sin errores.
ERROR	OUT	Bool	Si es TRUE, indica que la última petición se ha finalizado con errores. Además, si esta salida es TRUE, la salida STATUS contendrá los códigos de error respectivos.
STATUS	OUT	Word	Código de error

8.7.8 Instrucción SGN_GET

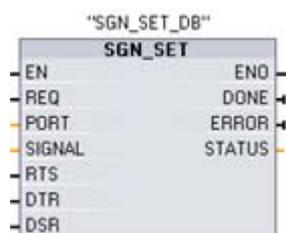


La instrucción SGN_GET (Consultar señales RS232) lee los estados actuales de las señales de comunicación RS232. Esta función es válida sólo para el CM (módulo de comunicación) RS232.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	Los valores de las señales RS232 se consultan cuando se produce un flanco ascendente en esta entrada
PORT	IN	PORT	Identificador del puerto de comunicación: Esta dirección lógica es una constante que puede referenciarse en la ficha "Constantes" de la tabla de variables predeterminadas.
NDR	OUT	Bool	TRUE durante un ciclo cuando se dispone de datos nuevos y la operación ha finalizado sin errores
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la operación con un error
STATUS	OUT	Word	Código de condición de ejecución
DTR	OUT	Bool	Terminal de datos disponible, módulo listo (salida)
DSR	OUT	Bool	Equipo de datos listo, interlocutor listo (entrada)
RTS	OUT	Bool	Petición de transmitir, módulo listo para transmitir (salida)
CTS	OUT	Bool	Listo para transmitir; el interlocutor puede recibir datos (entrada)
DCD	OUT	Bool	Detección de portadora de datos, nivel de señal de recepción (siempre FALSE, no se soporta)
RING	OUT	Bool	Indicación de timbre, señalización de una llamada entrante (siempre FALSE, no se soporta)

STATUS (W#16#...)	Descripción
80F0	El CM es un módulo RS485; no hay señales disponibles
80F1	Las señales no pueden activarse debido al control de flujo por hardware
80F2	Imposible activar DSR porque el módulo es DTE
80F3	Imposible activar DTR porque el módulo es DCE

8.7.9 Instrucción SGN_SET



La instrucción SGN_SET (Activar señales RS232) activa los estados de las señales de comunicación RS232. Esta función es válida sólo para el CM (módulo de comunicación) RS232.

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
REQ	IN	Bool	Inicia la activación de las señales RS232 cuando se produce un flanco ascendente en esta entrada
PORT	IN	PORT	Identificador del puerto de comunicación: Esta dirección lógica es una constante que puede referenciarse en la ficha "Constantes" de la tabla de variables predeterminadas.
SIGNAL	IN	Byte	Selecciona las señales que deben activarse: (se permiten varias) <ul style="list-style-type: none"> • 01H = Activar RTS • 02H = Activar DTR • 04H = Activar DSR
RTS	IN	Bool	Petición de transmitir, módulo listo para transmitir el valor que debe ajustarse (TRUE o FALSE)
DTR	IN	Bool	Terminal de datos disponible, módulo listo para transmitir el valor que debe ajustarse (TRUE o FALSE)
DSR	IN	Bool	Equipo de datos listo (aplicable sólo a las interfaces DCE) (no se utiliza)
DONE	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución

STATUS (W#16#...)	Descripción
80F0	El CM es un módulo RS485; no hay señales que puedan activarse
80F1	Las señales no pueden activarse debido al control de flujo por hardware
80F2	Imposible activar DSR porque el módulo es DTE
80F3	Imposible activar DTR porque el módulo es DCE

8.8 Errores

Valores de retorno de las instrucciones PtP

Toda instrucción PtP tiene tres salidas que indican el estado de finalización:

Parámetro	Tipo de datos	Valor predeterminado	Descripción
DONE	Booleano	FALSE	TRUE durante un ciclo indica que la última petición se ha finalizado sin errores.
ERROR	Booleano	FALSE	TRUE indica que la última petición se ha finalizado con errores. El código de error aplicable aparece en STATUS.
STATUS	Word	0	Dos bytes que contienen la clase y el número de error (si es aplicable). STATUS conserva su valor durante la ejecución de la función.

Errores y clases de errores comunes

Descripción de la clase	Clases de error	Descripción
Configuración del puerto	80Ax	Permite definir errores comunes de configuración del puerto
Configuración de la transmisión	80Bx	Permite definir errores comunes de configuración de la transmisión
Configuración de la recepción	80Cx	Permite definir errores comunes de configuración de la recepción
Tiempo de ejecución de la transmisión	80Dx	Permite definir errores comunes en tiempo de ejecución de la transmisión
Tiempo de ejecución de la recepción	80Ex	Permite definir errores comunes en tiempo de ejecución de la recepción
Procesamiento de señales	80Fx	Permite definir errores comunes en relación con el procesamiento de señales

Errores de configuración del puerto

Evento/ID de error	Descripción
0x80A0	El protocolo indicado no existe
0x80A1	La velocidad de transferencia indicada no existe
0x80A2	La paridad indicada no existe
0x80A3	El número de bits de datos indicado no existe
0x80A4	El número de bits de parada indicado no existe
0x80A5	El tipo de control de flujo indicado no existe

Errores de configuración de la transmisión

Evento/ID de error	Descripción
0x80B0	El protocolo indicado no existe
0x80B1	La velocidad de transferencia indicada no existe
0x80B2	La paridad indicada no existe
0x80B3	El número de bits de datos indicado no existe
0x80B4	El número de bits de parada indicado no existe
0x80B5	El tipo de control de flujo indicado no existe

Errores de configuración de la recepción

Evento/ID de error	Descripción
0x80C0	Error de condición de inicio
0x80C1	Error de condición de fin
0x80C3	Error de longitud máxima
0x80C4	Error del valor N (consulte N+LEN+M)
0x80C5	Error de tamaño de longitud (consulte MAXLEN o N+LEN+M)
0x80C6	Error del valor M (consulte N+LEN+M)
0x80C7	Error del valor N-longitud-M (consulte N+LEN+M)
0x80C8	Error de timeout de respuesta; no se ha recibido ningún mensaje durante el periodo de recepción indicado. (Consulte RCVTIME o MSGTIME)
0x80C9	Error de tiempo excedido entre caracteres (consulte CHARGAP)
0x80CA	Error de timeout por "idle line" (consulte IDLELINE)
0x80CB	Hay una secuencia de fin configurada, pero todos los caracteres son irrelevantes
0x80CC	Hay una secuencia de inicio configurada, pero todos los caracteres son irrelevantes

Errores de procesamiento de señales

Evento/ID de error	Descripción
0x80F0	El CM es un módulo RS485 y no hay señales disponibles
0x80F1	El CM es un módulo RS232, pero no es posible activar señales porque está habilitado el control de flujo por hardware
0x80F2	La señal DSR no se puede activar porque el módulo es un dispositivo DTE

Errores en tiempo de ejecución de la transmisión

Evento/ID de error	Descripción
Límite del búfer	Se ha excedido el búfer de transmisión disponible en total del CP
0x80D0	Se ha recibido una petición nueva estando activo el transmisor
0x80D1	El receptor ha lanzado una petición de control de flujo para suspender una transmisión activa y no ha habilitado nuevamente la transmisión en el tiempo de espera indicado. Este error también se genera durante el control de flujo por hardware cuando el receptor no confirma CTS en el tiempo de espera indicado.
0x80D2	La petición de transmisión se ha cancelado porque no se recibe ninguna señal DSR del DCE
0x80D3	Se ha excedido el búfer de transmisión disponible en total del CP
0x7000	La función de transmisión no está ocupada
0x7001	La función de transmisión está ocupada con la primera llamada
0x7002	La función de transmisión está ocupada con las llamadas posteriores (sondeos tras la primera llamada)

Valores de retorno en el tiempo de ejecución de la recepción

Evento/ID de error	Descripción
0x80E0	El mensaje se ha terminado porque el búfer de recepción está lleno
0x80E1	El mensaje se ha terminado debido a un error de paridad
0x80E2	El mensaje se ha terminado debido a un error de trama
0x80E3	El mensaje se ha terminado debido a un error de desbordamiento
0x80E4	El mensaje se ha terminado debido a que la longitud especificada excede el tamaño del búfer total
0x0094	El mensaje se ha terminado porque se ha recibido la longitud de caracteres máxima (MAXLEN)
0x0095	El mensaje se ha terminado porque el mensaje completo no se ha recibido en el tiempo indicado (MSGTIME)
0x0096	El mensaje se ha terminado porque no se ha recibido el siguiente carácter dentro del tiempo entre caracteres (CHARGAP)
0x0097	El mensaje se ha terminado porque el primer carácter no se ha recibido en el tiempo indicado (RCVTIME)
0x0098	El mensaje se ha terminado porque se ha cumplido la condición de longitud "n+len+m" (N+LEN+M)
0x0099	El mensaje se ha terminado porque se ha cumplido la secuencia de fin (ENDSEQ)

Errores diversos de parámetros

Evento/ID de error	Descripción
0x8n3A	Puntero no permitido en el parámetro n
0x8070	Se está utilizando toda la memoria de instancia interna
0x8080	Número de puerto no válido
0x8082	Ha fallado la parametrización porque ya se está parametrizando en segundo plano
0x8083	Desbordamiento del búfer. El CM ha devuelto más datos de lo permitido.
0x8085	El parámetro LEN está puesto a 0 o excede el valor máximo permitido
0x8088	El parámetro LEN excede el área de memoria especificada en DATA

Herramientas online y diagnóstico

9.1 LEDs de estado

La CPU y los módulos de E/S utilizan LEDs para indicar el estado operativo del módulo o de las E/S. La CPU incorpora los siguientes indicadores de estado:

- STOP/RUN
 - Luz naranja permanente indica el estado operativo STOP
 - Luz verde permanente indica el estado operativo RUN
 - Si parpadea (alternando entre verde y naranja), indica que la CPU está arrancando
- ERROR
 - Luz roja intermitente indica un error, p. ej. un error interno de la CPU, de la Memory Card o un error de configuración (los módulos no se corresponden)
 - Luz roja permanente indica que hay un fallo de hardware
- El LED MAINT (mantenimiento) parpadea cuando se inserta una Memory Card. La CPU pasa entonces a estado operativo STOP. Tras cambiar la CPU a estado operativo STOP, realice una de las funciones siguientes para iniciar la evaluación de la Memory Card:
 - Cambiar la CPU a estado operativo RUN
 - Realizar un borrado total (MRES)
 - Desconectar y volver a conectar la alimentación de la CPU

Descripción	STOP/RUN Naranja/verde	ERROR Rojo	MAINT Naranja
Alimentación desconectada	Off	Off	Off
Arranque, autotest, actualización de firmware	Parpadeo (alternando entre naranja y verde)	-	Off
Estado operativo STOP	On (naranja)	-	-
Estado operativo RUN	On (naranja)	-	-
Extracción de la Memory Card	On (naranja)	-	Parpadeo
Error	On (naranja o verde)	Parpadeo	-
Mantenimiento solicitado	On (naranja o verde)	-	On

9.1 LEDs de estado

Descripción	STOP/RUN Naranja/verde	ERROR Rojo	MAINT Naranja
Hardware averiado	On (naranja)	On	Off
Test de LEDs o firmware de la CPU defectuoso	Parpadeo (alternando entre naranja y verde)	Parpadeo	Parpadeo

La CPU incorpora asimismo dos LEDs que indican el estado de la comunicación PROFINET. Abra la tapa del bloque de terminales inferior para ver los LEDs PROFINET.

- Link (verde) se enciende para indicar una conexión correcta
- Rx/Tx (amarillo) se enciende para indicar la actividad de transmisión

La CPU y todos los módulos de señales (SM) digitales incorporan un LED I/O Channel para cada una de las entradas y salidas digitales. El LED I/O Channel (verde) se enciende o apaga para indicar el estado de la entrada o salida en cuestión.

Además, todo SM digital incorpora un LED DIAG que indica el estado del módulo:

- Verde indica que el módulo está operativo
- Rojo indica que el módulo está averiado o no operativo

Todo SM analógico incorpora un LED I/O Channel para cada una de las entradas y salidas analógicas.

- Verde indica que el canal se ha configurado y está activo
- Rojo indica una condición de error de la entrada o salida analógica en cuestión

Además, todo SM analógico incorpora un LED DIAG que indica el estado del módulo:

- Verde indica que el módulo está operativo
- Rojo indica que el módulo está averiado o no operativo

El SM detecta la presencia o ausencia de alimentación del módulo (alimentación de campo en caso necesario).

Descripción	DIAG (Rojo/verde)	I/O Channel (Rojo/verde)
Alimentación de campo desconectada	Rojo intermitente	Rojo intermitente
No se ha configurado o se está actualizando	Verde intermitente	Off
Módulo configurado sin errores	On (verde)	On (verde)
Condición de error	Rojo intermitente	-
Error de E/S (con diagnóstico habilitado)	-	Rojo intermitente
Error de E/S (con diagnóstico inhibido)	-	On (verde)

9.2 Establecer una conexión online con una CPU

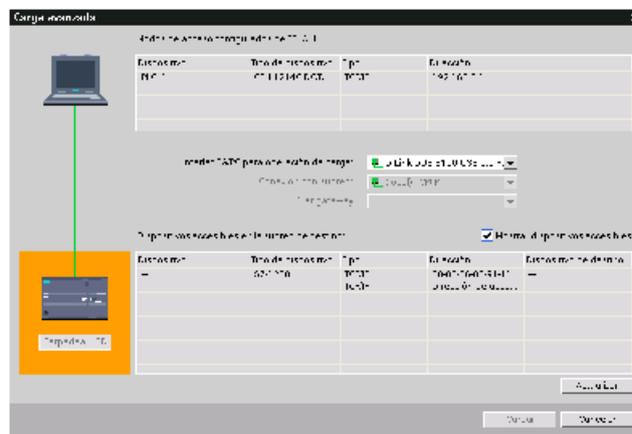
Una conexión online entre la programadora y un sistema de destino se requiere para cargar programas y datos de ingeniería del proyecto en el sistema de destino, así como para las actividades siguientes:

- Comprobar programas de usuario
- Visualizar y cambiar el estado operativo de la CPU
- Visualizar y ajustar la fecha y hora de la CPU
- Visualizar la información del módulo
- Comparar bloques online y offline
- Diagnosticar el hardware

Online y
diagnóstico



La Task Card "Herramientas online" permite acceder a los datos del sistema de destino en la vista online o de diagnóstico.



El estado online actual de un dispositivo se indica mediante un símbolo situado a la derecha del dispositivo en el árbol del proyecto.

El color naranja indica una conexión online.

Seleccione "Dispositivos accesibles" para buscar una CPU en la red.



Haga clic en "Conectar online" para establecer una conexión con una CPU en la red.

9.3 Ajustar la dirección IP y la hora

Es posible ajustar la dirección IP y la hora en la CPU online.

Tras establecer una conexión con una CPU online desde el área "Online y diagnóstico", es posible visualizar o cambiar la dirección IP.

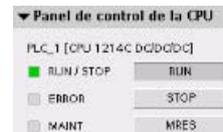
Encontrará más información en el apartado Dirección IP (Página 84).

También es posible visualizar o ajustar los parámetros de fecha y hora en la CPU online.



9.4 Panel de control de la CPU online

La Task Card "Panel de control de la CPU" muestra el estado operativo (STOP o RUN) de la CPU online: También indica si la CPU tiene un error o si se están forzando valores. El panel de control de la CPU permite cambiar el estado operativo de una CPU online.

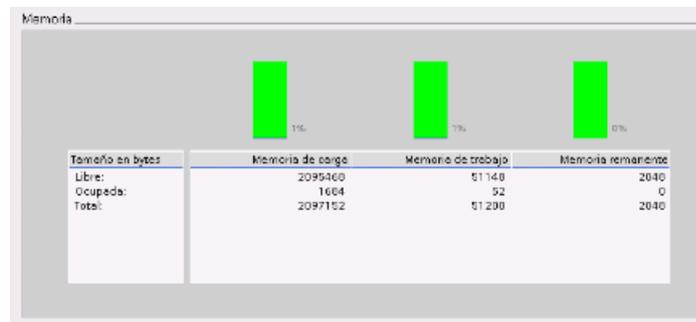
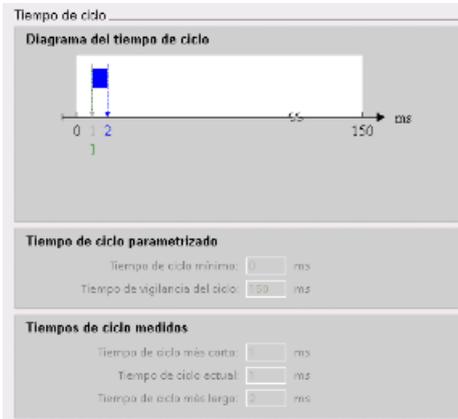


9.5 Vigilar el tiempo de ciclo y la carga de la memoria

Es posible vigilar el tiempo de ciclo y la carga de la memoria de una CPU online.

Tras establecer una conexión con la CPU online es posible visualizar los siguientes valores medidos:

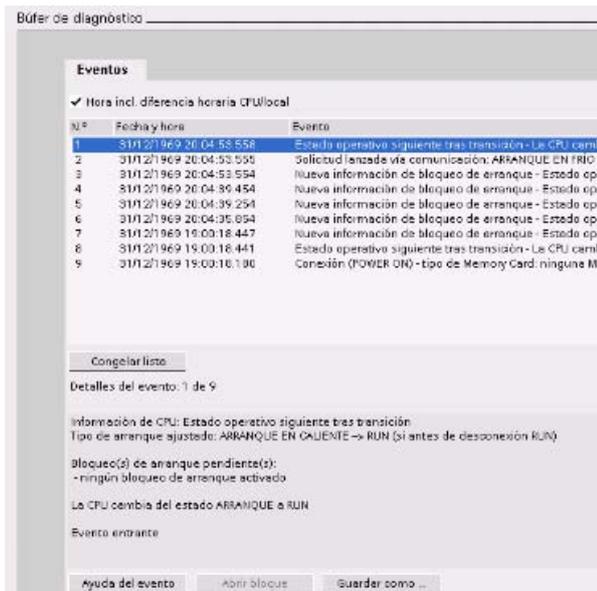
- Tiempo de ciclo
- Carga de la memoria



9.6 Visualizar los eventos de diagnóstico de la CPU

El búfer de diagnóstico permite consultar las actividades recientes de la CPU. El búfer de diagnóstico contiene las entradas siguientes:

- Eventos de diagnóstico
- Cambios del estado operativo de la CPU (transiciones a STOP o RUN)



La primera entrada contiene el evento más reciente. Toda entrada del búfer de diagnóstico incluye la fecha y hora de registro del evento, así como una descripción.

El número máximo de entradas depende de la CPU. Se soportan 50 entradas como máximo.

Sólo los 10 eventos más recientes del búfer de diagnóstico se almacenan de forma permanente. Si se restablece la configuración de fábrica de la CPU, se reinicializa el búfer de diagnóstico y se borran sus entradas.

9.7 Tablas de observación del programa de usuario

Una tabla de observación permite observar y forzar datos a medida que la CPU ejecuta el programa. Estos datos pueden ser de la memoria imagen de proceso (I o Q), datos físicos (I_:P o Q_:P), o bien datos M o DB, dependiendo de la función de observación o forzado.

La función de observación no modifica la secuencia del programa. Facilita información sobre la secuencia y los datos del programa en la CPU.

Las funciones de control permiten al usuario controlar la secuencia y los datos del programa. Hay que prestar atención al utilizar las funciones de forzado. Estas funciones pueden influir seriamente en la ejecución del programa de usuario/de sistema. Las tres funciones de control son: forzar, forzar permanentemente y desbloquear salidas en STOP.

La tabla de observación permite realizar las siguientes funciones online:

- Observar el estado de las variables
- Forzar los valores de las distintas variables
- Forzar permanentemente una variable a un valor determinado

Es posible seleccionar cuándo se debe observar o forzar la variable:

- Inicio del ciclo: Lee o escribe el valor al inicio del ciclo
- Fin del ciclo: Lee o escribe el valor al final del ciclo
- Cambiar a STOP

Para crear una tabla de observación, proceda del siguiente modo:

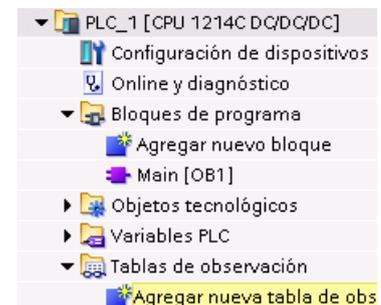
1. Haga doble clic en "Agregar nueva tabla de observación" para abrir una tabla de observación nueva.
2. Introduzca el nombre de la variable o agregue una variable a la tabla de observación.

Las siguientes opciones están disponibles para observar variables:

- Observar todos: Este comando inicia la observación de las variables visibles en la tabla de observación activa.
- Observar inmediatamente: Este comando inicia la observación de las variables visibles en la tabla de observación activa. La tabla de observación observa las variables inmediatamente y una sola vez.

