

“DISEÑO Y MONTAJE DE UN ENTRENADOR CON EL
CONTROLADOR LOGICO PROGRAMABLE PLC S7-200 DE
SIEMENS”

EDGAR ANTONIO PEREZ ROJAS

UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERIA ELECTRONICA
ESCUELA DE CIENCIAS BASICAS
BUCARAMANGA
2008

“DISEÑO Y MONTAJE DE UN ENTRENADOR CON EL
CONTROLADOR LOGICO PROGRAMABLE PLC S7-200 DE
SIEMENS”

Por

EDGAR ANTONIO PEREZ ROJAS

Tesis propuesta para obtener el titulo de ingeniero
electrónico

Asesor

JORGE ENRIQUE SANTAMARIA CARREÑO
Ingeniero Electrónico UPB. Esp.
Instructor Certificado Nacional Instruments – LabVIEW.

UNIVERSIDAD PONTIFICIA BOLIVARIANA
FACULTAD DE INGENIERIA ELECTRONICA
ESCUELA DE CIENCIAS BASICAS
BUCARAMANGA

2008

Nota de Aceptación

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

Bucaramanga, Agosto de 2008

A Mis Padres, A Mi Sobrinito

AGRADECIMIENTOS

Este es un paso muy importante en mi vida, la culminación del proceso y de una etapa en la formación personal y profesional. Sinceramente he llegado hasta esta etapa gracias a la ayuda de muchas personas principalmente.

A mi DIOS, pues el me dio la existencia y me dio la oportunidad de lograr este paso importante en la vida.

A mis apreciados padres, Gloria Maria Rojas de Pérez y Edgar Agustín Pérez por quererme y el respaldo incondicional hicieron posible alcanzar este gran logro, además de su sacrificio porque saliera adelante en la vida.

A mi sobrino Juan Sebastián Flórez Pérez quien me ha entregado amor, cariño amistad y apoyo sincero en esta etapa tan crucial en mi vida.

A mis amigos, compañeros, profesores y a todos los que de una u otra forma estuvieron presentes en este tiempo de formación.

EDGAR ANTONIO PEREZ ROJAS

CONTENIDO

	Pág.
INTRODUCCION	1
1.11PLANTEAMIENTO DEL PROBLEMA	2
2. OBJETIVOS	3
2.1 Objetivos generales	3
2.2 Objetivos específicos	3
3. JUSTIFICACION	4
4. FUNDAMENTO TEORICO	5
4.1 Instrumentos industriales	5
4.2 Clasificación de los procesos industriales	5
4.3 Lazo de control	6
4.4 Elemento final de control	8
4.4.1 Variador de Velocidad	8
4.4.2 Motores	9
4.4.3 Relés	9
4.4.4 Controladores lógicos programables (PLC's)	10
4.4.4.1 Partes del PLC	13
4.4.4.1.1 CPU (Unidad Central de Procesamiento)	14
4.4.4.1.2 Memoria	15
4.4.4.1.3 Unidades de entrada y salida (E/S, I/O)	15
4.4.4.1.4 Interface	15
4.4.4.1.5 Unidades de programación	16
4.4.4.1.6 Periféricos	16
4.4.4.2 Modelos de PLC's	16
4.4.4.2.1 Telemecanique	17
4.4.4.2.1.1 Zelio	17
4.4.4.2.1.2 Twido	18

4.4.4.2.2 Omron	18
4.4.4.2.2.1 CPM2A	19
4.4.4.2.2.2 CS1	20
4.4.4.2.3 Siemens	21
4.4.4.2.3.1 S7-200	22
4.4.4.2.3.2 S7-300	24
5. DISEÑO Y CONSTRUCCION DEL ENTRENADOR	26
5.1 Chasis del entrenador.	26
5.2 PLC S7-224XP	28
5.3 Cable de maestreo	31
5.4 Simulador de entradas	32
5.5 Modulo de salida	33
5.5.1 Accesorios potencia industrial Relay RT / RP / SR2M	33
5.5.2. PTM L0 524 EM07 red LED 6...24 VDC/VAC	34
Relé RT2	34
5.6 Fuente de alimentación	36
5.7 Variador de velocidad	37
5.7.1 Construcción	37
5.7.2 Características principales	37
5.8 MICROMASTER 420	39
5.9 Motor.	40
5.10 Borneras.	41
5.11 Conectores de cables para conecion	42
5.12 Marcaciones para borneras y cables	42
6. DESARROLLO METODOLÓGICO	44
6.1 Descripción general del Software STEP 7-Micro/WIN V4.0	44
6.2 Conexión del computador y la CPU S7-224	45
6.3 Instalación del software STEP 7-Micro/WIN V4	46
6.4 Ejecución del Software	46
6.4.1 Introducción de órdenes	51
6.4.1.1 Baras de Herramientas	51
6.4.1.2 Barras de Herramientas Comun	55
6.4.1.3 Barras de Herramientas Estandar	58
6.4.1.4 Barra de herramientas Test	59
7. APLICACIÓN DE LA PLATAFORMA STEP7- MICRO/WIN V.4.0	63
7.1 Operaciones lógicas con contactos	63
7.2 Operaciones lógicas con salidas	64
7.3 Operaciones de transferencia	66
7.4 Operaciones con temporizadores	65
7.5 Operación conteo	68

7.6	Operación	De	Comparación	69
8. APLICACIÓN DE LA PLATAFORMA STEP7- MICRO/WIN V.4.0				71
8.1	EJERCICIO 1			71
8.2	EJERCICIO 2			72
8.3	EJERCICIO 3			73
8.4	EJERCICIO 4			74
8.5	EJERCICIO 5			75
8.6	EJERCICIO 6			77
8.7	EJERCICIO 7			81
8.8	EJERCICIO 8			83
8.9	EJERCICIO 9			85
CONCLUSIONES Y RECOMENDACIONES				88
BIBLIOGRAFÍA				89
PAGINAS DE REFERENCIA				90
ANEXOS				
ANEXO A VARIADOR DE VELOCIDAD MICROMASTER 440				i
ANEXO B MOTOR				Xxx
ANEXO C REGLAS PARA MONTAR EL S7-200				Xxxix
ANEXO D MONTAR Y DESMONTAR EL S7-200				Xli
ANEXO E REGLAS DE PUESTA A TIERRA Y CABLEADO				Xlv
ANEXO F EJECUTAR LA LÓGICA DE CONTROL EN EL S7-200				Xlviii
ANEXO G SELECCIONAR EL MODO DE OPERACIÓN DEL S7-200				l
ANEXO H MANUAL DE SISTEMA DE AUTOMATIZACION DEL S7-200				li

LISTA DE FIGURAS

	Pág.
Figura 4 . 1 . Variador de Velocidad	6
Figura 4.2. Variador de velocidad	9
Figura 4.3. Estructura física de un motor	9
Figura 4.4. Estructura interna de un relé	10
Figura 1.5 Estructura Funcional de un PLC	11
Figura 4.6. Estructura externa compacta de un PLC	12
Figura 4.7. PLC modular	12
Figura 4.8. Ciclo de Trabajo de un PLC	14
Figura 4.9. Interfaz de un PLC	16
Figura 4.10. Modelo Zelio Logia de Telemecanique	17
Figura 4.11. Modelo de PLC Twido	18
Figura 4.12. PLC CPM2A	19
Figura 4.13. Descripción de las partes físicas del PLC CPM2A	19
Figura 4.14. Descripción del moldeo CMP2A	20
Figura 4 . 15. PLC CS1	21
Figura 4.16 Avances tecnológicos de los PLC`s	21
Figura 4.17 PLC`s S7-200	22
Figura 4.18 PLC`s S7-300	22
Figura 4.19 PLC`s S7-400	22
Figura 4.20 Modelos de la serie S7-200	23
Figura 4 . 2 1 . PLC S7-200	23
Figura 4.22 Modulo de expansión del S7-200	24
Figura 4.23. PLC S7-300	25
Figura 5.1 Diagrama de bloques del sistema	26
Figura 5.2 Dimensiones del chasis	27
Figura 5.3 Estructura completa del entrenado	27
Figura 5.4 Panel frontal	28
Figura 5.5 PLC S7-224XP	28
Figura 5.6 Especificaciones técnicas	29
Figura 5.7 Conexión de la alimentación del PLC S7-224XP	30

Figura 5.8 Diagrama de cableado de la CPU 224XP	31
Figura 5.9 Estructura del cable de muestreo.	32
Figura 5.10 Simulador de entrada.	32
Figura 5.11 Base del modulo de la salida.	33
Figura 5.12 Dimensiones del modulo de la salida.	33
Figura 5.13 Dimensiones del PTM L0 524 EM07.	34
Figura 5.14 Modulo indicador PTM L0 524 EM07 red LED 6...24 VDC/VAC.	34
Figura 5.15 Rele RT2.	34
Figura 5.16 Diseño PCB / terminal de cesión vista inferior a la soldadura de las patillas	35
Figura 5.17 Diseño PCB / terminal de cesión vista inferior.	35
Figura 5.18 Dimensiones del Reles	35
Figura 5.19 Fuente de alimentacion.	36
Figura 5.20 MICROMASTER 420	37
Figura 5.22 Tamaño constructivo 1/3 AC 200 V a 240 V 3 AC 380 V a 480 V	38
Figura 5.22 Conexión de la placa con pantallas	38
Figura 5.23 Esquemas de conexiones	39
Figura 5.24 Imagen del motor	40
Figura 5.25 Borneras porta fusibles. Diferentes vistas.	41
Figura 5.26 Borneras de uso general	41
Figura 5.27 Borneras para tierra	41
Figura 5.28 Conectores para cable	42
Figura 5.29 Marcaciones para borneras y cable	43
Figura 6.1. Entorno de trabajo del software STEP 7-Micro/WIN V4.	45
Figura 6.2. Comunicación del PC y el PLC	46
Figura 6.3. Acceso al software	47
Figura 6.4. Entorno de trabajo del software STEP 7-Micro/WIN V4	47
Figura 6.5. Selección del tipo de CPU a utilizar	48
Figura 6.6. Configuración del <i>TIPO DE CPU</i>	49
Figura 6.7. Ventana de configuración de la comunicación entre el PLC y el computador	49
Figura 6.8. Ventana de configuración del tipo de Cable de comunicación entre el computador y el PLC	50
Figura 6.9. Configuración de los parámetros PC/PPI	50
Figura 6.10. Selección del puerto de comunicación.	51
Figura 6.11. Barra de operación o de herramientas	52
Figura 6.12. Tipos de Contactos a insertar 0	53
Figura 6.13. Tipos de bobinas a insertar LÜJ	53

-D

Figura 6.14. Tipos de cuadros a insertar	54
Figura 6.15. Inserción de código de programación	55
Figura 6.16. Barra de Herramientas Común	55
Figura 6.172. Aplicación de los botones de los comentarios UOP y comentarios de segmento.	57
Figura 6.18. Visualización de la tabla de símbolos	57
Figura 6.19. Aplicación de las funciones de marcadores dentro de un Programa	58
Figura 6.20. Barra de herramientas Estándar	58
Figura 6.21. Cargar un programa	59
Figura 6.22. Barra de herramientas Test	60
Figura 6.23. Activa/desactiva el estado del programa	61
Figura 6.24. Tabla de estado	61
Figura 6.25. Dirección forzada	61
Figura 6.26. Curvas de estado de la CPU	62
Figura 8.1 Diagrama de la compuerta AND	71
Figura 8.2 Diagrama esquemático del ejercicio 1 en la plataforma STEP7-MICRO/WIN V.4.0	72
Figura 8.3 Diagrama de la compuerta OR	72
Figura 8.4 Diagrama esquemático del ejercicio 2 en la plataforma STEP7-MICRO/WIN V.4.0	72
Figura 8.5 Diagrama de la compuerta NAND	73
Figura 8.6 Diagrama esquemático del ejercicio 3 en la plataforma STEP7-MICRO/WIN V.4.0	74
Figura 8.7 Diagrama esquemático del ejercicio 4 en la plataforma STEP7-MICRO/WIN V.4.0	75
Figura 8.8 tabla de símbolos	75
Figura 8.9 Diagrama esquemático del ejercicio 5 en la plataforma STEP7-MICRO/WIN V.4.0	76
Figura 8.10 Diagrama del T37	76
Figura 8.11 Diagrama esquemático del ejercicio 6 en la plataforma STEP7-MICRO/WIN V.4.0	77
Figura 8.12 Diagrama esquemático del ejercicio 6 en la plataforma STEP7-MICRO/WIN V.4.0	78
Figura 8.13 Diagrama esquemático del ejercicio 6 en la plataforma STEP7-MICRO/WIN V.4.0	79
Figura 8.14 Diagrama de Incrementar contador	80
Figura 8.15 Tabla de símbolos	81
Figura 8.16 Areas de memoria y funciones del ST-200	81
Figura 8.17 Diagrama esquemático del ejercicio 7 en la plataforma STEP7-MICRO/WIN V.4.0	82
Figura 8.18 Diagrama de Transferir byte	82

Figura 8.19 Diagrama esquemático del ejercicio 7 en la plataforma STEP7-MICRO/WIN V.4.0	83
Figura 8.20 Diagrama esquemático del ejercicio 8 en la plataforma STEP7-MICRO/WIN V.4.0	84
Figura 8.21 Diagrama de llama Posicionar dominante biestable	84
Figura 8.22 Comportamiento de la función posicionar dominante biestable y rearmar dominante biestable.	85
Figura 8.23 Diagrama esquemático del ejercicio 8 en la plataforma STEP7-MICRO/WIN V.4.0	86
Figura 8.24 Diagrama de la Función de mover palabra	86

LISTA DE TABLAS

	Pág.
Tabla 4.1 Tipos de elementos primarios	7
Tabla 4.2 Clases de PLC`s S7-300	25
Tabla 5.1 Datos técnicos	36
Tabla 5.2 Datos técnicos del motor	40
Tabla 6.1 Descripción de los elementos de la barra de operación	52
Tabla 6.2 Descripción de los elementos de la barra de herramientas común	56
Tabla 6.3 Descripción de los elementos de la barra de herramientas estandar	59
Tabla 6.4 Descripción de los elementos de la barra de herramientas test	60
Tabla 7.1 Descripción del contacto estándar	63
Tabla 7.2 Contactos directos	64
Tabla 7.3 Contacto NOT	64
Tabla 7.4 Detección de flanco positivo y negativo	64
Tabla 7.5 Operaciones lógicas con salidas	65
Tabla 7.6 Operaciones de transferencia	66
Tabla 7.7 Tipos de temporizadores	67
Tabla 7.8 Resolución de los temporizados	67
Tabla 7.9 Operaciones de conteo	68
Tabla 7.10 Descripción de los operadores de comparación	69
Tabla 8.1 Tabla de la verdad de una compuesta AND	71
Tabla 8.2 Representación de la AND con contactores	71
Tabla 8.3 Tabla de la verdad de una compuesta OR	72
Tabla 8.4 Representación de la OR con contactores	73
Tabla 8.5 Tabla de la verdad de una compuesta NAND	74
Tabla 8.6 Representación de la NAND con contactores	74
Tabla 8.7 Tabla de verdad para la operación Posicionar dominante biestable	85

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: DISEÑO Y MONTAJE DE UN ENTRENADOR CON
EL CONTROLADOR LÓGICO PROGRAMABLE PLC
S7-200 SIEMENS

AUTOR(ES): EDGAR ANTONIO PÉREZ ROJAS

FACULTAD: Facultad de Ingeniería Electrónica

DIRECTOR(A): JORGE E. SANTAMARIA C.

RESUMEN

El entrenador que se construyó es un módulo para configurar, programar y usar un autómata, SIEMENS. El sistema consiste en un simulador de entradas, luces piloto; estos dispositivos se usan para que el PLC “teste y actúe” según su programación.

El autómata y los dispositivos mencionados anteriormente están dispuestos en una mesa con un “frame” el cual es una estructura que soporta el hardware. El sistema de supervisión y control se realiza desde el autómata, conjuntamente se puede realizar monitoreo en una PC.

Al implementar este entrenador, el practicante tendrá la posibilidad de familiarizarse de una forma práctica con el software y el lenguaje de programación que se implementa para este autómata en el área de la automatización industrial y tendrá unas bases sólidas para su buen desempeño en el campo laboral.

PALABRAS CLAVES: Automata Control Automatización

PLC STEP7

V° B° DIRECTOR DE TRABAJO DE GRADO

RESUMEN GENERAL DE TRABAJO DE GRADO

TITULO: **DESIGN AND INSTALLATION OF A TRAINER WITH
PROGRAMMABLE LOGIC CONTROLLER PLC S7-200
SIEMENS**

AUTOR(ES): EDGAR ANTONIO PÉREZ ROJAS

FACULTAD: Facultad de Ingeniería Electrónica

DIRECTOR(A): JORGE E. SANTAMARIA C.

RESUMEN

The entrenador is a module for configuring, programming and using a PLC, SIEMENS. The system consists of a simulator innings, pilot lights, these devices are used for the PLC "Testing and acting" in its programming.

The automaton and the aforementioned devices are arranged in a table with a "frame" which is a structure that supports the hardware. The monitoring and control system is carried out from the PLC, jointly monitoring can be performed on a PC.

By implementing this coach, the practitioner will be able to familiarise themselves in a practical way with the software and programming language that is implenta for this automaton in the area of industrial automation and have a solid foundation for their good performance at work .

PALABRAS CLAVES: Automata Control Automatización

PLC STEP7

V° B° DIRECTOR DE TRABAJO DE GRADO

INTRODUCCION

Con esta investigación se logra un aporte en el continuo trabajo que se realiza en la Universidad Pontificia Bolivariana (seccional Bucaramanga) en la facultad de ingeniería electrónica en construcción de entrenadores para aplicaciones en automatización y control basada en PC.

La automatización y control ha cumplido un papel importante en el progreso de la ingeniería, esta a su vez se ha convertido en parte fundamental e integral de los procesos industriales modernos.

Como los avances en la práctica y la teoría de la automatización aportan los medios para lograr un desempeño óptimo de los sistemas, mejora la productividad, simplifican las labores de muchas operaciones manuales, repetitivas y rutinarias, como otras actividades. Los ingenieros deben tener amplio y buen conocimiento de este campo que es muy trascendental.

Para diseñar y ejecutar sistemas de automatización y control de procesos la instrumentación virtual se ha convertido en una opción muy apetecida. Por este motivo varios sistemas están sacando partido de la tecnología del PC para las diferentes aplicaciones en las cuales el rendimiento, eficacia, aplicación y la ejecución son fundamentales. Las aplicaciones basadas en PC dan un rendimiento preciso que se necesita para los sistemas actuales; los PC son utilizados para la automatización industrial, por esta razón, se están usando aplicaciones con software de gran utilidad en la automatización (de la casa de Siemens). Como punto de partida un ambiente virtual usuarios como estudiante o docentes se ven directamente involucrados implementando un ambiente real por medio de un sistema para efectuar control y monitoreo mediante una Interfaz Gráfica de Usuario (GUI)

Al implementar este sistema de testeo basado en PC bajo plataforma Siemens, el estudiante de ingeniería que aplique en el área de automatización y control va a tener la posibilidad de familiarizarse de una forma práctica en ese campo y tendrá base práctica para un buen desempeño en el campo laboral.

De este modo, se pretende establecer un precedente para el fortalecimiento del área de control en la facultad de ingeniería electrónica de la Universidad Pontificia Bolivariana desarrollando entrenadores para la especialización y maestría en la mencionada área.

1. PLANTEAMIENTO DEL PROBLEMA

La Universidad Pontificia Bolivariana en el laboratorio de control cuenta con 6 módulos de proceso automatizados aproximadamente en los programas de Ingeniería Electrónica e Ingeniería Mecánica; y de las plataformas nuevas que se están implementando (Opto 22 y Compact Field Point), surge la necesidad de desarrollar proyectos con otras arquitecturas para el laboratorio.

Las variables que se le den a este entrenador se van a monitorear y controlar en un PC de propósito general bajo plataforma de la casa matriz de Siemens.

Siemens es en la actualidad una plataforma estándar de la industria de medida y test para el desarrollo de sistemas y control de instrumentación en el campo de la automatización para el monitoreo, adquisición de datos, análisis de procesos industriales

La visión desde la academia, es además, de estandarizar una plataforma, es tener una evolución en las diferentes áreas de trabajo como diseño, telecomunicaciones, tratamiento de señales, sistemas embebidos, simulación en tiempo real, etc.

Al implementar dicho sistema en la Universidad Pontificia Bolivariana da un paso arriba a campos nuevos en las tendencias del área de control de procesos digitales y analógicos, la cual es un área de crecimiento. Todos estos puntos hay que tenerlos en cuenta pues actualmente se abrirá una especialización en automatización y control en la universidad, lo que hace de esto una razón importante para fortalecer estos laboratorios.

2. OBJETIVOS

2.2 OBJETIVO GENERAL

- Diseñar y montar un entrenador para los controladores lógicos programables PLC S7-200 de Siemens.

2.3 OBJETIVOS ESPECÍFICOS

Diseñar la estructura del entrenador.

Seleccionar y adquirir los componentes y dispositivos que componen el entrenador.

Aprender a usar la plataforma de programación SIEMENS.

Estudiar el funcionamiento del autómatas adquirido para el proyecto.

Desarrollar pruebas de adquisición y generación de datos con el PLC.

Realizar una guía práctica que describa la configuración del controlador y su programación.

3. JUSTIFICACION

La Universidad Pontificia Bolivariana en su proceso de crecimiento y expansión ambiciona alcanzar y satisfacer la demanda a futuro de los alumnos en el laboratorio de control, en este momento se cuenta con aproximadamente seis módulos de procesos en los programas de Ingeniería Electrónica e Ingeniería Mecánica; y de las plataformas nuevas que se están implementando (Opto 22 y Compact Field Point), surge la necesidad de desarrollar proyectos con otras arquitecturas para el laboratorio de control.

Sistemas SCADA (Control Supervisorio y sistema de adquisición de datos) de la casa de SIEMENS no se han implementado en esta área, de esta manera se continua con las investigaciones que se están realizando en la Universidad en creación de módulos académicos.

Del mismo modo se pretende enseñar plataformas industriales en el laboratorio de control, tanto para los alumnos de pregrado como el de postgrado. Debido a que se está dictando la tercera cohorte del diplomado en instrumentación, automatización y control virtual, y que el próximo año inicia la especialización en control e instrumentación Industrial, este proyecto fortalecerá el laboratorio de control para el desarrollo de prácticas.

4. FUNDAMENTO TEORICO

Durante el desarrollo de este capítulo se realiza una descripción de los principales temas relacionados con el área de la instrumentación y procesos industriales, permitiendo al lector que no conozca a profundidad del tema, expandir sus conocimientos y darle una perspectiva estructurada en el tema.

4.1 Instrumentos industriales

Los elementos industriales hacen referencia a todos aquellos dispositivos eléctricos, mecánicos, electrónicos, neumáticos, hidráulicos, utilizados en un entorno industrial como parte de un proceso, cuyo uso permite la ejecución de acciones de control. Algunas de las funciones que realizan los instrumentos industriales son:

- Sensar o captar una variable.
- Acondicionar una variable dada.
- Transmitir una variable.
- Controlar una variable.
- Indicar la magnitud de una variable.
- Registrar una variable.
- Convertir una variable.
- Visualizar una alarma con respecto a la magnitud de una variable.
- Interrumpir o permitir una secuencia dada.
- Transmitir una señal.
- Amplificar una señal.
- Manipular una variable del proceso, etc.

4.2 Clasificación de los procesos industriales

Los procesos industriales se pueden clasificar de acuerdo a su localización, tecnología y aplicación, a continuación se describirá cada una de las clasificaciones:

Aplicación

- Neumáticos
- Hidráulicos
- Eléctricos
- Electrónicos
- Electromecánicos
- Mixtos
- Transductores
- Amplificadores

- Indicadores
- Analizadores
- Estación de operador
- Estación de control
- Estación de transferencia
- Relevador de cálculo.

Por localización

- Instalados en campo
- Instalados localmente
- Instalados en tablero principal
- Instalados remotamente.

Por tecnología

- Sistemas discretos
- Sistemas de control digital directo
- Sistemas de supervisión
- Sistemas de control supervisorio
- Sistemas de control supervisorio y adquisición de datos
- Sistemas de control distribuido
- Sistemas de control avanzado
- Sistemas de control adaptables
- Sistemas expertos.

4.3 Lazo de control

Un lazo de control se encuentra conformado por un elemento primario de medición (sensor) un elemento final de control (actuador), el controlador y el proceso a controlar. De acuerdo al proceso a controlar se dictaminan los sensores y actuadores a utilizar.

Figura 4.1 Representación de lazo de control.



- *Elemento primario de control*

Este elemento es el encargado de detectar el valor de la variable a medir, estos elementos pueden ser de origen eléctrico, mecánico, electrónico o una combinación de los mencionados. Entre los elementos primarios de medición más comunes para la medición de nivel, temperatura, presión y flujo están:

Tabla 4.1 . Tipos de elementos primarios

NIVEL	De Boya
	Capacitivo
	Inductivo
	Diferenciales de presión
	Para sólidos
	Para líquidos
TEMPERATURA	Termómetros Bimetálicos
	Termómetros de vástago de vidrio
	Pirómetros de radiación ópticos
	Pirómetros de radiación de infrarrojos
	Termómetros de cristal de cuarzo
	Termopares
	Resistencias eléctricas
PRESION	Tubo Bourdon
	Columnas
	Diferenciales
	Diafragma
	Fuelles
	Capsulas
	Campanas
FLUJO	Magnético
	Turbina
	Bomba Dosificadora
	Tubo de dali
	Vortex
	Remolino
	Placa de orificio
	Rotámetro

Fuente: Instrumentación Virtual Industrial. Henry Antonio Mendiburu. Perú, 2005. Pág. 20.

- *Planta*

La planta es la estructura física que conforma el proceso, está compuesto por diversos elementos físicos, algunos ejemplos de plantas son: un reactor químico, un dispositivo mecánico o un horno.

- *Proceso*

Es el desarrollo de operaciones continuas que conllevan a un resultado final (obtención de un elemento en particular).

4.4 Elemento final de control

El actuador o elemento final de control es quien recibe una señal (eléctrica, mecánica, neumática, etc) del controlador, la interpreta y ejecuta una labor que tiene como por objetivo eliminar el error y cumplir con las tareas de control propias del proceso llevado a cabo. Dentro de los actuadores más comunes se encuentran:

- Motores
- Relés
- Contactores
- Variadores de velocidad
- PLC's

4.4.1 Variador de velocidad

Este dispositivo puede ser de origen mecánico, hidráulico y/o eléctrico, es utilizado para realizar labores de control sobre la velocidad de giro de diferentes tipos de maquinas, por ejemplo motores. En la figura 4.2 se observa la fotografía física de un variador de velocidad.

Una de las principales razones de uso de este tipo de elementos es el ahorro de energía, dentro de las principales funciones de este dispositivo se tiene:

- Control de la aceleración.
- Cambio de las velocidades de operación de acuerdo a los requerimientos del proceso
- Compensación de variables en procesos variables.
- Permitir el posicionamiento de alta precisión.
- Control del par motor (torque).

Figura 4.2. Variador de velocidad



Fuente: http://www.es.sccdcwy.org/link-Variador_de_velocidad Visitada en enero 15 de 2008

4.4.2 Motores

El motor es un transductor, es decir, que de la energía de entrada, se genera energía mecánica, existen diversidad de motores, estos pueden ser:

- Motores eléctricos: El movimiento en este tipo de motores es obtenido a partir de una corriente eléctrica.
- Motores de combustión interna: Para la generación de los movimientos, estos motores utilizan combustibles como el petróleo, el alcohol, y aceites naturales como el de maíz (Biodiesel).

En la figura 4.3 se observa un motor eléctrico.

Figura 4.3. Apariencia física de un motor



Fuente: <http://www.herramientasconalco.com.ar/catalog/> Visitada en enero 18 de 2008

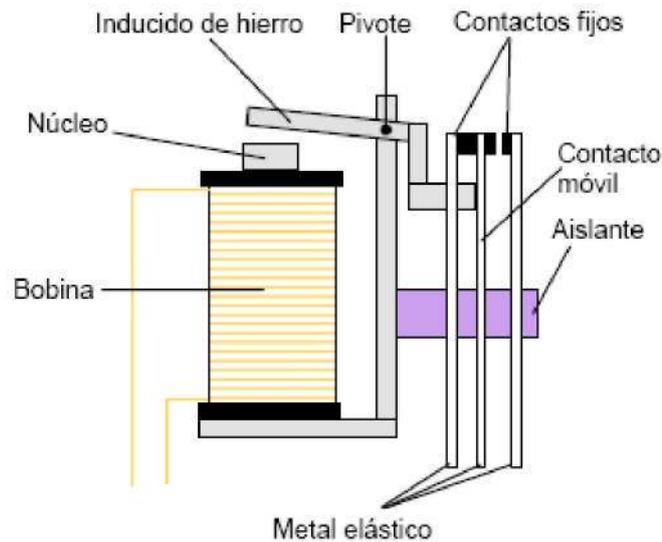
4.4.3 Relés

El relé que es un dispositivo mecánico utilizado como interruptor a partir de una pequeña tensión aplicada a su bobina. Al aplicarse una corriente sobre la bobina interna del relé, esta genera un campo magnético que acciona el interruptor mecánico. Ese interruptor es el encargado de manejar la potencia en sí, quedando al circuito electrónico la labor de "mover" la bobina. Este

dispositivo permite aislar mecánicamente la sección de potencia de la de control. Se encuentra formado por un contacto móvil o polo y un contacto fijo. Pero también hay relés que funcionan como un conmutador, porque disponen de un polo.