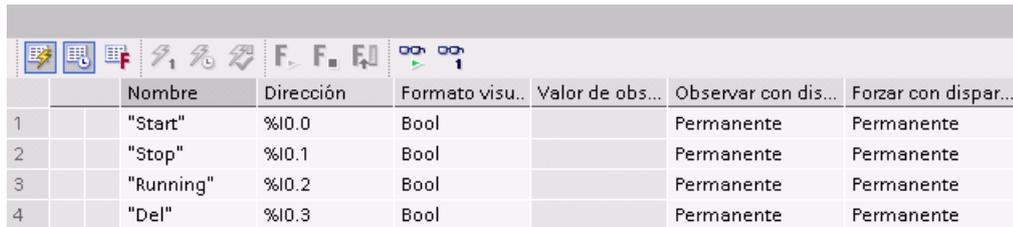
Las siguientes opciones están disponibles para forzar variables:

- "Forzar a 0" pone a "0" el valor de una dirección seleccionada.
- "Forzar a 1" pone a "1" el valor de una dirección seleccionada.
- "Forzar inmediatamente" cambia inmediatamente los valores de las direcciones seleccionadas durante un ciclo.



- "Forzar con disparador" cambia los valores de las direcciones seleccionadas.
Esta función no confirma si las direcciones seleccionadas se han forzado realmente. Si se requiere una confirmación del cambio, utilice la función "Forzar inmediatamente".
- "Habilitar salidas de periferia" desactiva el comando para inhibir las salidas y está disponible sólo cuando la CPU se encuentra en estado operativo STOP.

Para forzar las variables debe existir una conexión online con la CPU.



	Nombre	Dirección	Formato visu..	Valor de obs...	Observar con dis...	Forzar con dispar...
1	"Start"	%I0.0	Bool		Permanente	Permanente
2	"Stop"	%I0.1	Bool		Permanente	Permanente
3	"Running"	%I0.2	Bool		Permanente	Permanente
4	"Del"	%I0.3	Bool		Permanente	Permanente

Las distintas funciones pueden seleccionarse mediante los botones en el lado superior de la tabla de observación.

Introduzca el nombre de la variable que desea observar y seleccione un formato de visualización en la lista desplegable. Si existe una conexión online con la CPU y se hace clic en el botón "Observar", se visualizará el valor real del punto de datos en el campo "Valor de observación".

Utilizar un disparador para observar o forzar variables PLC

Si se utiliza un disparador es posible determinar en qué punto del ciclo debe observarse o forzarse la dirección seleccionada.

Tipo de disparador	Descripción
Permanente	Recoge los datos continuamente
Al inicio del ciclo	Permanente: Recoge los datos continuamente al inicio del ciclo, después de que la CPU lee las entradas
	Único: Recoge los datos al inicio del ciclo, después de que la CPU lee las entradas
Al final del ciclo	Permanente: Recoge los datos continuamente al final del ciclo, antes de que la CPU escribe en las salidas
	Único: Recoge los datos una vez al final del ciclo, antes de que la CPU escribe en las salidas
Transición a STOP	Permanente: Recoge los datos continuamente cuando la CPU pasa a STOP
	Único: Recoge los datos una vez cuando la CPU pasa a STOP

Para modificar una variable PLC en un disparador determinado, seleccione el inicio o final del ciclo.

- Forzar una salida: El mejor evento de disparo para forzar una salida es al final del ciclo, inmediatamente antes de que la CPU escribe en las salidas.

Observe el valor de las salidas al inicio del ciclo para determinar qué valor se escribe en las salidas físicas. Asimismo, observe las salidas antes de que la CPU escriba los valores en las salidas físicas para comprobar la lógica del programa y compararla con la reacción real de las E/S.

- Forzar una entrada: El mejor evento de disparo para forzar una entrada es al inicio del ciclo, inmediatamente después de que la CPU lee las entradas y antes de que el programa utiliza los valores de entrada.

Si está forzando entradas al inicio del ciclo, deberá observar el valor de las entradas al final del ciclo para asegurarse de que no ha cambiado desde el principio.. Si los valores son diferentes, es posible que el programa de usuario esté escribiendo en una entrada por error.

Para diagnosticar la causa de una transición a STOP de la CPU, utilice el disparador "Transición a STOP" para capturar los últimos valores de proceso.

Desbloquear las salidas en modo STOP

La tabla de observación permite escribir en las salidas cuando la CPU está en modo STOP. Esta función sirve para comprobar el cableado de las salidas y verificar que el cable conectado a un borne de salida lanza una señal "high" o "low" al terminal del dispositivo de proceso al que está conectado.

 ADVERTENCIA
--

Aunque la CPU está en estado operativo STOP, la habilitación de una salida física puede activar el punto del proceso a la que está conectada.

El estado de las salidas se puede cambiar en estado operativo STOP si están habilitadas las salidas. Si las salidas están inhibidas, no es posible modificarlas en estado operativo STOP.

- Para habilitar la modificación de las salidas en STOP, seleccione la opción "Habilitar salidas de periferia" del comando "Forzar" del menú "Online", o bien haga clic con el botón derecho del ratón en la fila correspondiente de la tabla de observación.
- Cuando la CPU pasa a estado operativo RUN se inhibe la opción "Habilitar salidas de periferia".
- Si alguna entrada o salida se fuerza permanentemente, la CPU no podrá desbloquear las salidas en estado operativo STOP. Primero es preciso cancelar la función de forzado permanentemente.

Forzar valores permanentemente en la CPU

La CPU permite forzar permanentemente las entradas y salidas. Para ello es preciso indicar la dirección de la entrada o salida física (I_:P o Q_:P) en la tabla de observación e iniciar el forzado permanente.

El valor forzado permanentemente sobrescribe las lecturas de las entradas físicas en el programa. El programa utiliza el valor forzado permanentemente para el procesamiento. Cuando el programa escribe en una salida física, el valor de forzado permanente sobrescribe el de la salida. El valor forzado permanentemente aparece en la salida física y es utilizado por el proceso.

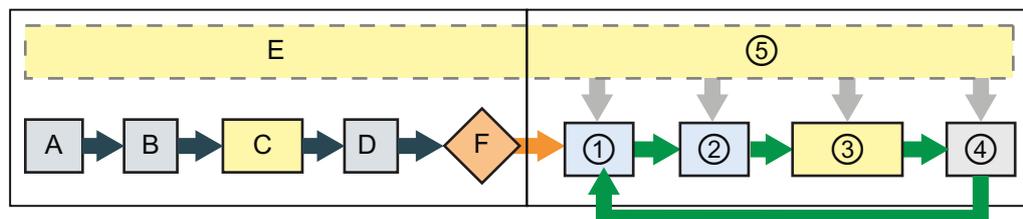
Cuando una entrada o salida se fuerza permanentemente en la tabla de observación, las acciones de forzado permanente se convierten en parte del programa de usuario. Aunque se haya cerrado el software de programación, las selecciones de forzado permanente permanecen activadas en el programa de la CPU hasta que son borradas al establecer una conexión online desde el software de programación y se para la función de forzado permanente. Los programas con entradas y/o salidas forzadas de forma permanente que se hayan cargado en una CPU diferente desde una Memory Card seguirán forzando permanentemente las E/S seleccionadas en el programa.

Si la CPU ejecuta el programa de usuario desde una Memory Card protegida contra escritura, el forzado permanente de una E/S no se puede iniciar ni cambiar desde una tabla de observación, ya que no es posible sobrescribir los valores en el programa de usuario protegido contra escritura. Todo intento de forzar permanentemente los valores protegidos contra escritura generará un error. Si se utiliza una Memory Card para transferir un programa de usuario, los elementos forzados permanentemente en esa Memory Card se transferirán a la CPU.

Nota

Las E/S digitales asignadas al HSC, PWM y PTO no se pueden forzar permanentemente

Las E/S digitales utilizadas por el contador rápido (HSC) y los dispositivos con modulación del ancho de pulso (PWM) y tren de impulsos (PTO) se asignan durante la configuración de dispositivos. Si se asignan direcciones de E/S digitales a dichos dispositivos, los valores de las direcciones de E/S asignadas no podrán modificarse utilizando la función de forzado permanente de la tabla de observación.



Arranque

- A La función de forzado permanente no afecta el borrado del área de memoria de las entradas (I).
- B La función de forzado permanente no afecta la inicialización de los valores de salida.
- C Durante la ejecución de los OBs de arranque, la CPU aplica el valor de forzado permanente cuando el programa de usuario accede a la entrada física.
- D El almacenamiento de los eventos de alarma en la cola de espera no se ve afectado.
- E La habilitación de escritura en las salidas no se ve afectada.

RUN

- ① Mientras escribe la memoria de las salidas (Q) en las salidas físicas, la CPU aplica el valor de forzado permanente a medida que se van actualizando las salidas.
- ② Al leer las entradas físicas, la CPU aplica los valores de forzado permanente inmediatamente antes de copiar las entradas en la memoria I.
- ③ Durante la ejecución del programa de usuario (OBs de ciclo), la CPU aplica el valor de forzado permanente cuando el programa de usuario accede a la entrada física o escribe en la salida física.
- ④ La función de forzado permanente no afecta el procesamiento de peticiones de comunicación ni los diagnósticos de autotest.
- ⑤ El procesamiento de las alarmas en cualquier parte del ciclo no se ve afectado.

Datos técnicos

A.1 Datos técnicos generales

Homologaciones

El sistema de automatización S7-1200 cumple las siguientes normas y especificaciones de test. Los criterios de test del sistema de automatización S7-1200 se basan en estas normas y especificaciones de test.

Homologación CE



El sistema de automatización S7-1200 satisface los requisitos y objetivos relacionados con la seguridad según las directivas CE indicadas a continuación y cumple las normas europeas (EN) armonizadas para controladores programables publicadas en los Diarios Oficiales de la Unión Europea.

- Directiva CE 2006/95/CE (Directiva de baja tensión) "Material eléctrico destinado a utilizarse con determinados límites de tensión"
 - EN 61131-2:2007 Autómatas programables - Requisitos y ensayos de los equipos
- Directiva CE 2004/108/CE (Directiva CEM) "Compatibilidad electromagnética"
 - Norma de emisión
EN 61000-6-4:2007: Entornos industriales
 - Norma de inmunidad
EN 61000-6-2:2005: Entornos industriales
- Directiva CE 94/9/CE (ATEX) "Equipos y sistemas de protección para uso en atmósferas potencialmente explosivas"
 - EN 60079-15:2005: Tipo de protección 'n'

La Declaración de conformidad CE se encuentra a disposición de las autoridades competentes en:

Siemens AG
IA AS RD ST PLC Amberg
Werner-von-Siemens-Str. 50
D92224 Amberg
Germany

Homologación cULus



Underwriters Laboratories Inc. cumple

- Underwriters Laboratories, Inc.: UL 508 Listed (Industrial Control Equipment)
- Canadian Standards Association: CSA C22.2 Number 142 (Process Control Equipment)

ATENCIÓN

La gama SIMATIC S7-1200 cumple la norma CSA.

El logotipo cULus indica que Underwriters Laboratories (UL) ha examinado y certificado el S7-1200 según las normas UL 508 y CSA 22.2 No. 142.

Homologación FM



Factory Mutual Research (FM):

Números de clase 3600 y 3611 de la norma de aprobación

Aprobado para ser utilizado en:

Class I, Division 2, Gas Group A, B, C, D, Temperature Class T4A Ta = 40° C

Class I, Zone 2, IIC, Temperature Class T4 Ta = 40° C

Homologación ATEX



EN 60079-0:2006: Atmósferas explosivas - Requisitos generales

EN 60079-15:2005: Material eléctrico para atmósferas de gas explosivas;

Tipo de protección 'n'

II 3 G Ex nA II T4

Las siguientes condiciones especiales deben cumplirse para el uso seguro del S7-1200:

- Los módulos deben montarse en una carcasa apropiada con un grado de protección mínimo de IP54 según EN 60529, considerando las condiciones ambientales en las que se utilizarán los equipos.
- Si, en condiciones nominales, la temperatura excede 70° C en el punto de entrada del cable, o bien 80° C en el punto de derivación de los conductores, la temperatura realmente medida deberá estar comprendida en el rango de temperatura admisible del cable seleccionado.
- Se deberán tomar las medidas necesarias para impedir que se exceda la tensión nominal en más de un 40% a causa de perturbaciones transitorias.

Aprobación C-Tick



El S7-1200 cumple los requisitos de las normas según AS/NZS 2064 (clase A).

Aprobación marina

Los productos S7-1200 se someten con regularidad a pruebas para obtener homologaciones especiales para aplicaciones y mercados específicos. Contacte con el representante de Siemens más próximo para obtener una lista de las homologaciones actuales y los respectivos números de referencia.

Sociedades de clasificación:

- ABS (American Bureau of Shipping)
- BV (Bureau Veritas)
- DNV (Det Norske Veritas)
- GL (Germanischer Lloyd)
- LRS (Lloyds Register of Shipping)
- Class NK (Nippon Kaiji Kyokai)

Entornos industriales

El sistema de automatización S7-1200 está diseñado para ser utilizado en entornos industriales.

Campo de aplicación	Requisitos respecto a la emisión de interferencias	Requisitos respecto a la inmunidad a interferencias
Industrial	EN 61000-6-4:2007	EN 61000-6-2:2005

Compatibilidad electromagnética

La compatibilidad electromagnética (también conocida por sus siglas CEM o EMC) es la capacidad de un dispositivo eléctrico para funcionar de forma satisfactoria en un entorno electromagnético sin causar interferencias electromagnéticas (EMI) sobre otros dispositivos eléctricos de ese entorno.

Compatibilidad electromagnética - Inmunidad según EN 61000-6-2	
EN 61000-4-2 Descargas electrostáticas	Descarga en el aire de 8 kV en todas las superficies Descarga en contactos de 6 kV en las superficies conductoras expuestas
EN 61000-4-3 Campos electromagnéticos radiados	80 a 1000 MHz, 10 V/m, 80% AM a 1 kHz 1-4 a 2,0 GHz, 3 V/m, 80% AM a 1 kHz 2,0 a 2,7 GHz, 1 V/m, 80% AM a 1 kHz
EN 61000-4-4 Transitorios eléctricos rápidos	2 kV, 5 kHz con red de conexión a la alimentación AC y DC 2 kV, 5 kHz con borne de conexión a las E/S
EN 61000-4-5 Inmunidad a ondas de choque	Sistemas AC - 2 kV en modo común, 1kV en modo diferencial Sistemas DC - 2 kV en modo común, 1kV en modo diferencial Para los sistemas DC (señales E/S, sistemas de alimentación DC) se requiere protección externa.
EN 61000-4-6 Perturbaciones conducidas	150 kHz a 80 MHz, 10 V RMS, 80% AM a 1kHz
EN 61000-4-11 Inmunidad a cortes e interrupciones breves	Sistemas AC 0% durante 1 ciclo, 40% durante 12 ciclos y 70% durante 30 ciclos a 60 Hz

Compatibilidad electromagnética - Emisiones conducidas y radiadas según EN 61000-6-4	
Emisiones conducidas EN 55011, clase A, grupo 1 0,15 MHz a 0,5 MHz 0,5 MHz a 5 MHz 5 MHz a 30 MHz	<79dB (µV) casi cresta; <66 dB (µV) valor medio <73dB (µV) casi cresta; <60 dB (µV) valor medio <73dB (µV) casi cresta; <60 dB (µV) valor medio
Emisiones radiadas EN 55011, clase A, grupo 1 30 MHz a 230 MHz 230 MHz a 1 GHz	<40dB (µV/m) casi cresta; medida a 10m <47dB (µV/m) casi cresta; medida a 10m

Condiciones ambientales

Condiciones ambientales - Transporte y almacenamiento	
EN 60068-2-2, ensayo Bb, calor seco y EN 60068-2-1, ensayo Ab, frío	-40° C a +70° C
EN 60068230, ensayo Dd, calor húmedo	25° C a 55° C, 95% de humedad
EN 60068-2-14, ensayo Na, choque de temperatura	-40° C a +70° C, tiempo de secado 3 horas, 2 ciclos
EN 60068232, caída libre	0,3 m, 5 veces, embalado para embarque
Presión atmosférica	1080 a 660h Pa (equivale a una altitud de -1000 a 3500m)

Condiciones ambientales - Funcionamiento	
Rango de temperatura ambiente (aire de entrada 25 mm bajo la unidad)	0° C a 55° C en montaje horizontal 0° C a 45° C en montaje vertical 95% de humedad no condensante
Presión atmosférica	1080 to 795 hPa (equivale a una altitud de -1000 a 2000m)
Concentración de contaminantes	SO ₂ : < 0,5 ppm; H ₂ S: < 0,1 ppm; RH < 60% no condensante
EN 60068214, ensayo Nb, cambio de temperatura	5° C a 55° C, 3° C/minuto
EN 60068227, choque mecánico	15 G, 11 ms impulso, 6 choques en c/u de 3 ejes
EN 6006826, vibración sinusoidal	Montaje en perfil DIN: 3,5 mm de 5 a 9 Hz, 1G de 9 a 150 Hz Montaje en panel: 7,0 mm de 5 a 9 Hz, 2G de 9 a 150 Hz 10 barridos por eje, 1 octava por minuto

Prueba de aislamiento para alta tensión	
Circuitos nominales de 24 V/5 V	520 V DC (ensayo de tipo de límites de aislamiento óptico)
Circuitos de 115/230 V a tierra	1.500 V AC (ensayo de rutina)/1950 V DC (ensayo de tipo)
Circuitos de 115/230 V a circuitos de 115/230 V	1.500 V AC (ensayo de rutina)/1950 V DC (ensayo de tipo)
Circuitos de 115 V/230V a circuitos de 24 V/5 V	1.500 V AC (ensayo de rutina)/3250 V DC (ensayo de tipo)

Clase de protección

- Clase de protección II según EN 61131-2 (el conductor protector no se requiere)

Grado de protección

- Protección mecánica IP20, EN 60529
- Protege los dedos contra el contacto con alta tensión, según ensayos realizados con sondas estándar. Se requiere protección externa contra polvo, impurezas, agua y objetos extraños de < 12,5mm de diámetro.

Tensiones nominales

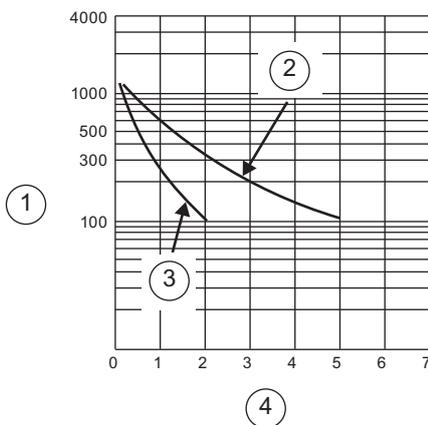
Tensión nominal	Tolerancia
24 V DC	20,4 V DC a 28,8 V DC
120/230 V AC	85 V AC a 264 V AC, 47 a 63 Hz

ATENCIÓN

Cuando un contacto mecánico aplica tensión a una CPU S7-1200, o bien a un módulo de señales digitales, envía una señal "1" a las salidas digitales durante aprox. 50 microsegundos. Considere ésto especialmente si desea utilizar dispositivos que reaccionen a impulsos de breve duración.

Vida útil de los relés

La figura siguiente muestra los datos típicos de rendimiento de los relés suministrados por el comercio especializado. El rendimiento real puede variar dependiendo de la aplicación. Un circuito de protección externo adaptado a la carga permite prolongar la vida útil de los contactos.