En la figura 4.4 se observa la estructura interna de un relé,

Figura 4.4. Estructura interna de un relé



Fuente: <http://platea.pntic.mec.es/~pcastela/tecno/documentos/apuntes/rele.pdf> Visitada en enero 20 de 2008

4.4.4 Controladores lógicos programables (PLC's)

El PLC es un dispositivo electrónico de funcionamiento digital basado en un microprocesador, que utiliza una memoria para el almacenamiento de las instrucciones de programa empleado para el control automático de maquinas y procesos, mediante la implementación de funciones específicas tales como operaciones lógicas, aritméticas, temporizaciones, cuentas y secuencias, a través de módulos de entrada y salida de tipo analógico / digital.

Los PLC se aplican en: instalaciones de aire acondicionado, calefacción, almacenamiento, cerámica, frío industrial, maquinado y retiro de virutas, plantas depuradoras de residuos, embotelladoras, seguridad, tratamientos térmicos, instalaciones eléctricas y de comando, industria de automoción, maquinaria de ensamblaje, maquinaria en procesos textiles y de confección, maquinarias en la industria del plástico, maquinaria en procesos de grava, arena y cemento, maquinaria industrial del mueble y madera, etc.

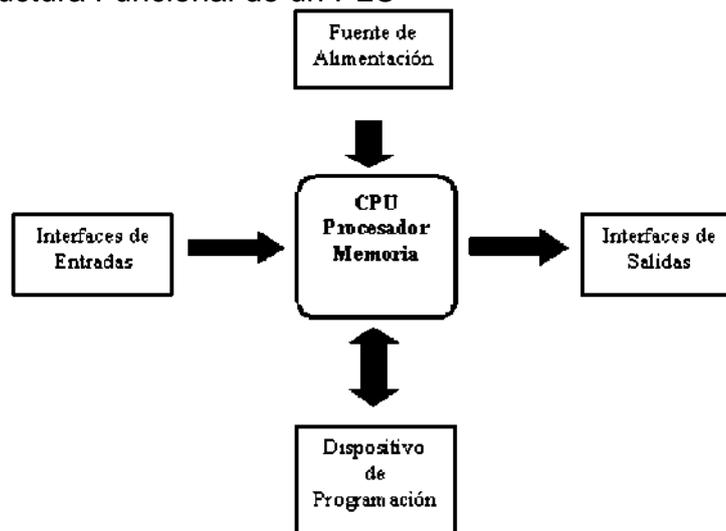
Para programar el PLC se conmuta al modo de programación y cuando se culmine dicho procedimiento, el usuario conmuta manualmente al modo ejecución a fin de que la CPU ejecute el programa en forma repetitiva. Existen PLC compactos que reúnen en una sola unidad, la fuente de poder, la CPU, la

memoria y las interfaces I/O. Estas versiones representan grandes ventajas en lo relacionado con costos, utilización de espacios reducidos; poseen un óptimo desempeño en condiciones hostiles del ambiente de trabajo, pueden ser programados mediante paquetes de software desde un computador.

La capacidad de expansión de los PLC's permiten usarse en redes de transferencia de datos semejantes al usado por los computadores, estas redes son usadas para almacenar datos desde cada maquina y generar reportes de estado de producción y performance para su manejo. Estas redes proveen comunicación entre cada PLC para el control de secuencia y sincronización de la planta.

Existen varios lenguajes de programación, tradicionalmente los más utilizados son el diagrama de escalera, lista de instrucciones y programación por estados, aunque se han incorporado lenguajes más intuitivos que permiten implementar algoritmos complejos mediante simples diagramas de flujo más fáciles de interpretar y mantener. En la figura 4.4 se observa la estructura funcional de un PLC.

Figura 1.5 Estructura Funcional de un PLC



Fuente: http://www.giotek.com.tw/sub_cat.php?cID=40&pID=17&subCID=2 Visitada en enero 21 de 2008

Existen dos formas estructurales externas para los PLC's, estas son:

- Compacta

Los elementos están acoplados una misma unidad (Figura 4.6).

Figura 4.6. Estructura externa compacta de un PLC

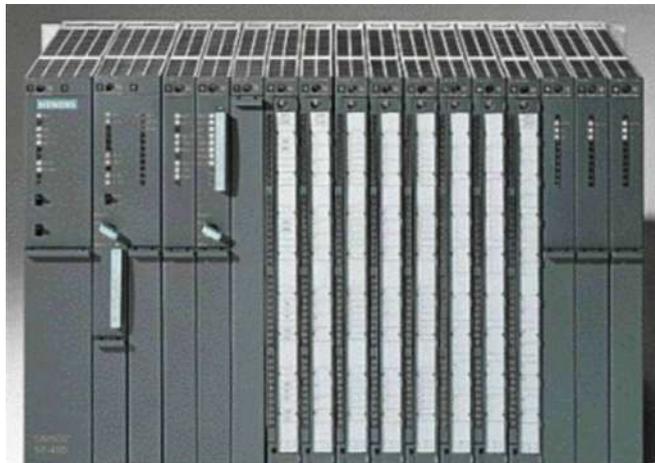


Fuente: http://www.giotek.com.tw/sub_catphp?cID=40&pID=17&subcID=2 Visitada en enero 21 de 2008

- Modular

En los que la CPU, la fuente de alimentación, las entradas, las salidas, etc., son cada una un módulo que se elige en función de la aplicación requerida. Para el caso de una estructura modular se dispone de la posibilidad de fijar los distintos módulos en raíles normalizados, para que el conjunto sea compacto y resistente.

Figura 4.7. PLC modular



Fuente: www.salesianos.edu/alcoy.juanxxiii/dpts/docs/automatas.pdf Visitada en enero 24 de 2008

El PLC (controlador lógico programable) se puede considerar como el corazón de un proceso industrial, este dispositivo reemplaza los sistemas de control basado en relés, contactores, cables, temporizadores. Un PLC se puede considerar como un computador hecho para soportar condiciones de trabajo pesado.

4.4.4.1 Partes del PLC

Sin importar el fabricante, el PLC esta conformado por:

- Sección de entradas

Esta parte corresponde a los elementos que envían datos al PLCs para que éste determine la acción a ejecutar, las entradas pueden ser de origen analógico o digital, dependiendo de los elementos de entrada (sensor, switch, pulsadores, etc).

- Sección de salidas

Son las líneas de salida, que determinan el elemento a manipular por el PLC, las salidas pueden ser de digitales o análogos.

- Unidad central de proceso (CPU)

Es la encargada de ejecutar el programa previamente desarrollado, la acción de ejecución es realizada de forma cíclica, para esto dispone de diferentes espacios de memoria, instrucciones de programa y registros. En algunos modelos de PLC (dependiendo del fabricante) se encuentran funciones integradas como controles PID, controladores de posición, reguladores, etc.

- Unidad de alimentación

Proporciona la corriente necesaria para que el PLC funcione, es decir, convierte la energía eléctrica disponible (de tipo AC en muchos casos) a niveles de tensión continua necesarios para la operación de los circuitos del procesador y las secciones de entrada y salida; algunos modelos de PLC la tienen incluida, a otros, se le tiene que adicionar como elemento externo.

- Consola de programación

Es el puerto con el que se programa, modifica, se introduce y supervisa el programa ha ser ejecutado por el PLC.

- Dispositivos periféricos

A esta sección pertenecen aquellas unidades entrada/salida (E/S), unidades de comunicaciones, y aquellos dispositivos particulares (dependiendo del fabricante).

- Interfaces

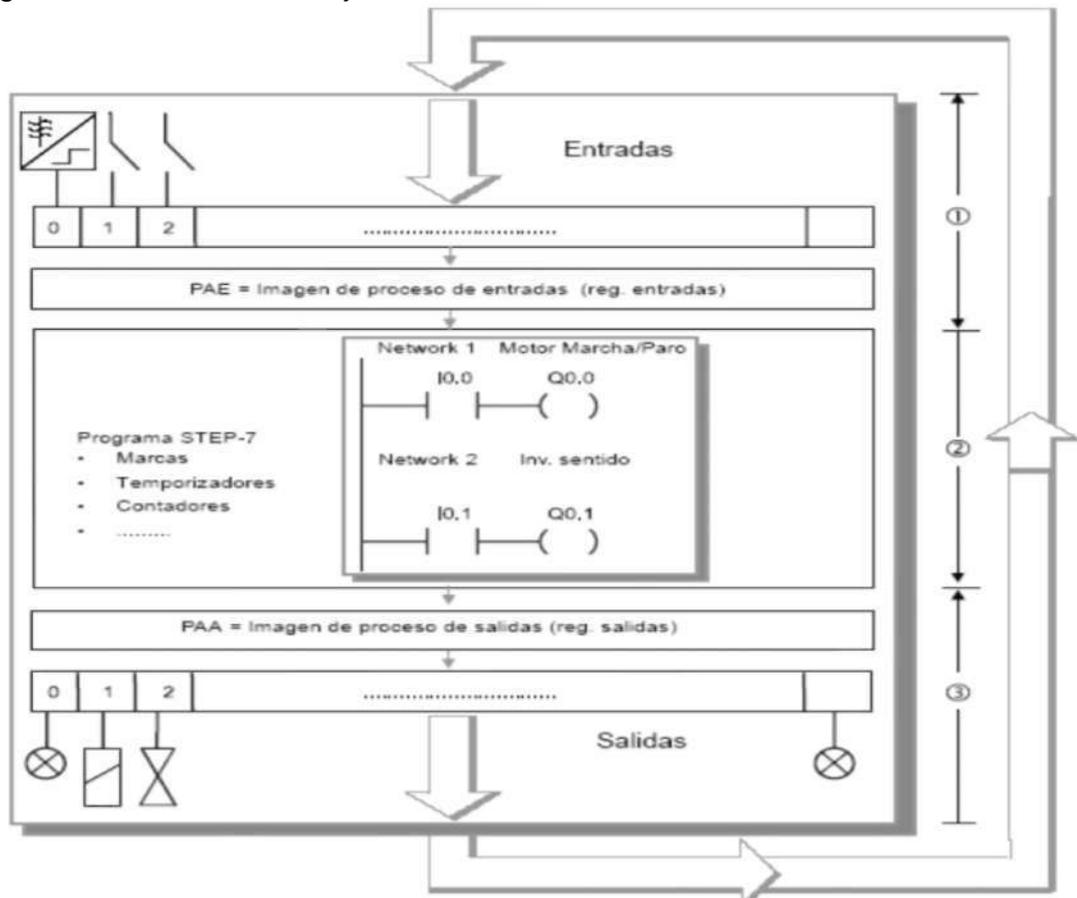
Permite la comunicación del PLC con periféricos, PC's, etc

4.4.4.1.1 CPU (Unidad Central de Procesamiento)

Es considerada como el elemento principal del PLC, es el cerebro del sistema, realiza la toma de decisiones y la transferencia de la información. En el procesador se distinguen tres bloques principales como son la CPU, la memoria y la interfaz de comunicaciones, tiene funciones como:

- Ejecutar el programa desarrollado por el usuario.
- Monitorear el tiempo de ejecución del programa desarrollado por el usuario, de manera que este no exceda su tiempo de ejecución. A esta labor se le conoce como *PERRO GUARDIAN* o *WATCHDOG*.
- Realiza una imagen de las entradas.
- Refresca el estado de las salidas, con respecto a las imágenes obtenidas al finalizar el ciclo de ejecución del programa.
- Revisar el comportamiento del sistema.

Figura 4.8. Ciclo de Trabajo de un PLC



Fuente: www.salesianos.edu/alcoy.juanxxiii/dpts/docs/automatas.pdf Visitada en enero 28 de 2008

Durante el ciclo de trabajo del PLC, como primera etapa se lee el estado en las entradas, grabándose en la imagen de proceso de estas. Con esta información el programa inicia su proceso de ejecución.

De acuerdo a la lógica desarrollada dentro de la programación se proceso con la toma de decisiones, este proceso es llevado a las salidas físicas del PLC; posteriormente se inicia un nuevo ciclo. Un ciclo tiene una duración aproximada de 3ms a 10ms, este tiempo depende del tipo y número de instrucción. El ciclo de trabajo del PLC es ejecutado solo cuando este se encuentra en modo RUN.

4.4.4.1.2 Memoria

Dentro del PLC existen diferentes tipos de memoria, y estas tiene una finalidad definida, estas son:

- Memoria del programa de usuario: esta memoria es usada para que el programa inicie a ejecutarse cíclicamente.
- Memoria de la tabla de datos: aquí se encuentra el tipo de datos, contadores, temporizadores, registros, etc
- Memoria del sistema: aquí se ejecuta el programa
- Memoria de almacenamiento: memoria externa utilizada para almacenar información en general relacionada con el programa a ejecutarse, este tipo de memoria suele ser EPROM, EEPROM o FLASH.

4.4.4.1.3 Unidades de entrada y salida (E/S, I/O)

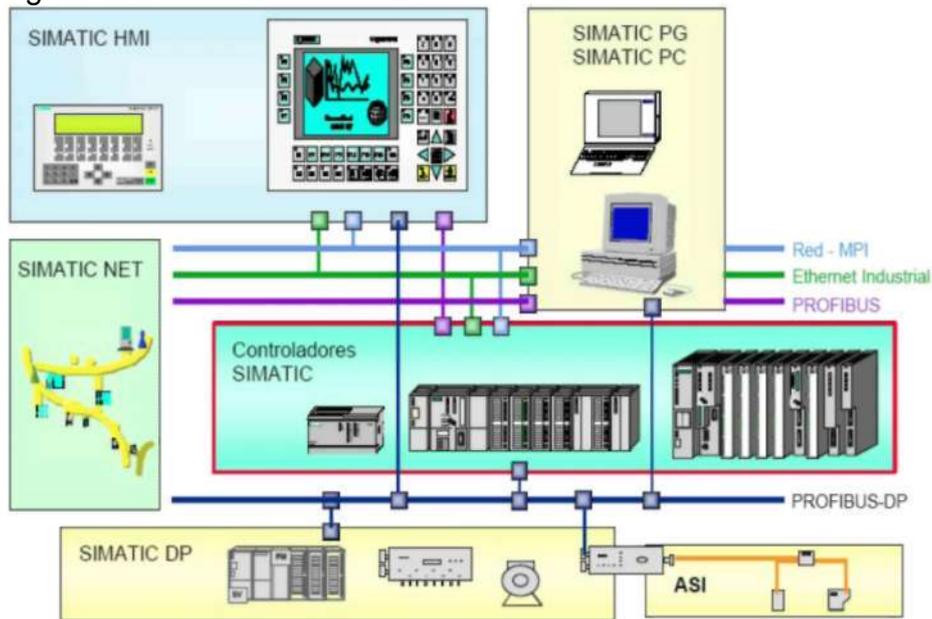
Dependiendo del origen de las entradas o salidas, se trabaja con dos referencias:

- Digitales: trabaja con niveles lógicos (1 y 0), Estas entras y salidas tienen un manejo a nivel de bit en el programa.
- Analógicas: tiene un valor cualquiera teniendo en cuenta un rango determinado especificado por el fabricante. Las señales se manejan a nivel de byte o por palabras 8 y 16 bits dentro del programa de usuario. Las entradas y salidas se leen incluyendo las imágenes dentro del área de memoria o mensajes a través de instrucciones específicas de las entradas y salidas.

4.4.4.1.4 Interfaces

Los PLC's se pueden comunicar con otros dispositivos como un computador, por lo general los PLC's poseen entradas y salidas serie del tipo RS-232, con este protocolo se maneja todas las características internas del PLC, incluyendo la programación del mismo suelen implementarse para monitorear el proceso

Figura 4.9. Interfaz de un PLC



Fuente: www.salesianos.edu/alcoy.juanxxiii/dpts/docs/automatas.pdf Visitada en febrero 1 de 2008

4.4.4.1.5 Unidades de programación

La programación de los PLC's se puede realizar de acuerdo a los siguientes elementos:

- Consola de programación: tiene forma de una calculadora
- Computador: Es la de mayor tendencia en la actualidad, se programa desde un computador.

4.4.4.1.6 Periféricos

El PLC tiene la capacidad de ser ampliado, esta posibilidad permite aumentar los módulos de entradas y salidas (analógicas/digitales), conexión a con otros PLC's, adicionar mayor memoria, etc.

4.4.4.2 Modelos de PLC's

En esta sección se describirá los modelos más representativos de PLC's de los fabricantes de mayor influencia en el mercado, con el objetivo de dar una visión con respecto a la amplia gama de PLC's que el usuario final puede seleccionar

4.4.4.2.1 Telemecanique

En 1924 se inventó el primer contactor sobre barras. Desde entonces, es especialista en productos y sistemas de automatización para la industria (protección y comando de potencia, detección de objetos, interfase hombre-máquina, autómatas programables, supervisión de procesos).¹

4.4.4.2.1.1 Zelio

Figura 4.10. Modelo Zelio Logic de Telemecanique



Fuente: <http://www.elec serrano.com.ar/schneider/plc/zelio/impresion2.htm> Visitada en Febrero 1 de 2008

Este tipo de PLC's es ideal para sistemas pequeños de automatización debido a su relé inteligente, su tamaño reducido, posee una estructura simple de programación, dado por el tipo de lenguaje que utiliza (Ladder y FBD). En la figura 4.10 se observa el modelo Zelio, Como descripciones generales de este tipo de PLC's, se tiene:

- Modelos monobloc de 10, 12, 20, entradas y salidas con visualizador o sin visualizador, tiene la opción de expandirse hasta 40 entradas y salidas.
- Programación en lenguaje de contactos Ladder o por bloque de función PDB.
- Pantalla LCD retro iluminada, con navegación contextual.
- Interfaz de comunicación con la cual se puede tener un control remoto.
- Instalación fácil.
- Especial para las industrias de procesos de materiales y plásticos, automatización de equipos auxiliares de maquinaria textil.
- Usados en el sector agrícola la automatización de bombeo, invernadero y riego etc.
- Control de acceso, automatización de cortinas eléctricas.
- Control y automatización de iluminación e instalación.
- Automatización de sistemas de aires acondicionados y compresores.

Tomado del portal electrónico: <http://www.schneider-electric.com.co/empresa/marcas.cfm>

4.4.4.2.1.2 Twido

Este tipo de PLC's se encuentra orientado a la automatización de máquinas pequeñas y instalaciones industriales, como principal característica presenta la reducción del espacio en tableros por su pequeño tamaño. La extensión de entradas y salidas y controladores ofrecen una variedad de opciones la hacer más fácil el cableado: conectores de resorte, borneras y módulos de precableado llamados Twido². En la figura 4.11 se observa este modelo.

Figura 4.11. Modelo de PLC Twido



Fuente: <http://www.elec serrano.com.ar/schneider/plc/twido/impresion.htm>. Visitado en Febrero 2 de 2008

Como las características más representativas de este modelo, se tienen:

- Entradas y salidas de nivel alto de 0V a 10V o de 4mA a 20mA.
- Entrada de bajo nivel termocuplas tipo T, K y J.
- Entradas de bajo nivel Pt 100 con rango de 100°C a 500°C.

Los módulos de salidas análogas son implementados en dirección de los preaccionadores en dispositivos físicos, entre ellos válvulas, aplicaciones que den control de procesos y variadores; tiene una resolución de 12 bits y su conexión es por medio de bornera de tornillo desconectable. Alimentación externa de 24Vcc por modulo análogo. Los módulos análogos tienen aislamiento de vías y tierra por medio optocoplador.

4.4.4.2.2 Omron

La corporación Omron, cuyas oficinas centrales están en Kyoto, Japón, es uno de los líderes mundiales en el campo de la automatización.

Establecida en 1933 y dirigida por su Presidente Hisao Sakuta, Omron tiene más de 23.000 empleados en más de 35 países trabajando para proporcionar productos y servicios en varios campos, incluyendo automatización industrial, industria de componentes electrónicos y atención sanitaria. La empresa tiene oficinas en Japón (Kyoto), Asia-Pacífico (Singapur), China (Hong Kong), Europa (Amsterdam) y EE.UU. (Chicago). La organización europea tiene sus

² Para mayor información consultar:

<http://twido.schneiderelectric.es/Twido/descargar/twido3234.pdf>

propias instalaciones de desarrollo y fabricación, y proporciona asistencia a cada cliente en todos los países de Europa.³

4.4.4.2.2.1 CPM2A

Tiene una interfaz de RS-232, permitiendo la interconexión de terminales para monitoreo de máquinas de una forma fácil, ajustar un punto, etc. Permite un sencillo mantenimiento debido a los bloques de terminales extraíbles, como expansores de entradas y salidas. En la Figura 4.12 se observa este tipo de PLC.

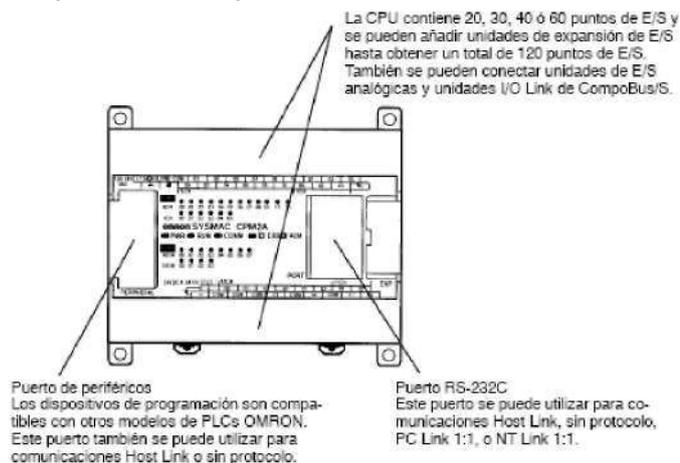
Figura 4.12. PLC CPM2A



Fuente: industrial.omron.es/es/products/catalogue/automation_systems/programmable_logic_controllers/compact_plc_series/cpm2a/default.html Visitado en febrero 4 de 2008

En este PLC existe un control sincronizado de pulso, salidas de pulso, funciones de reloj, selecciones análogas y entradas de interrupción. Este autómata es compacto de tal forma que tiene un gran y amplio rango de aplicaciones de control de máquinas, esta propiedad de ser compacto hace que se pueda integrar a la misma máquina como unidad de control. La CPU de este autómata tiene funciones de comunicación con ordenadores portátiles.

Figura 4.13. Descripción de las partes físicas del PLC CPM2A



Fuente: Autómatas programables industriales SYSMAC CPM1/CPM1A/CPM2A/CPM2C pdf

³ Tomado de: http://industrial.omron.es/es/company_info/about_omron/default.html Visitado en febrero 4 de 2008

En la figura 4.14 se observa las descripciones básicas de los modelos CPM2A.

Figura 4.14. Descripción del moldeo CMP2A



Número de puntos de E/S	Fuente de A.	Entradas	Salidas	Modelo
20 puntos de E/S (12 entradas y 8 salidas)	100 a 240 Vc.a.	24 Vc.c.	Relé	CPM2A-20CDR-A
	24 Vc.c.	24 Vc.c.	Relé	CPM2A-20CDR-D
		24 Vc.c.	Transistor negativo común o NPN	CPM2A-20CDT-D
		24 Vc.c.	Transistor positivo común o PNP	CPM2A-20CDT1-D
30 puntos de E/S (18 entradas y 12 salidas)	100 a 240 Vc.a.	24 Vc.c.	Relé	CPM2A-30CDR-A
	24 Vc.c.	24 Vc.c.	Relé	CPM2A-30CDR-D
		24 Vc.c.	Transistor negativo común o NPN	CPM2A-30CDT-D
		24 Vc.c.	Transistor positivo común o PNP	CPM2A-30CDT1-D
40 puntos de E/S (24 entradas y 16 salidas)	100 a 240 Vc.a.	24 Vc.c.	Relé	CPM2A-40CDR-A
	24 Vc.c.	24 Vc.c.	Relé	CPM2A-40CDR-D
		24 Vc.c.	Transistor negativo común o NPN	CPM2A-40CDT-D
		24 Vc.c.	Transistor positivo común o PNP	CPM2A-40CDT1-D
60 puntos de E/S (36 entradas y 24 salidas)	100 a 240 Vc.a.	24 Vc.c.	Relé	CPM2A-60CDR-A
	24 Vc.c.	24 Vc.c.	Relé	CPM2A-60CDR-D
		24 Vc.c.	Transistor negativo común o NPN	CPM2A-60CDT-D
		24 Vc.c.	Transistor positivo común o PNP	CPM2A-60CDT1-D

Fuente: Autómatas programables industriales SYSMAC CPM1/CPM1A/CPM2A/CPM2C pdf

4.4.4.2.2.2 CS1

Este autómata CS1 está disponible en dos velocidades de procesador y cada una con su diferente capacidad de memoria, hay dispuestas versiones para operaciones redundantes dual que admitan el intercambio de funcionamientos de entradas y salidas, estos autómatas tiene una ranura para una tarjeta dedicada con una conexión de bus de CPU directa, la cual se puede montar una tarjeta de comunicaciones con una de control de lazo, estas CPUs admiten lenguaje de diagrama de relés. Los bloques de programación ayudan a la programación y al tiempo el programador puede hacer sus propios bloques según lo requiera el. En la figura 4.15 se observa este modelo.

Figura 4 .15. PLC CS1

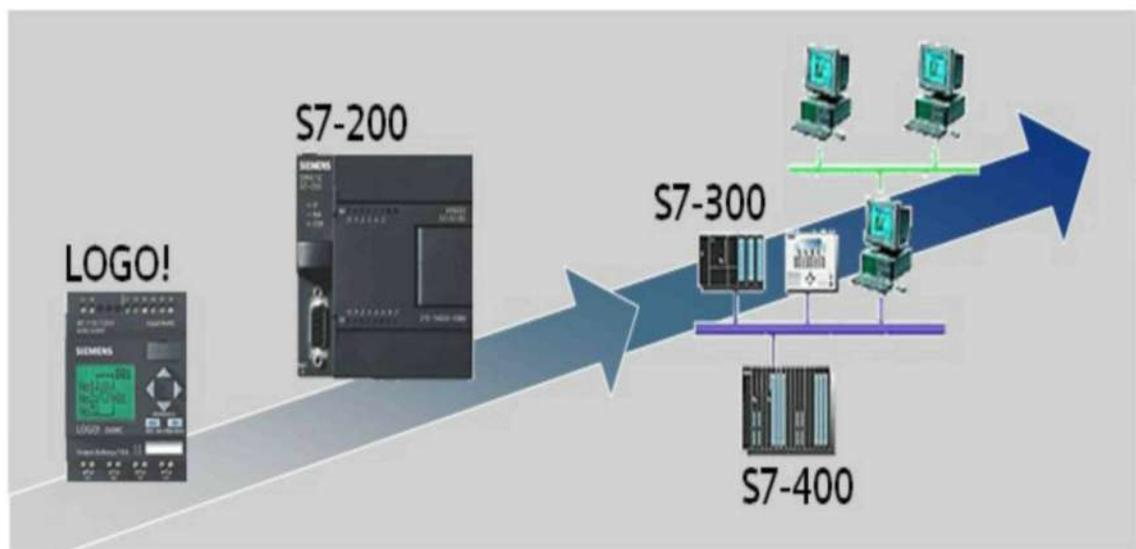


Fuente:http://industrial.omron.es/es/products/catalogue/automation_systems/programmable_logic_controllers/rack_plc_series/cs1d/cpu_units/default.html Visitada en febrero 4 de 2008

4.4.4.2.3 Siemens

Siemens es una empresa de origen Alemán, con gran versatilidad y trayectoria en el campo de los PLC's, tiene una gran amplia de modelos, todos ellos buscando solucionar las necesidades del consumidor final.

Figura 4.16 Avances tecnológicos de los PLC`s



Los PLC`s de Siemens se dividen según características del proceso industrial, y las diferentes clases que se encuentran en el mercado son el PLC`s S7-200 (Precio económico, Software propio (MicroWin), Posibilidad de expansión de hasta siete módulos, "Paquete total" el equipo contiene fuente de alimentación, CPU y entradas/salidas), PLC`s S7-300 (tiene una configuración

modular, compuesta por fuente de alimentación, CPU, modulo de señales, cable de bus PROFIBUS, cable PG) y el PLC`s S7-400.

Figura 4.17 PLC`s S7-200



Figura 4.18 PLC`s S7-300



Figura 4.19 PLC`s S7-400



4.4.4.2.3.1 S7-200

Esta gama de S7-200 es la que comprende los Micro- PLC's, este modelo de autómata cambia el estado de sus salidas dependiendo del programa ejecutado, incluye operaciones de lógica booleana, contadores y temporizadores, aritméticas, también incluye operaciones complejas como la

comunicación con otros autómatas o aparatos inteligentes. Tiene un diseño compacto, configuración flexible y tiene un amplio juego de operaciones.

Figura 4.20 Modelos de la serie S7-200

S7-200 CPU	Order Number
CPU 221 DC/DC/DC 6 entradas/4 salidas	6ES7 211-0AA23-0XB0
CPU 221 AC/DC/relé 6 entradas/4 salidas de relé	6ES7 211-0BA23-0XB0
CPU 222 DC/DC/DC 8 entradas/6 salidas	6ES7 212-1AB23-0XB0
CPU 222 AC/DC/relé 8 entradas/6 salidas de relé	6ES7 212-1BB23-0XB0
CPU 224 DC/DC/DC 14 entradas/10 salidas	6ES7 214-1AD23-0XB0
CPU 224 AC/DC/relé 14 entradas/10 salidas de relé	6ES7 214-1BD23-0XB0
CPU 224XP DC/DC/DC 14 entradas/10 salidas	6ES7 214-2AD23-0XE0
CPU 224XP AC/DC/relé 14 entradas/10 salidas de relé	6ES7 214-2BD23-0XE0

Figura 4.21 . PLC S7-200



Fuente: http://user.chol.com/~sis1226/product_info/s7200.htm visitada el 5 de febrero del 2008

Figura 4.22 Modulo de expansión del S7-200



Como las características más representativas de este modelo, se destacan:

- Soportes de hardware de las CPU: para parar la edición en modo RUN. La CPU tiene entradas y salidas integradas y dos puertos de comunicaciones, en algunas versiones se incluyen filtros de entradas y captura de impulsos.
- Soporte de cartucho de memoria: El PLC transfiere al cartucho de memoria comparaciones y selecciones de programa.
- Trabaja sobre el software el STEP 7-Micro/WIN (versión 4.0) de 32 bits.

4.4.4.2.3.2 S7-300

Como tecnología de CPU, tiene funciones de monitoreo y control, estas se encuentran integradas en una CPU SIMATIC. Este tipo de PLC es ideal para:

- Control de tareas y necesidades en materia de tecnología principalmente relacionados con control y monitoreo en el SIMATIC S7-300
- Propuesta de tareas hasta ocho ejes acoplados o único ejes

Figura 4.23. PLC S7-300



Fuente: <http://www.automationtraining.ca/siemens.asp> visitada el 5 de febrero del 2008

Tabla 4.2 Clases de PLC`s S7-300

CLASES DE PLC'S	CPU313	CPU 314	CPU 315 Í315-2 DP)	CPU316-2DP	CPU 318-2
Memoria usuario/datos	4 K/- 12KB	8 K/- 24 KB	1 G (21)K/- 48 (G4) KB	42 K/- 128 KB	84 K/256 KB 512KB
Direcc. libre.	no	no	si	si	si
ED/SD	25G	1024	1024	1024	1024
EA/SA	64	256	256	256	256
Tiemp. trat. 1K inst. binaria	0,6 ms	0,3 ms	0,3 ms	0,3 ms	0,1 ms
Marcas	2048	2048	2048	2048	8192
Contadores	64	64	64	64	512
Tiempos	128	128	128	128	512
Comunicación			Conexión PROFIBUS-DP maestro/esclavo (con 64 estaciones DP)	Conexión PROFIBUS-DP maestro/esclavo (con 64 estaciones DP)	Conexión PROFIBUS-DP (maestro/esclavo; 125 estaciones DP); MPI útil, como DP (12MBaud)

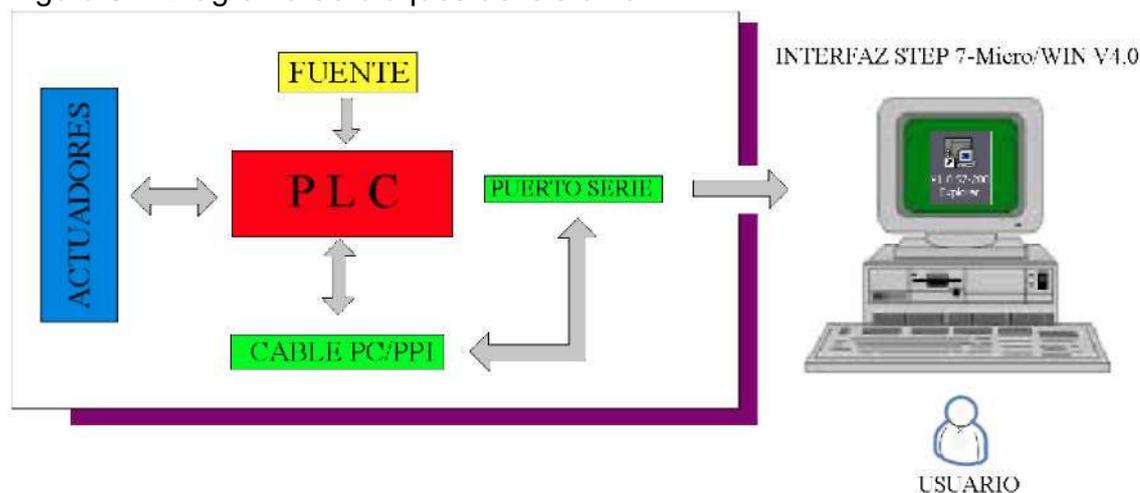
Este modelo está diseñado para máquinas de flujo, el procesamiento y montaje de líneas, equipos de etiquetado, su entrada de tambor o pórticos simples (sin interpolación).

5. DISEÑO Y CONSTRUCCION DEL ENTRENADOR

El objetivo principal del diseño del entrenador es de tipo práctico, ya que por medio de este, el estudiante tiene un contacto real con los procesos industriales.

El entrenador se diseño con una estructura de hierro la cual tendrá una forma de un escritorio donde se tendrán dos niveles horizontales, el nivel inferior el cual servirá para soportar un motor, el nivel superior es para trabajo del usuario donde puede colocar un PC o cualquier dispositivo que necesite para el desarrollo de su actividad, un panel vertical en donde se va a montar los respectivos elementos que conforman este entrenador que son el PLC S7-224xp de siemens, la fuente de poder, un grupo de relé, un variador de velocidad, un simulador de entradas, borneras de potencia y control y este entrenador contara con un cable multimaestro RS-232/PPI que el encargado de la comunicación entre el PLC y el PC de este modo el usuario pueda hacer uso de autómeta y programarlo de la forma que el usuario lo requiera. En la siguiente figura se muestra un diagrama de bloques del sistema.

Figura 5.1 Diagrama de bloques del sistema

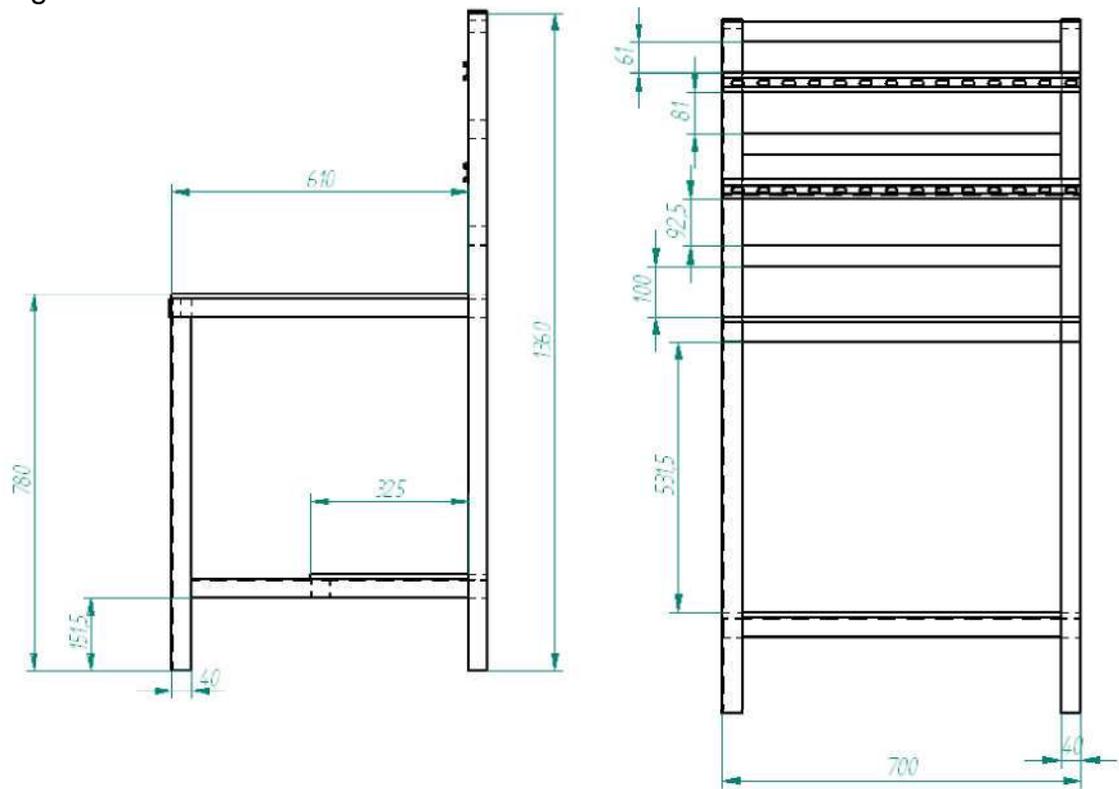


En este diagrama de bloques se puede ver como el usuario desde la PC puede programar el autómeta mediante el cable PC/PPI y por medio del software STEP 7-Micro/WIN V4.0, al igual se ve la alimentación del sistema y que el PLC se encarga de hacer activar los actuadores que son los relés que tienen unos indicadores, o hacer la activación del Variador de velocidad para poder tener manejo sobre el motor.

5.1 Chasis del entrenador.

El chasis del entrenador se creó para tener un espacio en el cual se puede manipular de forma libre los elementos de uso. Para la fabricación del chasis se utilizó hierro DE 4mm de espesor.

Figura 5.2 Dimensiones del chasis



Las dimensiones que se utilizaron son mostradas en la figura 5.2

Figura 5.3 Estructura completa del entrenado

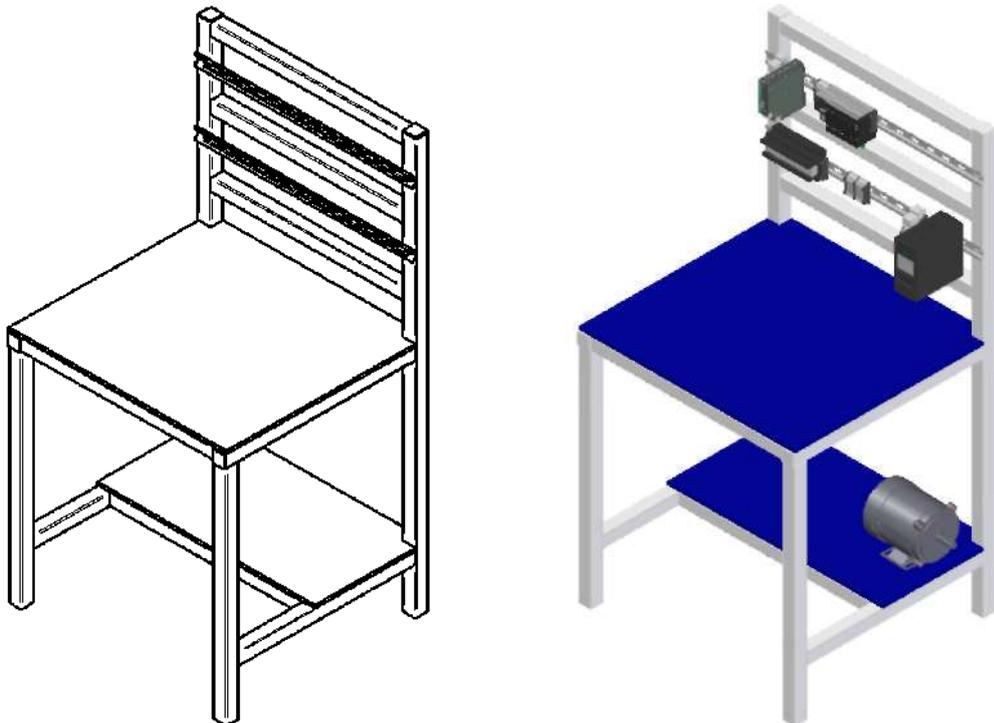
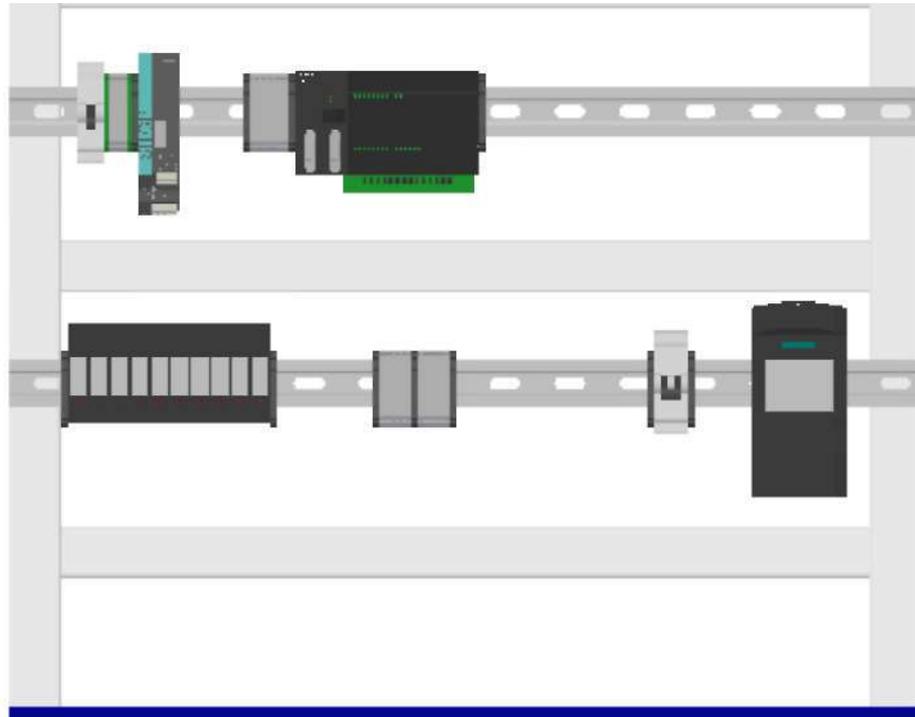


Figura 5.4 Panel frontal



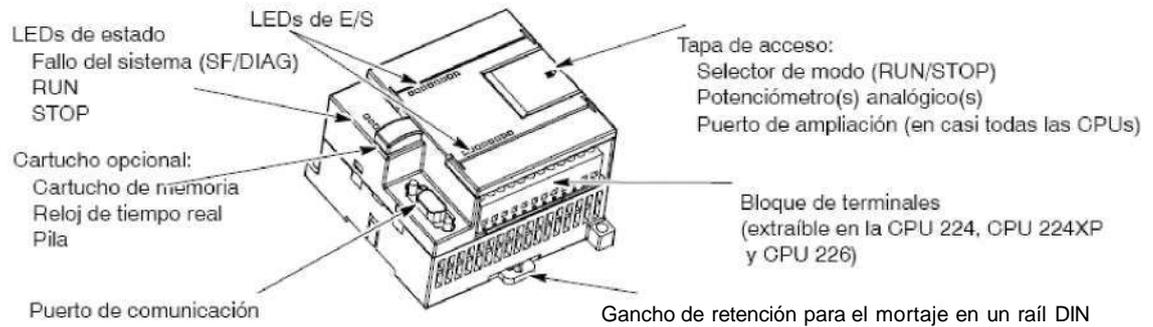
5.2 PLC S7-224XP

Figura 5.5 PLC S7-224XP

El modelo de CPU 224XP tiene una alimentación nominal de 24Vc.c, contiene 14 entradas digitales, 10 salidas digitales, 2 puertos COM, 2 entradas analógicas, 1 salida analógica, contiene un bloque de terminales extraíbles.

Sus dimensiones físicas es de 140mm de largo, 80mm de ancho y 62mm de profundidad; con un peso de 390g. La capacidad de disipación es de 8W. A una tensión de +5Vc.c. produce una corriente continua de 600mA y cuando la tensión aumenta a +24Vc.c. se produce una corriente continua de 280mA.

Figura 5.6 Especificaciones técnicas



CPU 224XP	
Memoria	
Tamaño del programa de usuario (EEPROM) con edición en modo RUN sin edición en modo RUN	12286bytes 16384 bytes
Datos de usuario (EEPROM)	10 240 bytes (remanentes)
Respaldo (condensador de alto rendimiento) (pila opoional)	Típ. 100 horas (mín. 70 horas a 40°C) Típ. 200 días
Entradas y salidas (E.-S)	
E/S de ampliación	14 E/10 S
E/S analógicas	2 E/1 S
Tamaño de la imagen de E/S digitales	256 (128 E/128 S)
Tamaño de la imagen de E/S analógicas	64 (32 E/32 S)
N- máx. de módulos de ampliación	7 módulos ¹
N=máx. de módulos inteligentes	7 módulos ¹
Entradas de captura de impulsos	14
Contadores rápidos Fase simple Dos fases	6 contadores (total) 4 a 30 kHz 2 a 200kHz 3 a 20 kHz 1 a 100 kHz
Salidas de impulsos	2 a 100 kHz (sólo en salidas c.c.)

Dsitos generales	CPU 224XP
Temporizadores	256 temporizadores en total: 4 temporizadores de 1 ms, 16 temporizadores de 10 ms y 236 temporizadores de 100 ms
Contadores	256 (respaldo por condensador de alto rendimiento o pila)
Marcas internas almacenadas al desconectar la CPU	256 (respaldo por condensador de alto rendimiento o pila) 112 (almacenamiento en EEPROM)
Interrupciones temporizadas	2 con resolución de 1 ms
Interrupciones de flanco	4 flancos positivos y/o 4 flancos negativos
Potenciómetros analógicos	2 con resolución de 8 bits
Velocidad de ejeouoióon booleana	0,22 ns por operación
Reloj de tiempo real	Incorporado
Cartuchos opcionales	Memoria y pila
Comunicación integrada	CPU 224XP
Puertos (potencia limitada)	2 puertos RS-485
Velocidades de transferencia PPI , DP/T	9,6, 19,2 y 187,5 kbit/s
Velocidades de transferencia Freeport	1.2 kbitfs a 115.2 kbit/s
Longitud máx, del cable por segmento	Con repetidor aislado: 1000 m hasta 187,5 kbit/s, 1200 m hasta 38,4 kbitfs Sin repetidor aislado: 50 m
N- máximo de estaciones	32 por segmento, 126 por red
N- máximo de maestros	32
Punto a punto (modo maestro PPI)	Sí (NETR/NETW)
Enlaces MPI	4 en total. 2 reservados (1 para una PG y 1 para un OP)

Figura 5.7 Conexión de la alimentación del PLC S7-224XP

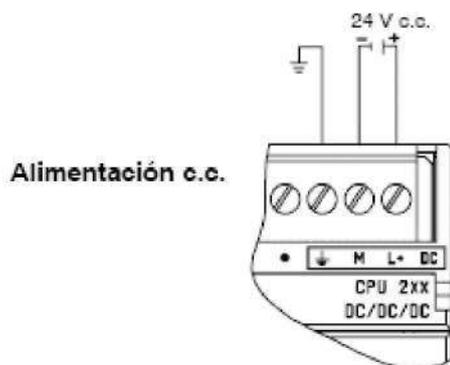
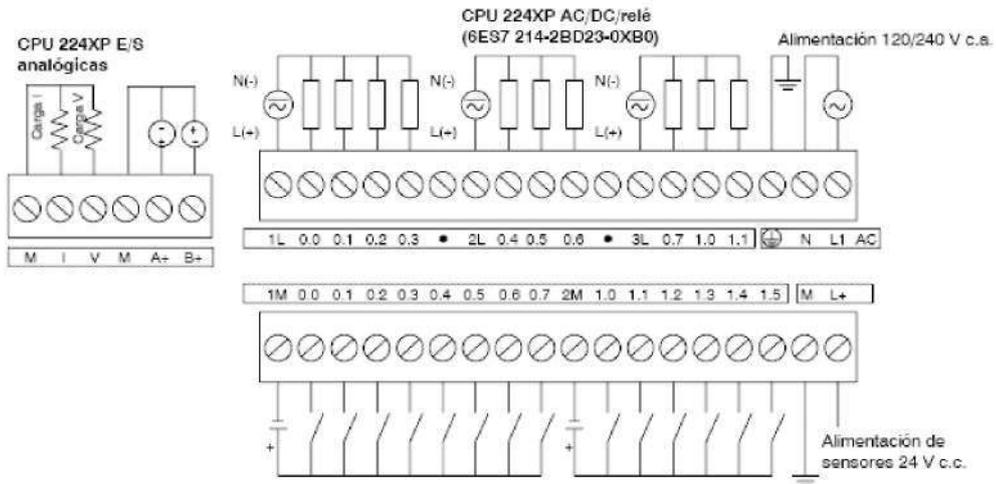


Figura 5.8 Diagrama de cableado de la CPU 224XP



5.3 Cable de maestreo

Para conectar el cable directamente al PC:

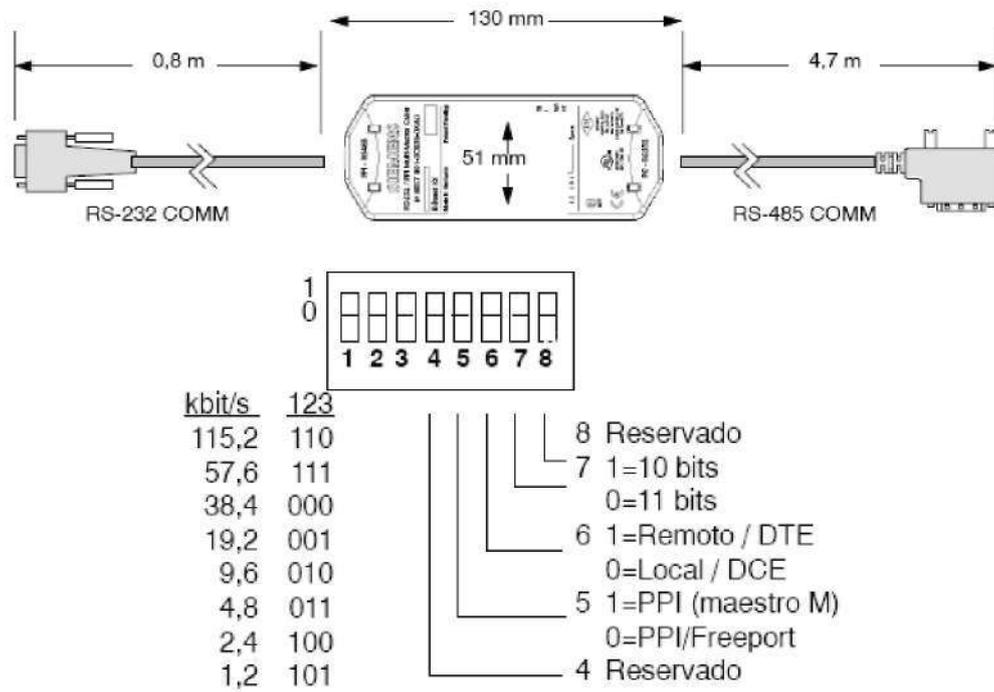
1. Ajuste el modo PPI (interruptor 5=1).
2. Ajuste el modo local (interruptor 6=0).

Para conectar el cable a un módem:

1. Ajuste el modo PPI (interruptor 5=1).
2. Ajuste el modo remoto (interruptor 6=1).

La figura 5.8 muestra las dimensiones, la etiqueta y los LEDs del cable multimaestro RS-232/PPI S7-200.

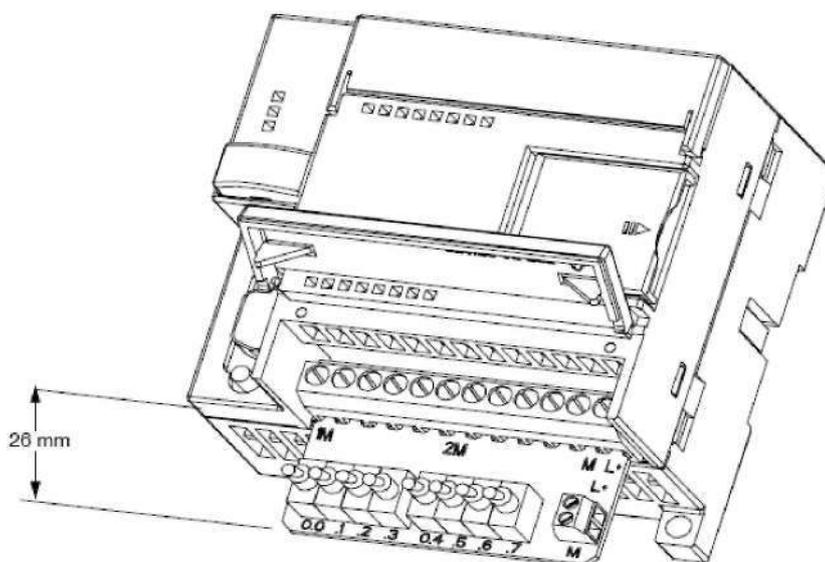
Figura 5.9 Estructura del cable de muestreo.



LED	Color	Descripción
Tx	Verde	Indicador de transmisión RS-232
Rx	Verde	Indicador de recepción RS-232
PPI	Verde	Indicador de transmisión RS-485

5.4 Simulador de entradas

Figura 5.10 Simulador de entrada.



Nº de referencia	Simulador de 14 entradas 6ES7 274-1XH00-0XA0
Dimensiones (l x a x p)	91,5 x 36,5 x 22 mm
Peso	0,03 kg
Entradas	14

5.5 Modulo de salida

Esta parte del entrenador está compuesta por 3 partes que juntas hacen optimizar las salidas del PLC, teniendo un modulo indicador del estado de las salidas como accesorios potencia industrial Relay RT / RP / SR2M, un modulo indicador PTM L0 524 EM07 red LED 6...24 VDC/VAC y un relé RT2 y a continuación veremos las dimensiones de los elementos.

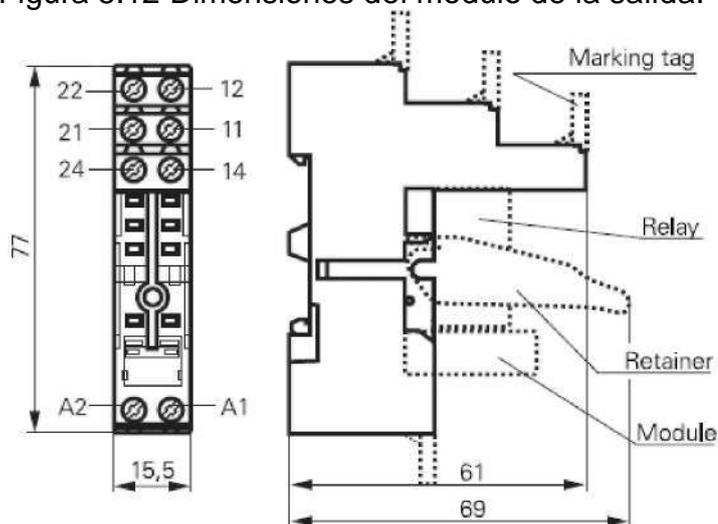
5.5.1 Accesorios potencia industrial Relay RT / RP / SR2M

RT 78 726 Socket con bornes de tornillo, depositadas 5 mm para montaje en carril DIN

Figura 5.11 Base del modulo de la salida.



Figura 5.12 Dimensiones del modulo de la salida.



5.5.2 PTM L0 524 EM07 red LED 6...24 VDC/VAC

Figura 5.13 Dimensiones del PTM L0 524 EM07.

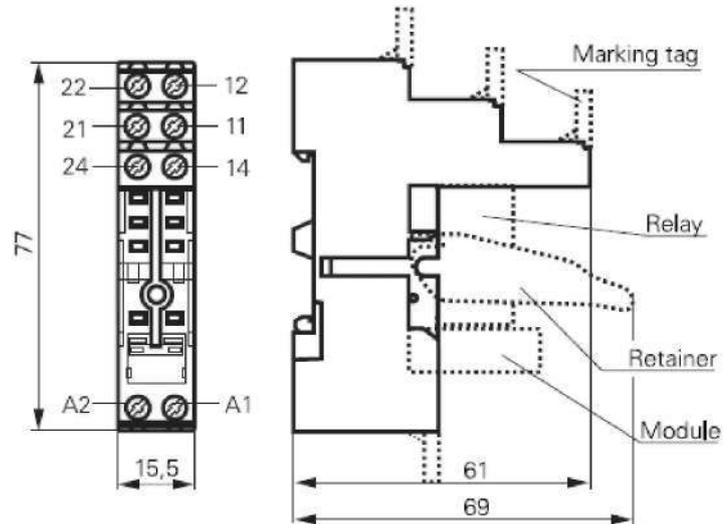


Figura 5.14 Modulo indicador PTM L0 524 EM07 red LED 6...24 VDC/VAC.