- ① Vida útil ($\times 10^3$ operaciones)
- ② 250 V AC de carga resistiva
30 V DC de carga resistiva
- ③ 250 V AC de carga inductiva (p.f.=0,4)
30 V DC de carga inductiva (L/R=7 ms)
- ④ Intensidad normal de servicio (A)

A.2 CPUs

A.2.1 Datos técnicos de la CPU 1211C

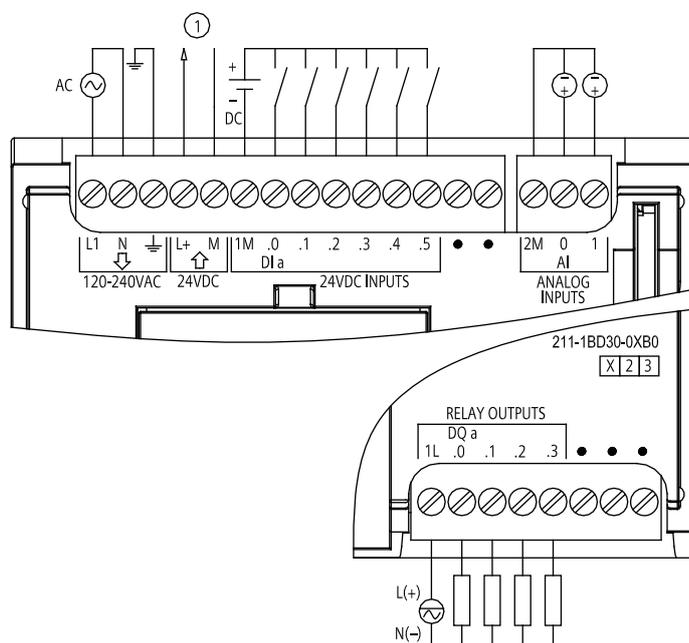
Datos técnicos			
Modelo	CPU 1211C AC/DC/relé	CPU 1211C DC/DC/relé	CPU 1211C DC/DC/DC
Referencia (MLFB)	6ES7 211-1BD30-0XB0	6ES7 211-1HD30-0XB0	6ES7 211-1AD30-0XB0
General			
Dimensiones A x A x P (mm)	90 x 100 x 75		
Peso	420 gramos	380 gramos	370 gramos
Disipación de potencia	10 W	8 W	
Intensidad disponible (bus CM)	750 mA máx. (5 V DC)		
Intensidad disponible (24 V DC)	300 mA máx. (alimentación de sensores)		
Consumo de corriente de las entradas digitales (24 V DC)	4 mA/entrada utilizada		
Características de la CPU			
Memoria de usuario	25 KB de memoria de trabajo / 1 MB de memoria de carga / 2 KB de memoria remanente		
E/S digitales integradas	6 entradas/4 salidas		
E/S analógicas integradas	2 entradas		
Tamaño de la memoria imagen de proceso	1024 bytes de entradas (I)/1024 bytes de salidas (Q)		
Área de marcas (M)	4096 bytes		
Ampliación con módulos de señales	Ninguna		
Ampliación con Signal Boards	1 SB máx.		
Ampliación con módulos de comunicación	3 CMs máx.		
Contadores rápidos	3 en total Fase simple: 3 a 100 kHz Fase en cuadratura: 3 a 80 kHz		
Salidas de impulsos	2		
Entradas de captura de impulsos	6		
Alarmas de retardo/cíclicas	4 en total con resolución de 1 ms		
Alarmas de flanco	6 ascendentes y 6 descendentes (10 y 10 con Signal Board opcional)		
Memory Card	SIMATIC Memory Card (opcional)		
Precisión del reloj en tiempo real	+/- 60 segundos/mes		
Tiempo de respaldo del reloj en tiempo real	10 días típ./6 días mín. a 40°C (condensador de alto rendimiento sin mantenimiento)		
Rendimiento			
Velocidad de ejecución booleana	0,1 µs/instrucción		
Velocidad de ejecución de transferencia de palabras	12 µs/instrucción		

Datos técnicos			
Modelo	CPU 1211C AC/DC/relé	CPU 1211C DC/DC/relé	CPU 1211C DC/DC/DC
Velocidad de ejecución de funciones matemáticas con números reales	18 µs/instrucción		
Comunicación			
Número de puertos	1		
Tipo	Ethernet		
Conexiones	<ul style="list-style-type: none"> • 3 para HMI • 1 para la programadora • 8 para instrucciones Ethernet en el programa de usuario • 3 para CPU a CPU 		
Transferencia de datos	10/100 Mb/s		
Aislamiento (señal externa a lógica del PLC)	Aislado por transformador, 1500 V DC		
Tipo de cable	CAT5e apantallado		
Fuente de alimentación			
Rango de tensión	85 a 264 V AC	20,4 a 28,8 V DC	
Frecuencia de línea	47 a 63 Hz	--	
Intensidad de entrada CPU sólo a carga máx.	60 mA a 120 V AC 30 mA a 240 V AC	300 mA a 24 V DC	
CPU con todos los accesorios de ampliación a carga máx.	180 mA a 120 V AC 90 mA a 240 V AC	900 mA a 24 V DC	
Corriente de irrupción (máx.)	20 A a 264 V AC	12 A a 28,8 V DC	
Aislamiento (potencia de entrada a lógica)	1500 V AC	Sin aislamiento	
Corriente de fuga a tierra, línea AC a tierra funcional	0,5 mA máx.	-	
Tiempo de mantenimiento (pérdida de potencia)	20 ms a 120 V AC 80 ms a 240 V AC	10 ms a 24 V DC	
Fusible interno, no reemplazable por el usuario	3 A, 250 V, de acción lenta		
Alimentación de sensores			
Rango de tensión	20,4 a 28,8 V DC	L+ menos 4 V DC mín.	
Intensidad de salida nominal (máx.)	300 mA (protegido contra cortocircuito)		
Ruido de rizado máx. (<10 MHz)	< 1 V de pico a pico	Igual a la línea de entrada	
Aislamiento (lógica de la CPU a alimentación de sensores)	Sin aislamiento		
Entradas digitales			
Número de entradas	6		
Tipo	Sumidero/fuente (tipo 1 IEC sumidero)		
Tensión nominal	24 V DC a 4 mA, nominal		
Tensión continua admisible	30 V DC, máx.		
Sobretensión transitoria	35 V DC durante 0,5 seg.		
Señal 1 lógica (mín.)	15 V DC a 2,5 mA		

Datos técnicos			
Modelo	CPU 1211C AC/DC/relé	CPU 1211C DC/DC/relé	CPU 1211C DC/DC/DC
Señal 0 lógica (máx.)	5 V DC a 1 mA		
Aislamiento (campo a lógica)	500 V AC durante 1 minuto		
Grupos de aislamiento	1		
Tiempos de filtro	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms (seleccionable en grupos de 4)		
Frecuencias de entrada de reloj HSC (máx.) (señal 1 lógica = 15 a 26 V DC)	Fase simple: 100 KHz Fase en cuadratura: 80 KHz		
Número de entradas ON simultáneamente	6		
Longitud de cable (metros)	500 apantallado, 300 no apantallado, 50 apantallado para entradas HSC		
Entradas analógicas			
Número de entradas	2		
Tipo	Tensión (asimétrica)		
Rango	0 a 10 V		
Rango total (palabra de datos)	0 a 27648 (consulte Representación de entradas analógicas para tensión (Página 346))		
Rango de sobreimpulso (palabra de datos)	27.649 a 32.511 (consulte Representación de entradas analógicas para tensión (Página 346))		
Desbordamiento (palabra de datos)	32.512 a 32767 (consulte Representación de entradas analógicas para tensión (Página 346))		
Resolución	10 bits		
Tensión de resistencia al choque máxima	35 V DC		
Alisamiento	Ninguno, débil, medio o fuerte (consulte los tiempos de respuesta de paso en Tiempos de respuesta de las entradas analógicas (Página 346))		
Rechazo de interferencias	10, 50 ó 60 Hz (consulte las frecuencias de muestreo en Tiempos de respuesta de las entradas analógicas (Página 346))		
Impedancia	≥100 KΩ		
Aislamiento (campo a lógica)	Ninguno		
Precisión (25°C / 0 a 55°C)	3,0% / 3,5% de rango máximo		
Rechazo en modo común	40 dB, DC a 60 Hz		
Rango de señales operativo	La tensión de señal más la tensión en modo común debe ser menor que +12 V y mayor que -12 V		
Longitud de cable (metros)	10 m, par trenzado apantallado		
Salidas digitales			
Número de salidas	4		
Tipo	Relé, contacto seco	Estado sólido - MOSFET	
Rango de tensión	5 a 30 V DC ó 5 a 250 V AC		20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	--		20 V DC mín.
Señal 0 lógica con carga de 10 KΩ	--		0,1 V DC máx.
Intensidad (máx.)	2,0 A		0,5 A
Carga de lámparas	30 W DC/200 W AC		5 W
Resistencia en estado ON	Máx. 0,2 Ω (si son nuevas)		0,6 Ω máx.

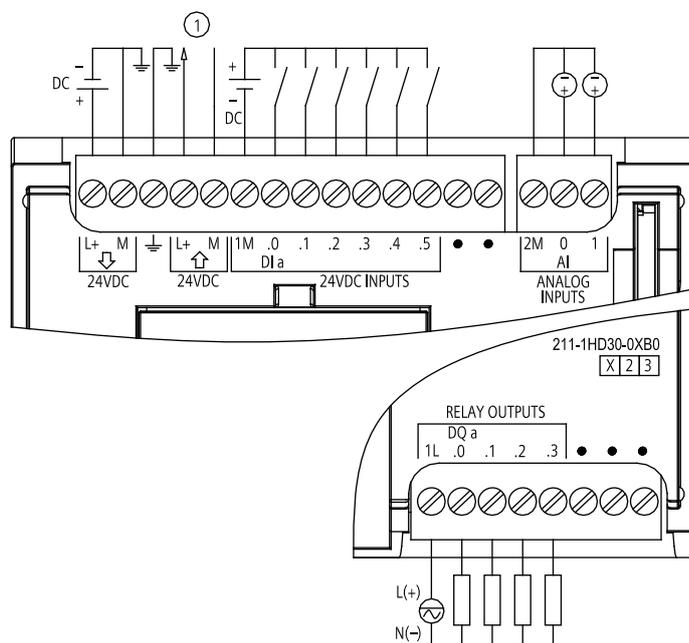
Datos técnicos			
Modelo	CPU 1211C AC/DC/relé	CPU 1211C DC/DC/relé	CPU 1211C DC/DC/DC
Corriente de fuga por salida	--		10 µA máx.
Sobrecorriente momentánea	7 A si están cerrados los contactos		8 A durante máx. 100 ms
Protección contra sobrecargas	No		
Aislamiento (campo a lógica)	1500 V AC durante 1 minuto (bobina a contacto) Ninguno (bobina a lógica)		500 V AC durante 1 minuto
Resistencia de aislamiento	100 MΩ mín. si son nuevas		--
Aislamiento entre contactos abiertos	750 V AC durante 1 minuto		--
Grupos de aislamiento	1		1
Tensión de bloqueo inductiva	--		L+ menos 48 V DC, disipación de 1 W
Retardo de conmutación (Qa.0 a Qa.3)	10 ms máx.		1,0 µs máx., OFF a ON 3,0 µs máx., ON a OFF
Frecuencia de tren de impulsos (Qa.0 y Qa.2)	No recomendado		100 KHz máx., 2 Hz mín.
Vida útil mecánica (sin carga)	10.000.000 ciclos abiertos/cerrados		--
Vida útil de los contactos bajo carga nominal	100.000 ciclos abiertos/cerrados		--
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)		
Número de salidas ON simultáneamente	4		
Longitud de cable (metros)	500 apantallado, 150 no apantallado		

Diagramas de cableado



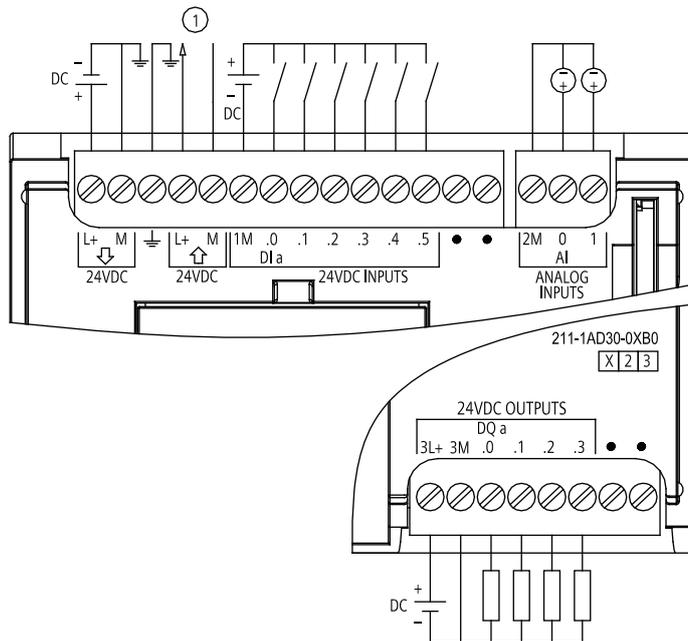
① Alimentación de sensores 24 V DC

Figura A-1 CPU 1211C AC/DC/relé (6ES7 211-1BD30-0XB0)



① Alimentación de sensores 24 V DC

Figura A-2 CPU 1211C DC/DC/relé (6ES7 211-1HD30-0XB0)



① Alimentación de sensores 24 V DC

Figura A-3 CPU 1211C DC/DC/DC (6ES7 211-1AD30-0XB0)

A.2.2 Datos técnicos de la CPU 1212C

Datos técnicos			
Modelo	CPU 1212C AC/DC/relé	CPU 1212C DC/DC/relé	CPU 1212C DC/DC/DC
Referencia	6ES7 212-1BD30-0XB0	6ES7 212-1HD30-0XB0	6ES7 212-1AD30-0XB0
General			
Dimensiones A x A x P (mm)	90 x 100 x 75		
Peso	425 gramos	385 gramos	370 gramos
Disipación de potencia	11 W	9 W	
Intensidad disponible (SM y bus CM)	1000 mA máx. (5 V DC)		
Intensidad disponible (24 V DC)	300 mA máx. (alimentación de sensores)		
Consumo de corriente de las entradas digitales (24 V DC)	4 mA/entrada utilizada		
Características de la CPU			
Memoria de usuario	25 KB de memoria de trabajo / 1 MB de memoria de carga / 2 KB de memoria remanente		
E/S digitales integradas	8 entradas/6 salidas		
E/S analógicas integradas	2 entradas		
Tamaño de la memoria imagen de proceso	1024 bytes de entradas (I)/1024 bytes de salidas (Q)		
Área de marcas (M)	4096 bytes		
Ampliación con módulos de señales	2 SMs máx.		

Datos técnicos			
Modelo	CPU 1212C AC/DC/relé	CPU 1212C DC/DC/relé	CPU 1212C DC/DC/DC
Ampliación con Signal Boards	1 SB máx.		
Ampliación con módulos de comunicación	3 CMs máx.		
Contadores rápidos	4 en total Fase simple: 3 a 100 kHz y 1 a 30 kHz de frecuencia de reloj Fase en cuadratura: 3 a 80 kHz y 1 a 20 kHz de frecuencia de reloj		
Salidas de impulsos	2		
Entradas de captura de impulsos	8		
Alarmas de retardo/cíclicas	4 en total con resolución de 1 ms		
Alarmas de flanco	8 ascendentes y 8 descendentes (12 y 12 con Signal Board opcional)		
Memory Card	SIMATIC Memory Card (opcional)		
Precisión del reloj en tiempo real	+/- 60 segundos/mes		
Tiempo de respaldo del reloj en tiempo real	10 días típ./6 días mín. a 40°C (condensador de alto rendimiento sin mantenimiento)		
Rendimiento			
Velocidad de ejecución booleana	0,1 µs/instrucción		
Velocidad de ejecución de transferencia de palabras	12 µs/instrucción		
Velocidad de ejecución de funciones matemáticas con números reales	18 µs/instrucción		
Comunicación			
Número de puertos	1		
Tipo	Ethernet		
Conexiones	<ul style="list-style-type: none"> • 3 para HMI • 1 para la programadora • 8 para instrucciones Ethernet en el programa de usuario • 3 para CPU a CPU 		
Transferencia de datos	10/100 Mb/s		
Aislamiento (señal externa a lógica del PLC)	Aislado por transformador, 1500 V DC		
Tipo de cable	CAT5e apantallado		
Fuente de alimentación			
Rango de tensión	85 a 264 V AC	20,4 a 28,8 V DC	
Frecuencia de línea	47 a 63 Hz	--	
Intensidad de entrada CPU sólo a carga máx.	80 mA a 120 V AC 40 mA a 240 V AC	400 mA a 24 V DC	
CPU con todos los accesorios de ampliación a carga máx.	240 mA a 120 V AC 120 mA a 240 V AC	1200 mA a 24 V DC	
Corriente de irrupción (máx.)	20 A a 264 V AC	12 A a 28,8 V DC	
Aislamiento (potencia de entrada a lógica)	1500 V AC	Sin aislamiento	
Corriente de fuga a tierra, línea AC a tierra funcional	0,5 mA máx.	-	

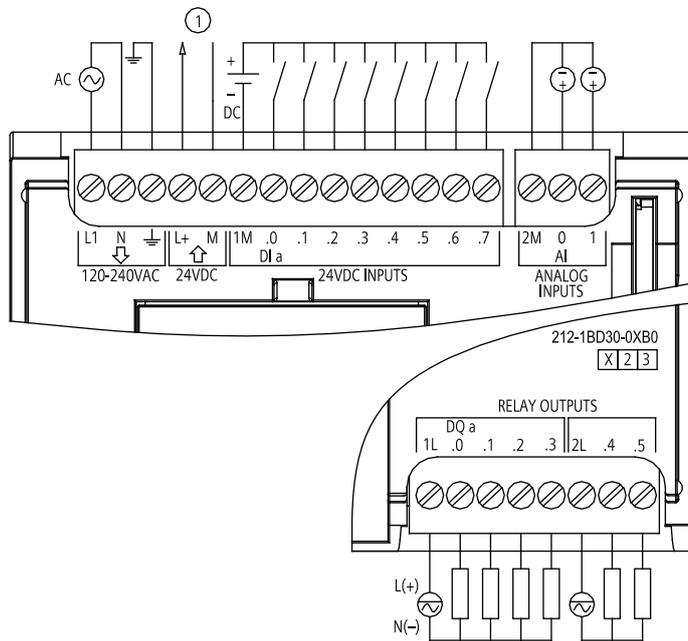
Datos técnicos

A.2 CPUs

Datos técnicos			
Modelo	CPU 1212C AC/DC/relé	CPU 1212C DC/DC/relé	CPU 1212C DC/DC/DC
Tiempo de mantenimiento (pérdida de potencia)	20 ms a 120 V AC 80 ms a 240 V AC	10 ms a 24 V DC	
Fusible interno, no reemplazable por el usuario	3 A, 250 V, de acción lenta		
Alimentación de sensores			
Rango de tensión	20,4 a 28,8 V DC	L+ menos 4 V DC mín.	
Intensidad de salida nominal (máx.)	300 mA (protegido contra cortocircuito)		
Ruido de rizado máx. (<10 MHz)	< 1 V de pico a pico	Igual a la línea de entrada	
Aislamiento (lógica de la CPU a alimentación de sensores)	Sin aislamiento		
Entradas digitales			
Número de entradas	8		
Tipo	Sumidero/fuente (tipo 1 IEC sumidero)		
Tensión nominal	24 V DC a 4 mA, nominal		
Tensión continua admisible	30 V DC, máx.		
Sobretensión transitoria	35 V DC durante 0,5 seg.		
Señal 1 lógica (mín.)	15 V DC a 2,5 mA		
Señal 0 lógica (máx.)	5 V DC a 1 mA		
Aislamiento (campo a lógica)	500 V AC durante 1 minuto		
Grupos de aislamiento	1		
Tiempos de filtro	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms (seleccionable en grupos de 4)		
Frecuencias de entrada de reloj HSC (máx.) (señal 1 lógica = 15 a 26 V DC)	Fase simple: 100 KHz (la.0 a la.5) y 30 KHz (la.6 a la.7) Fase en cuadratura: 80 KHz (la.0 a la.5) y 20 KHz (la.6 a la.7)		
Número de entradas ON simultáneamente	8		
Longitud de cable (metros)	500 apantallado, 300 no apantallado, 50 apantallado para entradas HSC		
Entradas analógicas			
Número de entradas	2		
Tipo	Tensión (asimétrica)		
Rango	0 a 10 V		
Rango total (palabra de datos)	0 a 27648 (consulte Representación de entradas analógicas para tensión (Página 346))		
Rango de sobreimpulso (palabra de datos)	27.649 a 32.511 (consulte Representación de entradas analógicas para tensión (Página 346))		
Desbordamiento (palabra de datos)	32.512 a 32767 (consulte Representación de entradas analógicas para tensión (Página 346))		
Resolución	10 bits		
Tensión de resistencia al choque máxima	35 V DC		
Alisamiento	Ninguno, débil, medio o fuerte (consulte los tiempos de respuesta de paso en Tiempos de respuesta de las entradas analógicas (Página 346))		