5.5.3 Relé RT2

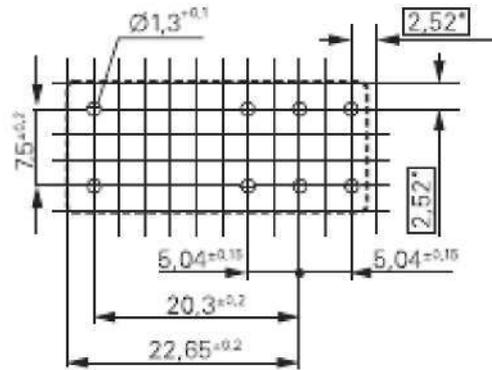
Figura 5.15 Rele RT2



Especificaciones del rele RT2:

- 2 polo 8 A, 2 o CO 2 NO contactos o DC-AC-bobina.
- Sensible bobina de 400 mW.
- Aislamiento reforzado.
- WG versión: Producto de acuerdo a IEC60335-1.
- Compatible con RoHS (Directiva 2002/95/CE).
- Como por producto, código de fecha 0413.

Figura 5.16 Diseño PCB / terminal de cesión vista inferior a la soldadura de las patillas



Con el PCB recomienda un tamaño de agujero de la red patrón de 2,5 mm a 2,54 mm puede ser utilizado.

Figura 5.17 Diseño PCB / terminal de cesión vista inferior.

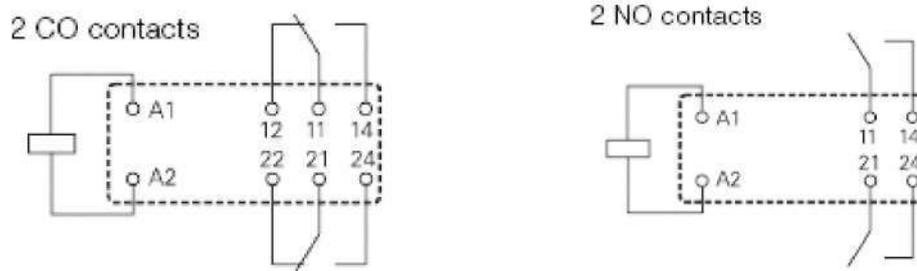
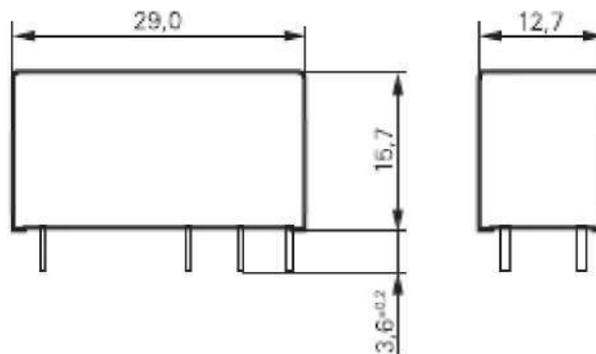


Figura 5.18 Dimensiones del Relé.



5.6 Fuente de alimentación

Figura 5.19 Fuente de alimentación.



Tabla 5.1 Datos técnicos

SITOP	24VDC/2.5 ^a
ORDEN	6EP1332-2BA10
Nominales de tensión de entrada - Rango	120/230 V AC 85...132/170...264 V AC
Línea buffering	> 20 ms (at 93/187 V AC)
Frecuencia nominal de red	50/60 Hz
Irrupción actual (25 ° C)	< 27 A
Recomendado en miniatura para romper el circuito	A partir de 3 A características C
Nominal de voltaje de salida	24 V DC
Tolerancia	± 3%
Rango de ajuste	22.8...28 V DC
Corriente de salida nominal	2.5 A (3 A up to +45 °C)
Eficiencia en valores nominales aprox.	85%
Puede conmutarse en paralelo para impulsar el rendimiento	Si
Electrónica corto-circuito protección	Sí. Constante, aprox. 1,3 x potencia nominal actual. Extra Potencia: 1,5 x potencia nominal actual durante 5 segundos / minuto
RI especificación (EN 55022)	Clase B
Línea de la limitación de armónicos en conformidad con EN 61000-3-2	No se aplica
Grado de protección EN 6052	IP20
Temperatura ambiente (sin derating)	0...+60 °C
Dimensiones (W x alto x D) en mm	32.5 x 125 x 125
Peso, aprox.	0.4 kg
Certificados	CE, UL, CSA, GL; ATEX; Ubicación peligrosos Clase I Div 2 Grupos A, B, C & D, T4

5.7 Variador de velocidad

El convertidor MICROMASTER 420 se puede usar en numerosas aplicaciones de accionamiento de velocidad variable. Es especialmente idóneo para aplicaciones con bombas, ventiladores y en manutención y transporte. Este convertidor se caracteriza particularmente por su funcionalidad. El gran margen de tensión de alimentación de la red permite aplicarlo en todo el mundo.

Figura 5.20 MICROMASTER 420



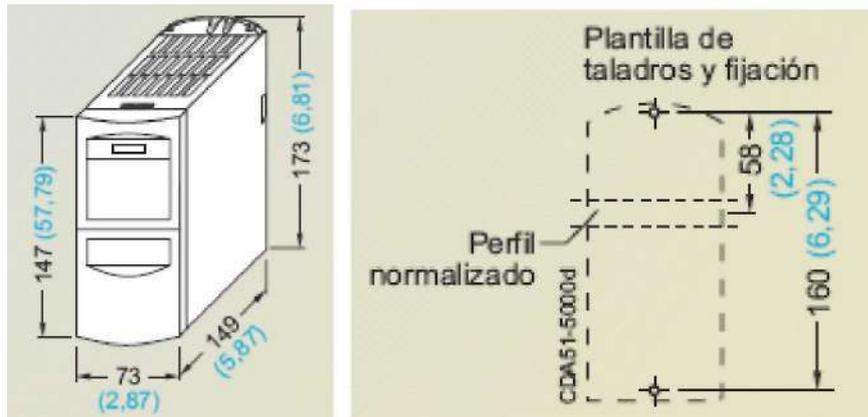
5.7.1 Construcción

El convertidor MICROMASTER 420 tiene una estructura modular. Los paneles de operador y los módulos de comunicaciones se pueden sustituir sin herramienta.

5.7.2 Características principales

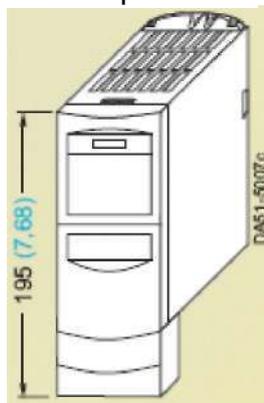
- Puesta en servicio rápida y sencilla
- Configuración particularmente flexible gracias a la construcción modular
- 3 entradas digitales libremente parametrizables y aisladas galvánicamente
- Una entrada analógica (0 V a 10 V, escalable); a elección aplicable como cuarta entrada digital
- Una salida analógica parametrizable (0 mA a 20 mA)
- Una salida a relé parametrizable (DC 30 V/5 A carga óhmica; AC 250 V/2 A carga inductiva)
- Funcionamiento silencioso del motor gracias a altas frecuencias de pulsación, ajustable (dado el caso, observar la reducción de potencia)
- Protección para motor y convertidor.

Figura 5.21 Tamaño constructivo 1/3 AC 200 V a 240 V 3 AC 380 V a 480 V



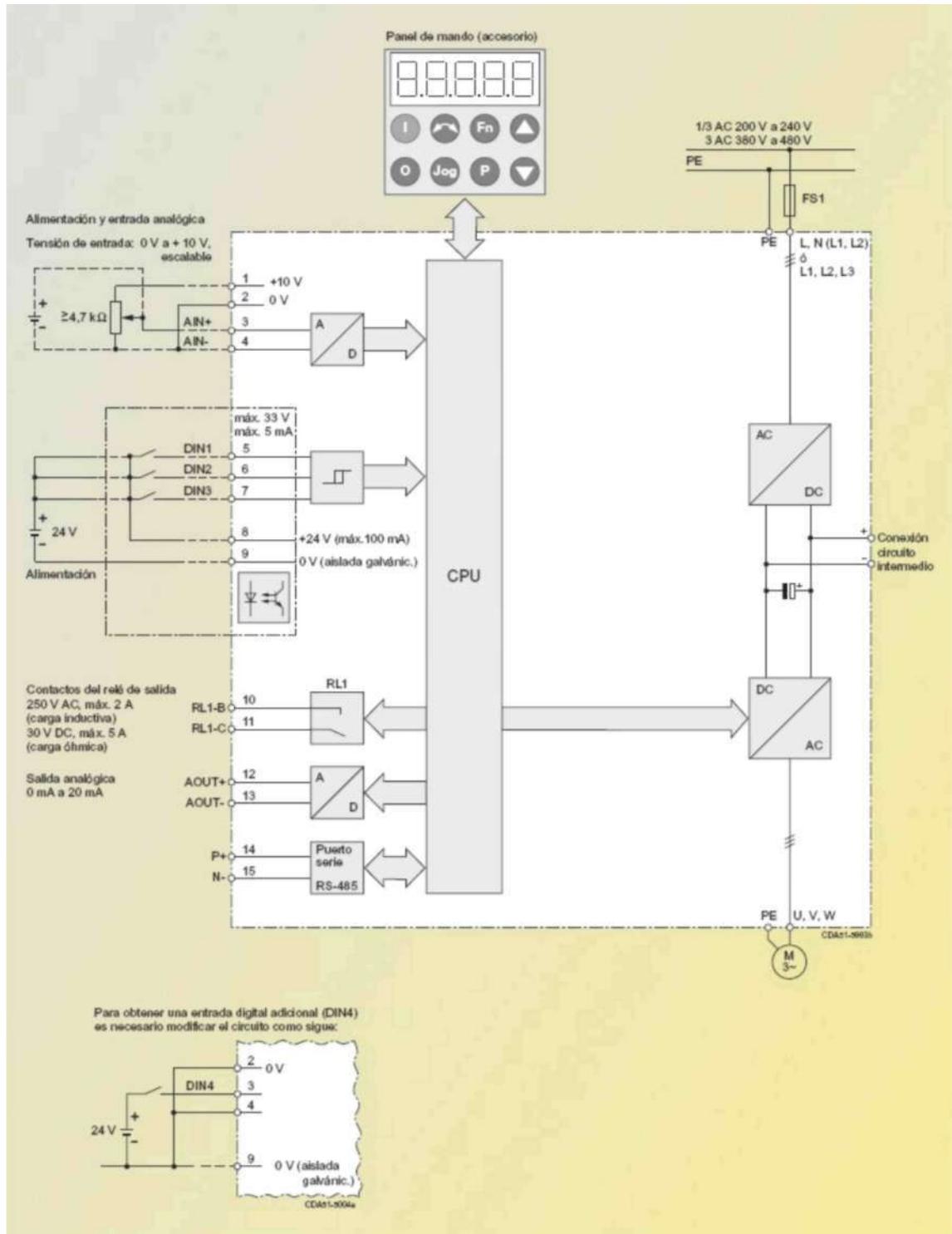
Fijación con
2 tornillos M4, 2 tuercas M4,
2 arandelas M4
o abrochado en perfil
Par de apriete con arandela colocada: 2,5 Nm
Espacio libre necesario para ventilación arriba y abajo: 100 mm

Figura 5.22 Conexión de la placa con pantallas



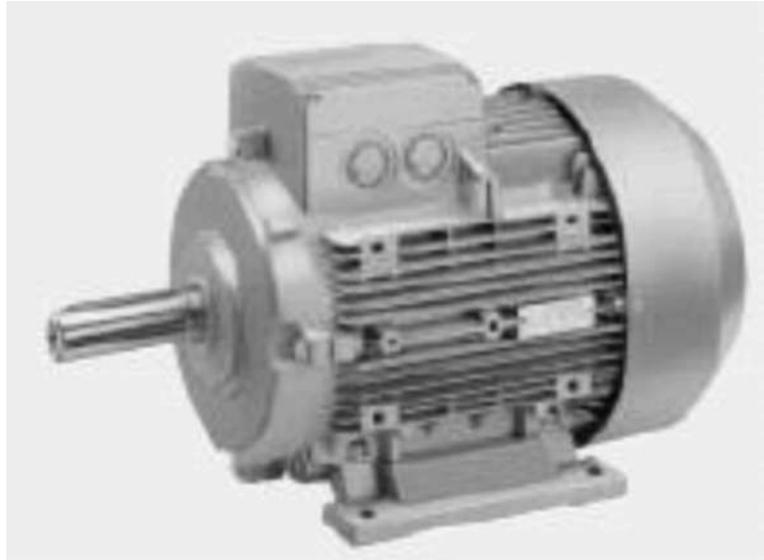
5.8 MICROMASTER 420

Figura 5.23 Esquemas de conexiones



5.9 Motor.

Figura 5.24 Imagen del motor



Ejecuciones más habituales de motores 1LA7 disponibles de almacén. Los escudos de los tipos 1LA7 de almacén a partir del tamaño 90 son de fundición; en el resto de los casos son de aluminio. Las protecciones de ventilador son de chapa de acero.

Tabla 5.2 Datos técnicos del motor

Tamaños constructivos	56 a 225 M
Potencias (4 polos)	Potencias (4 polos)
Clase F	Utilización en clase B
Alto rendimiento	Clases de eficiencia eff1 y eff2
Apropiado para funcionamiento con convertidor	Tiempos de rampa $t_S > 0,1 \mu s$ con U 500 V

5.10 Borneras.

Para la parte del cableado y acondicionamiento de todos los dispositivos implementados, se dispuso trabajar con borneras de marca *Legrand*, que tiene como característica principal buena calidad, fácil manejo, gran manejo eléctrico esperado.

Figura 5.25 Borneras porta fusibles. Diferentes vistas.



Figura 5.26 Borneras de uso general

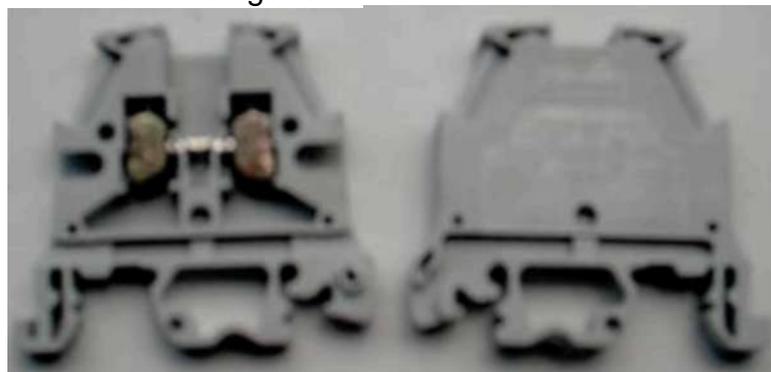
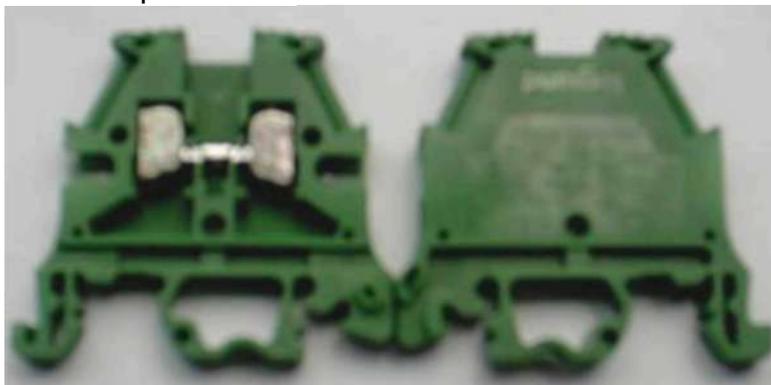


Figura 5.27 Borneras para tierra



5.11 Conectores de cables para conexión.

Estos ponchadores facilitan y garantizan una conexión fuerte, compacta y segura en el cableado industrial. Evitan los cortos por desdeñamiento de hilos del cable que se presenta tan comúnmente en todas las conexiones. Además le dan la posibilidad de marcar los cables, por medio de una lengüeta anexa que tiene cada uno, de ahí se partió para escoger la marcación lo que facilita la no confusión en el momento de identificar cada dispositivo.

Figura 5.28 Conectores para cable



5.12 Marcaciones para borneras y cables.

Las marcaciones para borneras son diferentes a la de los cables. Para borneras se adquirió marcación marca *Dexon* ya que son de uso fácil y económico, además tienen una presentación agradable al usuario y son resistentes a altas temperaturas.

La marcación de los cables es marca *Telemecanique*, muy conocida en la industria, por su resistencia a altas temperaturas, por tener una marcación ajustada, difícil de desprender, desteñirse, derretirse, etc.

Garantizando una mayor seguridad a la hora de hacer un cambio de cables, arreglo o algún tipo de reparación. En casos de incendio, difícilmente se pierden pues están hechos de un material especial que no deja perder la marcación para que se vuelva a construir o investigar de qué conexión provino el error.

Estas marcaciones además de ser de alta calidad son de gran ayuda, ya que sirven como guía de excelente visibilidad en el momento de su utilización por parte del usuario.

Figura 5.29 Marcaciones para borneras y cable



6. DESARROLLO METODOLÓGICO

Durante el planteamiento de este capítulo se describirá los aspectos tenidos presentes durante la elaboración del entrenador, se hablara de aspectos técnicos relacionados con el software de desarrollo, materiales utilizados, se detallara un registro fotográfico, etc, con el objeto de dar al lector una mejor comprensión del proceso de creación e implementación del entrenador.

6.1 Descripción general del Software STEP 7-Micro/WIN V4.0

Este es un software desarrollado por SIEMENS, que permite el desarrollo de programas para la familia de PLC's SIEMENS, se debe contar los siguientes requisitos de sistemas para que STEP 7-Micro/WIN V4.0⁴ tenga un desarrollo normal:

Sistema operativo: Windows NT, Windows 2000 o Windows XP (Professional o Home), se deberá ejecutar en sesión con privilegios de *ADMINISTRADOR*.

100 MB libres en el disco duro (Como mínimo)

Mouse (recomendado)

512MB mínimo

Unidad de CD ROM

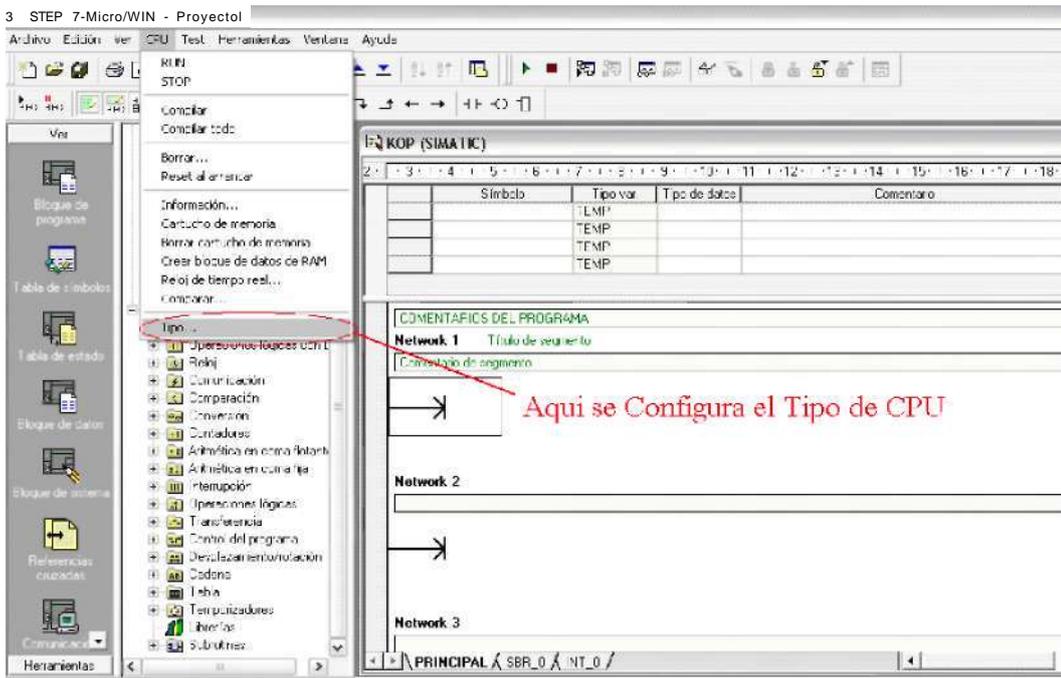
Como procesador se requiere de un PENTIUM 3 o superior, para el caso de los procesadores AMD se requiere ATLHON o superior

En la figura 6.1 se observa el entorno de trabajo del software, como se puede apreciar en dicha figura, la GUI es interactiva y con una buena interfaz graficas, además de que con el tiempo resulta intuitiva para el usuario.

⁴ Para mayor información consultar:

<http://support.automation.siemens.com> Visitada en febrero 4 de 2008

Figura 6.1. Entorno de trabajo del software STEP 7-Micro/WIN V4.



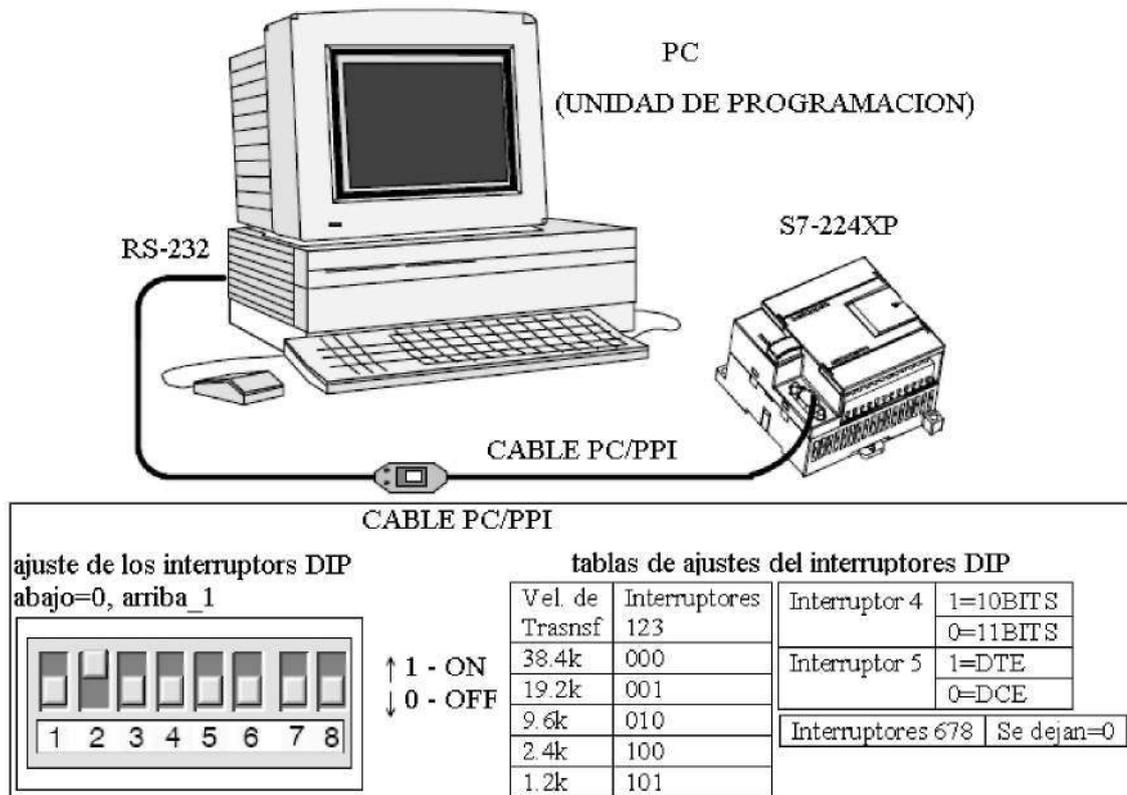
6.2 Conexión del computador y la CPU S7-224

La Conexión del computador y la CPU S7-224 se realiza por medio de un PC/PPI⁵, se debe realizar el siguiente proceso:

- Ajustar los interruptores DIP para la velocidad de transferencia así como los bits auxiliares de comunicación (“DCE” y “11 bits”), figura 6.2.
- Conectar el extremo correspondiente al PC (RS-232) del cable PC/PPI.
- Conectar al puerto de comunicación de la CPU el cable PC/PPI (RS-485) que es el vínculo con PPI.

⁵ Para mayor información consultar: <http://sputnik.epsj23.net/~omiro/Memoria.htm> Visitada en febrero 4 de 2008

Figura 6.2. Comunicación del PC y el PLC



Fuente: <http://sputnik.epsj23.net/~omiro/Memoria.htm> Visitada en febrero 4 de 2008

6.3 Instalación del software STEP 7-Micro/WIN V4

Una vez con el CD de instalación que proporciona SIEMENS en el momento que se adquiere el PLC, se procede con la instalación; al introducir el CD en la unidad, se inicia un programa automático de instalación (AutoRUN)

6.4 Ejecución del Software

Finalizado el proceso de instalación, se crea dos accesos directos, uno en el escritorio y el otro en menú de todos los programas, en este caso se accede a programa dando doble clic sobre el icono del menú de programas como se detalla en la figura 6.3. Una vez se inicialice el programa este tiene un entorno de menús gráficos e iconos (figura 6.4), se cuenta con barra de herramientas, menú, pestañas de eventos, ventana de programación, etc

Figura 6.3. Acceso al software

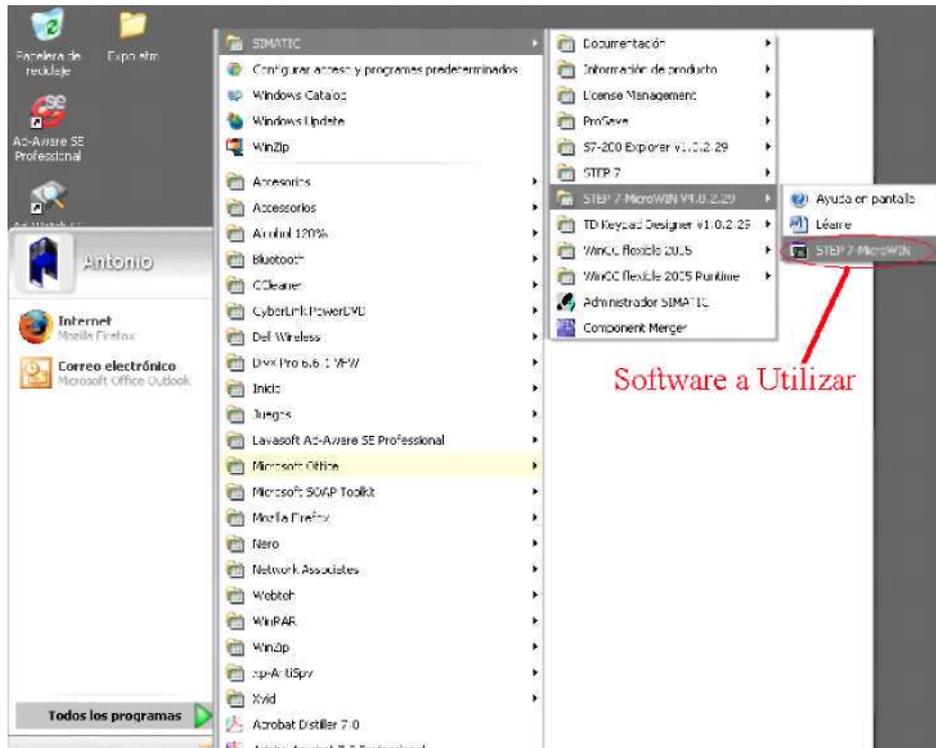
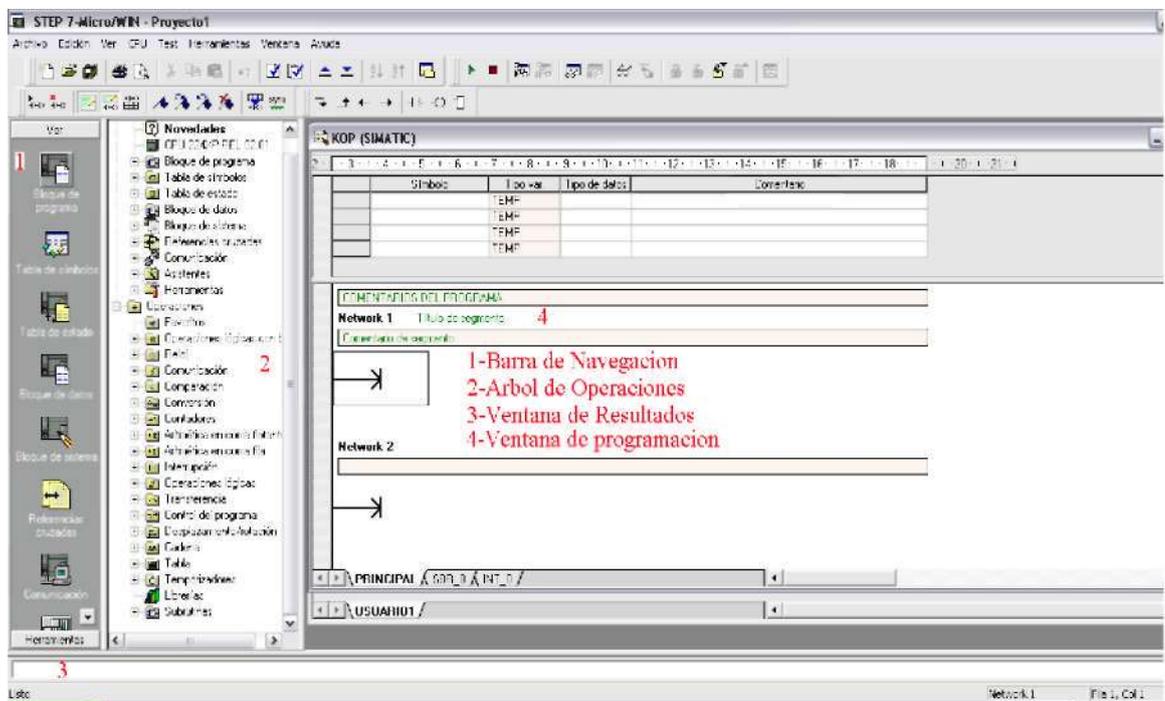


Figura 6.4. Entorno de trabajo del software STEP 7-Micro/WIN V4



A continuación se describen los principales componentes el entorno del Software:

1. Barra de navegación: Permite acceder de forma rápida a las opciones más comunes.
2. Árbol de operaciones: Se encuentra los comandos de programación aceptadas por el PLC.
3. Ventana de resultados: Se visualiza el estado de la compilación del programa, errores, etc.
4. Ventana de programación: Es la zona que posee las divisiones para las líneas de programación (Networks); sobre estas se estructura el programa a desarrollar dependiendo del tipo de lengua que se elija (AWL, KOP, FLIP),

Una vez con el software ejecutándose, se dirige a la configuración del tipo de CPU, esto se realiza en el menú *CPU*, opción *TIPO* (figura 6.5).

Figura 6.5. Selección del tipo de CPU a utilizar

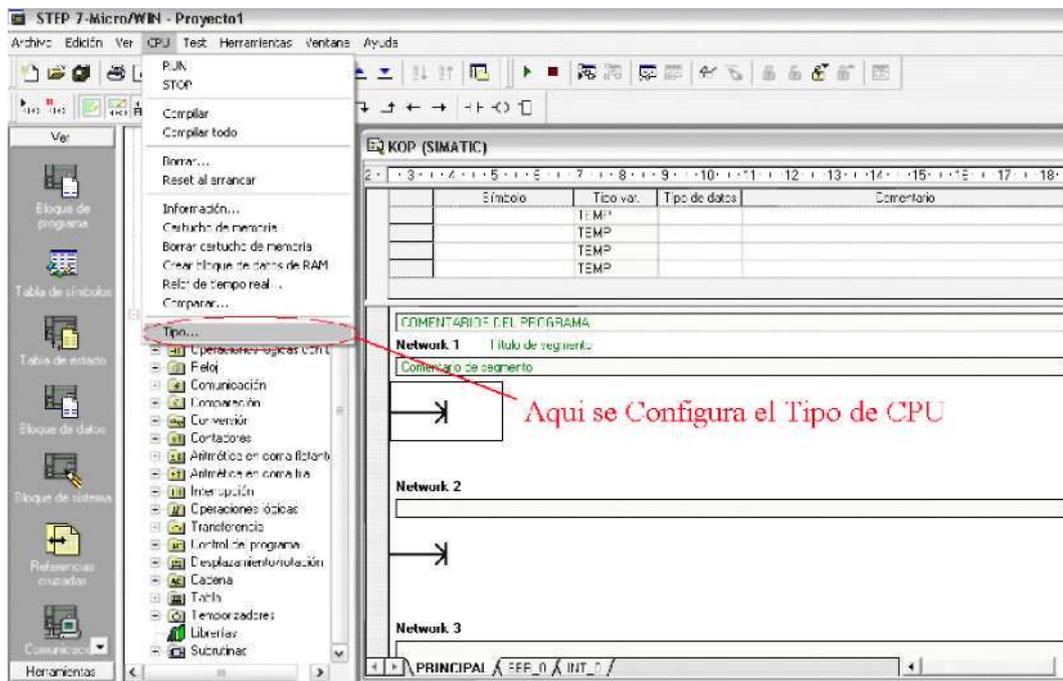
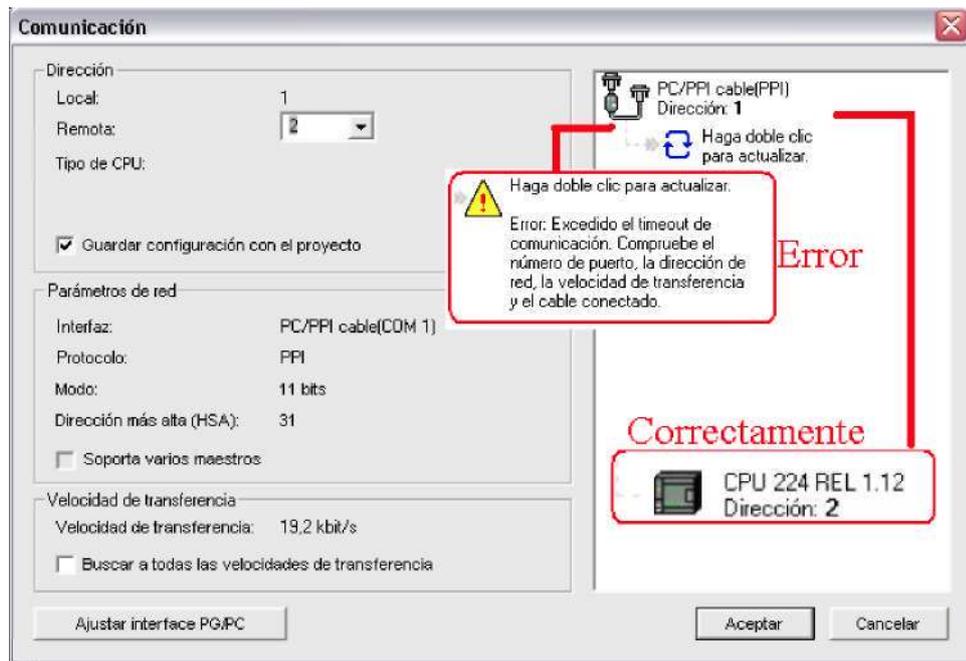


Figura 6.6. Configuración del TIPO DE CPU



Posteriormente se debe inicializar el proceso de comunicación del computador y el PLC, para ello en la ventana de CPU (Figura 6.6) se da clic en el botón COMUNICACIÓN, posteriormente se despliega una venta en la cual se configura los parámetros de comunicaciones (Figura 6.7)

Figura 6.7. Ventana de configuración de la comunicación entre el PLC y el computador



Al realizar clic en el botón ACTUALIZAR, se realiza el reconocimiento del PLC hacia el computador, de no ocurrir esto, se debe establecer la comunicación de forma manual, para ello se da doble clic sobre el botón CABLE PPI (figura 6.8)

Figura 6.8. Ventana de configuración del tipo de Cable de comunicación entre el computador y el PLC



Se selecciona la opción *PC/PPI cable (PPI)*, y se da clic en el botón propiedades, en la ventana que aparece realizada esta operación se configura los siguientes parámetros (figura 6.9):

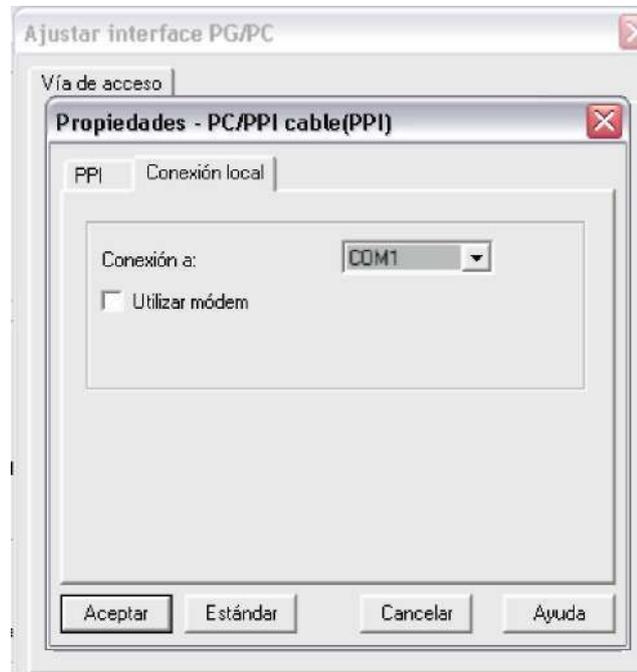
- Dirección de CPU 1.
- Timeout 1 s.
- Velocidad de transferencia 19.2 kbits/s.
- Dirección de estación más alta 31

Figura 6.9. Configuración de los parámetros PC/PPI



Al dar clic en la pestaña *conexión local* (figura 6.10), se selecciona el puerto que la interfaz va a utilizar, esta etapa es muy importante porque de lo contrario la comunicación nunca se implementará.

Figura 6.10. Selección del puerto de comunicación.



Finalizado esa operación, y al dar clic en **ACEPTAR** en la ventana **PROPIEDADES PC/PPI**, nuevamente se da clic en el botón **ACTUALIZAR**, con esta operación ya el PC debe reconocer el PLC, en caso de que el intento se fallido, hay que revisar las conexiones físicas (cableado del PC al PLC, configuración del DIP SW).

6.4.1 Introducción de órdenes

Durante el lapso de este tema se describirá los elementos, funciones, etc más relevantes para el desarrollo de programas. El software trabaja con el lenguaje KOP.

6.4.1.1 Baras de Herramientas

En la figura 6.11 se observa los elementos de barra de herramientas, dichos elementos son explicados en la tabla 6.1.

Figura 6.11. Barra de operación o de herramientas

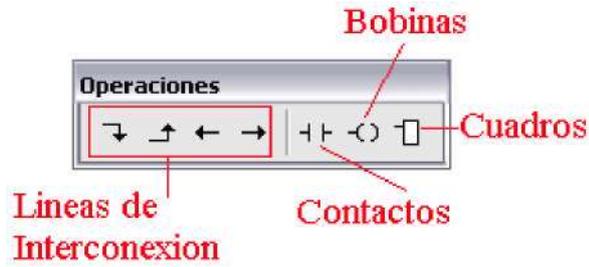


Tabla 6.1. Descripción de los elementos de la barra de operación

ICONO	DESCRIPCION
^J INSERTA LINEA HACIA ABAJO	Permite interconectar elementos en sentido inferior
L£J INSERTA LINEA HACIAN ARRIBA	Permite interconectar elementos en sentido superior
LtJ INSERTA LINEA HACIA IZQUIERDA	Permite interconectar elementos en la derecha
L^J INSERTA LINEA HACIA DERECHA	Permite interconectar elementos en hacia izquierda
^ INSERTA CONTACTO	Permite la insertar diferentes tipos de contactos,
Lili INSERTA BOBINA	Permite la insertar diferentes tipos de bobinas
"o INSERTA CUADRO	Permite la insertar diferentes tipos de cuadros

En la figuras 6.12 hasta la figura 6.14 se observa cada uno de los contactos, bobinas y cuadros que permiten insertar el software.

Figura 6.12. Tipos de Contactos a insertar **0**

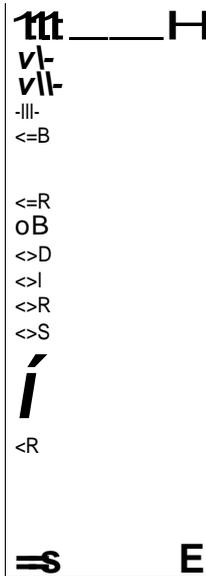


Figura 6.13. Tipos de bobinas a insertar **-0**



Figura 6.14. Tipos de cuadros a insertar



Para acceder a estos elementos de forma rápida con teclado, se tiene la siguiente equivalencia:

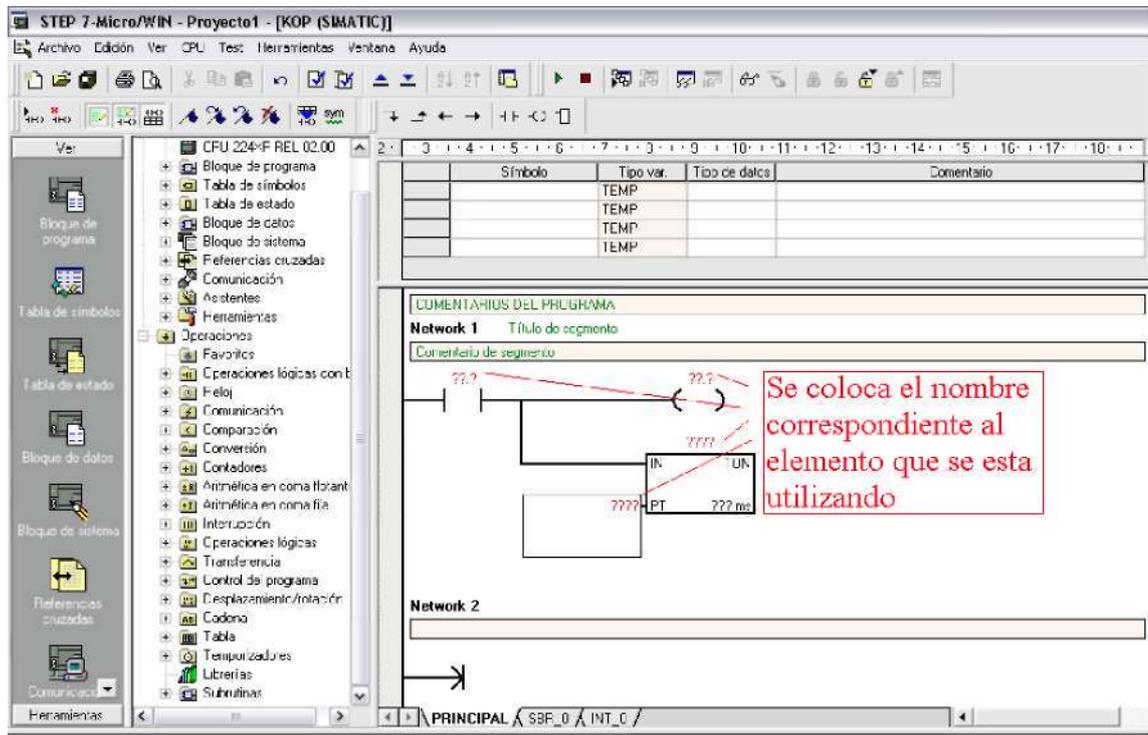
F4: Contacto

F6: Bobina

F9: Cuadro

Al seleccionar el elemento a introducir, se le debe asignar un nombre, para ello se adiciona interrogantes en la parte superior del elemento y se teclea la estructura. Para realizar combinaciones (serie, paralelo, mixto.) de funciones/elementos se utilizan las *líneas*, que permiten realizar ramificaciones

Figura 6.15. Inserción de código de programación



6.4.1.2 Barras de Herramientas Común

Esta barra permite la insercción de segmentos u cometarios.

Figura 6.16. Barra de Herramientas Común

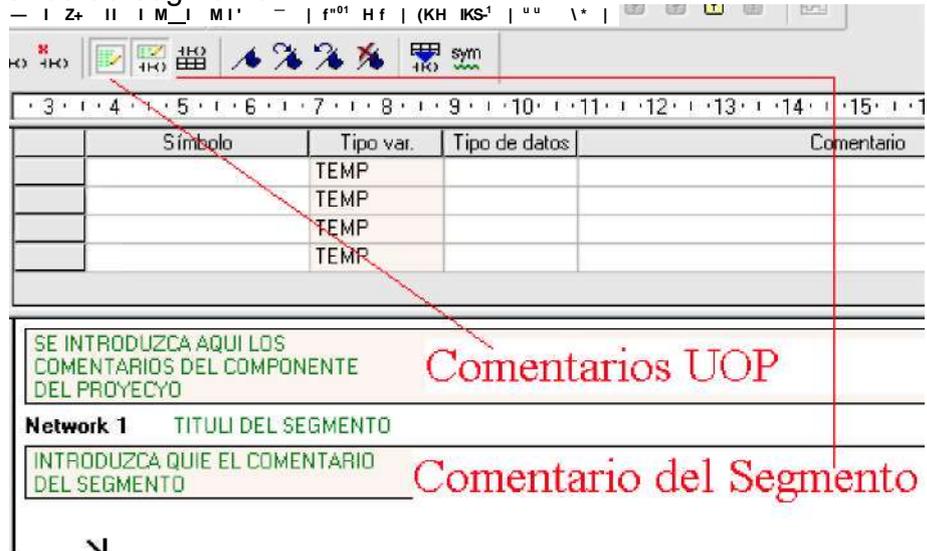


Tabla 6.2. Descripción de los elementos de la barra de herramientas común

ICONO	DESCRIPCION
 Insertar Segmento	Agrega un segmento a la estructura de programación
 Borra segmento:	Elimina aquellos segmentos no deseados
 Comentario UOP	Permite la introducción de comentarios de las unidades de organización del programa
 Comentario de segmento	Permite la inserción de comentarios para identificar las sección del programa
 Mostrar/ocultar las tablas de símbolo de información de cada segmento	Visualiza ó oculta la información del símbolo
 Aplicar todos los símbolos al proyecto	Actualiza el proyectos con los cambios de nombres, etc realizados
 Mostar ó ocultar marcadores	Visualiza ó oculta un marcador en el segmento donde se encuentre el cursor.
 Marcador Siguiente	Se desplaza hacia el siguiente marcador de una forma rápida
 Marcador Posterior	Se desplaza hacia el anterior marcador de una forma rápida
 Borra marcadores	Elimina los marcadores del programa

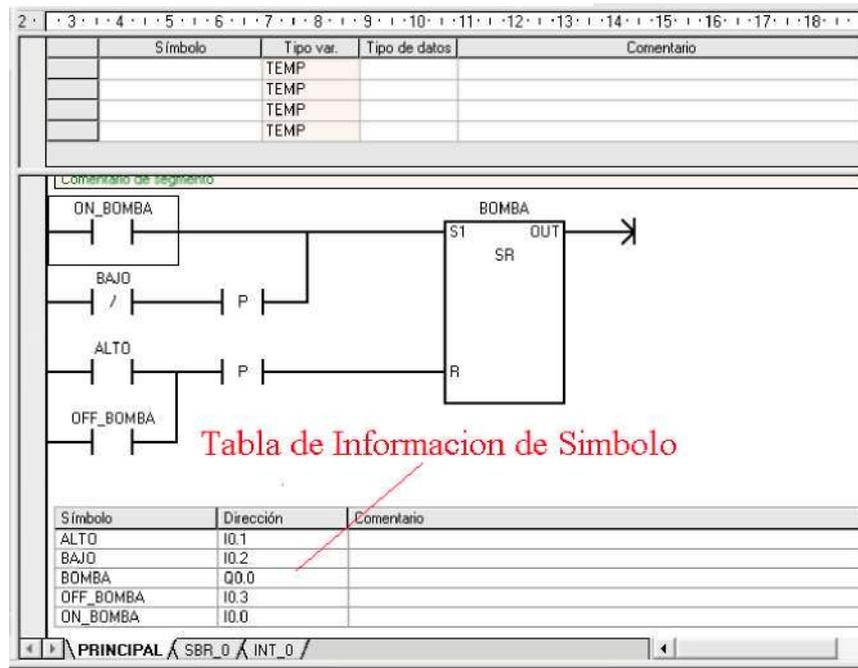
En la figura 6.17 se observa la presentación de la aplicación de los botones de los comentarios UOP y comentarios de segmento.

Figura 6.172. Aplicación de los botones de los comentarios UOP y comentarios de segmento.



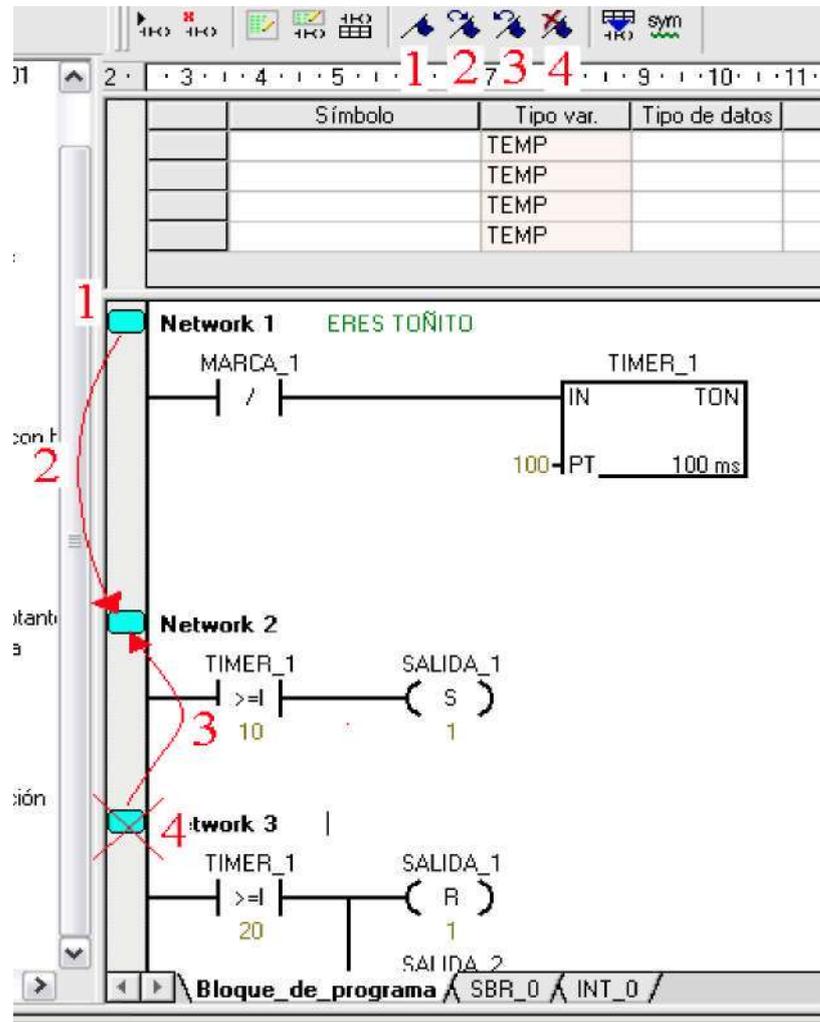
En la figura 6.18 se observa la tabla de símbolos.

Figura 6.18. Visualización de la tabla de símbolos



Las funciones mencionadas con marcadores tienen una aplicación durante el desarrollo de un programa, este aspecto se observa en la figura 6.19.

Figura 6.19. Aplicación de las funciones de marcadores dentro de un programa



6.4.1.3 Barras de Herramientas Estándar

Esta barra permite la elaboración de las funciones básicas de un programa, permite imprimir, crear nuevo archivo, editar, visualizar, etc, su apariencia se observa en la figura 6.20.

Figura 6.20. Barra de herramientas Estándar



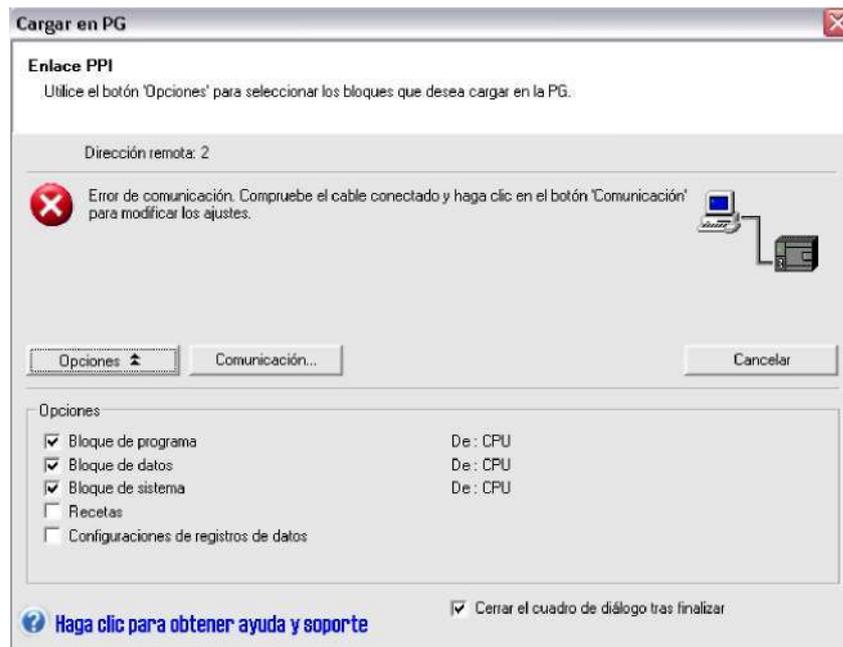
En la tabla 6.3 se observa la descripción de los elementos que componen la barra de herramientas.

Tabla 6.3. Descripción de los elementos de la barra de herramientas estándar

ICONO	DESCRIPCION
 Compilar el bloque del programa o bloque de datos	Encuentra errores que se puedan generar en un bloque y da una respuesta en la ventana de resultados
 Compila todo	Encuentra los errores de todos los bloques del programa
 cargar el proyecto	Carga el PG deberá funcionar correctamente la comunicación de la CPU. Sub Menú de carga de un proyecto, permite el despliegue de las opciones de configuración (Figura 5.21)

Opciones

Figura 6.21. Cargar un programa



6.4.1.4 Barra de herramientas Test

Esta barra permite la compilación, ejecución de un programa, así como el detenerlos y funciones básicas relacionadas con la ejecución. Esta barra se observa en la figura 6.22.