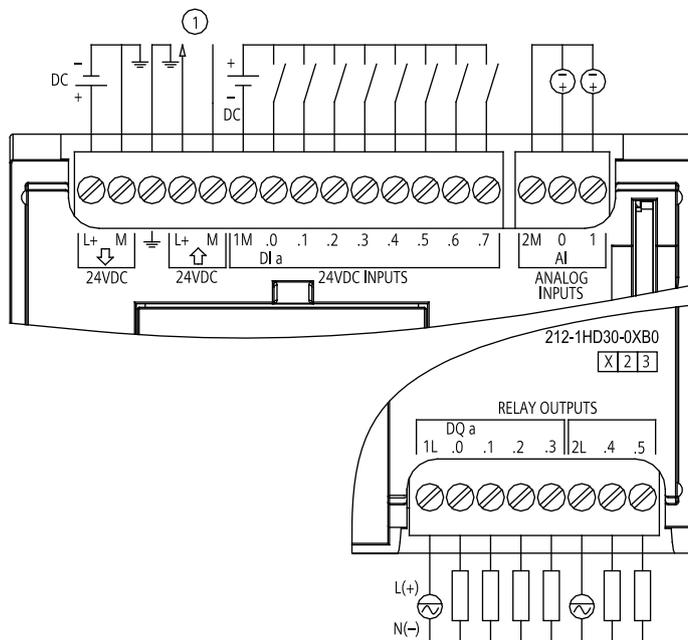
Datos técnicos			
Modelo	CPU 1212C AC/DC/relé	CPU 1212C DC/DC/relé	CPU 1212C DC/DC/DC
Rechazo de interferencias	10, 50 ó 60 Hz (consulte las frecuencias de muestreo en Tiempos de respuesta de las entradas analógicas (Página 346))		
Impedancia	≥100 KΩ		
Aislamiento (campo a lógica)	Ninguno		
Precisión (25°C / 0 a 55°C)	3,0% / 3,5% de rango máximo		
Rechazo en modo común	40 dB, DC a 60 Hz		
Rango de señales operativo	La tensión de señal más la tensión en modo común debe ser menor que +12 V y mayor que -12 V		
Longitud de cable (metros)	10 trenzado y apantallado		
Salidas digitales			
Número de salidas	6		
Tipo	Relé, contacto seco		Estado sólido - MOSFET
Rango de tensión	5 a 30 V DC ó 5 a 250 V AC		20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	--		20 V DC mín.
Señal 0 lógica con carga de 10 KΩ	--		0,1 V DC máx.
Intensidad (máx.)	2,0 A		0,5 A
Carga de lámparas	30 W DC/200 W AC		5 W
Resistencia en estado ON	Máx. 0,2 Ω (si son nuevas)		0,6 Ω máx.
Corriente de fuga por salida	--		10 μA máx.
Sobrecorriente momentánea	7 A si están cerrados los contactos		8 A durante máx. 100 ms
Protección contra sobrecargas	No		
Aislamiento (campo a lógica)	1500 V AC durante 1 minuto (bobina a contacto) Ninguno (bobina a lógica)		500 V AC durante 1 minuto
Resistencia de aislamiento	100 MΩ mín. si son nuevas		--
Aislamiento entre contactos abiertos	750 V AC durante 1 minuto		--
Grupos de aislamiento	2		1
Tensión de bloqueo inductiva	--		L+ menos 48 V DC, disipación de 1 W
Retardo de conmutación (Qa.0 a Qa.3)	10 ms máx.		1,0 μs máx., OFF a ON 3,0 μs máx., ON a OFF
Retardo de conmutación (Qa.4 a Qa.5)	10 ms máx.		50 μs máx., OFF a ON 200 μs máx., ON a OFF
Frecuencia de tren de impulsos (Qa.0 y Qa.2)	No recomendado		100 KHz máx., 2 Hz mín.
Vida útil mecánica (sin carga)	10.000.000 ciclos abiertos/cerrados		--
Vida útil de los contactos bajo carga nominal	100.000 ciclos abiertos/cerrados		--
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)		
Número de salidas ON simultáneamente	6		
Longitud de cable (metros)	500 apantallado, 150 no apantallado		

Diagramas de cableado



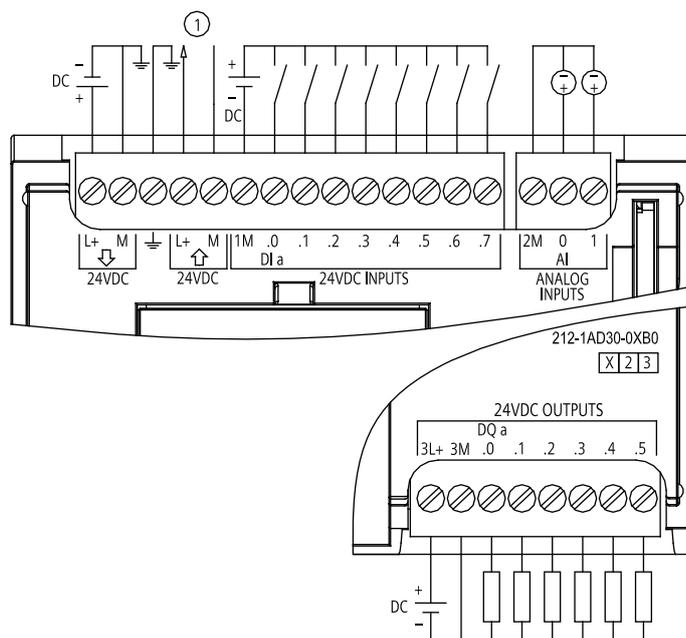
① Alimentación de sensores 24 V DC

Figura A-4 CPU 1212C AC/DC relé (6ES7 212-1BD30-0XB0)



① Alimentación de sensores 24 V DC

Figura A-5 CPU 1212C DC/DC/relé (6ES7 212-1HD30-0XB0)



① Alimentación de sensores 24 V DC

Figura A-6 CPU 1212C DC/DC/DC (6ES7 212-1AD30-0XB0)

A.2.3 Datos técnicos de la CPU 1214C

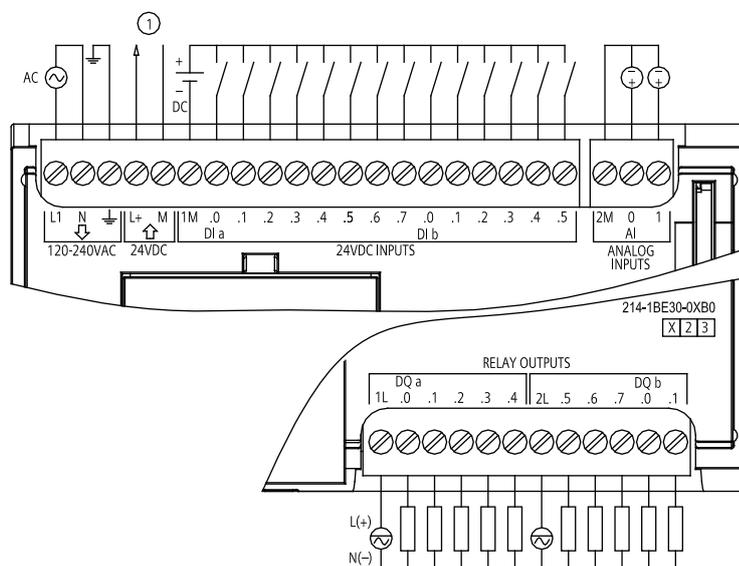
Datos técnicos			
Modelo	CPU 1214C AC/DC/relé	CPU 1214C DC/DC/relé	CPU 1214C DC/DC/DC
Referencia	6ES7 214-1BE30-0XB0	6ES7 214-1HE30-0XB0	6ES7 214-1AE30-0XB0
General			
Dimensiones A x A x P (mm)	110 x 100 x 75		
Peso	475 gramos	435 gramos	415 gramos
Disipación de potencia	14 W	12 W	
Intensidad disponible (SM y bus CM)	1600 mA máx. (5 V DC)		
Intensidad disponible (24 V DC)	400 mA máx. (alimentación de sensores)		
Consumo de corriente de las entradas digitales (24 V DC)	4 mA/entrada utilizada		
Características de la CPU			
Memoria de usuario	50 KB de memoria de trabajo / 2 MB de memoria de carga / 2 KB de memoria remanente		
E/S digitales integradas	14 entradas/10 salidas		
E/S analógicas integradas	2 entradas		
Tamaño de la memoria imagen de proceso	1024 bytes de entradas (I)/1024 bytes de salidas (Q)		
Área de marcas (M)	8192 bytes		
Ampliación con módulos de señales	8 SMs máx.		

Datos técnicos			
Modelo	CPU 1214C AC/DC/relé	CPU 1214C DC/DC/relé	CPU 1214C DC/DC/DC
Ampliación con Signal Boards	1 SB máx.		
Ampliación con módulos de comunicación	3 CMs máx.		
Contadores rápidos	6 en total Fase simple: 3 a 100 kHz y 3 a 30 kHz de frecuencia de reloj Fase en cuadratura: 3 a 80 kHz y 3 a 20 kHz de frecuencia de reloj		
Salidas de impulsos	2		
Entradas de captura de impulsos	14		
Alarmas de retardo/cíclicas	4 en total con resolución de 1 ms		
Alarmas de flanco	12 ascendentes y 12 descendentes (14 y 14 con Signal Board opcional)		
Memory Card	SIMATIC Memory Card (opcional)		
Precisión del reloj en tiempo real	+/- 60 segundos/mes		
Tiempo de respaldo del reloj en tiempo real	10 días típ./6 días mín. a 40°C (condensador de alto rendimiento sin mantenimiento)		
Rendimiento			
Velocidad de ejecución booleana	0,1 µs/instrucción		
Velocidad de ejecución de transferencia de palabras	12 µs/instrucción		
Velocidad de ejecución de funciones matemáticas con números reales	18 µs/instrucción		
Comunicación			
Número de puertos	1		
Tipo	Ethernet		
Conexiones	<ul style="list-style-type: none"> • 3 para HMI • 1 para la programadora • 8 para instrucciones Ethernet en el programa de usuario • 3 para CPU a CPU 		
Transferencia de datos	10/100 Mb/s		
Aislamiento (señal externa a lógica del PLC)	Aislado por transformador, 1500 V DC		
Tipo de cable	CAT5e apantallado		
Fuente de alimentación			
Rango de tensión	85 a 264 V AC	20,4 a 28,8 V DC	
Frecuencia de línea	47 a 63 Hz	--	
Intensidad de entrada CPU sólo a carga máx.	100 mA a 120 V AC 50 mA a 240 V AC	500 mA a 24 V DC	
CPU con todos los accesorios de ampliación a carga máx.	300 mA a 120 V AC 150 mA a 240 V AC	1500 mA a 24 V DC	
Corriente de irrupción (máx.)	20 A a 264 V AC	12 A a 28,8 V DC	
Aislamiento (potencia de entrada a lógica)	1500 V AC	Sin aislamiento	
Corriente de fuga a tierra, línea AC a tierra funcional	0,5 mA máx.	-	

Datos técnicos			
Modelo	CPU 1214C AC/DC/relé	CPU 1214C DC/DC/relé	CPU 1214C DC/DC/DC
Tiempo de mantenimiento (pérdida de potencia)	20 ms a 120 V AC 80 ms a 240 V AC	10 ms a 24 V DC	
Fusible interno, no reemplazable por el usuario	3 A, 250 V, de acción lenta		
Alimentación de sensores			
Rango de tensión	20,4 a 28,8 V DC	L+ menos 4 V DC mín.	
Intensidad de salida nominal (máx.)	400 mA (protegido contra cortocircuito)		
Ruido de rizado máx. (<10 MHz)	< 1 V de pico a pico	Igual a la línea de entrada	
Aislamiento (lógica de la CPU a alimentación de sensores)	Sin aislamiento		
Entradas digitales			
Número de entradas	14		
Tipo	Sumidero/fuente (tipo 1 IEC sumidero)		
Tensión nominal	24 V DC a 4 mA, nominal		
Tensión continua admisible	30 V DC, máx.		
Sobretensión transitoria	35 V DC durante 0,5 seg.		
Señal 1 lógica (mín.)	15 V DC a 2,5 mA		
Señal 0 lógica (máx.)	5 V DC a 1 mA		
Aislamiento (campo a lógica)	500 V AC durante 1 minuto		
Grupos de aislamiento	1		
Tiempos de filtro	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms (seleccionable en grupos de 4)		
Frecuencias de entrada de reloj HSC (máx.) (señal 1 lógica = 15 a 26 V DC)	Fase simple: 100 KHz (Ia.0 a Ia.5) y 30 KHz (Ia.6 a Ib.5) Fase en cuadratura: 80 KHz (Ia.0 a Ia.5) y 20 KHz (Ia.6 a Ib.5)		
Número de entradas ON simultáneamente	14		
Longitud de cable (metros)	500 apantallado, 300 no apantallado, 50 apantallado para entradas HSC		
Entradas analógicas			
Número de entradas	2		
Tipo	Tensión (asimétrica)		
Rango	0 a 10 V		
Rango total (palabra de datos)	0 a 27648 (consulte Representación de entradas analógicas para tensión (Página 346))		
Rango de sobreimpulso (palabra de datos)	27.649 a 32.511 (consulte Representación de entradas analógicas para tensión (Página 346))		
Desbordamiento (palabra de datos)	32.512 a 32767 (consulte Representación de entradas analógicas para tensión (Página 346))		
Resolución	10 bits		
Tensión de resistencia al choque máxima	35 V DC		
Alisamiento	Ninguno, débil, medio o fuerte (consulte los tiempos de respuesta de las etapas en Tiempos de respuesta de las entradas analógicas (Página 346))		

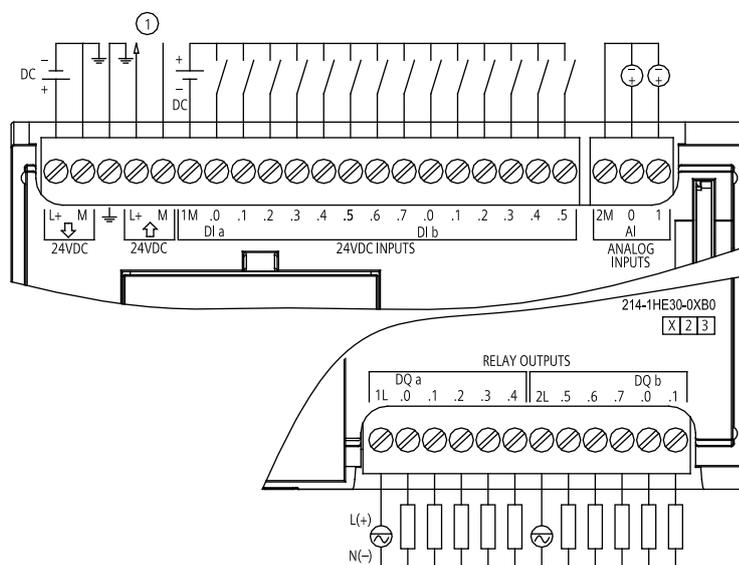
Datos técnicos			
Modelo	CPU 1214C AC/DC/relé	CPU 1214C DC/DC/relé	CPU 1214C DC/DC/DC
Rechazo de interferencias	10, 50 ó 60 Hz (consulte las frecuencias de muestreo en Tiempos de respuesta de las entradas analógicas (Página 346))		
Impedancia	≥100 KΩ		
Aislamiento (campo a lógica)	Ninguno		
Precisión (25°C / 0 a 55°C)	3,0% / 3,5% de rango máximo		
Rechazo en modo común	40 dB, DC a 60 Hz		
Rango de señales operativo	La tensión de señal más la tensión en modo común debe ser menor que +12 V y mayor que -12 V		
Longitud de cable (metros)	10 trenzado y apantallado		
Salidas digitales			
Número de salidas	10		
Tipo	Relé, contacto seco		Estado sólido - MOSFET
Rango de tensión	5 a 30 V DC ó 5 a 250 V AC		20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	--		20 V DC mín.
Señal 0 lógica con carga de 10 KΩ	--		0,1 V DC máx.
Intensidad (máx.)	2,0 A		0,5 A
Carga de lámparas	30 W DC/200 W AC		5 W
Resistencia en estado ON	Máx. 0,2 Ω (si son nuevas)		0,6 Ω máx.
Corriente de fuga por salida	--		10 μA máx.
Sobrecorriente momentánea	7 A si están cerrados los contactos		8 A durante máx. 100 ms
Protección contra sobrecargas	No		
Aislamiento (campo a lógica)	1500 V AC durante 1 minuto (bobina a contacto) Ninguno (bobina a lógica)		500 V AC durante 1 minuto
Resistencia de aislamiento	100 MΩ mín. si son nuevas		--
Aislamiento entre contactos abiertos	750 V AC durante 1 minuto		--
Grupos de aislamiento	2		1
Tensión de bloqueo inductiva	--		L+ menos 48 V DC, disipación de 1 W
Retardo de conmutación (Qa.0 a Qa.3)	10 ms máx.		1,0 μs máx., OFF a ON 3,0 μs máx., ON a OFF
Retardo de conmutación (Qa.4 a Qb.1)	10 ms máx.		50 μs máx., OFF a ON 200 μs máx., ON a OFF
Frecuencia de tren de impulsos (Qa.0 y Qa.2)	No recomendado		100 KHz máx., 2 Hz mín.
Vida útil mecánica (sin carga)	10.000.000 ciclos abiertos/cerrados		--
Vida útil de los contactos bajo carga nominal	100.000 ciclos abiertos/cerrados		--
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)		
Número de salidas ON simultáneamente	10		
Longitud de cable (metros)	500 apantallado, 150 no apantallado		

Diagramas de cableado



① Alimentación de sensores 24 V DC

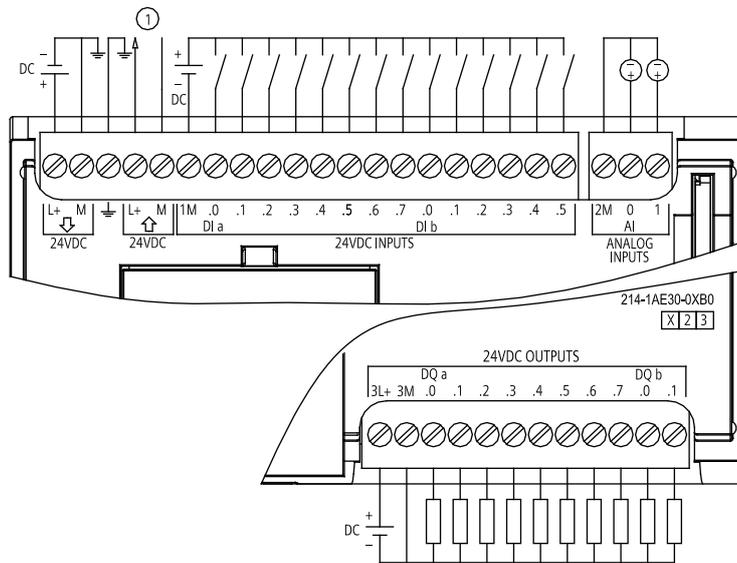
Figura A-7 CPU 1214C AC/DC/relé (6ES7 214-1BE30-0XB0)



① Alimentación de sensores 24 V DC

Figura A-8 CPU 1214C DC/DC/relé (6ES7 214-1HE30-0XB0)

A.3 Módulos de señales digitales (SMs)



① Alimentación de sensores 24 V DC

Figura A-9 CPU 1214C DC/DC/DC (6ES7 214-1AE30-0XB0)

A.3 Módulos de señales digitales (SMs)

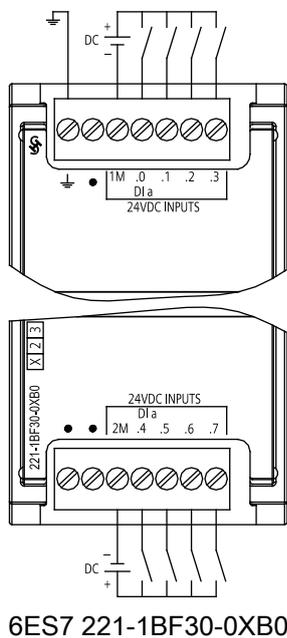
A.3.1 Datos técnicos del módulo de entradas digitales SM 1221

Datos técnicos		
Modelo	SM 1221 DI 8x24 V DC	SM 1221 DI 16x24 VD C
Referencia	6ES7 221-1BF30-0XB0	6ES7 221-1BH30-0XB0
General		
Dimensiones A x A x P (mm)	45 x 100 x 75	
Peso	170 gramos	210 gramos
Disipación de potencia	1,5 W	2,5 W
Consumo de corriente (bus SM)	105 mA	130 mA
Consumo de corriente (24 V DC)	4 mA/entrada utilizada	4 mA/entrada utilizada
Entradas digitales		
Número de entradas	8	16
Tipo	Sumidero/fuente (tipo 1 IEC sumidero)	
Tensión nominal	24 V DC a 4 mA, nominal	
Tensión continua admisible	30 V DC, máx.	
Sobretensión transitoria	35 V DC durante 0,5 seg.	
Señal 1 lógica (mín.)	15 V DC a 2,5 mA	
Señal 0 lógica (máx.)	5 V DC a 1 mA	

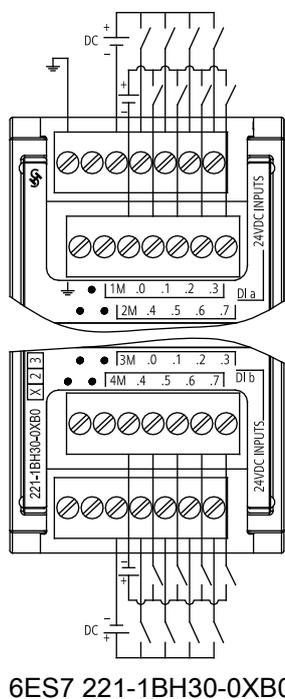
Datos técnicos		
Modelo	SM 1221 DI 8x24 V DC	SM 1221 DI 16x24 VD C
Aislamiento (campo a lógica)	500 V AC durante 1 minuto	
Grupos de aislamiento	2	4
Tiempos de filtro	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms (seleccionable en grupos de 4)	
Número de entradas ON simultáneamente	8	16
Longitud de cable (metros)	500 apantallado, 300 no apantallado	