Figura 6.22. Barra de herramientas Test



Tabla 6.4. Descripción de los elementos de la barra de herramientas test

ICONO	DESCRIPCION
LiJ Modo RUN	Permite la ejecucion del programa
^ Modo STOP	Detiene el la ejecucion del programa
S Activa/desactiva el estado del programa	Permite la visualziacion de los operandos durante el ciclo del programa(figura 5.23).
•^ Activar/detener la funcion detener	Permite realizar una pausa durante la ejecucion con el objetivo de que el usuario pueda examinar los valores del estado en un segmento.
JE Activa/deasactiva el estado de tabla:	Permite recoger informacion de estado (Figura 5.24)
1 Lectura sencilla de la tabla de estado	Permite la actualizacion de los valores del programa en la tabla de datos
^ Escribir en la tabla de datos	Permite modificar valores en la tabla de datos
® Forzar datos en la CPU	Permite forzar direccion de registros en la tabla de datos (Figura 5.25)
ÉL Desforzar datos en la CPU	permite eliminar la opcion de forzar los datos en en un bloque de programa.
^ Desforzar todo en la tabla de estado	Desfuerza todas as direcciones
M Activa/desactiva la vista de la curva:	Permite ver las curvas de estado de la CPU y se puede hacer esto durante un periodo definido (Figura 5.26)

Figura 6.23. Activa/desactiva el estado del programa

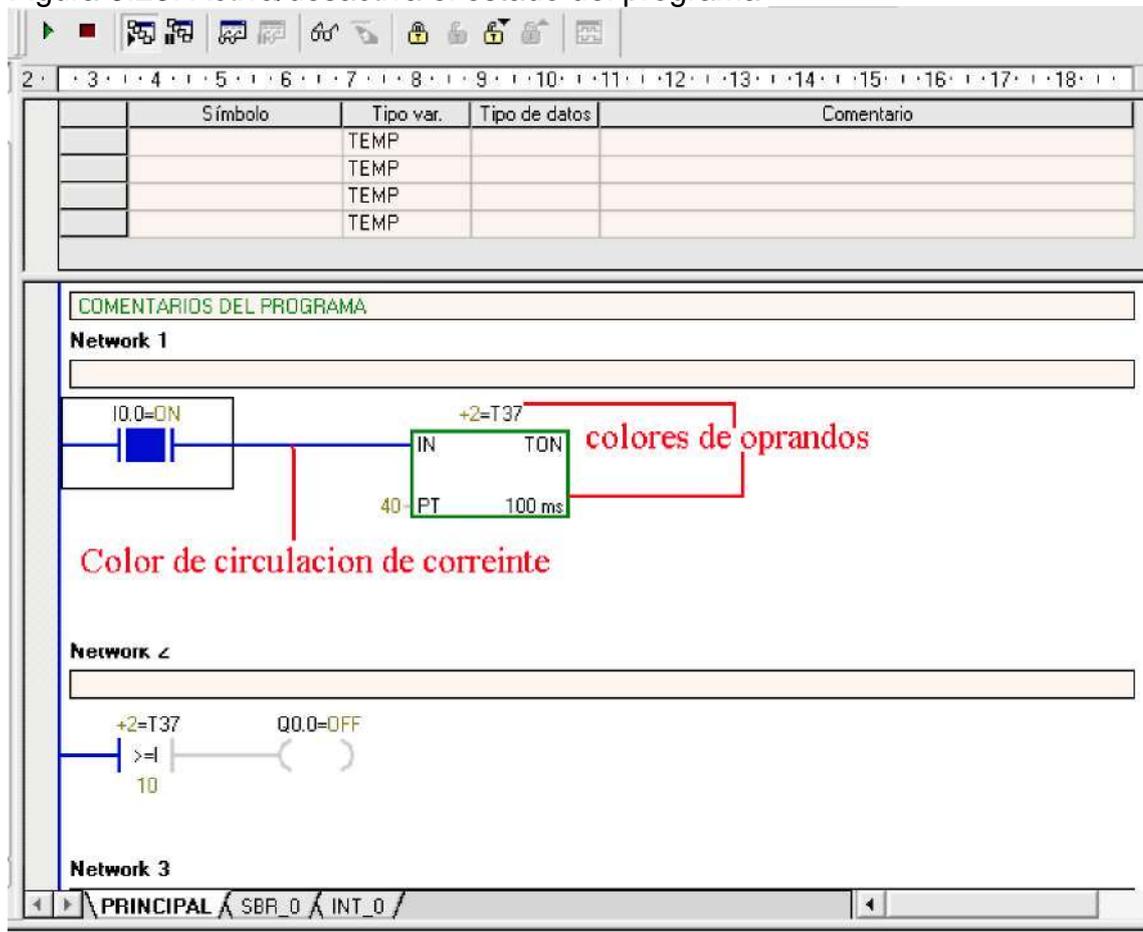


Figura 6.24. Tabla de estado

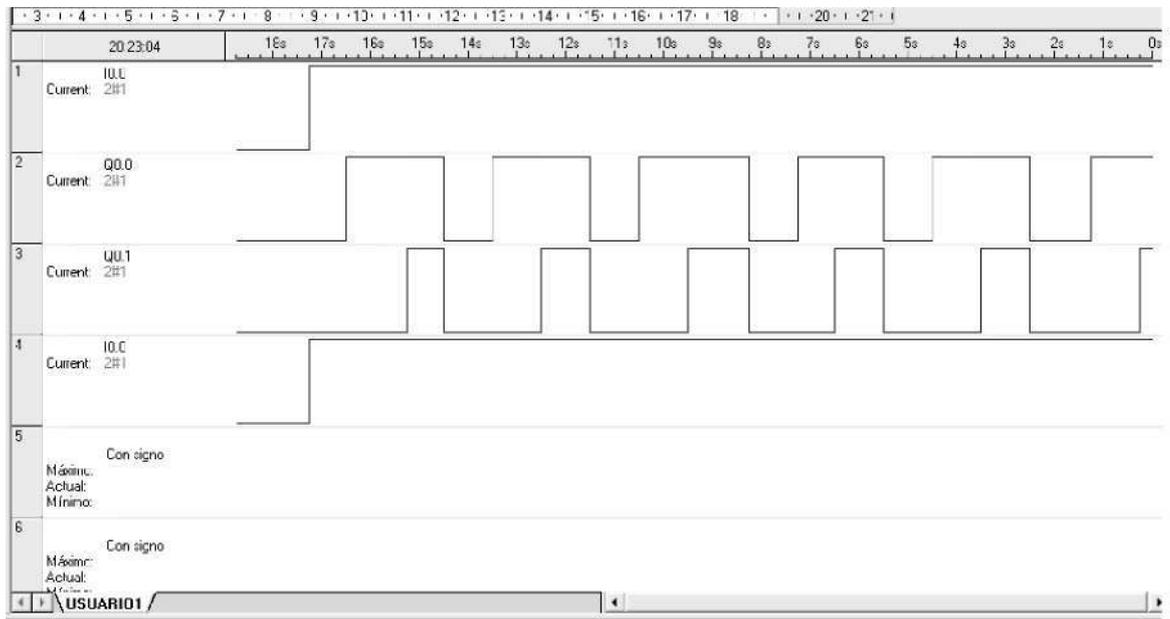
	Dirección	Formato	Valor actual	Nuevo valor
1	Q0.0	Bit	2tt0	
2	Q0.1	Bit	280	
3	I0.0	Bit	2tt0	11
4		Con signo		
5		Con signo		

Figura 6.25. Dirección forzada

	Dirección	Formato	Valor actual	Nuevo valor
1	Q0.0	Bit	2tt1	
2	Q0.1	Bit	2tt0	
3	I0.0	Bit	& 281	
4		Con signo		
5		Con signo		

Indica que esta forzada

Figura 6.26. Curvas de estado de la CPU



7. APLICACIÓN DE LA PLATAFORMA STEP7- MICRO/WIN V.4.0

En este capítulo se describirá las principales operandos utilizados para la realización del código dentro de la plataforma *STEP 7- MICRO/WIN V.4.0*, permitiendo al lector obtener un fundamento sólido relacionado con la estructura de programación. El desarrollo del capítulo iniciará con las operaciones lógicas, operaciones de transferencia, operaciones con temporizadores, consolidando una base sólida para el lector,

7.1 Operaciones lógicas con contactos

El objetivo de realizar operaciones con contactores es la creación y conexión circuitos lógicos, el lenguaje KOP permite que estos contactos sean normalmente cerrados y normalmente abiertos. A la ejecución de una operación directa en el lenguaje KOP inicia con la letra *I* (inmediata), al ejecutarse la operación directa o contacto leerá el valor direccionándolo a la entrada física, teniendo en cuenta que la imagen del proceso no se actualiza. En la operación no directa o contacto leer el valor direccionando en la imagen del proceso que es actualizado por el autómata antes de cada ciclo. En las tablas 7.1 a 7.4 se observa los diferentes tipos de contactos y su respetivo manejo.

Tabla 7.1 . Descripción del contacto estándar

Contactos estándar	
Esta operación lee el valor direccionado de la memoria del proceso si el tipo de dato es I o Q. mediante contactos se representan las operaciones de contacto normalmente abierto y contacto normalmente cerrado	
10.0	Contacto normalmente abierto se cierra (ON) si el bit es igual a 1.
	Contacto normalmente cerrado se cierra (ON) si el bit es igual a 0.

Tabla 7.2. Contactos directos

Contactos directos	
Al ejecutar esta operación el valor físico de la entrada es el que se lee, no obstante el proceso no se actualiza. No depende del ciclo del autómeta para su actualización, si no que inmediatamente se actualice.	
$\overset{10.0}{\text{—}1\ 1\ 1\text{—}}$	Contacto directo abierto se cierra (activado) si en la entrada física el bit es 1.
$\overset{10.0}{\text{—}1\ " \ 1\text{—}}$	Contacto directo cerrado se cierra (activado) si en la entrada física el bit es 0.

Tabla 7.3. Contacto NOT

NOT

Niega la entrada, cambia el valor.



Contacto NOT cambia la entrada. Al alcanzar el contacto NOT la corriente se detiene y si no alcanza el contacto NOT va a circular corriente.

Tabla 7.4. Detección de flanco positivo y negativo

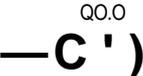
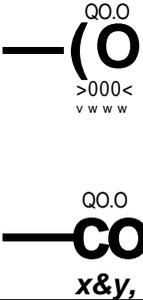
Detección de flanco positivo y flanco negativo	
Esta operación necesita de una transición de bajo a alto o viceversa, el flanco no es detectado en el primer ciclo.	
$\text{—} \cdot \text{—}$	Contacto detectar flanco positivo cada vez que produce un cambio de bajo a alto (0 a 1) permite que circule corriente durante el ciclo.
$\text{—} N \text{—}$	Contacto detectar flanco negativo cada vez que produce un cambio de alto a bajo (1 a 0) permite que circule corriente durante el ciclo.

7.2 Operaciones lógicas con salidas

El objetivo de este tipo de operación es realizar operaciones lógicas con las salidas digitales. En el lenguaje KOP, se activa o desactiva la bobina de salida debido a la lógica con la cual se controla la circulación de la corriente. La corriente excita la bobina y por esto se activa la salida, entonces la operación se pondrá en alto (1) y de lo contrario la operación quedara en bajo (0).

Se implementa / para indicar que la operación va a ser directa. Durante el transcurso de la ejecución del programa se escribe en el proceso y en la salida física el valor direccionado de la salida. Con diferencia de las operaciones no directas en estas solo se escribe el valor en el proceso. A continuación se observara las operaciones lógicas con salidas.

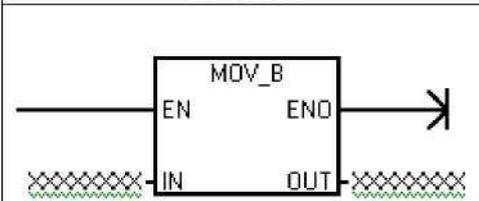
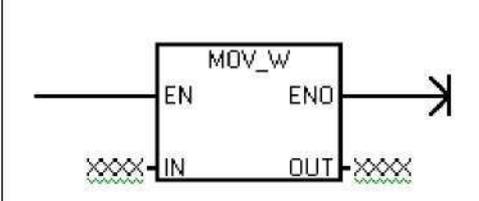
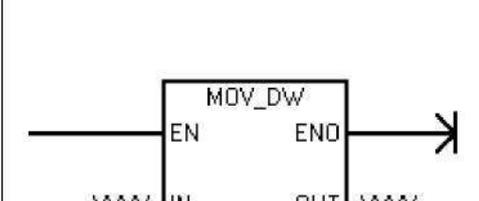
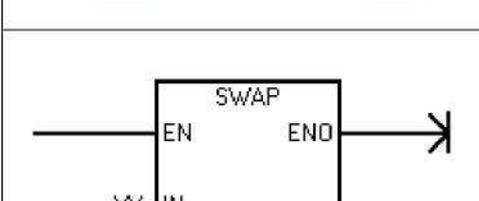
Tabla 7.5. Operaciones lógicas con salidas

ICONO	FUNCION	DESCRIPCION
	Asignar	Esta operación se encarga de asignar el nuevo valor de salida en el proceso. El ejecutar esta operación activa o desactiva el bit de salida en el proceso.
	Asignar directamente	Escribirá el valor actualizado de la operación en el proceso como en la salida física. Al ejecutar esta operación la salida se ajusta directamente y equivalente en la circulación de corriente. Se indica con "I" la operación ejecutada es directa.
	Poner a 1, Poner a 0 (N bits)	Esta función activa y desactiva correspondientemente un número indicado de entradas y salidas, esto es a partir de una dirección indicada, el rango establecido de entradas es de 8 bits (0- 255)
	Poner a 1 directamente, poner a 0 directamente (N bits)	Activan o desactivan correspondientemente un número indicado de entradas y salidas dependiendo de una dirección indicada. Tiene un rango de entradas y salidas que corresponde a 7 bits (0 – 128) Esta operación es directa "I" se escribe en la dirección del proceso como en la salida física el nuevo valor. En las operaciones que no son directas, el nuevo valor es tomado en el proceso.

7.3 Operaciones de transferencia

Estas operación permite la transferencia de datos de una dirección a otra, también realizar la operación SWAP, intercambiando los 8 bit menos significativos por los 8 bit mas significativos.

TABLA 7.6 Operaciones de transferencia

ICONO	FUNCION	DESCRIPCION
	Transferir BYTE	Transfiere el byte de entrada al byte de salida.
	Transferir palabra	Transfiere la palabra de la entrada a la palabra de salida.
	Transferir palabra Doble	Realiza la transferencia de una palabra doble de entrada a una palabra doble de salida
	Invertir BYTE de una palabra	Esta operación se encarga de intercambiar los 8 bit menos significativos y los 8 bit mas significativos de la palabra de entrada.

7.4 Operaciones con temporizadores

La operación de temporización de 1ms, 10ms y 100ms tiene algunas diferencias. Cuando el valor actual de un temporizador bit T de 1 ms es posible actualizarlo varias veces en el mismo ciclo. Cuando se utiliza un temporizador

de 100ms en una subrutina o en una interrupción que no se ejecute dentro del todo el ciclo no se actualizará correctamente, se le suma al valor actual solamente los valores de la base de tiempo que pertenece al ciclo actual. Los temporizadores de 100ms se deben implementar cuando se vaya a ejecutar exactamente una vez por ciclo. El ciclo de trabajo y la base de tiempo del sistema actúa de forma asíncrona.

Tabla 7.7. Tipos de temporizadores

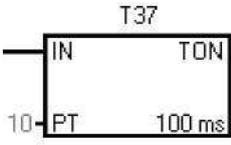
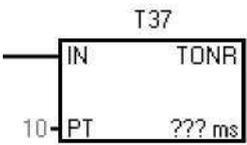
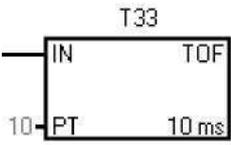
ICONO	FUNCION	DESCRIPCION
	Temporizador como retardo a la conexión (TON)	El valor actual se borra cuando la entrada de habilitación esta desactivada.
	Temporizador como retardo a la conexión con memoria (TONR)	Se utiliza para hacer una acumulación de varios periodos de tiempo de la entrada. Para borrar esta temporización se utiliza la operación <i>Poner a 0</i> .
	Temporizador como retardo a la desconexión (TOF)	Es implementada para retardar la puesta a 0 de una salida durante un periodo determinado tras haberse desactivado una entrada

Tabla 7.8. Resolución de los temporizados

Temporizador	Resolución	Valor máximo	Nº de temporizador
TONR	1 ms	32,767s	T0,T64
	10 ms	327,67s	T1-T4,T65-T68
	100 ms	3276,7 s	T5-T31, T69-T95
TON, TOF	1 ms	32,767s	T32,T96
	10 ms	327,67s	T33-T36,T97-T100
	100 ms	3276,7 s	T37-T63, T101-T255

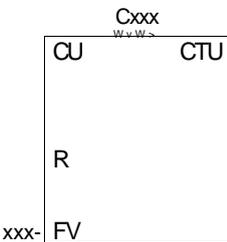
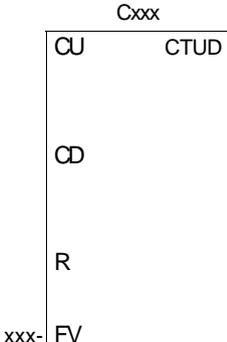
Al seleccionar un temporizador se debe tener presente:

- En el nombre del temporizador, introducir un número de acuerdo a su rango de trabajo
- Cuando el número del temporizador no es valido, se indicara el valor “???”.
- Seleccionar el temporizador de acuerdo a las necesidades dentro del programa desarrollado
- Cuando es valida la información introducida por el programador el valor de la base de tiempo se indicara en el cuadro del temporizador por ejemplo “100ms”.
- La operación de temporizadores como retardo a la conexión con memoria se implementa para acumular varios intervalos temporizados
- El TONR solo es inicializado con la operación poner a 0.

7.5 Operación conteo

Cada contador dispone de solo un valor actual, por ende no se podrá asignar el mismo número a dos o más contadores. Las operaciones de incrementar o disminuir un contador acceden al mismo valor actual.

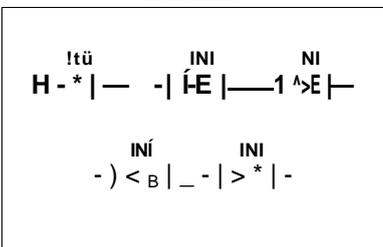
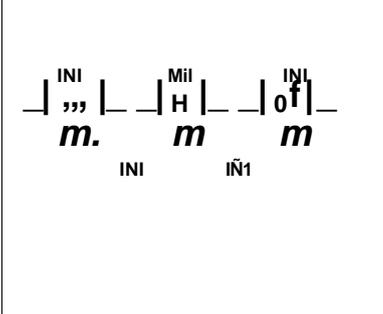
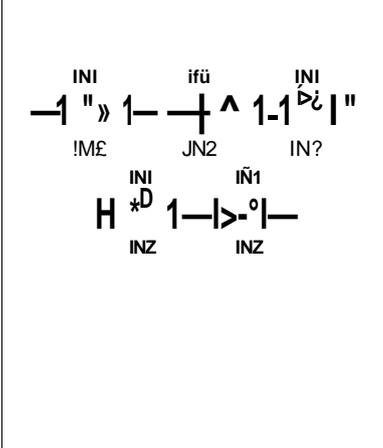
Tabla 7.9. Operaciones de conteo

ICONO	FUNCION	DESCRIPCION
	Incrementar contador	<p>Inicia el conteo hasta el valor máximo a partir del valor actual cuando se produce un flanco positivo en la entrada del contador.</p> <p>Rangos de conteo Cxxx= C0 hasta C255</p>
	Aumentar/disminuir un contador	<p>Comienza el conteo adelante al producirse el flanco positivo en la entrada de contar ascendente CU, y si se produce un flanco positivo en la entrada de disminuir contador CD el contador comienza a decrementarse, el valor. El bit de conteo solo se borra cuando se activa la entrada R.</p>

7.6 Operación De Comparación

El objetivo de estos es la de permitir tomar decisiones en el control, este tipo de lenguaje utiliza contactos de comparación para realizar comparación de valores de bytes, palabras, palabras dobles, las posible comparaciones son <=, =, >=. Al realizar comparaciones <>, > y < hay que utilizar una NOT junto con las operaciones <=, = o >=. Las operaciones de comparación de palabras y palabras dobles llevan signo y las comparaciones de bytes no necesitan signo.

Tabla 7.10. Descripción de los operadores de comparación

ICONO	FUNCION	DESCRIPCION
	Comparar BYTE's	Es utilizada para realizar la comparación de dos valores IN1 e IN2, esta operación no lleva signo.
	Comparar Enteros	Esta operación se implementa en una comparación entre dos valores IN1 e IN2, esta comparación de enteros sencillos lleva signo (16#7FFF > 16#8000).
	Comparar enteros dobles	Esta operación se implementa para una comparación entre dos valores IN1 e IN2, esta comparación de enteros dobles lleva signo (16#7FFFFFFF > 16#80000000), y en este lenguaje solo se activa cuando la comparación es verdadera.

Los siguientes errores detendrán de forma inmediata la ejecución del programa en el autómeta:

- Cuando detecta una dirección indirecta no valida esto se da en todas las comparaciones.
- Cuando detecta un número real no valido esto sucede en la operación de comparar reales

Al evitar las anteriores condiciones de error se inicializa correctamente los puertos y el valor que contenga los números antes de la ejecución de operaciones de comparación que utilicen estos valores; las comparaciones se ejecutan sin tomar en cuenta el estado de la señal

8. APLICACIÓN DE LA PLATAFORMA STEP7- MICRO/WIN V.4.0

8.1 EJERCICIO 1

Realizar la representación de una compuerta lógica AND (figura 8.1) por medio de contadores. El comportamiento de una compuerta AND esta dada por la tabla 8.1 de la verdad.

Figura 8.1 Diagrama de la compuerta AND

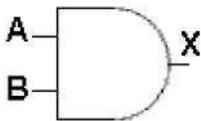
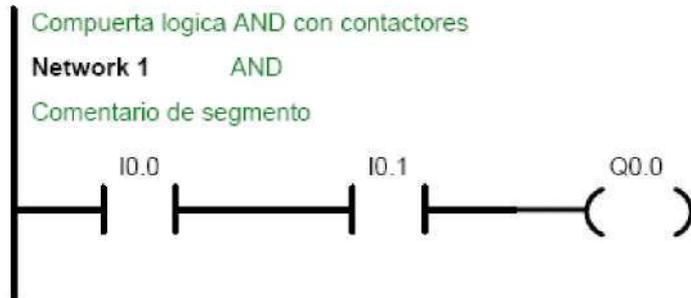


Tabla 8.1 Tabla de la verdad de una compuerta AND		
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

La entrada de la compuerta A= I0.0 de la entrada B= I0.1 y la salida X = Q0.0 Así como se observa en la tabla 8.2.

Tabla 8.2 Representación de la AND con contactores		
I0.0	I0.1	Q0.0
0	0	0
0	1	0
1	0	0
1	1	1

Figura 8.2 Diagrama esquemático del ejercicio 1 en la plataforma STEP7-MICRO/WIN V.4.0

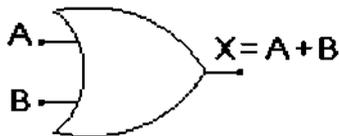


Network 1// Los contactos normalmente abiertos I0.0 e I0.1 deben estar cerrados (ON) para poder activar Q0.0; de lo contrario no va a ser activada la salida Q0.0

8.2 EJERCICIO 2

Realizar la representación de una compuerta lógica OR por medio de contactores. El comportamiento de una compuerta OR esta dada por la tabla de la verdad, ver tabla 8.3.

Figura 8.3 Diagrama de la compuerta OR

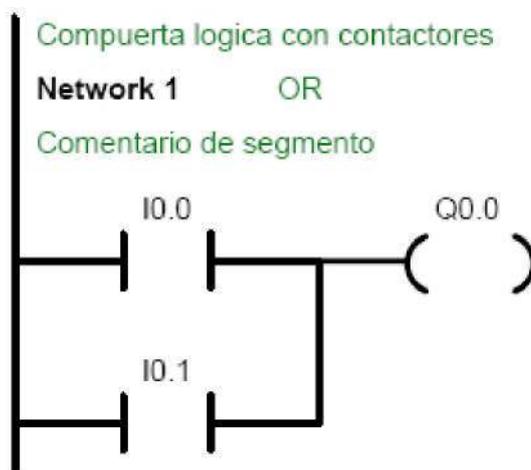


A	B	$X=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

La entrada de la comporta A= I0.0 de la entrada B= I0.1 y la salida ($X=A+B$) = Q0.0, este comportamiento se observa en la tabla 8.4.

Tabla 8.4 Representación de la OR con contactores		
I0.0	I0.1	Q0.0
0	0	0
0	1	1
1	0	1
1	1	1

Figura 8.4 Diagrama esquemático del ejercicio 2 en la plataforma STEP7-MICRO/WIN V.4.0



Network 1// Los contactos normalmente abiertos I0.0 e I0.1 están en paralelo que llevan a una misma salida, se da como alternativa I0.0 o I0.1 que si están cerrados (ON) alguna de estas entradas se activar Q0.0; si ninguna de las entradas se activa pues entonces la salida Q0.0 no se activara.

8.3 EJERCICIO 3

Desarrollar la representación de una compuerta lógica NAND por medio de contactores. La compuerta NAND presenta un comportamiento dado por la tabla 8.5.

Figura 8.5 Diagrama de la compuerta NAND

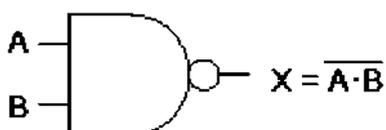
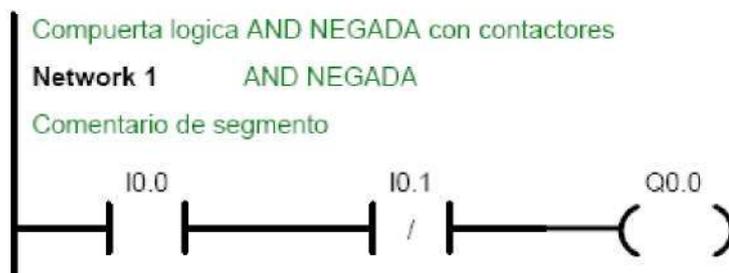


Tabla 8.5 Tabla de la verdad de una compuesta NAND		
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

La entrada de la comporta A= I0.0 de la entrada B= I0.1 y la salida (X=A+B) = Q0.0 (tabla 8.6)

Tabla 8.6 Representación de la NAND con contactores		
I0.0	I0.1	Q0.0
0	0	1
0	1	1
1	0	1
1	1	0

Figura 8.6 Diagrama esquemático del ejercicio 3 en la plataforma STEP7-MICRO/WIN V.4.0



Network 1//los contactos están I0.0 normalmente abierto y el contacto I0.1 normalmente cerrado, entonces para poder activar la salida Q0.0 se necesita que el contacto I0.1 no este activo y el contacto I0.0 este activo.

8.4 EJERCICIO 4

En los anteriores ejercicios se estaba implementado las entradas como interruptores, es decir, que la salida permanecía activa si la entrada está activa, pero que ocurre si se utiliza la entrada como un pulsador y se requiere que la salida se mantenga activa permanentemente, además de que esta la posibilidad de una salida con un paro de emergencia.

Figura 8.7 Diagrama esquemático del ejercicio 4 en la plataforma STEP7-MICRO/WIN V.4.0



Network 1// el contacto ENTRADA_1 va a actuar como un pulsador entonces la SALIDA_1 se activa y se va mantener gracias a que se cierra el contacto SALIDA_1 que esta en paralelo con el contacto de la ENTRADA_1 y la única manera de salirse de esta operación es con el contacto PARO_EMERGEN que va a hacer que se habrá el circuito.

Como en este ejercicio se observa se han etiquetado los contactos con nombres representativos al proceso que se realiza y para ello se implementa la función de tabla de símbolos en la cual se personalizan los contactos.

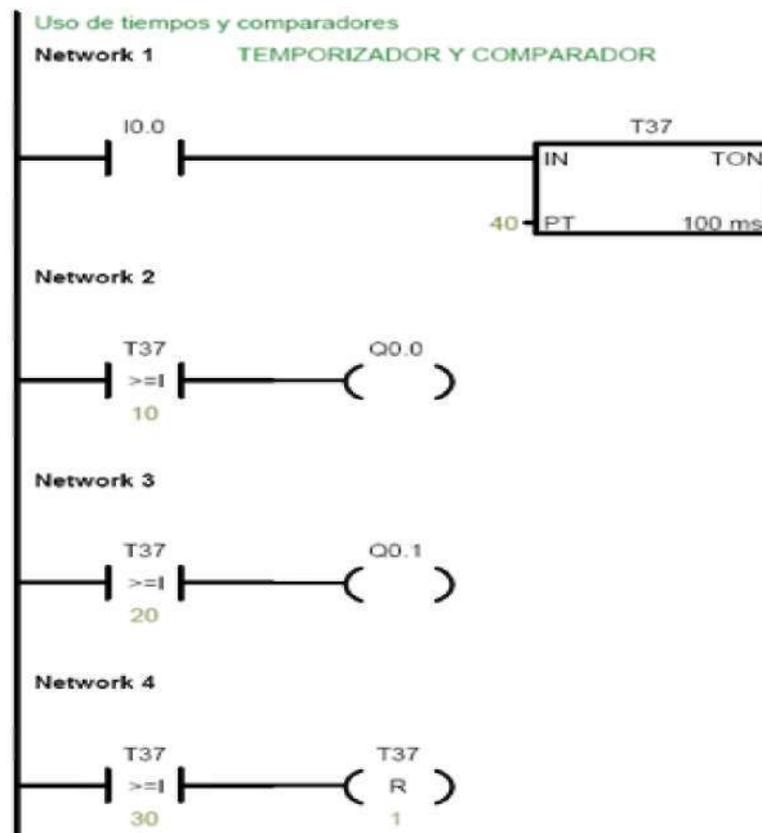
Figura 8.8 tabla de símbolos

Tabla de símbolos				
		Símbolo	Dirección	Comentario
1		ENTRADA_1	I0.0	ENTRADA 1
2		PARO_EMERGENCIA	I0.1	
3		SALIDA_1	Q0.0	
4				
5				

8.5 EJERCICIO 5

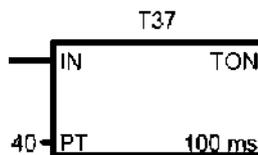
Desarrollar un control para la activación de dos salida con un tiempo predispuesto entre ella para posteriormente desactivarlas con la dependencia de tiempos

Figura 8.9 Diagrama esquemático del ejercicio 5 en la plataforma STEP7-MICRO/WIN V.4.0



Network 1// El contacto I0.0 se activa va a accionar el temporizador como retardo a la conexión TON y va a entre un tiempo predispuesto por el usuario de la siguiente forma:

Figura 8.10 Diagrama del T37



Se configura un T37 que es de 100ms y un valor de PT de 40 esto nos indica que se multiplica los 100ms por 40 y nos dará 4s; vale aclarar que el programa solo tomara decisiones con respecto al valor PT para cualquier tipo de control.

Network 2// se implementa un contacto de comparar enteros de T37 esto con fin de hacer un control que cuando el T37 sea \geq a 10 se activara la salida Q0.0.