Diagramas de cableado

SM 1221 DI 8 x 24 V DC



SM 1221 DI 16 x 24 V DC

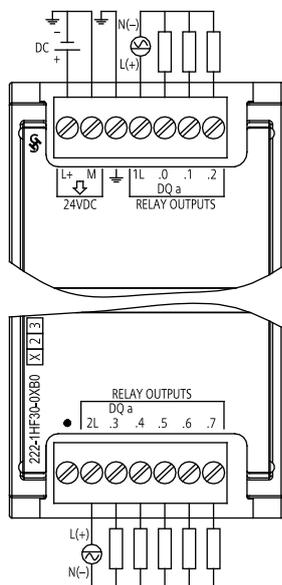


A.3.2 Datos técnicos del módulo de salidas digitales SM 1222

Datos técnicos				
Modelo	SM 1222 DQ 8xrelé	SM1222 DQ 16xrelé	SM1222 DQ 8x24 V DC	SM1222 DQ 16x24 V DC
Referencia	6ES7 222-1HF30-0XB0	6ES7 222-1HH30-0XB0	6ES7 222-1BF30-0XB0	6ES7 222-1BH30-0XB0
General				
Dimensiones A x A x P (mm)	45 x 100 x 75			
Peso	190 gramos	260 gramos	180 gramos	220 gramos
Disipación de potencia	4,5 W	8,5 W	1,5 W	2,5 W
Consumo de corriente (bus SM)	120 mA	135 mA	120 mA	140 mA
Consumo de corriente (24 V DC)	11 mA/bobina de relé utilizada		--	
Salidas digitales				
Número de salidas	8	16	8	16
Tipo	Relé, contacto seco		Estado sólido - MOSFET	
Rango de tensión	5 a 30 V DC ó 5 a 250 V AC		20,4 a 28,8 V DC	
Señal 1 lógica a intensidad máx.	--		20 V DC mín.	
Señal 0 lógica con carga de 10K Ω	--		0,1 V DC máx.	
Intensidad (máx.)	2,0 A		0,5 A	
Carga de lámparas	30 W DC/200 W AC		5W	
Resistencia en estado ON (contactos)	Máx. 0,2 Ω (si son nuevas)		0,6 Ω máx.	
Corriente de fuga por salida	--		10 μ A máx.	
Sobrecorriente momentánea	7 A si están cerrados los contactos		8 A durante máx. 100 ms	
Protección contra sobrecargas	No			
Aislamiento (campo a lógica)	1500 V AC durante 1 minuto (bobina a contacto) Ninguno (bobina a lógica)		500 V AC durante 1 minuto	
Resistencia de aislamiento	100 M Ω mín. si son nuevas		--	
Aislamiento entre contactos abiertos	750 V AC durante 1 minuto		--	
Grupos de aislamiento	2	4	1	1
Intensidad por neutro (máx.)	10 A		4 A	8 A
Tensión de bloqueo inductiva	--		L+ menos 48 V, disipación de 1 W	
Retardo de conmutación	10 ms máx.		50 μ s máx. OFF a ON 200 μ s máx. ON a OFF	
Vida útil mecánica (sin carga)	10.000.000 ciclos abiertos/cerrados		--	
Vida útil de los contactos bajo carga nominal	100.000 ciclos abiertos/cerrados		--	
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)			
Número de salidas ON simultáneamente	8	16	8	16
Longitud de cable (metros)	500 apantallado, 150 no apantallado			

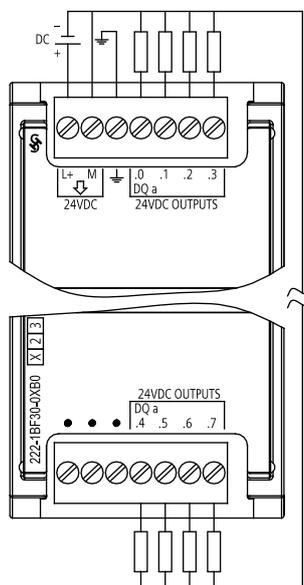
Diagramas de cableado

SM 1222 DQ 8 x relé



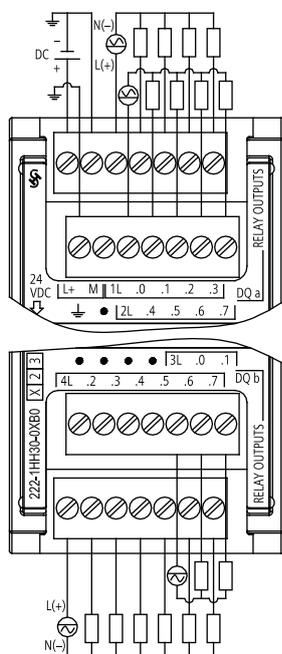
6ES7 222-1HF30-0XB0

SM 1222 DQ 8 x 24 V DC



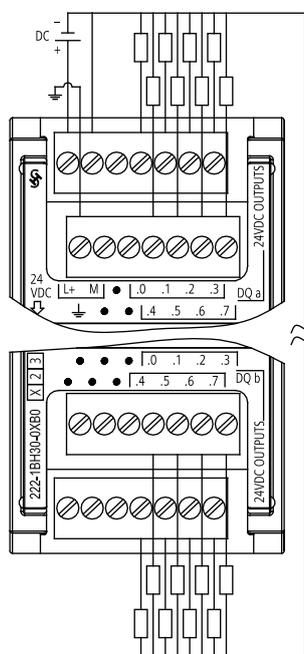
6ES7 222-1BF30-0XB0

SM 1222 DQ 16 x relé



6ES7 222-1HH30-0XB0

SM 1222 DQ 16 x 24 V DC



6ES7 222-1BH30-0XB0

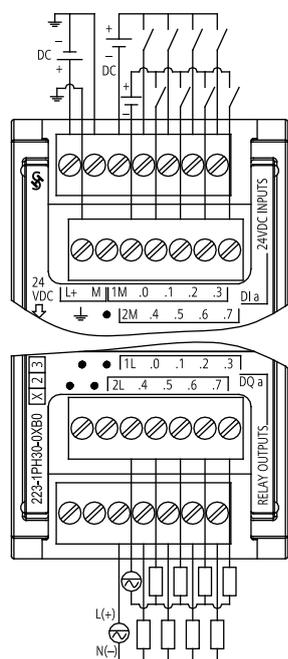
A.3.3 Datos técnicos del módulo de entradas/salidas digitales SM 1223

Datos técnicos				
Modelo	SM 1223 DI 8x24 V DC, DQ 8xrelé	SM 1223 DI 16x24 V DC, DQ 16xrelé	SM 1223 DI 8x24 V DC, DQ 8x24 V DC	SM 1223 DI 16x24 V DC, DQ16x24 V DC
Referencia	6ES7 223-1PH30-0XB0	6ES7 223-1PL30-0XB0	6ES7 223-1BH30-0XB0	6ES7 223-1BL30-0XB0
Dimensiones A x A x P (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75	70 x 100 x 75
Peso	230 gramos	350 gramos	210 gramos	310 gramos
Disipación de potencia	5,5 W	10 W	2,5 W	4,5 W
Consumo de corriente (bus SM)	145 mA	180 mA	145 mA	185 mA
Consumo de corriente (24 V DC)	4 mA/entrada utilizada 11 mA/bobina de relé utilizada		4 mA/entrada utilizada	
Entradas digitales				
Número de entradas	8	16	8	16
Tipo	Sumidero/fuente (tipo 1 IEC sumidero)			
Tensión nominal	24 V DC a 4 mA, nominal			
Tensión continua admisible	30 V DC máx.			
Sobretensión transitoria	35 V DC durante 0,5 seg.			
Señal 1 lógica (mín.)	15 V DC a 2,5 mA			
Señal 0 lógica (máx.)	5 V DC a 1 mA			
Aislamiento (campo a lógica)	500 V AC durante 1 minuto			
Grupos de aislamiento	2	2	2	2
Tiempos de filtro	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms, seleccionable en grupos de 4			
Número de entradas ON simultáneamente	8	16	8	16
Longitud de cable (metros)	500 apantallado, 300 no apantallado			
Salidas digitales				
Número de salidas	8	16	8	16
Tipo	Relé, contacto seco		Estado sólido - MOSFET	
Rango de tensión	5 a 30 V DC ó 5 a 250 V AC		20,4 a 28,8 V DC	
Señal 1 lógica a intensidad máx.	--		20 V DC mín.	
Señal 0 lógica con carga de 10 K Ω	--		0,1 V DC, máx.	
Intensidad (máx.)	2,0 A		0,5 A	
Carga de lámparas	30 W DC/200 W AC		5 W	
Resistencia en estado ON (contactos)	Máx. 0,2 Ω (si son nuevas)		0,6 Ω máx.	
Corriente de fuga por salida	--		10 μ A máx.	
Sobrecorriente momentánea	7 A si están cerrados los contactos		8 A durante máx. 100 ms	
Protección contra sobrecargas	No			
Aislamiento (campo a lógica)	1500 V AC durante 1 minuto (bobina a contacto) Ninguno (bobina a lógica)		500 V AC durante 1 minuto	
Resistencia de aislamiento	100 M Ω mín. si son nuevas		--	

Datos técnicos				
Modelo	SM 1223 DI 8x24 V DC, DQ 8xrelé	SM 1223 DI 16x24 V DC, DQ 16xrelé	SM 1223 DI 8x24 V DC, DQ 8x24 V DC	SM 1223 DI 16x24 V DC, DQ16x24 V DC
Aislamiento entre contactos abiertos	750 V AC durante 1 minuto		--	
Grupos de aislamiento	2	4	1	1
Intensidad por neutro	10A	8 A	4 A	8 A
Tensión de bloqueo inductiva	--		L+ menos 48 V, disipación de 1 W	
Retardo de conmutación	10 ms máx.		50 µs máx. OFF a ON 200 µs máx. ON a OFF	
Vida útil mecánica (sin carga)	10.000.000 ciclos abiertos/cerrados		--	
Vida útil de los contactos bajo carga nominal	100.000 ciclos abiertos/cerrados		--	
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)			
Número de salidas ON simultáneamente	8	16	8	16
Longitud de cable (metros)	500 apantallado, 150 no apantallado			

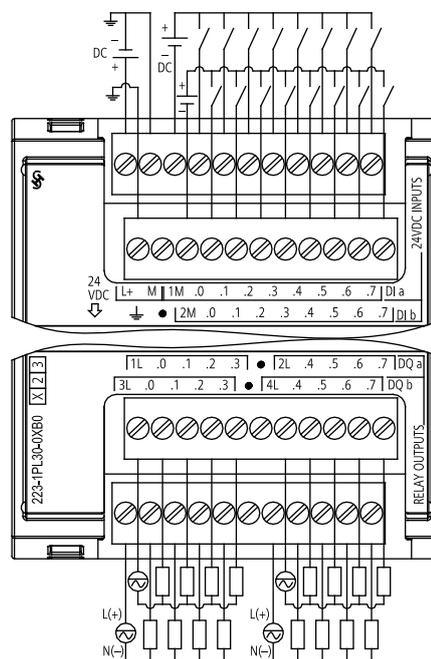
Diagramas de cableado

SM 1223 DI 8 x 24 V DC, DQ 8 x relé



6ES7 223-1PH30-0XB0

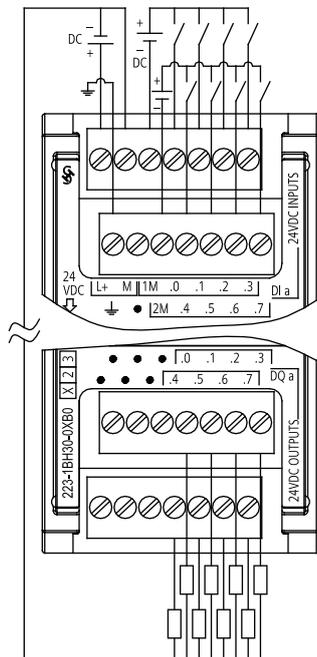
SM1223 DI 16 x 24 V DC, DQ 16 x relé



6ES7 223-1PL30-0XB0

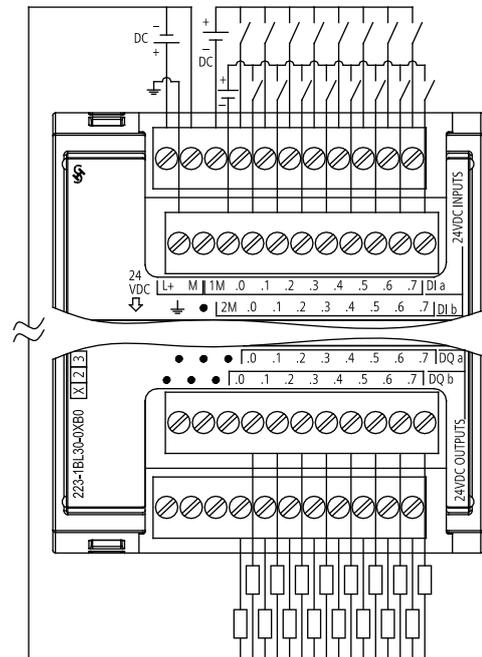
A.4 Módulos de señales analógicas (SMs)

SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC



6ES7 223-1BH30-0XB0

SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC



6ES7 223-1BL30-0XB0

A.4 Módulos de señales analógicas (SMs)

A.4.1 Datos técnicos de los módulos de señales analógicas SM 1231, SM 1232, SM 1234

Datos técnicos			
Modelo	SM 1231 AI 4x13bit	SM 1231 AI 8x13bit	SM 1234 AI 4x13bit AQ 2x14bit
Referencia	6ES7 231-4HD30-0XB0	6ES7 231-4HF30-0XB0	6ES7 234-4HE30-0XB0
General			
Dimensiones A x A x P (mm)	45 x 100 x 75	45 x 100 x 75	45 x 100 x 75
Peso	180 gramos	180 gramos	220 gramos
Disipación de potencia	1,5 W	1,5 W	2,0 W
Consumo de corriente (bus SM)	80 mA	90 mA	80 mA
Consumo de corriente (24 V DC)	45 mA	45 mA	60 mA (sin carga)
Entradas analógicas			
Número de entradas	4	8	4
Tipo	Tensión o intensidad (diferencial): Seleccionable en grupos de 2		
Rango	±10 V, ±5 V, ±2,5 V ó 0 a 20 mA		

Datos técnicos			
Modelo	SM 1231 AI 4x13bit	SM 1231 AI 8x13bit	SM 1234 AI 4x13bit AQ 2x14bit
Rango total (palabra de datos)	-27.648 a 27.648		
Rango de sobreimpulso/subimpulso (palabra de datos)	Tensión: 32.511 a 27.649 / -27.649 a -32.512 Intensidad: 32.511 a 27.649 / 0 a -4864 (Consulte Representación de entradas analógicas para tensión, representación de entradas analógicas para intensidad (Página 346))		
Rebase por exceso/defecto (palabra de datos)	Tensión: 32.767 a 32.512 / -32.513 a -32.768 Intensidad: 32.767 a 32.512 / -4865 a -32.768 (Consulte Representación de entradas analógicas para tensión, representación de entradas analógicas para intensidad (Página 346))		
Resolución	12 bits + bit de signo		
Tensión/intensidad de resistencia al choque máxima	±35 V / ±40 mA		
Alisamiento	Ninguno, débil, medio o fuerte (consulte los tiempos de respuesta de paso en Tiempos de respuesta de las entradas analógicas (Página 346))		
Rechazo de interferencias	400, 60, 50 ó 10 Hz (consulte las frecuencias de muestreo en Tiempos de respuesta de las entradas analógicas (Página 346))		
Impedancia	≥ 9 MΩ (tensión) / 250 Ω (intensidad)		
Aislamiento (campo a lógica)	Ninguno		
Precisión (25°C / 0 a 55°C)	±0,1% / ±0,2% de rango máximo		
Tiempo de conversión analógica/digital	625 μs (rechazo de 400 Hz)		
Rechazo en modo común	40 dB, DC a 60 Hz		
Rango de señales operativo	La tensión de señal más la tensión en modo común debe ser menor que +12 V y mayor que -12 V		
Longitud de cable (metros)	100 metros, trenzado y apantallado		
Diagnóstico			
Rebase por exceso/defecto	Sí ¹	Sí ¹	Sí ¹
Cortocircuito a tierra (sólo en modo de tensión)	No aplicable	No aplicable	Sí en las salidas
Rotura de hilo (sólo en modo de intensidad)	No aplicable	No aplicable	Sí en las salidas
24 V DC, baja tensión	Sí	Sí	Sí

¹ Si se aplica una tensión superior a +30 V DC o inferior a -15 V DC a la entrada, el valor resultante se desconocerá y es posible que no se active el rebase por exceso o por defecto correspondiente.

Datos técnicos			
Modelo	SM 1232 AQ 2x14bit	SM 1232 AQ 4x14bit	SM 1234 AI 4x13bit AQ 2x14bit
Referencia	6ES7 232-4HB30-0XB0	6ES7 232-4HD30-0XB0	6ES7 234-4HE30-0XB0
General			
Dimensiones A x A x P (mm)	45 x 100 x 75	45 x 100 x 75	45 x 100 x 75
Peso	180 gramos	180 gramos	220 gramos
Disipación de potencia	1,5 W	1,5 W	2,0 W

Datos técnicos			
Modelo	SM 1232 AQ 2x14bit	SM 1232 AQ 4x14bit	SM 1234 AI 4x13bit AQ 2x14bit
Consumo de corriente (bus SM)	80 mA	80 mA	80 mA
Consumo de corriente (24 V DC)	45 mA (sin carga)	45 mA (sin carga)	60 mA (sin carga)
Salidas analógicas			
Número de salidas	2	4	2
Tipo	Tensión o intensidad		
Rango	±10 V ó 0 a 20 mA		
Resolución	Tensión: 14 bits; intensidad: 13 bits		
Rango total (palabra de datos)	Tensión: -27.648 a 27.648; intensidad: 0 a 27.648 (Consulte Representación de entradas analógicas para tensión y representación de entradas analógicas para intensidad) (Página 346)		
Precisión (25°C / 0 a 55°C)	±0,3% / ±0,6% de rango máximo		
Tiempo de estabilización (95% del nuevo valor)	Tensión: 300 µS (R), 750 µS (1 uF); intensidad: 600 µS (1 mH), 2 ms (10 mH)		
Impedancia de carga	Tensión: ≥ 1000 Ω; intensidad: ≤ 600 Ω		
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)		
Aislamiento (campo a lógica)	Ninguna		
Longitud de cable (metros)	100 metros, trenzado y apantallado		
Diagnóstico			
Rebase por exceso/defecto	Sí	Sí	Sí ¹
Cortocircuito a tierra (sólo en modo de tensión)	Sí	Sí	Sí en las salidas
Rotura de hilo (sólo en modo de intensidad)	Sí	Sí	Sí en las salidas
24 V DC, baja tensión	Sí	Sí	Sí

¹ Si se aplica una tensión superior a +30 V DC o inferior a -15 V DC a la entrada, el valor resultante se desconocerá y es posible que no se active el rebase por exceso o por defecto correspondiente.

Tiempos de respuesta de las entradas analógicas

Respuesta de paso de los módulos analógicos SM (en ms)				
0V a 10V medido a 95%				
Selección de alisamiento	Frecuencia de rechazo			
	400 Hz	60 Hz	50 Hz	10 Hz
Ninguno	4	18	22	100
Débil	9	52	63	320
Medio	32	203	241	1200
Fuerte	61	400	483	2410
Frecuencia de muestreo				
• 4 canales	• 0.625	• 4.17	• 5	• 25
• 8 canales	• 1.25	• 4.17	• 5	• 25

Respuesta de paso de entradas analógicas de la CPU (en ms)			
0V a 10V medido a 95%			
Selección de alisamiento	Frecuencia de rechazo		
	60 Hz	50 Hz	10 Hz
Ninguno	63	65	130
Débil	84	93	340
Medio	221	258	1210
Fuerte	424	499	2410
Frecuencia de muestreo	4.17	5	25

Representación de entradas analógicas para tensión

Sistema	Rango de medida de tensión						
Decimal	Hexadecimal	±10 V	±5 V	±2,5 V		0 a 10 V	
32767	7FFF	11,851 V	5,926 V	2,963 V	Rebase por exceso	11,851V	Rebase por exceso
32512	7F00						
32511	7EFF	11,759 V	5,879 V	2,940 V	Rango de sobreimpulso	11,759 V	Rango de sobreimpulso
27649	6C01						
27648	6C00	10 V	5 V	2,5 V	Rango nominal	10 V	Rango nominal
20736	5100	7,5 V	3,75 V	1,875 V		7,5 V	
1	1	361,7 µV	180,8 µV	90,4 µV		361,7 µV	
0	0	0 V	0 V	0 V		0 V	
-1	FFFF					Los valores negativos no se soportan	
-20736	AF00	-7,5 V	-3,75 V	-1,875 V			
-27648	9400	-10 V	-5 V	-2,5 V			
-27649	93FF						
-32512	8100	-11,759 V	-5,879 V	-2,940 V	Rango de subimpulso		
-32513	80FF				Rebase por defecto		
-32768	8000	-11,851 V	-5,926 V	-2,963 V			

Representación de entradas analógicas para intensidad

Sistema	Rango de medida de intensidad		
Decimal	Hexadecimal	0 mA a 20 mA	
32767	7FFF	23,70 mA	Rebase por exceso
32512	7F00		
32511	7EFF	23,52 mA	Rango de sobreimpulso
27649	6C01		
27648	6C00	20 mA	Rango nominal
20736	5100	15 mA	
1	1	723,4 nA	

Sistema		Rango de medida de intensidad	
Decimal	Hexadecimal	0 mA a 20 mA	
0	0	0 mA	
-1	FFFF		Rango de subimpulso
-4864	ED00	-3,52 mA	
-4865	ECFF		Rebase por defecto
-32768	8000		

Representación de salidas analógicas para tensión

Sistema		Rango de salida de tensión	
Decimal	Hexadecimal	± 10 V	
32767	7FFF	V. nota 1	Rebase por exceso
32512	7F00	V. nota 1	
32511	7EFF	11,76 V	Rango de sobreimpulso
27649	6C01		
27648	6C00	10 V	Rango nominal
20736	5100	7,5 V	
1	1	361,7 μ V	
0	0	0 V	
-1	FFFF	-361,7 μ V	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		Rango de subimpulso
-32512	8100	-11,76 V	
-32513	80FF	V. nota 1	Rebase por defecto
-32768	8000	V. nota 1	

¹ . En una condición de rebase por exceso o por defecto, la reacción de las salidas analógicas corresponderá a las propiedades ajustadas en la configuración de dispositivos para el módulo de señales analógico. En el parámetro "Reacción a STOP de la CPU", seleccione: "Aplicar valor sustitutivo" o "Mantener último valor".

Representación de salidas analógicas para intensidad

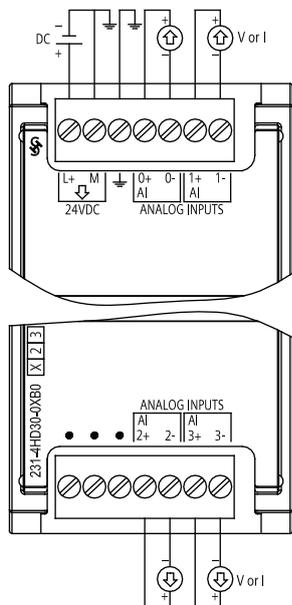
Sistema		Rango de salida de intensidad	
Decimal	Hexadecimal	± 20 mA	
32767	7FFF	V. nota 1	Rebase por exceso
32512	7F00	V. nota 1	
32511	7EFF	23,52 mA	Rango de sobreimpulso
27649	6C01		
27648	6C00	20 mA	Rango nominal
20736	5100	15 mA	
1	1	723,4 nA	

Sistema		Rango de salida de intensidad	
Decimal	Hexadecimal	± 20 mA	
0	0	0 mA	
-1	FFFF		Rango de subimpulso
-32512	8100		
-32513	80FF	V. nota 1	Rebase por defecto
-32768	8000	V. nota 1	

1. En una condición de rebase por exceso o por defecto, la reacción de las salidas analógicas corresponderá a las propiedades ajustadas en la configuración de dispositivos para el módulo de señales analógico. En el parámetro "Reacción a STOP de la CPU", seleccione: "Aplicar valor sustitutivo" o "Mantener último valor".

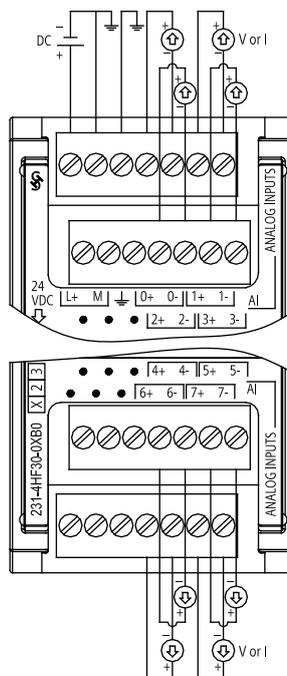
Diagramas de cableado

SM 1231 AI 4 x 13 bit



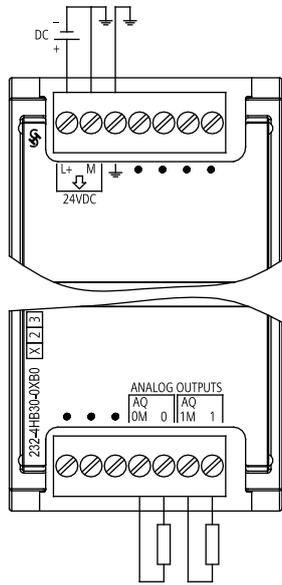
6ES7 231-4HD30-0XB0

SM 1231 AI 8 x 13 bit



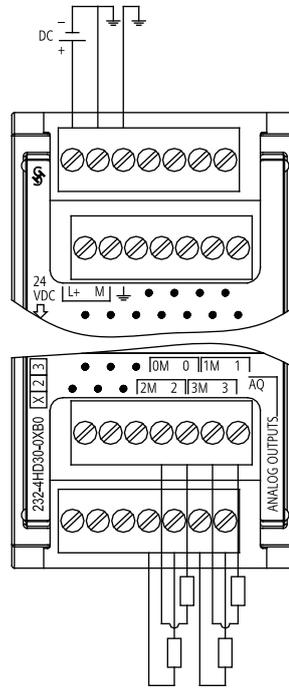
6ES7 231-4HF30-0XB0

SM 1232 AQ 2 x 14 bit



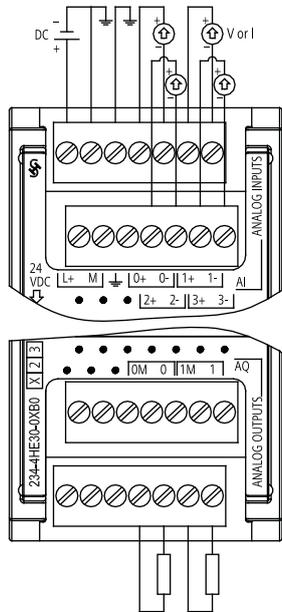
6ES7 232-4HB30-0XB0

SM 1232 AQ 4 x 14 bit



6ES7 232-4HD30-0XB0

SM 1234 AI 4 x 13 bit / AQ 2 x 14 bit



6ES7 234-4HE30-0XB0

A.5 Signal Boards (SBs)

A.5.1 Datos técnicos de la SB 1223 2 entradas x 24 V DC / 2 salidas x 24 V DC

Datos técnicos de la Signal Board digital

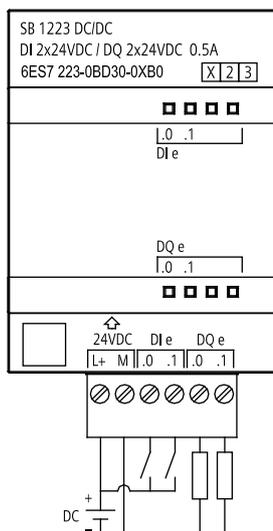
Datos técnicos	
Modelo	SB 1223 DI 2x24 V DC, DQ 2x24 V DC
Referencia	6ES7 223-0BD30-0XB0
General	
Dimensiones A x A x P (mm)	38 x 62 x 21
Peso	40 gramos
Disipación de potencia	1,0 W
Consumo de corriente (bus SM)	50 mA
Consumo de corriente (24 V DC)	4 mA/entrada utilizada
Entradas digitales	
Número de entradas	2
Tipo	Tipo 1 IEC sumidero
Tensión nominal	24 V DC a 4 mA, nominal
Tensión continua admisible	30 V DC, máx.
Sobretensión transitoria	35 V DC durante 0,5 seg.
Señal 1 lógica (mín.)	15 V DC a 2,5 mA
Señal 0 lógica (máx.)	5 V DC a 1 mA
Frecuencias de entrada de reloj HSC (máx.)	20 kHz (15 a 30 V DC) 30 kHz (15 a 26 V DC)
Aislamiento (campo a lógica)	500 V AC durante 1 minuto
Grupos de aislamiento	1
Tiempos de filtro	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms Seleccionable en grupos de 2
Número de entradas ON simultáneamente	2
Longitud de cable (metros)	500 apantallado, 300 no apantallado
Salidas digitales	
Número de salidas	2
Tipo de salida	Estado sólido - MOSFET
Rango de tensión	20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	20 V DC mín.
Señal 0 lógica con carga de 10K Ω	0,1 V DC máx.
Intensidad (máx.)	0,5 A
Carga de lámparas	5 W
Resistencia en estado ON (contactos)	0,6 Ω máx.
Corriente de fuga por salida	10 μ A máx.
Frecuencia de tren de impulsos	20 KHz máx., 2 Hz mín.

Datos técnicos

A.5 Signal Boards (SBs)

Datos técnicos	
Modelo	SB 1223 DI 2x24 V DC, DQ 2x24 V DC
Sobrecorriente momentánea	5 A durante máx. 100 ms
Protección contra sobrecargas	No
Aislamiento (campo a lógica)	500 V AC durante 1 minuto
Grupos de aislamiento	1
Intensidad por neutro	1 A
Tensión de bloqueo inductiva	L+ menos 48 V, disipación de 1 W
Retardo de conmutación	2 µs máx. OFF a ON 10 µs máx. ON a OFF
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)
Número de salidas ON simultáneamente	2
Longitud de cable (metros)	500 apantallado, 150 no apantallado

Diagrama de cableado de la SB 1223 2 entradas x 24 V DC / 2 salidas x 24 V DC

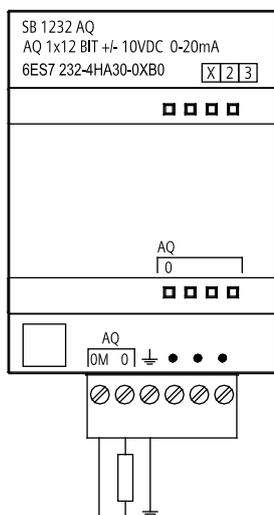


A.5.2 Datos técnicos de la SB 1232 de 1 salida analógica

Datos técnicos de la Signal Board analógica

Datos técnicos	
Modelo	SB 1223 AQ 1x12bit
Referencia	6ES7 232-4HA30-0XB0
General	
Dimensiones A x A x P (mm)	38 x 62 x 21 mm
Peso	40 gramos
Disipación de potencia	1,5 W
Consumo de corriente (bus SM)	15 mA
Consumo de corriente (24 V DC)	40 mA (sin carga)
Salidas analógicas	
Número de salidas	1
Tipo	Tensión o intensidad
Rango	±10 V ó 0 a 20 mA
Resolución	Tensión: 12 bits Intensidad: 11 bits
Rango total (palabra de datos)	Tensión: -27.648 a 27.648 Intensidad: 0 a 27.648
Precisión (25°C / 0 a 55°C)	±0.5% / ±1% de rango máximo
Tiempo de estabilización (95% del nuevo valor)	Tensión: 300 µS (R), 750 µS (1 uF) Intensidad: 600 µS (1 mH), 2 ms (10 mH)
Impedancia de carga	Tensión: ≥ 1000 Ω Intensidad: ≤ 600 Ω
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)
Aislamiento (campo a lógica)	Ninguno
Longitud de cable (metros)	10 metros, trenzado y apantallado
Diagnóstico	
Rebase por exceso/defecto	Sí
Cortocircuito a tierra (sólo en modo de tensión)	Sí
Rotura de hilo (sólo en modo de intensidad)	Sí

Diagrama de cableado de la SB 1232 de 1 salida analógica

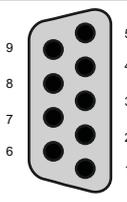


A.6 Módulos de comunicación (CMs)

A.6.1 Datos técnicos del CM 1241 RS485

Tabla A- 1 Módulo de comunicación CM 1241 RS485

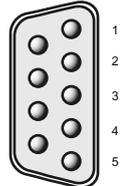
Datos técnicos	
Referencia	6ES7 241-1CH30-0XB0
Dimensiones y peso	
Dimensiones	30 x 100 x 75 mm
Peso	150 gramos
Transmisor y receptor	
Rango de tensión en modo común	-7 V a +12 V, 1 segundo, 3 VRMS continuo
Tensión de salida diferencial del transmisor	2 V mín. a $R_L = 100 \Omega$ 1,5 V mín. a $R_L = 54 \Omega$
Terminación y polarización	10K Ω a +5 V en B, pin PROFIBUS 3 10K Ω a GND en A, pin PROFIBUS 8
Impedancia de entrada del receptor	5,4K Ω mín. incluyendo terminación
Umbral/sensibilidad del receptor	+/- 0,2 V mín., 60 mV de histéresis típica
Aislamiento Señal RS485 a conexión a masa Señal RS485 a lógica de la CPU	500 V AC durante 1 minuto
Longitud de cable, apantallado	1000 m máx.
Fuente de alimentación	
Pérdida de potencia (disipación)	1,1 W
De +5 V DC	220 mA

Pin	Descripción	Conector (hembra)	Pin	Descripción
1 GND	Masa lógica o de comunicación		6 PWR	+5V con resistor en serie de 100 ohmios: Salida
2	Sin conexión		7	Sin conexión
3 TxD+	Señal B (RxD/TxD+): Entrada/salida		8 TXD-	Señal A (RxD/TxD-): Entrada/salida
4 RTS	Petición de transmitir (nivel TTL): Salida		9	Sin conexión
5 GND	Masa lógica o de comunicación		SHELL	Conexión a masa

A.6.2 Datos técnicos del CM 1241 RS232

Módulo de comunicación CM 1241 RS232

Datos técnicos	
Referencia	6ES7 241-1AH30-0XB0
Dimensiones y peso	
Dimensiones	30 x 100 x 75 mm
Peso	150 gramos
Transmisor y receptor	
Tensión de salida del transmisor	+/- 5 V mín. a $R_L = 3K \Omega$
Tensión de salida del transmisor	+/- 15 V DC máx.
Impedancia de entrada del receptor	3 K Ω mín.
Umbral/sensibilidad del receptor	0,8 V mín. bajo, 2,4 máx. alto histéresis típica: 0,5 V
Tensión de entrada del receptor	+/- 30 V DC máx.
Aislamiento Señal RS 232 a conexión a masa Señal RS 232 a lógica de la CPU	500 V AC durante 1 minuto
Longitud de cable, apantallado	10 m máx.
Fuente de alimentación	
Pérdida de potencia (disipación)	1,1 W
De +5 V DC	220 mA

Pin	Descripción	Conector (macho)	Pin	Descripción
1 DCD	Detección de portadora de datos: Entrada		6 DSR	Equipo de datos listo: Entrada
2 RxD	Datos recibidos de DCE: Entrada		7 RTS	Petición de transmitir Salida
3 TxD	Datos transmitidos a DCE: Salida		8 CTS	Listo para transmitir: Entrada
4 DTR	Terminal de datos disponible: Salida		9 RI	Indicación de timbre (no utilizado)
5 GND	Masa lógica		SHELL	Conexión a masa

A.7 SIMATIC Memory Cards

Datos técnicos de la Memory Card

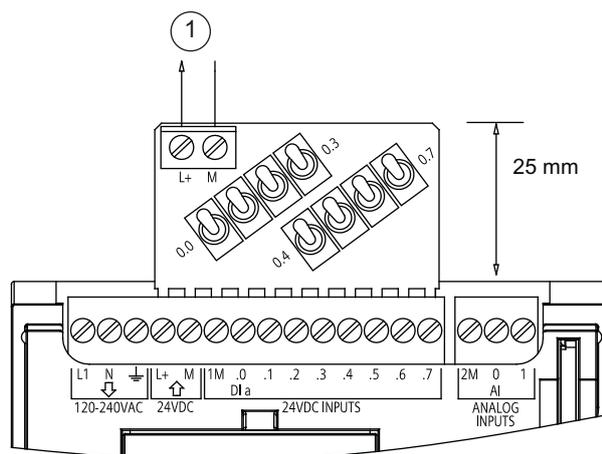
Referencia	Capacidad
6ES7 954-8LF00-0AA0	24 MB
6ES7 954-8LB00-0AA0	2 MB

A.8 Simuladores de entradas

Modelo	Simulador de 8 entradas	Simulador de 14 entradas
Referencia	6ES7 274-1XF30-0XA0	6ES7 274-1XH30-0XA0
Dimensiones A x A x P (mm)	43 x 35 x 23	67 x 35 x 23
Peso	20 gramos	30 gramos
Entradas	8	14
Uso con CPU	CPU 1211C, CPU 1212C	CPU 1214C

 ADVERTENCIA
<p>Estos simuladores de entradas no están aprobados para ser utilizados en ubicaciones peligrosas ("hazardous locations") Class I DIV 2 o Class I Zone 2. Los interruptores pueden producir chispas o explotar si se utilizan en ubicaciones Class I DIV 2 o Class I Zone 2.</p>

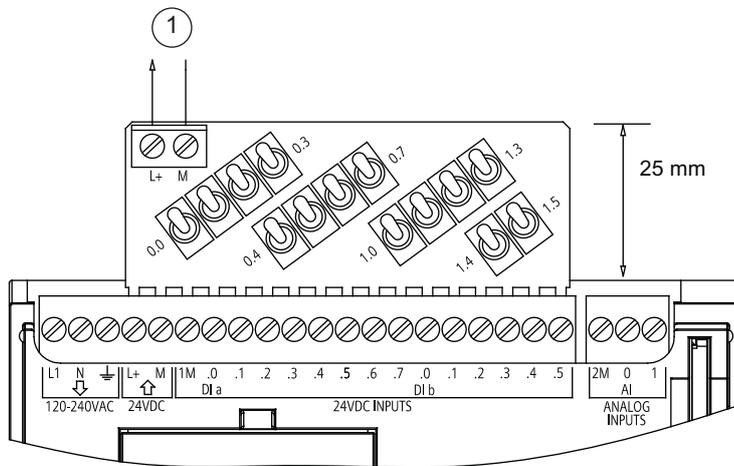
Simulador de 8 entradas



① Alimentación de sensores de 24 V DC

6ES7 274-1XF30-0XA0

Simulador de 14 entradas



① Alimentación de sensores de 24 V DC

6ES7 274-1XH30-0XA0

A.9 Cable para módulos de ampliación

Datos técnicos	
Referencia	6ES7 290-6AA30-0XA0
Longitud del cable	2 m
Peso	200 g

El cable para módulos de ampliación tiene un conector macho y uno hembra.

1. Conecte el conector macho al conector de bus en el lado derecho del módulo de señales.
2. Conecte el conector hembra al conector de bus en el lado izquierdo del módulo de señales.
 - Inserte la extensión con gancho del conector hembra en la carcasa en el conector de bus
 - Empuje el conector hembra hacia dentro del conector de bus.

Calcular la corriente necesaria

La CPU incorpora una fuente de alimentación interna capaz de abastecer la CPU, los módulos de ampliación y otros consumidores de 24 V DC.

Hay tres tipos de módulos de ampliación, a saber:

- Los módulos de señales (SM) se montan a la derecha de la CPU. Toda CPU permite conectar un número máximo posible de módulos de señales, sin considerar la corriente disponible.
 - La CPU 1214 permite conectar 8 módulos de señales
 - La CPU 1212 permite conectar 2 módulos de señales
 - La CPU 1211 no permite conectar módulos de señales
- Los módulos de comunicación (CM) se montan a la izquierda de la CPU. Se permiten como máximo 3 módulos de comunicación para cualquier CPU, sin considerar la corriente disponible.
- Las Signal Boards (SB) se montan en el lado superior de la CPU. Se permite como máximo 1 Signal Board para cualquier CPU.

Utilice la información siguiente como guía para determinar cuánta energía (o corriente) puede suministrar la CPU a la configuración.

Toda CPU suministra alimentación tanto de 5 V DC como de 24 V DC:

- La CPU suministra 5 V DC a los módulos de ampliación cuando son conectados. Si el consumo de 5 V DC de los módulos de ampliación excede la corriente que ofrece la CPU, habrá que desconectar tantos módulos de ampliación como sea necesario para no excederla.
- Toda CPU incorpora una fuente de alimentación de sensores de 24 V DC que puede suministrar 24 V DC a las entradas locales, o bien a las bobinas de relé de los módulos de ampliación. Si el consumo de 24 V DC excede la corriente disponible de la CPU, es posible agregar una fuente de alimentación externa de 24 V DC para suministrar 24 V DC a los módulos de ampliación. La alimentación de 24 V DC debe conectarse manualmente a las entradas o bobinas de relé.

ADVERTENCIA

Si se conecta una fuente de alimentación externa de 24 V DC en paralelo con la fuente de alimentación DC de sensores, podría surgir un conflicto entre ambas fuentes, ya que cada una intenta establecer su propio nivel de tensión de salida.