Network 3// se implementa un contacto de comparar enteros de T37 y cuando T37 sea \geq a 20 se activara la salida Q0.1.

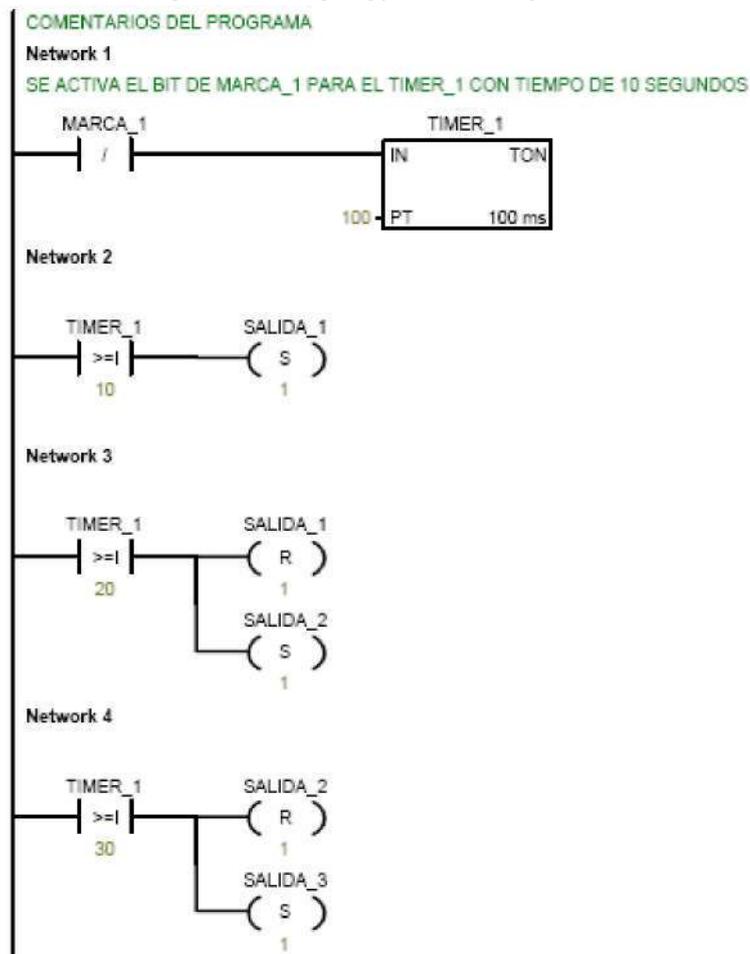
Network 4// se implementa un contacto de comparar enteros de T37 y cuando T37 sea \geq a 30 se activara la bobina de T37 que es una bobina de Poner a 0

(R) que se encargara de colocar el contador en cero y de esta forma desactivar todas las salidas anteriormente activadas.

8.6 EJERCICIO 6

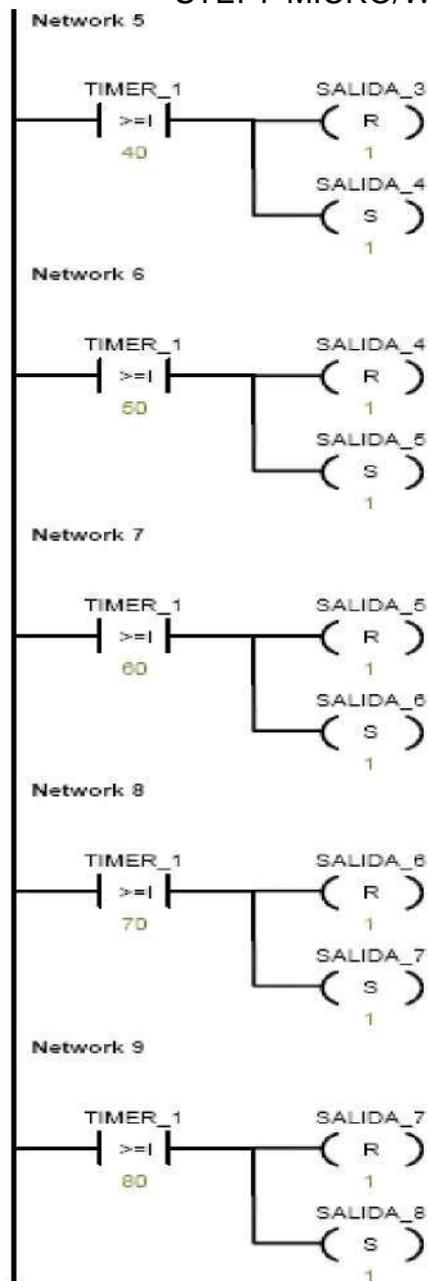
En la industria se presenta variedad de procesos en los cuales se debe hacer uso de temporizadores y comparaciones para tomar decisiones dentro de este mismo. Un proceso donde se de una primera orden donde esta va a activar un timer y luego se llega a una comparación la cual hace que se active un proceso el timer sigue su curso, llega otra comparación con respecto al mismo timer, y cuando se llegue a la comparación verdadera se activara otro proceso y el anterior proceso se para, de esta misma forma al llegar a la siguiente comparación se realizara los mismo que en la anterior comparación se activa un proceso y el anterior se para así consecutivamente en una cantidad de procesos determinados se repetirá el procesos y cuando se llegue al ultimo proceso se retomara desde el principio y se volverá ha hacer la misma rutina una cantidad determina de veces y cuando llegue a esa cantidad determinada se activara un indicador y para todo el proceso.

Figura 8.11 Diagrama esquemático del ejercicio 6 en la plataforma STEP7-MICRO/WIN V.4.0



Network 1// al iniciar este proceso se activara el TIMIER_1 de 10 segundos.
 Network2// se implementa un contacto de comparar enteros de TIMER_1 esto con fin de hacer un control que cunado el TIMER_1 sea \geq a 10 active la SALIDA_1.
 Network3// se implementa un contacto de comparar enteros de TIMER_1 esto con fin de hacer un control que cunado el TIMER_1 sea \geq a 20 active la SALIDA_2 y a su ves se desactive la SALIDA_1.
 Network4// se implementa un contacto de comparar enteros de TIMER_1 esto con fin de hacer un control que cunado el TIMER_1 sea \geq a 30 active la SALIDA_3 y a su ves se desactive la SALIDA_2.

Figura 8.12 Diagrama esquematico del ejercicio 6 en la plataforma STEP7-MICRO/WIN V.4.0



Network5// se implementa un contacto de comparar enteros de TIMER_1 esto con fin de hacer un control que cuando el TIMER_1 sea \geq a 40 active la SALIDA_4 y a su vez se desactive la SALIDA_3.

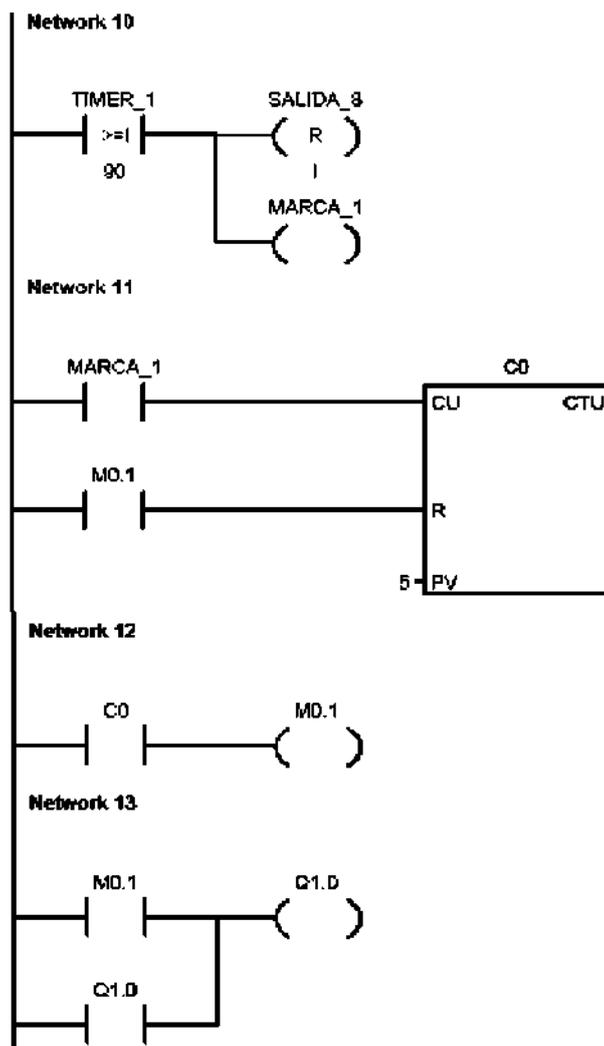
Network6// se implementa un contacto de comparar enteros de TIMER_1 esto con fin de hacer un control que cuando el TIMER_1 sea \geq a 50 active la SALIDA_5 y a su vez se desactive la SALIDA_4.

Network7// se implementa un contacto de comparar enteros de TIMER_1 esto con fin de hacer un control que cuando el TIMER_1 sea \geq a 60 active la SALIDA_6 y a su vez se desactive la SALIDA_5.

Network8// se implementa un contacto de comparar enteros de TIMER_1 esto con fin de hacer un control que cuando el TIMER_1 sea \geq a 70 active la SALIDA_7 y a su vez se desactive la SALIDA_6.

Network9// se implementa un contacto de comparar enteros de TIMER_1 esto con fin de hacer un control que cuando el TIMER_1 sea \geq a 80 active la SALIDA_8 y a su vez se desactive la SALIDA_7.

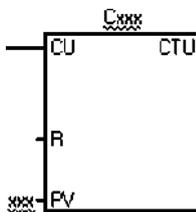
Figura 8.13 Diagrama esquemático del ejercicio 6 en la plataforma STEP7-MICRO/WIN V.4.0



Network10// se implementa un contacto de comparar enteros de TIMER_1 esto con fin de hacer un control que cuando el TIMER_1 sea \geq a 90 para desactivar la SALIDA_8 y a su vez se activa la MARCA_1. Esto lo que hará es abrir el contacto de la MARCA_1 del Network1 y al mismo tiempo cierra el contacto de la MARCA_1 de Network11.

Network11// se cierra el contacto de la MARCA_1 mientras dure la comparación del TIMER_1 verdadera esto con el fin de producir pulsos positivos a la entrada del CD (Incrementar contador) y este contador va a ejecutarse por un valor predispuesto por el usuario de la siguiente forma:

Figura 8.14 Diagrama de Incrementar contador



Se configura un T37 que es de 100ms y un valor de PT de 40 esto nos indica que se multiplica los 100ms por 40 y nos dará 4s; vale aclarar que el programa solo tomara decisiones con respecto al valor PT para cualquier tipo de control

Se configura CD que es un contador (Incrementar contador) donde PV el valor predispuesto por el usuario y nos indicara las veces que este contador se incrementará y el contador se inicializa cuando se active la entrada R.

Network12// cuando el conteo sea igual al que predispuesto por el usuario PV se activara el contacto de CD esto con el fin de que se active la bobina de M0.1 y esta hará que se inicialice el contador CD en el Network11 y activara el contacto de M0.1 del Network13

Network13// se activara el contacto M0.1 esto para activar a Q1.0 y al activarse se cerrara el contacto Q1.0 que se encargara de dar la alerta que se a terminado con el proceso.

Para facilitar al usuario a la hora de identificar las respectivas funciones se utilizaron etiquetas y esto se hace por medio de la función de tabla de símbolos.

Figura 8.15 Tabla de símbolos

Índice	Símbolo	Dirección	Comentario
1	SALIDA_1	Q0.0	
2	SALIDA_2	Q0.1	
3	SALIDA_3	Q0.2	
4	SALIDA_4	Q0.3	
5	SALIDA_5	Q0.4	
6	SALIDA_6	Q0.5	
7	SALIDA_7	Q0.6	
8	SALIDA_8	Q0.7	
9	MÁRCA_1	M0.0	
10	TIMER_1	T37	TIMER DE 10 SEGUNDOS

8.7 EJERCICIO 7

En este ejercicio se implementara los potenciómetros que trae integrado el autómatas para poder realizar algún tipo de control en un proceso industrial que se presente.

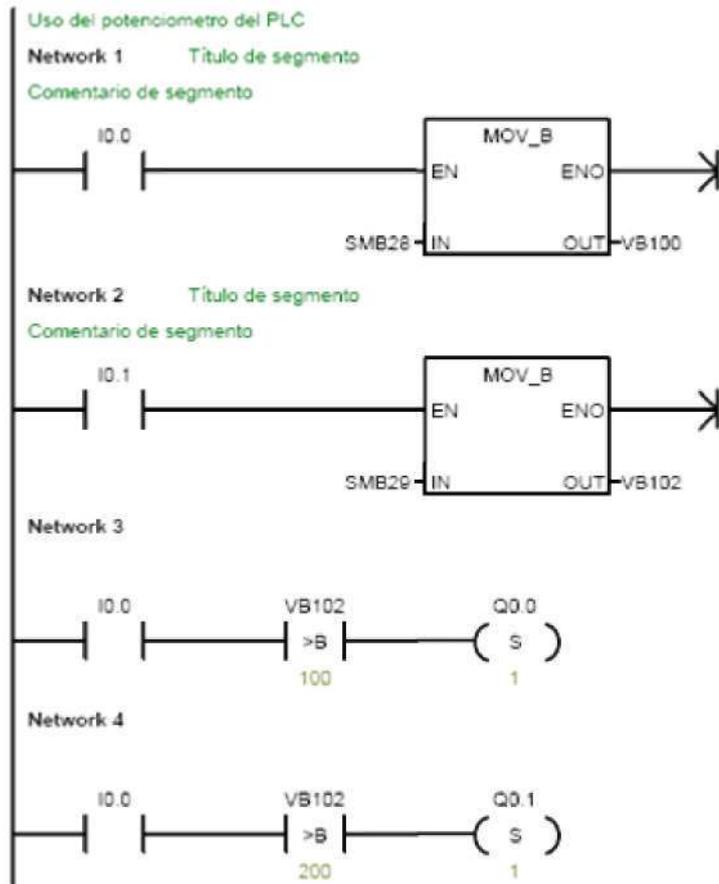
Para esto se tiene que conocer algunas cosas como lo son memorias de variables (VB) que la se usan en este ejercicio, esta función nos dará el rango de direcciones.

Figura 8.16 Areas de memoria y funciones del ST-200

Descripción	CPU 224XP
Memoria de variables (V)	VB0 a VB10239

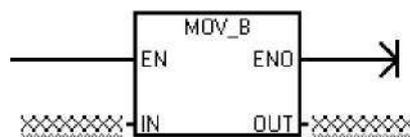
Debido a la utilización de este rango, el número de la localidad de la memoria donde se va a guardar el dato tiene un tamaño de 16 bit por eso se coloca en el número de memoria de variable 100, 102.

Figura 8.17 Diagrama esquemático del ejercicio 7 en la plataforma STEP7-MICRO/WIN V.4.0



Network1// se cierra el contacto I0.0 y esto activara la función de transferencia de bit donde hay que configurarlo de la siguiente forma:

Figura 8.18 Diagrama de transferir byte



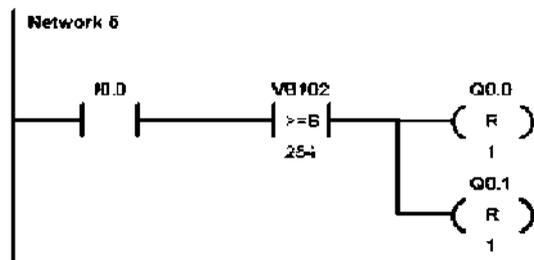
Donde se debe configurar lo siguiente el valor de IN que para nuestro caso es SM28 que es la localidad de uno de los potenciómetros al igual que se debe configurar el valor de OUT que en este valor debe ir VB que es 100 que anteriormente fue explicado.

Network2// se cierra el contacto I0.1 que va a activar una función de transferencia de bit como se efectuó en el Network1 solo que en este caso el valor de IN se coloca SM29 y el valor de VB se coloca 102.

Network3// se cierra el contacto I0.0 y se inicia una comparación con el contacto VB102 de una forma manual se manipulará el potenciómetro integrado en el autómata y de esta forma llegar al valor predispuesto por la comparación para así poder ejecutar la acción que se hace cuando la comparación es verdadera y cuando la comparación es verdadera se activa la salida Q0.0

Network4// se cierra el contacto I0.0 y se inicia una comparación con el contacto VB102 de una forma manual se manipulará el potenciómetro integrado en el autómata y de esta forma llegar al valor predispuesto por la comparación para así poder ejecutar la acción que se hace cuando la comparación es verdadera y cuando la comparación es verdadera se activa la salida Q0.1

Figura 8.19 Diagrama esquemático del ejercicio 7 en la plataforma STEP7-MICRO/WIN V.4.0

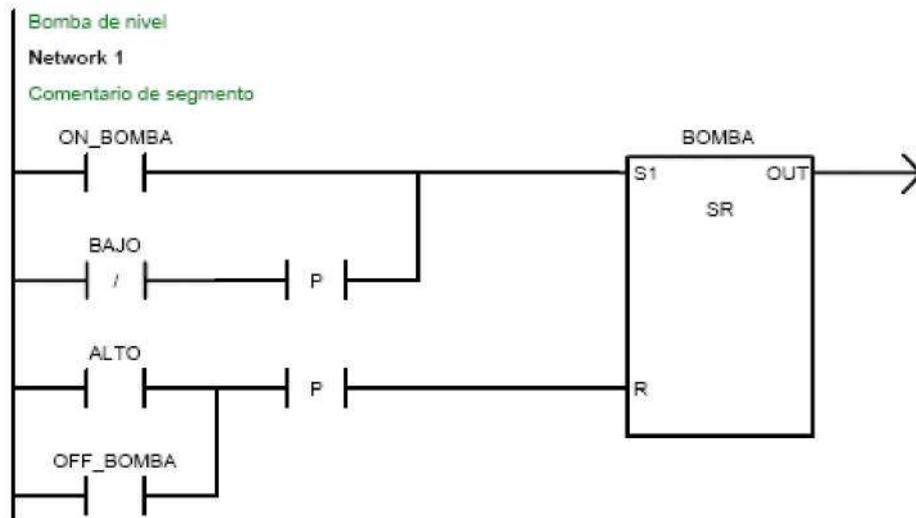


Network5// se cierra el contacto I0.0 y se inicia una comparación con el contacto VB102 de una forma manual se manipulará el potenciómetro integrado en el autómata y de esta forma llegar al valor predispuesto por la comparación para así poder ejecutar la acción que se hace cuando la comparación es verdadera y cuando la comparación es verdaderas se van a desactivar simultáneamente las salidas Q0.0 y Q0.1

8.8 EJERCICIO 8

Este ejercicio mostrará una situación en un proceso industrial donde se reflejará una situación con respecto a unos sensores que son los encargados de hacer que el proceso se desarrolle dependiendo las condiciones del este.

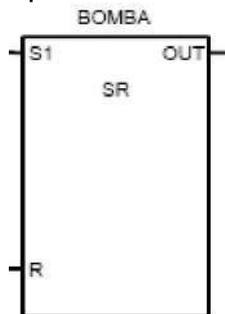
Figura 8.20 Diagrama esquemático del ejercicio 8 en la plataforma STEP7-MICRO/WIN V.4.0



Network1// cuando se cierre el contacto ON_BOMBA inicia el proceso y se comienza a depender del comportamiento de los sensores que están representado en contactos como contacto BAJO y contacto ALTO, se activa el contacto BAJO sigue su proceso, hasta cuando se activa el contacto ALTO se para el proceso hasta que se vuelva a activar el contacto BAJO y retoma el proceso, para este ejercicio se implementa una función que se representa como BOMBA. El comportamiento de esta función se explica de la siguiente forma.

La función se llama posicionar dominante biestable representada en el ejercicio como BOMBA.

Figura 8.21 Diagrama de llama posicionar dominante biestable

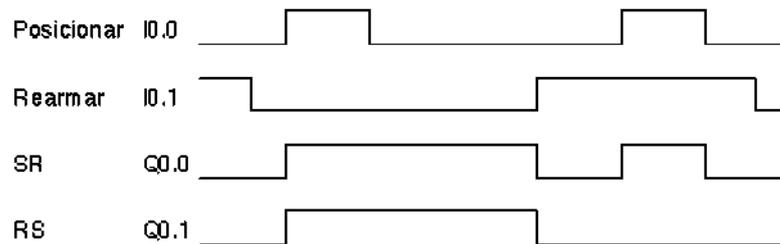


La operación posicionar dominante biestable (SR) es un flip-flop en el que domina la señal "posicionar". Si tanto la señal "posicionar" (S1) como la señal "rearmar" (R) son verdaderas, la salida (OUT) será verdadera.

Tabla 8.7 Tabla de verdad para la operación Posicionar dominante biestable

Operación	S1	R	Out
SR	0	0	Estado anterior
	0	1	0
	1	0	1
	1	1	1

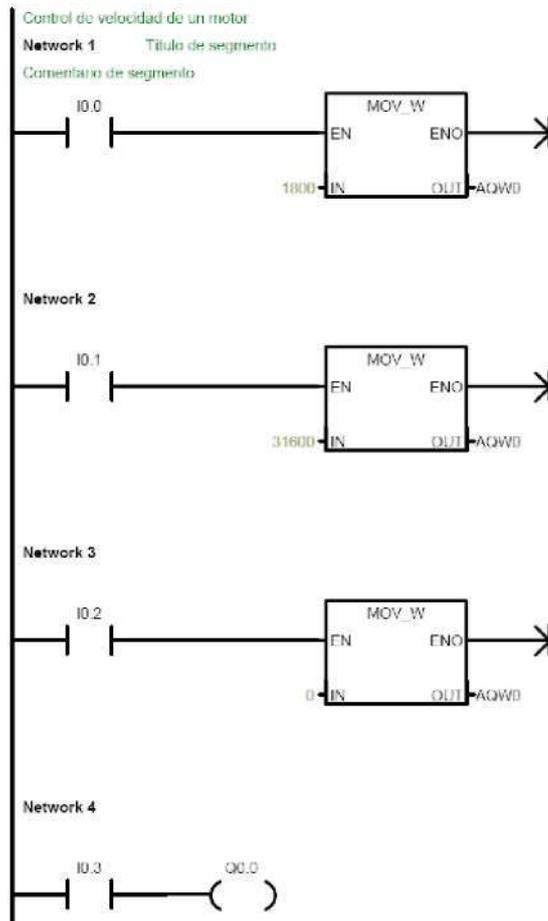
Figura 8.22 Comportamiento de la función posicionar dominante biestable y rearmar donimante biestable.



8.9 EJERCICIO 9

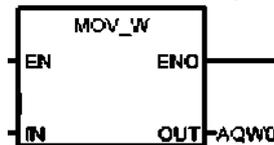
Este ejercicio plantea la utilización de la salida análoga para hacer un control de un variador de velocidad que nos dará una muestra de cómo hacerle control a un motor y esto se hará atraes de contactores y de la función de transferencia de palabra

Figura 8.23 Diagrama esquemático del ejercicio 8 en la plataforma STEP7-MICRO/WIN V.4.0



Netwrk1// al cerrar I0.0 se activa la función de Transferir palabra donde esta función se comporta de la siguiente manera

Figura 8.24 Diagrama de la función de mover palabra



La operación de transferencia de palabra doble (MOVW) realiza la transferencia de una palabra doble de entrada a una palabra doble de salida.

En este caso se envía la salida a través de AQW0 que es una localidad de memoria que se va a ser disecionada por la parte análoga del PLC y así poder llevar el valor que se programa al variador de velocidad.

Netwrk2// se realiza lo mismo que en el Netwrk1// pero con un valor diferente de IN

Netwrk3// se realiza lo mismo que en los dos anteriores Netwrk// con la diferencia que en este se dispone un valor de 0 para así poder detener el proceso

Netwrk4// se asigna la entrada I0.3 y de esta forma vamos a colocar en on off el variador de velocidad o dependiendo de la programación que se le haya dado al variador de velocidad.

CONCLUSIONES Y RECOMENDACIONES

- Este trabajo de grado comprendió el diseñar y construcción de un entrenador de PLC el cual permite realizar practicas de software y hardware para conocer su programación realizando pruebas de entrada salida análogas y digitales.
- Con el desarrollo e implementación de este proyecto para la academia, el alumno aplica y aprovecha no solo los conocimientos del área de electrónica sino también, procedimientos como el diseño, lo cual es un aspecto fundamental y de gran importancia para que el alumno se familiarice completamente con equipos de instrumentación industrial y control de procesos que puede llegar a encontrarse en su desempeño laboral y profesional.
- Este tipo de entrenadores le sirven a la Universidad Pontificia Bolivariana para implementar laboratorios para los alumnos de pregrado y posgrado dando un paso importante en los campos del área de control de procesos, debido a que estos sistemas actuales están experimentado una evolución permanente por que cada día se mejora la eficiencia debido a los diversos equipos que continuamente salen al mercado.
- Se selecciono el PLC S7-224xp de siemens para el entrenador debido a que es muy utilizado en las empresas en sus aplicaciones industriales de automatización y control y así darle al alumno desde pregrado una experiencia la cual le puede ser útil a la hora que se desempeñe en este tipo de campo.
- Se hace una guía práctica para que el alumno siga paso a paso las experiencias desde lo más básico hasta el nivel medio logrando así conocer el manejo del PLC.
- Luego de haber reconocido el entrenador se puede retomar este mismo y realizar prácticas con un propósito específico para las diferentes materias de laboratorio para que así el alumno a la hora de desarrollarlas se desenvuelva con su propia forma de programación y de esta forma crearles criterio de manejo de PLCs.
- Después de lo anteriormente dicho se pude llegar a hacer una integración de diferentes arquitecturas para hacer GUI de supervisión de lo que se programa en el PLC S7-224xp de siemens.

BIBLIOGRAFÍA

- > Manuales de S7 - 200 de SIEMENS.
- > CREUS, Antonio. Instrumentación Industrial.6ª Edición. Alfaomega Marcombo.
- > KATSUHIKO, Ogata. Ingeniería de Control Moderna. 4ª Edición. Prentice-Hall.
- > AUTOMATAS ORIGRMABLES ENTORNO Y APLICACIONES, Enrique Mandado Pérez, Jorge Marcos Acevedo, Celso Fernández Silva, José I. Armesto Quiroga, Serafín Pérez López

ANEXO A

VARIADOR DE VELOCIDAD MICROMASTER 440

SIEMENS

MICROMASTER 420
0,12 kW - 11 kW

Instrucciones de servicio (resumen)

Edición 10/06



Documentación de usuario

Advertencias, precauciones y notas

Las advertencias, precauciones y notas siguientes están pensadas para su seguridad y como medio para prevenir daños en el producto o en componentes situados en las máquinas conectadas.

Las **advertencias**, precauciones y notas **específicas** aplicables a actividades particulares se listan al comienzo de los capítulos o apartados correspondientes y se repiten o añaden en puntos críticos a lo largo de dichos capítulos o apartados. Rogamos leer cuidadosamente la información ya que se entrega para su seguridad personal y le ayudará a prolongar la vida útil de su convertidor MICROMASTER 420 y el equipo que conecte al mismo.

ADVERTENCIA

- > Este equipo contiene tensiones peligrosas y controla partes mecánicas en rotación potencialmente peligrosas. No respetar las advertencias o no seguir las instrucciones contenidas en este manual puede provocar la muerte, lesiones graves o daños materiales considerables.
- > En este equipo sólo deberá trabajar personal adecuadamente cualificado y sólo una vez familiarizado con todas las consignas de seguridad, procedimientos de instalación, operación y mantenimientos contenidos en este Manual. El funcionamiento exitoso y seguro de este equipo depende de si ha sido manipulado, instalado, operado y mantenido adecuadamente.
- > En el circuito intermedio de todos los módulos permanece un nivel de tensión peligroso durante 5 minutos tras que hayan sido desconectadas todas las tensiones. Por ello, una vez desconectado el convertidor de la fuente de alimentación, espere 5 minutos antes de efectuar trabajos en cualquier módulo MICROMASTER. El equipo se descarga en ese tiempo.
- > Este equipo es capaz de proporcionar una protección de sobrecarga del motor interna de acuerdo con UL5D8C sección 42. Véase PD610 (nivel 3) y P0335. La protección de sobrecarga del motor también se puede realizar utilizando una sonda externa tipo PTC mediante una entrada digital.
- > Este equipo es apto para utilizarlo en redes equilibradas capaces de entregar como máximo 10.000 amperios (eficaces), para tensión máxima de 230/460 V, si está protegido por fusible de tipo H o K, por un interruptor automático o mediante una línea derivada para el motor protegida (Per ulteriori dettagli vedi Manuale operativo Appendice F).
- > Utilice solamente cables de cobre clase 1 60/75 °C con la sección que se dictamine en las instrucciones de uso.
- > Las entradas de red. los bornes de corriente continua y del motor pueden estar sometidas a tensión peligrosa aunque no esté funcionando el convertidor; antes de realizar cualquier trabajo de instalación, esperar 5 minutos para que la unidad pueda descargarse después de haberse desconectada de la fuente de alimentación.