Este conflicto puede reducir la vida útil u ocasionar la avería inmediata de una o ambas fuentes de alimentación y, en consecuencia, el funcionamiento imprevisible del sistema PLC. El funcionamiento imprevisible puede producir la muerte, lesiones corporales graves y/o daños materiales.

La fuente de alimentación DC de sensores de la CPU y cualquier fuente de alimentación externa deben alimentar diferentes puntos. Se permite una conexión común de los cables neutros.

Algunos puertos de entrada de alimentación de 24 V del sistema PLC están interconectados y tienen un circuito lógico común que conecta varios bornes M. La fuente de alimentación de 24V de la CPU, la entrada de alimentación de las bobinas de relé de un SM, o bien una fuente de alimentación analógica sin aislamiento galvánico son ejemplos de circuitos interconectados si no tienen aislamiento galvánico según las hojas de datos técnicos. Todos los bornes M sin aislamiento galvánico deben conectarse al mismo potencial de referencia externo.

 **ADVERTENCIA**

Si los bornes M sin aislamiento galvánico se conectan a diferentes potenciales de referencia, circularán corrientes indeseadas que podrían averiar o causar reacciones inesperadas en el PLC y los equipos conectados.

Las averías o reacciones inesperadas podrían producir la muerte, lesiones corporales graves y/o daños materiales.

Asegúrese que todos los bornes M sin aislamiento galvánico de un sistema PLC están conectados al mismo potencial de referencia.

Para más información sobre la corriente disponible de las distintas CPUs y el consumo de corriente de los módulos de señales, consulte los datos técnicos (Página 319).

Nota

Si se excede la corriente disponible de la CPU, es posible que no se pueda conectar el número máximo de módulos soportado.

B.2 Ejemplo de cálculo del consumo de corriente

El ejemplo siguiente muestra el cálculo del consumo de corriente de un PLC que incluye una CPU 1214C AC/DC/relé, 3 módulos de señales SM 1223 de 8 entradas DC y 8 salidas de relé, así como un módulo de señales SM 1221 de 8 entradas DC. Este ejemplo incluye 46 entradas y 34 salidas en total.

Nota

La CPU ya ha asignado la corriente necesaria para accionar las bobinas de relé internas. Por tanto, no es necesario incluir el consumo de corriente de las bobinas de relé internas en el cálculo.

La CPU de este ejemplo suministra suficiente corriente de 5 V DC a los SMs, pero la alimentación de sensores no suministra suficiente corriente de 24 V DC a todas las entradas y bobinas de relé de ampliación. Las E/S requieren 448 mA, pero la CPU sólo puede suministrar 400 mA. Para esta configuración se necesita una fuente de alimentación adicional de 48 mA (como mínimo) a 24 V DC para operar todas las entradas y salidas de 24 V DC.

Corriente disponible de la CPU	5 V DC	24 V DC
CPU 1214C AC/DC/relé	1600 mA	400 mA
<i>Menos</i>		
Consumo del sistema	5 V DC	24 V DC
CPU 1214C, 14 entradas	-	14 * 4 mA = 56 mA
3 SM 1223, 5 V de corriente	3 * 145 mA = 435 mA	-
1 SM 1221, 5 V de corriente	1 * 105 mA = 105 mA	-
3 SM 1223, 8 entradas c/u	-	3 * 8 * 4 mA = 96 mA
3 SM 1223, 8 salidas de relé c/u	-	3 * 8 * 11 mA = 264 mA
1 SM 1221, 8 entradas	-	8 * 4 mA = 32 mA
Consumo total	540 mA	448 mA
<i>Igual a</i>		
Balance de corriente	5 V DC	24 V DC
Balance total de corriente	1060 mA	(48 mA)

Referencias

CPUs		Referencia
CPU 1211C	CPU 1211C DC/DC/DC	6ES7 211-1AD30-0XB0
	CPU 1211C AC/DC/relé	6ES7 211-1BD30-0XB0
	CPU 1211C DC/DC/relé	6ES7 211-1HD30-0XB0
CPU 1212C	CPU 1212C DC/DC/DC	6ES7 212-1AD30-0XB0
	CPU 1212C AC/DC/relé	6ES7 212-1BD30-0XB0
	CPU 1212C DC/DC/relé	6ES7 212-1HD30-0XB0
CPU 1214C	CPU 1214C DC/DC/DC	6ES7 214-1AE30-0XB0
	CPU 1214C AC/DC/relé	6ES7 214-1BE30-0XB0
	CPU 1214C DC/DC/relé	6ES7 214-1HE30-0XB0

Módulos de señales, módulos de comunicación y Signal Boards		Referencia
Módulos de señales	SM 1221 8 entradas x 24 V DC	6ES7 221-1BF30-0XB0
	SM 1221 16 entradas x 24 V DC	6ES7 221-1BH30-0XB0
	SM 1222 8 salidas x 24 V DC	6ES7 222-1BF30-0XB0
	SM 1222 16 salidas x 24 V DC	6ES7 222-1BH30-0XB0
	SM 1222 8 salidas de relé	6ES7 222-1HF30-0XB0
	SM 1222 16 salidas de relé	6ES7 222-1HH30-0XB0
	SM 1223 8 entradas x 24 V DC / 8 salidas x 24 V DC	6ES7 223-1BH30-0XB0
	SM 1223 16 entradas x 24 V DC / 16 salidas x 24 V DC	6ES7 223-1BL30-0XB0
	SM 1223 8 entradas x 24 V DC / 8 salidas de relé	6ES7 223-1PH30-0XB0
	SM 1223 16 entradas x 24 V DC / 16 salidas de relé	6ES7 223-1PL30-0XB0
	SM 1231 4 entradas analógicas	6ES7 231-4HD30-0XB0
	SM 1231 8 entradas analógicas	6ES7 231-4HF30-0XB0
	SM 1232 2 salidas analógicas	6ES7 232-4HB30-0XB0
	SM 1232 4 salidas analógicas	6ES7 232-4HD30-0XB0
	SM 1234 4 entradas analógicas / 2 salidas analógicas	6ES7 234-4HE30-0XB0
Módulos de comunicación	CM 1241 RS232	6ES7 241-1AH30-0XB0
	CM 1241 RS485	6ES7 241-1CH30-0XB0
Signal Boards	SB 1223 2 entradas x 24 V DC / 2 salidas x 24 V DC	6ES7 223-0BD30-0XB0
	SB 1232 de 1 salida analógica	6ES7 232-4HA30-0XB0

Dispositivos HMI	Referencia
KTP400 Basic (Mono, PN)	6AV6 647-0AA11-3AX0
KTP600 Basic (Mono, PN)	6AV6 647-0AB11-3AX0
KTP600 Basic (Color, PN)	6AV6 647-0AD11-3AX0
KTP1000 Basic (Color, PN)	6AV6 647-0AF11-3AX0
TP1500 Basic (Color, PN)	6AV6 647-0AG11-3AX0

Paquete de programación	Referencia
STEP 7 Basic v10.5	6ES7 822-0AA0-0YA0

Memory Cards, otros dispositivos de hardware y repuestos		Referencia
Memory Cards	SIMATIC MC 2 MB	6ES7 954-8LB00-0AA0
	SIMATIC MC 24 MB	6ES7 954-8LF00-0AA0
Otros dispositivos de hardware	Fuente de alimentación PSU 1200	6EP1 332-1SH71
	Switch Ethernet CSM 1277 - 4 puertos	6GK7 277-1AA00-0AA0
	Simulador (1214C/1211C - 8 entradas)	6ES7 274-1XF30-0XA0
	Simulador (1214C - 14 entradas)	6ES7 274-1XH30-0XA0
	Cable para módulos de ampliación, 2 m	6ES7 290-6AA30-0XA0
Repuestos	Bloque de conectores, 7 terminales, estañado	6ES7 292-1AG30-0XA0
	Bloque de conectores, 8 terminales, estañado (4/pq)	6ES7 292-1AH30-0XA0
	Bloque de conectores, 11 terminales, estañado (4/pq)	6ES7 292-1AL30-0XA0
	Bloque de conectores, 12 terminales, estañado (4/pq)	6ES7 292-1AM30-0XA0
	Bloque de conectores, 14 terminales, estañado (4/pq)	6ES7 292-1AP30-0XA0
	Bloque de conectores, 20 terminales, estañado (4/pq)	6ES7 292-1AV30-0XA0
	Bloque de conectores, 3 terminales, dorado (4/pq)	6ES7 292-1BC0-0XA0
	Bloque de conectores, 6 terminales, dorado (4/pq)	6ES7 292-1BF30-0XA0
	Bloque de conectores, 7 terminales, dorado (4/pq)	6ES7 292-1BG30-0XA0
Bloque de conectores, 11 terminales, dorado (4/pq)	6ES7 292-1BL30-0XA0	

Documentación	Referencia
Manual de sistema del controlador programable S7-1200 <ul style="list-style-type: none">• Alemán• Inglés• Francés• Español• Italiano• Chino	<ul style="list-style-type: none">• 6ES7 298-8FA30-8AH0• 6ES7 298-8FA30-8BH0• 6ES7 298-8FA30-8CH0• 6ES7 298-8FA30-8DH0• 6ES7 298-8FA30-8EH0• 6ES7 298-8FA30-8FH0
S7-1200 Easy Book <ul style="list-style-type: none">• Alemán• Inglés• Francés• Español• Italiano• Chino	<ul style="list-style-type: none">• 6ES7 298-8FA30-8AQ0• 6ES7 298-8FA30-8BQ0• 6ES7 298-8FA30-8CQ0• 6ES7 298-8FA30-8DQ0• 6ES7 298-8FA30-8EQ0• 6ES7 298-8FA30-8FQ0

Índice alfabético

A

AC

Cargas inductivas, 39

Acceso a la ayuda en pantalla, 17

Agregar un dispositivo

CPU sin especificar, 79

Alarmas

Sinopsis, 47

Ampliar la ventana de ayuda en pantalla, 18

Ampliar las prestaciones del S7-1200, 13

Aprobación C-Tick, 321

Aprobación marina, 321

Arquitectura de sondeo, 281

Arquitectura de sondeo del esclavo, 282

Arquitectura de sondeo del maestro, 281

Ayuda, 17

Ampliar, 18

Desacoplar, 18

Imprimir, 20

Mostrar el contenido e índice, 18

Ayuda contextual, 17

Ayuda desplegable, 17

Ayuda en pantalla, 17

Ampliar la ventana de ayuda, 18

Desacoplar, 18

Imprimir, 20

Mostrar el contenido e índice, 18

B

Bits de parada, 272

Bloque

Tipos, 41

Bloque de administración de datos (DHB), 95

Bloque de datos

Bloque de datos global, 60, 95

Bloque de datos instancia, 60

Bloque de organización (OB), 92

Bloque de datos (DB), 95

Bloque de datos global, 60, 95

Bloque de datos instancia, 60

Bloque de función (FB)

Bloque de datos instancia, 94

Parámetros de salida, 94

Valor inicial, 94

Bloque de organización

Clases de prioridad, 47

Configurar el funcionamiento, 93

Crear, 93

De ciclo, varios, 93

Función, 47

Llamada, 47

Procesar, 92

Bloque de transferencia (bloque T), 257

Bloque lógico

DB (bloque de datos), 95

FB (bloque de función), 94

FC (función), 93

Protección de know-how, 99

Bloques

Bloques de datos (DBs), 41

Bloques de función (FBs), 41

Bloques de organización (OBs), 41, 47

Funciones (FCs), 41

Bloques lógicos, 90

Búfer de diagnóstico, 56, 312

C

Carácter de fin del mensaje, 278

Carácter de inicio del mensaje, 276

Cargar en dispositivo, 252

Cargas de lámpara, 39

Cargas inductivas, 39

Clase de prioridad

Sinopsis, 47

Clase de protección, 323

Códigos de estado USS, 216

Colas de espera, 48

Compatibilidad electromagnética (CEM), 322

Comprobar el programa, 102

Comunicación

Arquitectura de sondeo, 281

Carga, 54

Conexión de hardware, 243

Control de flujo, 272

Dirección IP, 84, 249

Librerías, 269

Parámetros de transmisión y recepción, 274

Red, 242

Comunicación de red, 242

Comunicación Ethernet, 241

Comunicación PtP, 269

- Comunicación punto a punto, 269
- Comunicación serie, 269
- Comunicación TCP/IP, 241
- Condiciones ambientales, 323
- Condiciones de fin, 278
- Condiciones de inicio, 276
- Conector
 - Montaje y desmontaje, 35
- Conector de bus, 14
- Conector del bloque de terminales
 - Montaje, 35
- Conexión de red, 83
 - Varias CPUs, 257
- Configuración
 - Comunicación entre PLCs, 256
 - Determinar, 79
 - Dirección IP, 84, 249
 - HMI a CPU, 255
 - HSC (contador rápido), 124
 - Interfaces de comunicación, 271
 - Parámetros de arranque, 43, 71
 - PROFINET, 84, 249
 - Puerto Industrial Ethernet, 84, 249
 - Puertos, 271
 - Recepción de mensajes, 275
 - Tiempo de ciclo, 54
- Configuración de dispositivos, 77, 243
 - Agregar módulos, 81
 - Agregar nuevo dispositivo, 78
 - Conexión de red, 83
 - Configurar la CPU, 80
 - Configurar los módulos, 82
 - Determinar, 79
 - PROFINET, 84, 249
 - Puerto Ethernet, 84, 249
- Configuración de hardware, 77
 - Agregar módulos, 81
 - Agregar nuevo dispositivo, 78
 - Conexión de red, 83
- Configuración de la instrucción TRCV_C, 263
- Configuración de la instrucción TSEND_C, 259
- Configuración de la recepción de mensajes, 275
- Configuración de la transferencia de mensajes, 274
- Configuración de los parámetros de recepción, 263
- Configuración de mensajes
 - Instrucciones, 280
 - Recibir, 275
 - Transmitir, 274
- Configuración de parámetros
 - Recibir, 263
 - Transmitir, 259
- Configuración de parámetros de transmisión, 259
- Configuración del envío de mensajes, 274
- Configuración del puerto, 271
 - Instrucciones, 280
- Configuración hardware
 - Configurar la CPU, 80
 - Configurar los módulos, 82
 - Determinar, 79
 - PROFINET, 84, 249
 - Puerto Ethernet, 84, 249
- Configurar parámetros
 - CPU, 80
 - Módulos, 82
 - PROFINET, 84, 249
 - Puerto Ethernet, 84, 249
- Consumo de corriente
 - Calcular, 363, 364
- Contador rápido, 121
- Contadores
 - Rápidos (HSC), 121
 - Rápidos (HSC): Configurar, 124
- Contraseña, 60
- Contraseña olvidada, 60
- Control de flujo, 272
 - Configuración, 272
- Control de flujo por hardware, 272
- Control de flujo por software, 273
- Corriente disponible, 361
 - Ejemplo, 363, 364
- Corriente necesaria, 25
- CPU
 - Agregar módulos, 81
 - Agregar nuevo dispositivo, 78
 - Cargar en dispositivo, 252
 - Cargas de lámpara, 39
 - Cargas inductivas, 39
 - Conectar online, 309
 - Conexión de red, 83
 - Configuración de dispositivos, 77
 - Configurar la comunicación con HMI, 253
 - Configurar parámetros, 80
 - Configurar varias, 256
 - Consumo de corriente, 361
 - Contraseña olvidada, 60
 - Corriente necesaria, 25
 - CPU sin especificar, 79
 - Crear una tarjeta de programa, 74
 - Crear una tarjeta de transferencia, 71
 - Datos técnicos 1211C, 325
 - Datos técnicos 1212C, 330
 - Datos técnicos 1214C, 335
 - Diagramas de cableado 1211C, 329
 - Diagramas de cableado 1212C, 334

- Diagramas de cableado 1214C, 339
 - Dirección IP, 84, 249
 - Dirección MAC, 266
 - Directrices de aislamiento galvánico, 37
 - Directrices de cableado, 36, 38
 - Ejecución del programa, 42
 - Estado operativo STOP, 316
 - Estados operativos, 44
 - Niveles de protección, 59
 - Online, 310
 - Panel de control online, 310
 - Parámetros de arranque, 43, 71
 - Procedimientos de montaje, 29
 - Procesamiento del arranque, 45
 - PROFINET, 84, 249
 - Protección por contraseña, 59
 - Puerto Ethernet, 84, 249
 - Puesta a tierra, 38
 - Recuperación si se olvida la contraseña, 60
 - Signal Board (SB), 13
 - Sinopsis, 11
 - Tabla de comparación, 12
 - Tablas de observación, 313
 - Tarjeta de programa, 74
 - Tarjeta de transferencia, 71
 - Tarjeta de transferencia vacía, 60
 - Tiempo de ciclo, 54
 - Zona de disipación, 26
 - CPU online, 310
 - Panel de control, 310
 - Vigilancia de la carga de la memoria, 311
 - Vigilancia del tiempo de ciclo, 311
 - CPU sin especificar, 79
 - Crear una conexión de red, 83
 - CTS, 272
 - Customer support, 3
- D**
- Datos técnicos, 319
 - Aprobación C-Tick, 321
 - Aprobación marina, 321
 - Compatibilidad electromagnética (CEM), 322
 - Condiciones ambientales, 323
 - CPU 1211C, 325
 - CPU 1212C, 330
 - CPU 1214C, 335
 - Diagrama de cableado del SM 1221, 341
 - Diagrama de cableado del SM 1222, 343
 - Diagrama de cableado del SM 1223, 345
 - Diagramas de cableado: SM 1231, 1232, 1234, 351
 - Entornos, 321
 - Generales, 319
 - Homologación ATEX, 320
 - Homologación CE, 319
 - Homologación cULus, 320
 - Homologación FM, 320
 - Memory Cards, 359
 - Módulo de comunicación CM 1241 RS232, 358
 - Módulo de comunicación CM 1241 RS485, 357
 - Módulo de señales SM 1221, 340
 - Módulo de señales SM 1222, 342
 - Módulo de señales SM 1223, 344
 - Módulos de señales analógicos, 346
 - Protección, 323
 - SB 1223, 353
 - SB 1223, 355
 - Signal Boards digitales (SBs), 353
 - Simuladores de entradas, 359
 - Tensión del módulo de señales analógico, 348
 - Tensiones nominales, 324
 - Vida útil de los relés, 324
 - Datos técnicos de la Memory Card, 359
 - Datos técnicos de la SB 1223, 353, 355
 - Datos técnicos de la Signal Board (SB) digital, 353
 - Datos técnicos de los módulos de señales analógicos, 346
 - Datos técnicos del CM 1241 RS232, 358
 - Datos técnicos del CM 1241 RS485, 357
 - Datos técnicos generales, 319
 - DB (bloque de datos), 95
 - DC
 - Cargas inductivas, 39
 - Desacoplar la ayuda en pantalla, 18
 - Determinar, 79
 - Diagrama de cableado de la SB 1223, 354
 - Diagrama de cableado de la SB 1232, 356
 - Diagramas de cableado
 - CPU 1211C, 329
 - CPU 1212C, 334
 - CPU 1214C, 339
 - Módulo de señales SM 1221, 341
 - Módulo de señales SM 1222, 343
 - Módulo de señales SM 1223, 345
 - SB 1223, 354
 - SB 1232, 356
 - SM 1231, 1232, 1234, 351
 - Dirección IP, 84, 85, 249, 250
 - Asignar, 244, 251
 - Asignar online, 247
 - Configurar, 84, 249
 - Dirección IP del router, 85, 250
 - Dirección IP, ajustar en la CPU online, 310
 - Dirección MAC, 84, 249, 266

Directrices

- Aislamiento galvánico, 37
- Cargas de lámpara, 39
- Cargas inductivas, 39
- Directrices de cableado, 36, 38
- Montaje, 23
- Procedimientos de montaje, 27
- Puesta a tierra, 38
- Directrices de aislamiento galvánico, 37
- Directrices de cableado
 - Puesta a tierra, 38
 - Requisitos, 36
- Diseñar un sistema PLC, 87, 88
- Dispositivos HMI
 - Conexión de red, 83
 - Sinopsis, 21
- Documentación, 17

E

E/S

- Cargas inductivas, 39
- Direccionamiento, 65
- Indicadores de estado analógicos, 308
- Indicadores de estado digitales, 308
- Ejecución de eventos, 48
- Ejecución del programa, 41
- EN y ENO (flujo de corriente), 98
- Entornos
 - Industrial, 321
- Errores
 - Errores de diagnóstico, 52
 - Errores de tiempo, 51
 - Instrucciones PtP, 302
- Errores de configuración de la recepción, 303
- Errores de configuración de la transmisión, 303
- Errores de configuración del puerto, 302
- Errores de procesamiento de señales, 303
- Errores diversos de parámetros PtP, 305
- Errores en tiempo de ejecución de la transmisión, 304
- Espacio de montaje, 24
- Estado operativo RUN, 44, 46
- Estado operativo STOP, 44, 316
- Estructura del programa, 90
- Ethernet
 - Conexión de red, 83
 - Dirección IP, 84, 249

F

- FB (bloque de función), 94

- FC (función), 93
- Fin del mensaje, 278
- Función (FC), 93
- FUP (Diagrama de funciones), 98

G

Getting Started

- Ayuda contextual, 17
- Ayuda desplegable, 17
- Ayuda en pantalla, 17
- Documentación, 17
- Sistema de información, 17
- Tooltips, 17
- Tooltips en cascada, 17
- Vistas del portal y del proyecto, 16

H

HMI

- Configurar la comunicación PROFINET, 253
- Homologación ATEX, 320
- Homologación CE, 319
- Homologación cULus, 320
- Homologación FM, 320
- Hora, ajustar en la CPU online, 310
- Hotline, 3
- HSC (contador rápido), 121
 - Configuración, 124

I

- Idle Line, 274, 276
- Imprimir temas de ayuda, 20
- Indicadores de estado de las E/S analógicas, 308
- Indicadores de estado de las E/S digitales, 308
- Indicadores LED, 282, 307
- Información de contacto, 3
- Inicio del mensaje, 276
- Insertar un dispositivo
 - CPU sin especificar, 79
- Instalación
 - STEP 7, 15
 - TIA Portal, 15
- Instancia individual
 - Ejemplo, 95
- Instrucción "Ajustar valor límite", 134
- Instrucción "Cambiar disposición", 140
- Instrucción ABS (Calcular valor absoluto), 132
- Instrucción Acumulador de tiempo (TONR), 112
- Instrucción ATTACH, 193

- Instrucción CEIL (Crear el siguiente número entero superior a partir del número en coma flotante), 143
- Instrucción Comprobar invalidez, 127
- Instrucción Comprobar validez, 127
- Instrucción Consultar flanco de señal ascendente de un operando, 109
- Instrucción Consultar flanco de señal descendente de un operando, 109
- Instrucción Copiar valor, 137
- Instrucción CTRL_PWM, 202
- Instrucción de alarma de retardo CAN_DINT, 196
- Instrucción de alarma de retardo SRT_DINT, 196
- Instrucción DEC (Decrementar), 131
- Instrucción DECO (Decodificar), 148
- Instrucción Desactivar salida, 106
- Instrucción DETACH, 193
- Instrucción DIS_AIRT, 199
- Instrucción Dividir (DIV), 128
- Instrucción EN_AIRT, 199
- Instrucción ENCO (Codificar), 148
- Instrucción FILL_BLK (Rellenar área), 139
- Instrucción FLOOR (Crear el siguiente número entero inferior a partir del número en coma flotante), 143
- Instrucción HSC (Controlar contador rápido), 119
- Instrucción Impulso (TP), 112
- Instrucción INC (Incrementar), 131
- Instrucción Inicializar temporizador (RT), 112
- Instrucción INV (Complemento a uno), 147
- Instrucción JMPN, 145
- Instrucción Marca de salto, 145
- Instrucción matemática en coma flotante, 135
- Instrucción MAX (Determinar máximo), 133
- Instrucción MIN (Determinar mínimo), 133
- Instrucción MOD (Obtener resto de división), 129
- Instrucción MOVE_BLK (Copiar área), 137
- Instrucción Multiplicar (MUL), 128
- Instrucción MUX (Multiplexar), 149
- Instrucción NEG (Generar complemento a dos), 130
- Instrucción NORM_X (Normalizar), 144
- Instrucción PID_Compact, 199
- Instrucción PORT_CFG (configuración de puerto), 284
- Instrucción RCV_CFG (recibir configuración), 288
- Instrucción RCV_PTP (recepción punto a punto), 298
- Instrucción RCV_RST (inicializar receptor), 299
- Instrucción RE_TRIGR, 174
- Instrucción Restar (SUB), 128
- Instrucción RET (Retroceder), 146
- Instrucción Retardo al conectar (TON), 112
- Instrucción Retardo al desconectar (TOF), 112
- Instrucción ROL (Rotar a la izquierda), 152
- Instrucción ROR (Rotar a la derecha), 152
- Instrucción ROUND (Redondear número), 142
- Instrucción RT (inicializar temporizador), 112
- Instrucción S_CONV, 159
- Instrucción Saltar al bloque si la señal es 1 (condicional) (JMP), 145
- Instrucción SCALE_X (Escalar), 144
- Instrucción SEL (Seleccionar), 149
- Instrucción SEND_CFG (enviar configuración), 286
- Instrucción SEND_PtP (enviar datos punto a punto), 294
- Instrucción SGN_GET (consultar señales RS232), 300
- Instrucción SGN_SET (activar señales RS232), 301
- Instrucción SHL (Desplazar a la izquierda), 151
- Instrucción SHR (Desplazar a la derecha), 151
- Instrucción STP (Parar ciclo del PLC), 175
- Instrucción STRG_VAL, 159
- Instrucción Sumar (ADD), 128
- Instrucción T_ADD, 154
- Instrucción T_CONV, 154
- Instrucción T_DIFF, 154
- Instrucción T_SUB, 154
- Instrucción TCON, 184
- Instrucción TDISCON, 184
- Instrucción TOF (retardo al desconectar), 112
- Instrucción TON (retardo al conectar), 112
- Instrucción TONR (acumulador de tiempo), 112
- Instrucción TP (impulso), 112
- Instrucción TRCV, 184
- Instrucción TRCV_C, 178, 262
- Instrucción TRUNC (Truncar a entero doble), 142
- Instrucción TSEND, 184
- Instrucción TSEND_C, 178, 258
- Instrucción UFILL_BLK (Rellenar área sin interrupciones), 139
- Instrucción UMOVE_BLK (Copiar área sin interrupciones), 137
- Instrucción USS_DRV, 209
- Instrucción USS_PORT, 212
- Instrucción USS_RPM, 213
- Instrucción USS_WPM, 214
- Instrucción VAL_STRG, 159
- Instrucción Valor dentro del rango, 127
- Instrucción Valor fuera del rango, 127
- Instrucciones
- Activar, 106
 - Alarma: ATTACH, 193
 - Alarma: CAN_DINT, 196
 - Alarma: DETACH, 193
 - Alarma: DIS_AIRT, 199
 - Alarma: EN_AIRT, 199
 - Alarma: SRT_DINT, 196
 - Calcular valor absoluto (ABS), 132
 - Calendario, 154

- Cambiar disposición, 140
- Codificar (ENCO), 148
- Códigos de estado USS, 216
- Comparación, 126
- Complemento a uno (INV), 147
- Contador, 116
- Contador rápido (HSC), 119
- Convertir, 141
- Convertir cadena en valor: S_CONV, 159
- Convertir cadena en valor: STRG_VAL, 159
- Convertir valor en cadena: S_CONV, 159
- Convertir valor en cadena: VAL_STRG, 159
- Copiar área (MOVE_BLK), 137
- Copiar área sin interrupciones (UMOVE_BLK), 137
- Crear el siguiente número entero inferior a partir del número en coma flotante (FLOOR), 143
- Crear el siguiente número entero superior a partir del número en coma flotante (CEIL), 143
- CTRL_PWM, 202
- DEC (Decrementar), 131
- Decodificar (DECO), 148
- Dentro del rango, 127
- Desactivar, 106
- Desplazamiento, 137
- Desplazar a la derecha (SHR), 151
- Desplazar a la izquierda (SHL), 151
- Dividir (DIV), 128
- Escalar (SCALE_X), 144
- Fecha, 154
- Flanco ascendente, 109
- Flanco descendente, 109
- Fuera del rango, 127
- GET_ERROR, 178
- Hora, 154
- INC (Incrementar), 131
- Límite, 134
- Lógicas con bits, 103
- Marca de salto, 145
- Matemáticas en coma flotante, 135
- MAX (Determinar máximo), 133
- MIN (Determinar mínimo), 133
- MOD (Obtener resto de división), 129
- Multiplexar (MUX), 149
- Multiplicar (MUL), 128
- NEG (Generar complemento a dos), 130
- Normalizar (NORM_X), 144
- NOT_OK, 127
- O, 147
- OK, 127
- PID_Compact, 199
- PORT_CFG (configuración de puerto), 284
- RCV_CFG (recibir configuración), 288
- RCV_PtP (recepción punto a punto), 298
- RCV_RST (inicializar receptor), 299
- RE_TRIGR, 53, 174
- Redondear, 142
- Rellenar área (FILL_BLK), 139
- Rellenar área sin interrupciones (UFILL_BLK), 139
- Reloj, 156
- Reloj: Escribir hora del sistema (WR_SYS_T), 156
- Reloj: Leer hora del sistema (RD_SYS_T), 156
- Reloj: Leer hora local (RD_LOC_T), 156
- Restar (SUB), 128
- Retroceder (RET), 146
- Rotar a la derecha (ROR), 152
- Rotar a la izquierda (ROL), 152
- Saltar al bloque si la señal es 1 (condicional) (JMP), 145
- Seleccionar (SEL), 149
- SEND_CFG (enviar configuración), 286
- SEND_PtP (enviar datos punto a punto), 294
- SGN_GET (consultar señales RS232), 300
- SGN_SET (activar señales RS232), 301
- STP (Parar ciclo del PLC), 175
- Sumar (ADD), 128
- T_ADD, 154
- T_CONV, 154
- T_DIFF, 154
- T_SUB, 154
- TCON, 184
- TDISCON, 184
- Temporizador, 112
- Temporizador: RT (inicializar temporizador), 112
- Temporizador: TOF (retardo al desconectar), 112
- Temporizador: TON (retardo al conectar), 112
- Temporizador: TONR (acumulador de tiempo), 112
- Temporizador: TP (impulso), 112
- TRCV, 184
- TRCV_C, 178, 262
- Truncar a entero doble (TRUNC), 142
- TSEND, 184
- TSEND_C, 178, 258
- USS_DRV, 209
- USS_PORT, 212
- USS_RPM, 213
- USS_WPM, 214
- XOR (O-exclusiva), 147
- Y, 147
- Instrucciones con contadores, 116
- Instrucciones con temporizadores, 112
- Instrucciones de calendario, 154
- Instrucciones de comparación, 126
- Instrucciones de conversión, 141

Instrucciones de conversión de cadenas en valores, 159
 Instrucciones de conversión de valores en cadenas, 159
 Instrucciones de fecha, 154
 Instrucciones de hora, 154
 Instrucciones de reloj, 156
 Escribir hora del sistema (WR_SYS_T), 156
 Leer hora del sistema (RD_SYS_T), 156
 Leer hora local (RD_LOC_T), 156
 Instrucciones Ethernet
 TCON, 184
 TDISCON, 184
 TRCV, 184
 TRCV_C, 178
 TSEND, 184
 TSEND_C, 178
 Instrucciones lógicas con bits, 103
 Instrucciones matemáticas, 128
 Interfaces de comunicación
 Configuración, 271
 Programación, 280
 Interfaz PROFINET
 Propiedades de direcciones Ethernet, 85, 250
 Propiedades de sincronización horaria, 268

J

Juego de instrucciones, 106

K

KOP (Esquema de contactos), 97

L

Latencia de alarmas, 50
 Librería del protocolo USS, 205
 Librería global
 USS, 205
 Llamada de bloque
 Llamada como instancia individual o multiinstancia, 94
 Principios básicos, 41
 Longitud
 Mensaje, 279
 Longitud del mensaje, 278
 Longitud m, 279
 Longitud máxima del mensaje, 278
 Longitud n, 279

M

Máscara de subred, 84, 249
 MB_COMM_LOAD, 217
 MB_MASTER, 219
 MB_SLAVE, 231
 Memoria
 I (memoria imagen de proceso de las entradas), 62
 L (memoria local), 60
 M (área de marcas), 63
 Marcas de ciclo, 57
 Marcas de sistema, 57
 Memoria de carga, 54
 Memoria de trabajo, 54
 Memoria remanente, 54
 Memoria temporal, 64
 Q (memoria imagen de proceso de las salidas), 63
 Memory Card
 Configurar los parámetros de arranque, 71
 Contraseña olvidada, 60
 Tarjeta de programa, 74
 Tarjeta de transferencia, 71
 Tarjeta de transferencia vacía, 60
 MODBUS, 217
 MB_Master, 219
 MB_SLAVE, 231
 Módulo de comunicación
 Agregar módulos, 81
 Agregar nuevo dispositivo, 78
 Configuración de dispositivos, 77
 Módulo de comunicación (CM), 282
 Consumo de corriente, 361
 Datos técnicos, 357
 Desmontaje, 33
 Montaje, 33
 Recepción de datos, 298
 Sinopsis, 14
 Tabla de comparación, 13
 Módulo de comunicación, librería USS, 206
 Módulo de señales (SM)
 Agregar módulos, 81
 Agregar nuevo dispositivo, 78
 Configuración de dispositivos, 77
 Consumo de corriente, 361
 Desmontaje, 31
 Montaje, 31
 Sinopsis, 14
 Tabla de comparación, 13
 Módulos
 Configurar parámetros, 82
 Módulo de comunicación (CM), 14
 Módulo de señales (SM), 14
 Signal Board (SB), 13

- Tabla de comparación, 13
- Zona de disipación, 26
- Módulos de comunicación
 - RS232 y RS485, 270
- Módulos de comunicación RS232 y RS485, 270
- Módulos de E/S
 - Tablas de observación, 313
- Módulos de señales
 - Datos técnicos SM 1221, 340
 - Datos técnicos SM 1222, 342
 - Datos técnicos SM 1223, 344
- Montaje
 - Aislamiento galvánico, 37
 - Cargas de lámpara, 39
 - Cargas inductivas, 39
 - Conector del bloque de terminales, 35
 - Corriente necesaria, 25
 - CPU, 29
 - Dimensiones, 26
 - Dimensiones de montaje, 26
 - Directrices, 23
 - Directrices de aislamiento galvánico, 37
 - Directrices de cableado, 36, 38
 - Espacio, 24
 - Módulo de comunicación (CM), 33
 - Módulo de señales (SM), 31
 - Puesta a tierra, 38
 - Signal Board (SB), 34
 - Sinopsis, 23, 27
 - Zona de disipación, 26
- Mostrar el contenido e índice (ayuda en pantalla), 18

N

- Network Time Protocol (NTP), 268
- Nivel de protección
 - Bloque lógico, 99
 - Contraseña olvidada, 60
 - CPU, 59
- Números
 - Coma flotante, 66
 - Real, 66

O

- Observar el programa, 102
- Online, conectar online, 309
- Operación lógica O, 147
- Operación lógica O-exclusiva (XOR), 147
- Operación lógica Y, 147

P

- Parametrización, 94
- Parámetros de arranque, 43, 71
- Parámetros de salida, 94
- Paridad, 272
- Pausa, 274, 276
- Perfil DIN, 27
- PLC
 - Diseño del sistema, 87
 - Sinopsis, 11
 - Usar bloques, 88
- Posición de carácter
 - Longitud del mensaje, 279
- Posiciones de memoria, 60, 62
- Prioridades de procesamiento, 48
- PROFINET, 241
 - Comprobar una red, 251
 - Conexión de red, 83
 - Dirección IP, 84, 249
- Programación
 - CPU sin especificar, 79
 - Estructurada, 89
 - Flujo de corriente (EN y ENO), 98
 - FUP (Diagrama de funciones), 98
 - Instrucciones PtP, 280
 - KOP (Esquema de contactos), 97
 - Lineal, 89
- Programación estructurada, 89, 90
- Programación lineal, 89
- Programación PtP, 280
- Programación punto a punto, 280
- Protección anticopia, 99
- Protección de know-how, 99
- Protección por contraseña
 - Bloque lógico, 99
 - Contraseña olvidada, 60
 - CPU, 59
 - Tarjeta de transferencia vacía, 60
- Protocolo
 - Comunicación, 269
 - Freeport, 269
- Protocolo Freeport, 269
- Proyecto
 - Contraseña olvidada, 60
 - Proteger un bloque lógico, 99
 - Restringir el acceso a la CPU, 59
 - Tarjeta de programa, 74
 - Tarjeta de transferencia, 71
 - Tarjeta de transferencia vacía, 60
- PTO (tren de impulsos), 202
- PWM
 - Instrucción CTRL_PWM, 202

R

Reloj
 Reloj en tiempo real, 56
 Retardo RTS OFF, 274
 Retardo RTS ON, 274
 Router IP, 85, 250
 RTS, 272
 RTS conmutado, 272
 RTS siempre ON, 273

S

S7-1200
 Agregar módulos, 81
 Agregar nuevo dispositivo, 78
 Ampliar las prestaciones, 13
 Cargas de lámpara, 39
 Cargas inductivas, 39
 Conector del bloque de terminales, 35
 Conexión de red, 83
 Configuración de dispositivos, 77
 Configurar los módulos, 82
 Configurar los parámetros de la CPU, 80
 Contraseña olvidada, 60
 Corriente necesaria, 25
 CPU, 11
 Dimensiones de montaje, 26
 Dirección IP, 84, 249
 Directrices de aislamiento galvánico, 37
 Directrices de cableado, 36, 38
 Dispositivos HMI, 21
 Espacio, 24
 Módulo de comunicación (CM), 14
 Módulo de señales (SM), 14
 Montaje de la CPU, 29
 Montaje de un CM, 33
 Montaje de un SM, 31
 Montaje de una SB, 34
 Parámetros de arranque, 43, 71
 PROFINET, 84, 249
 Protección por contraseña, 59
 Puerto Ethernet, 84, 249
 Puesta a tierra, 38
 Signal Board (SB), 13
 Sinopsis del montaje, 27
 Tabla de comparación de los modelos de CPUs, 12
 Tarjeta de programa, 74
 Tarjeta de transferencia, 71
 Tarjeta de transferencia vacía, 60
 Tiempo de ciclo, 54
 Zona de disipación, 26

Secuencia de caracteres

Fin del mensaje, 279
 Inicio del mensaje, 277

Seguridad

Bloque lógico, 99
 Contraseña olvidada, 60
 CPU, 59

Signal Board (SB)

Agregar módulos, 81
 Configuración de dispositivos, 77
 Consumo de corriente, 361
 Desmontaje, 34
 Montaje, 34
 Sinopsis, 13
 Tabla de comparación, 13

Signal board (SM)

Agregar nuevo dispositivo, 78

Simuladores de entradas, 359

Sistema de información, 17

Ampliar, 18
 Desacoplar, 18
 Imprimir, 20
 Mostrar el contenido e índice, 18

STEP 7

Agregar módulos, 81
 Agregar nuevo dispositivo, 78
 Conexión de red, 83
 Configuración de dispositivos, 77
 Configurar la CPU, 80
 Configurar los módulos, 82
 Instalación, 15
 PROFINET, 84, 249
 Puerto Ethernet, 84, 249
 Vista del portal, 16
 Vista del proyecto, 16

Support, 3

T

Tabla de comparación

Dispositivos HMI, 21
 Modelos de CPUs, 12

Tabla de comparación de módulos, 13

Tablas de observación, 102, 313

Tarjeta de programa, 74

Configurar los parámetros de arranque, 71

Tarjeta de transferencia, 71

Configurar los parámetros de arranque, 71
 Contraseña olvidada, 60

Tarjeta de transferencia vacía, 60

Technical support, 3

Tensión del módulo de señales analógico, 348

Tensiones nominales, 324
TIA Portal
 Agregar módulos, 81
 Agregar nuevo dispositivo, 78
 Conexión de red, 83
 Configuración de dispositivos, 77
 Configurar la CPU, 80
 Configurar los módulos, 82
 Instalación, 15
 PROFINET, 84, 249
 Puerto Ethernet, 84, 249
 Vista del portal, 16
 Vista del proyecto, 16
Tiempo de ciclo, 53, 54
Tiempo de espera, 272
Tiempo excedido entre caracteres, 278
Tipo de datos STRING, 67
Tipo de datos DTL, 68
Tipo de datos DTL (Data and Time Long), 68
Tipos de datos, 65
 DTL, 68
 Matrices, 67
 STRING, 67
Tooltips, 17
Transición de RUN a STOP, 58
Transmisión de datos, iniciar, 294
Tren de impulsos (PTO), 202
TSAP (Transport Service Access Points), 259, 263

V

Valores de retorno
 Instrucciones PtP, 302
Valores de retorno de instrucciones PtP, 302
Valores de retorno en el tiempo de ejecución de la recepción, 304

Velocidad de transferencia, 271
Vida útil de los relés, 324
Vigilancia de la carga de la memoria, CPU online, 311
Vigilancia del tiempo de ciclo, CPU online, 311
Vista del portal, 16
 Agregar módulos, 81
 Agregar nuevo dispositivo, 78
 Configurar el puerto Ethernet, 84, 249
 Configurar la CPU, 80
 Configurar los módulos, 82
 PROFINET, 84, 249
Vista del proyecto, 16
 Agregar módulos, 81
 Agregar nuevo dispositivo, 78
 Conexión de red, 83
 Configuración de dispositivos, 77
 Configurar el puerto Ethernet, 84, 249
 Configurar los módulos, 82
 Configurar los parámetros de la CPU, 80
 PROFINET, 84, 249

W

Watchdog, 174

X

XON/XOFF, 273

Z

Zona de disipación, 26