NOTAS

- > Antes de efectuar cualquier tipo de trabajo de instalación y puesta en servicio es necesario leer todas las instrucciones y advertencias de seguridad, incluyendo los rótulos de advertencia fijados al equipo. Asegurarse de que los rótulos de advertencia se mantengan en condición legible y sustituir los rótulos perdidos o dañados.
- > La temperatura ambiente máxima permitida será de 50 °C

1 Instalación

1.1 Distancias para el montaje

Los convertidores se pueden montar adosándolos unos a otros. Sin embargo, si se montan uno sobre otro deberá dejarse un huelgo de 100 mm por encima y por debajo de cada convertidor.

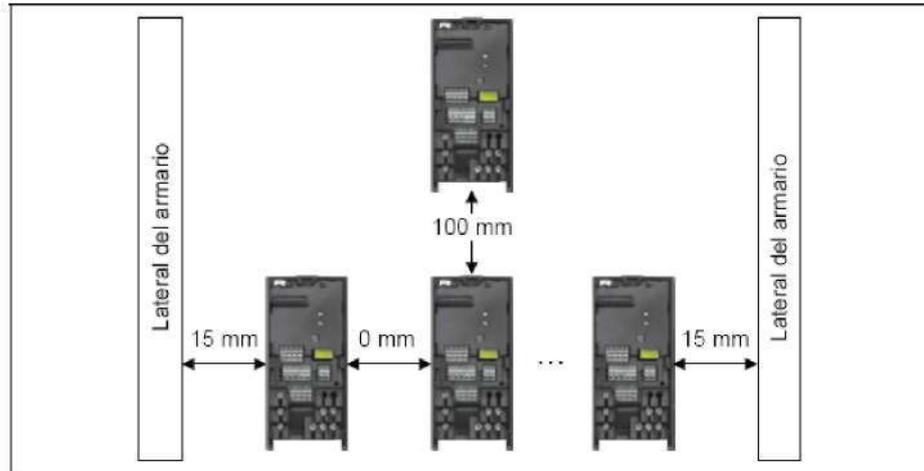


Figura 1-1 Distancias para el montaje

1.2 Medidas para el montaje

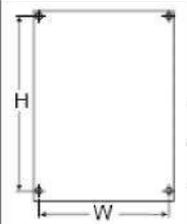
	Tamaño constructivo	Medidas perforaciones		Par de apriete	
		H mm (Inch)	W mm (Inch)	Tornillos	Nm (ibf.in)
	A	160 (6.30)	–	2xM4	2,5 (22.12)
	B	174 (6.85)	138 (5.43)	4xM4	
	C	204 (8.03)	174 (6.85)	4xM4	

Figura 1-2 Medidas el para montaje

2 Instalación eléctrica

2.1 Datos técnicos

1 AC 200 V – 240 V

Referencia 6SE6420-	2AB 2UC	11- 2AA1	12- 5AA1	13- 7AA1	15- 5AA1	17- 5AA1	21- 1BA1	21- 5BA1	22- 2BA1	23- 0CA1
Tamaño constructivo		A					B			C
Potencia nominal	kW hp	0,12 0,16	0,25 0,33	0,37 0,5	0,55 0,75	0,75 1,0	1,1 1,5	1,5 2,0	2,2 3,0	3,0 4,0
Corriente de entrada	A	1,8	3,2	4,6	6,2	8,2	11,0	14,4	20,2	35,5
Corriente de salida	A	0,9	1,7	2,3	3,0	3,9	5,5	7,4	10,4	13,6
Fusible recomendado	A 3NA	10 3803	10 3803	10 3803	10 3803	16 3805	20 3807	20 3807	32 3812	40 3817
Cable de entrada	mm ² AWG	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	2,5-6,0 13-9	2,5-6,0 13-9	4,0-6,0 11-9	6,0-10 9-7
Cable de salida	mm ² AWG	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-6,0 17-9	1,0-6,0 17-9	1,0-6,0 17-9	1,5-10 15-7
Par de apriete	Nm (lbf.in)	1,1 (10)					1,5 (13.3)			2,25 (20)

3 AC 200 V – 240 V

Referencia 6SE6420-	2AC 2UC	11- 2AA1	12- 5AA1	13- 7AA1	15- 5AA1	17- 5AA1	21- 1BA1	21- 5BA1	22- 2BA1	23- 0CA1	24- 0CA1	25- 5CA1
Tamaño constructivo		A					B			C		
Potencia nominal	kW hp	0,12 0,16	0,25 0,33	0,37 0,5	0,55 0,75	0,75 1,0	1,1 1,5	1,5 2,0	2,2 3,0	3,0 4,0	4,0 5,0	5,5 7,5
Corriente de entrada	A	1,1	1,9	2,7	3,6	4,7	6,4	8,3	11,7	15,6	19,7	26,3
Corriente de salida	A	0,9	1,7	2,3	3,0	3,9	5,5	7,4	10,4	13,6	17,5	22,0
Fusible recomendado	A 3NA	10 3803	10 3803	10 3803	10 3803	10 3803	16 3805	16 3805	20 3807	25 3810	32 3812	35 3814
Cable de entrada	mm ² AWG	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-6,0 17-9	1,0-6,0 17-9	1,0-6,0 17-9	2,5-10 13-7	2,5-10 13-7	4,0-10 11-7
Cable de salida	mm ² AWG	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-6,0 17-9	1,0-6,0 17-9	1,0-6,0 17-9	1,5-10 15-7	2,5-10 13-7	4,0-10 11-7
Par de apriete	Nm (lbf.in)	1,1 (10)					1,5 (13.3)			2,25 (20)		

3 AC 380 V – 480 V

Referencia 6SE6420-	2AD 2UD	13- 7AA1	15- 5AA1	17- 5AA1	21- 1AA1	21- 5AA1	22- 2BA1	23- 0BA1	24- 0BA1	25- 5CA1	27- 5CA1	31- 1CA1
Tamaño constructivo		A					B			C		
Potencia nominal	kW hp	0,37 0,5	0,55 0,75	0,75 1,0	1,1 1,5	1,5 2,0	2,2 3,0	3,0 4,0	4,0 5,0	5,5 7,5	7,5 10,0	11,0 15,0
Corriente de entrada	A	2,2	2,8	3,7	4,9	5,9	8,8	11,1	13,6	17,3	23,1	33,8
Corriente de salida	A	1,2	1,6	2,1	3,0	4,0	5,9	7,7	10,2	13,2	19,0	26,0
Fusible recomendado	A 3NA	10 3803	10 3803	10 3803	10 3803	10 3803	16 3805	16 3805	20 3807	20 3807	25 3810	35 3814
Cable de entrada	mm ² AWG	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-6,0 17-9	1,0-6,0 17-9	1,5-6,0 15-9	2,5-10 13-7	4,0-10 11-7	6,0-10 9-7
Cable de salida	mm ² AWG	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-2,5 17-13	1,0-6,0 17-9	1,0-6,0 17-9	1,0-6,0 17-9	1,5-10 15-7	2,5-10 13-7	4,0-10 11-7
Par de apriete	Nm (lbf.in)	1,1 (10)					1,5 (13.3)			2,25 (20)		

2.2 Conexiones de red y del motor

Retirando las tapas se accede a los bornes de red y del motor



Figura 2-1 Sacar las tapas

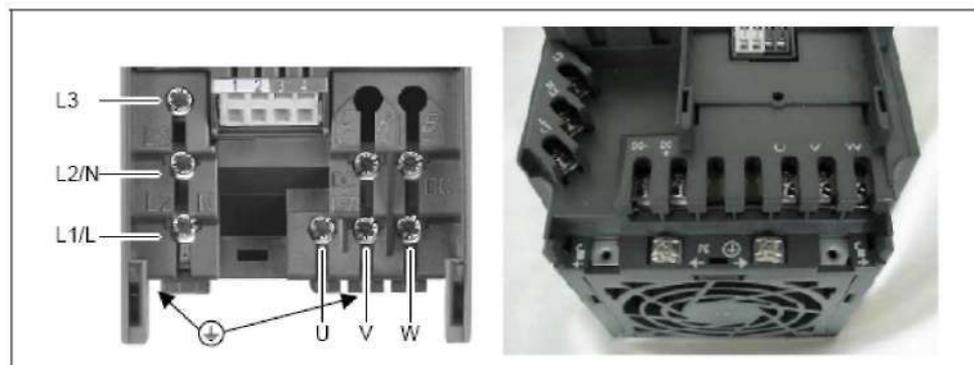
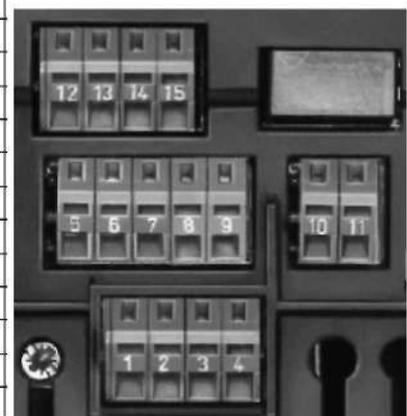


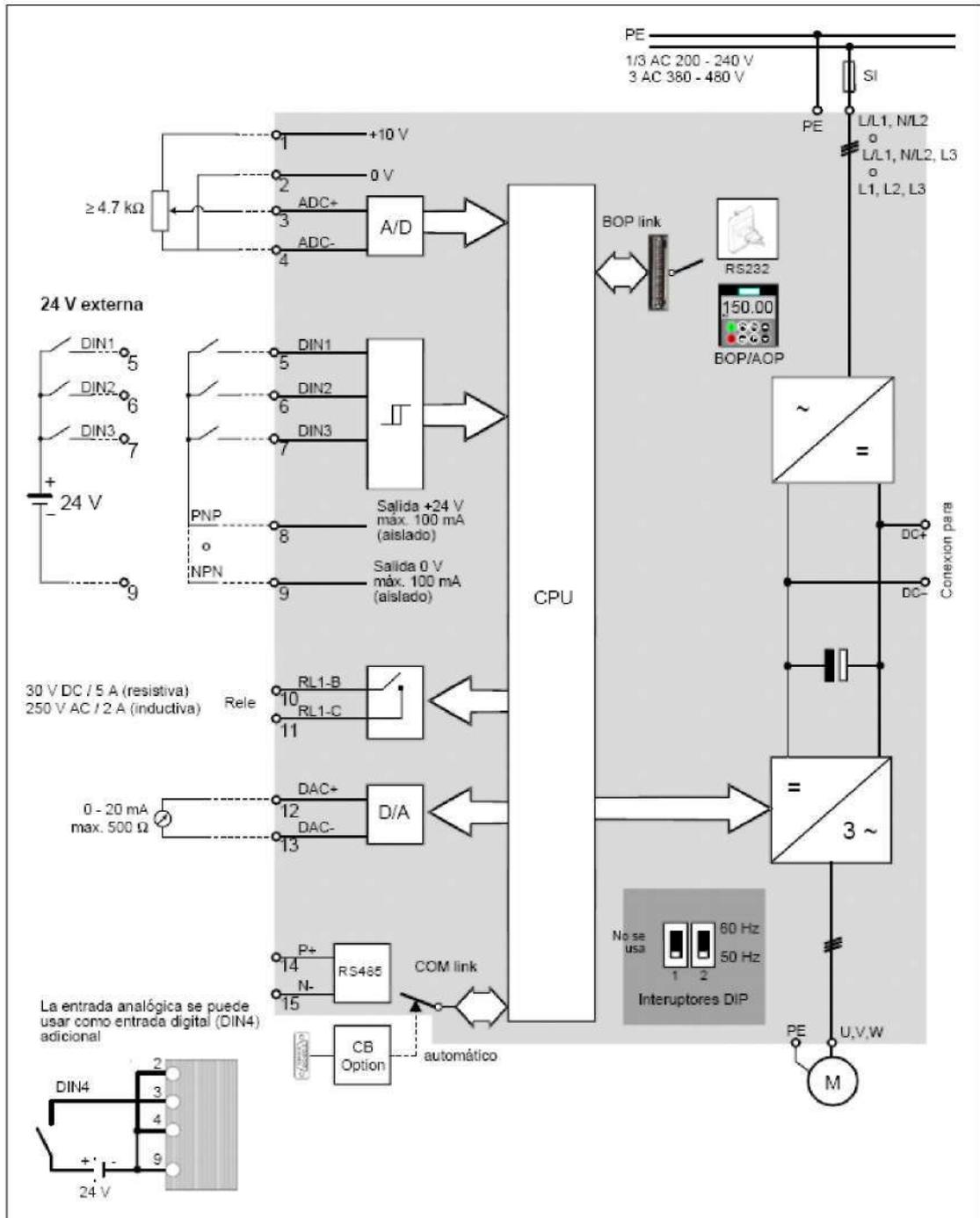
Figura 2-2 Bornes de red y del motor

2.3 Bornes

Borne	Significado	Funciones
1	-	Entrada +10 V
2	-	Entrada 0 V
3	ADC+	Entrada analógica (+)
4	ADC-	Entrada analógica (-)
5	DIN1	Entrada digital 1
6	DIN2	Entrada digital 2
7	DIN3	Entrada digital 3
8	-	Salida aislada +24 V / máx. 100 mA
9	-	Salida aislada 0 V / máx. 100 mA
10	RL1-B	Salida digital / contacto de trabajo
11	RL1-C	Salida digital / conmutador
12	DAC+	Salida analógica (+)
13	DAC-	Salida analógica (-)
14	P+	Conexión RS485
15	N-	Conexión RS485



2.4 Esquema de bloques



3 Ajuste de fábrica

El convertidor MICROMASTER 420 sale de fabrica ajustado para poder funcionar sin necesidad de parametrizaciones adicionales. Los parámetros (P0304, P0305, P0307, P0310) se han ajustado para un motor de Siemens de 4 polos 1LA7 y hay que cotejarlos con los datos asignados del motor a conectar (véase la placa de características).

Otros ajuste de fábrica:

- Fuente de órdenes P0700 = 2 (Entrada digital, véase Figura 3-1)
- Fuente de consignas P1000 = 2 (Entrada analógica, véase Figura 3-1)
- Refrigeración del motor P0335 = 0
- Factor sobrecarga motor P0640 = 150 %
- Frecuencia mínima P1080 = 0 Hz
- Frecuencia máx. P1082 = 50 Hz
- Tiempo de aceleración P1120 = 10 s
- Tiempo de deceleración P1121 = 10 s
- Modo de control P1300 = 0

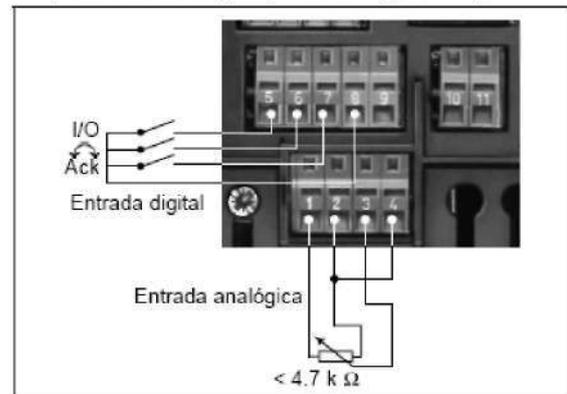


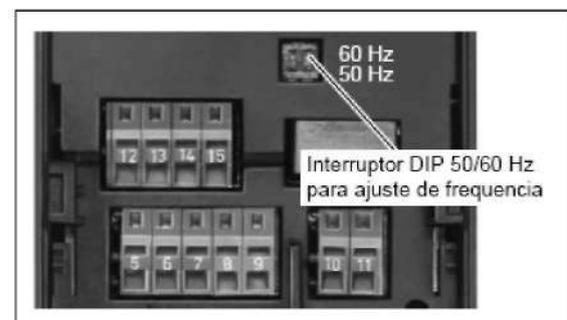
Figura 3-1 Entrada / Salida

Entrada / Salida	Bornes	Parámetro	Ajuste por defecto
Entrada digital 1	5	P0701 = 1	ON / OFF1 (I/O)
Entrada digital 2	6	P0702 = 12	Inversión (↕)
Entrada digital 3	7	P0703 = 9	Acuse de fallo (Ack)
Entrada digital	8	-	Alimentación entrada digital
Entrada analógica	3/4	P1000 = 2	Consigna de frecuencia
	1/2	-	Alimentación entrada analógica
Relé de salida	10/11	P0731 = 52.3	Indicación de fallo
Salida analógica	12/13	P0771 = 21	Frecuencia de salida

3.1 Interruptor DIP 50/60 Hz

Los convertidores están preajustados para motores con una frecuencia nominal de 50 Hz, pero se pueden adaptar a motores dimensionados con una frecuencia nominal de 60 Hz con un interruptor DIP.

- Posición Off: Ajustes europeos por defecto (50 Hz, kW etc.)
- Posición On: Ajustes norteamericanos por defecto (60 Hz, hp etc.)

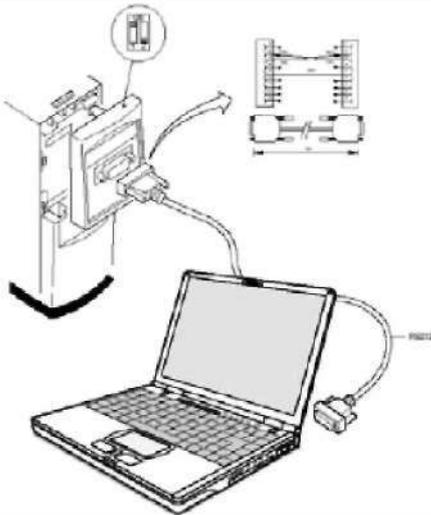


4 Comunicación

4.1 Estructura de la comunicación MICROMASTER 420 ↔ STARTER

La comunicación entre el STARTER y el MICROMASTER 420 requiere de los siguientes componentes opcionales:

- Kit de conexión PC-convertidor
- BOP, hay que modificar los valores estándar USS (véase sección 6.2.1 "Interface en serie (USS)") en los convertidores MICROMASTER 420.

	MICROMASTER 420 Ajustes USS véase sección 6.2.1 "Interface en serie (USS)"
	STARTER Menú Extras --> Ajustar interface PG/PC --> Seleccionar "Puerta COM del PC (USS)" --> Propiedades --> Interface "COM1", seleccionar velocidad de transmisión
	NOTA Los ajustes de parámetros USS en los convertidores MICROMASTER 420 tienen que concordar con los del STARTER

4.2 Estructura de la comunicación MICROMASTER 420 ↔ AOP

- La comunicación entre el AOP y el MM420 se basa en el protocolo USS, de forma análoga a la comunicación entre STARTER y MM420.
- Al contrario que con el BOP, si la detección de interface no se ha efectuado, se tienen que ajustar los parámetros de comunicación, tanto en el MM420 como en el AOP (véase Tabla 4-1).
- Con los componentes opcionales se puede enlazar el AOP a la interface de comunicación (véase Tabla 4-1).

Tabla 4-1

	AOP en conexión BOP	AOP en conexión COM
Parámetros MM420 - Velocidad de transmisión - Dirección de bus	P2010[1] -	P2010[0] P2011
Parámetros AOP - Velocidad de transmisión - Dirección de bus	P8553 -	P8553 P8552
Opciones - Conexión directa - Conexión indirecta	No se necesita opción BOP/AOP (Kit puerta simple) (6SE6400-0PM00-0AA0)	No es posible AOP (Kit puerta múltiple) (6SE6400-0MD00-0AA0)

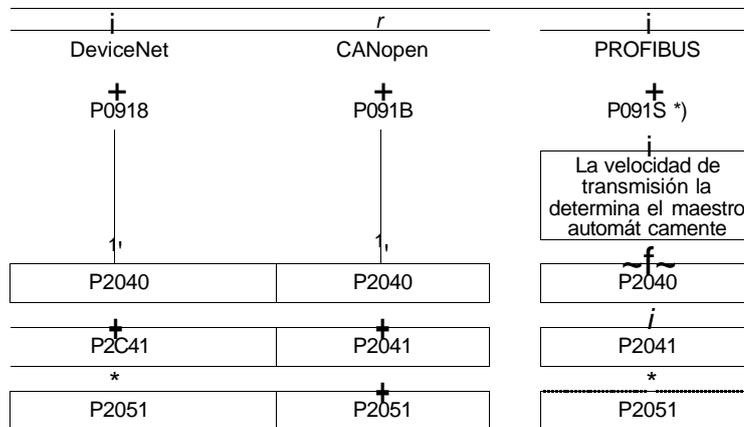
AOP como unidad de control

Parámetro / Botones		AOP en conexión BOP		AOP en conexión COM	
Fuente de órdenes 	P0700	4		5	
Fuente de consignas (MOP) 	P1000			I	
	P1035	2032.13 (2032.D)		2036.13(2036.0)	
	P1G36	2032.14 (2032.E)		2036.14 (2036.E)	
	9			FAft^S AI Fn BTB1 P	OMrta l.l>_*
0			STOPPED F • [IAH2 J r000 V=DOU T		
		MOP elevar frecuencia de salida			
		MOP bajar frecuencia de salida			
Acuse de fallo 	P2104	2032.7		2036.7	

* Se pueden acusar fallos con el AOP independientemente de P0700 y P1000.

4.3 Interface de bus (CB)

Interface de bus (CB)



*) El interruptor DIP es importante para direccionar el hardware

	DeviceNet	CANopen	PROFIBUS
P2041[0]	Longitud PZD palabra de estado/ valor real	Tipo de transmisión de T PD0 1.T PD0 5	No es necesario
P2041[1]	Longitud PZD palabra de mando/consigna	Tipo de transmisión	T_PD0_6 R_PD0_1 R_PD0_5 R_PD0_6
P2041[2]	Velocidad de transm. 0: 125kBaud 1:250kBaud 2: 500 kBaud	Mapping de CANopen	MM4 Véanse las instrucciones de servicio "Tarjetas opcionales PROFIBUS"
P2041[3]	Diagnóstico	Mapping de CANopen < -> MM4	
P2041 [4]		- Reacción a error de comunicación - Velocidad de transmisión	

BOP/AOP (Opción)

I 150.00

5.1 Botones y sus funciones en los paneles

Panel Botón	Función	Efectos
r0000	Indicación de estado	Lo parral a de c::a lícu ce: Tíuesra les a_Listes actúa es de convencí:;'. I HL
O	Marcha	Al pulsar este botón se arranca el convertidor. Por defecto está bloqueado. Para activarlo: BOP: P0700 = 1 o P0719 = 10 ...16 AOP: P0700 = 4 o P0719 = 40 .46 en conexión BOP P0700 = 5 o P0719 = 50 ... 56 en conexión COM
O	Parada	OFF1 Pulsando este botón se para el motor siguiendo la rampa de deceleración seleccionada. Para activarlo: véase botón "Marcha" OFF2 Pulsando el botón dos veces (o una vez prolongada) el motor se para de forma natural (inercia hasta parada). BOP: Esta función está constantemente activada (independientemente de P0700 y P0719).
e	Invertir sentido	Pulsar este botón para cambiar el sentido de giro del motor. El inverso se indica mediante un signo negativo (-) o un punto decimal intermitente. Por defecto está bloqueado Para activarlo: véase botón "Marcha" .
^	Jog motor	Pulsando este botón en estado "listo" el motor arranca y gira a la frecuencia Jog preseleccionada. Pulsando este botón mientras el convertidor no tiene salida hace que el motor arranque y gire a la frecuencia Jog preseleccionada. El motor se detiene cuando se suelta el botón. Pulsar este botón cuando el motor está funcionando carece de efecto.
O	Función	Este botón sirve para visualizar información adicional. Funciona pulsándolo y manteniéndolo apretado. Muestra lo siguiente comenzando por cualquier parámetro durante la operación: 1. Tensión en circuito intermedio (indicado mediante d - unidades en V). 2. Corriente de salida. (A) 3. Frecuencia de salida (Hz) 4. Tensión de salida (o - unidades en V). 5. El valor seleccionado en P0005. (Si P0005 se ha configurado de tal forma que se muestra uno de los datos indicados arriba (1 - 4), no aparece el valor correspondiente de nuevo). Cualquier pulsación adicional hace que vuelva a visualizarse ía sucesión indicada anteriormente. Función de salto Pulsando brevemente el botón Fn es posible saltar desde cualquier parámetro (rxxx. o Pxxxx) a r0000. lo que permite, si se desea, modificar otro parámetro. Una vez retornado a r0000, si pulsa el botón Fn irá de nuevo a su punto inicial. Acusar Cuando aparecen mensajes de alarma y error , se pueden acusar, pulsando el botón Fn.
O	Acceder a parámetros	Pulsando este botón es posible accederá los parámetros.
O	Subir valor	Pulsando este botón se sube el valor visualizada.
O	Bajar valor	Pulsando este botón se baja el valor visualizado.
© O	Menú AOP	Llamada del menú en el AOP (solo si de dispone de AOP).

5.2 Modificación de parámetros, ejemplo P0003 "nivel de acceso"

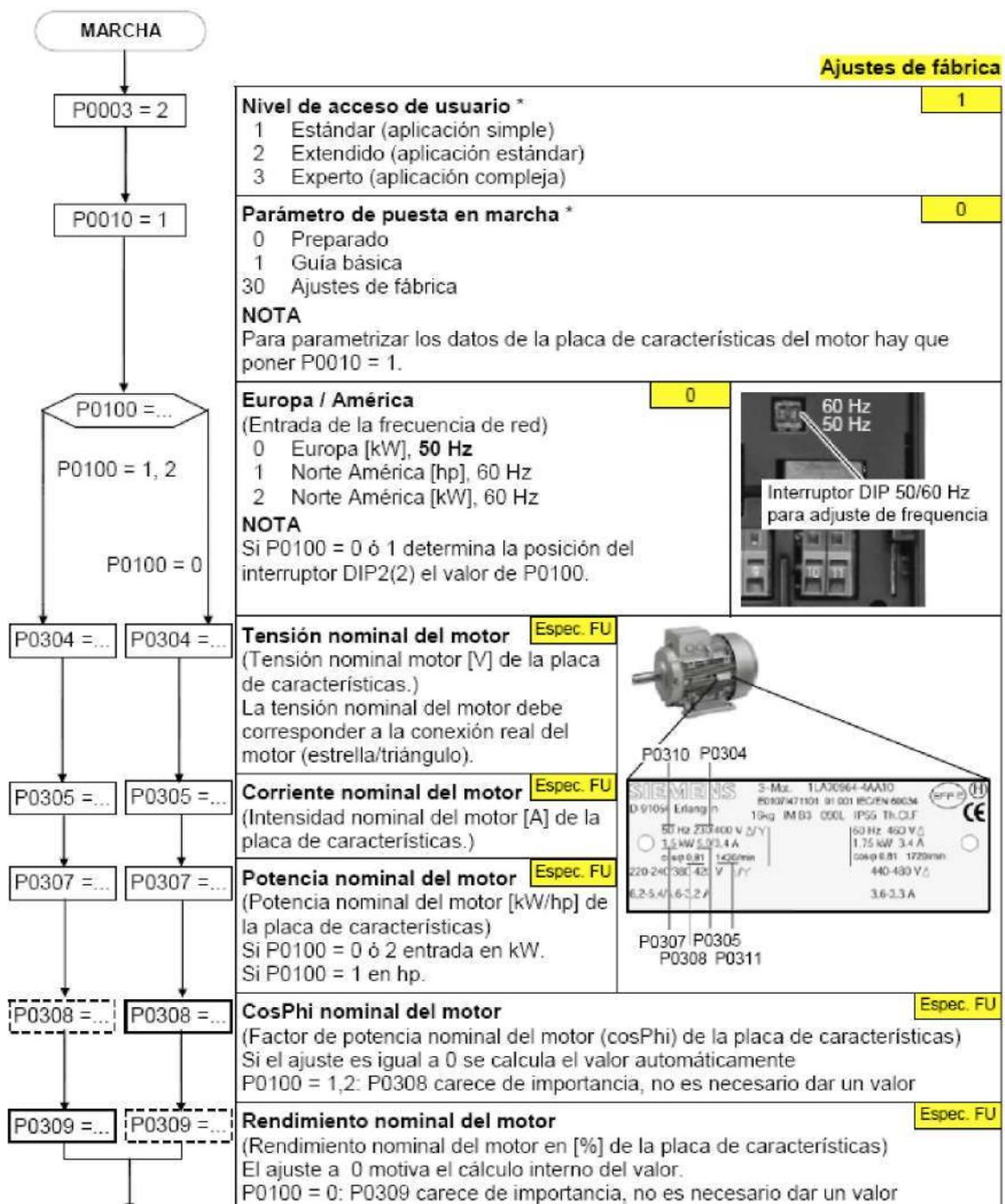
Paso		Resultado en pantalla
1	Pulsar P para acceder a parámetros	r0000
2	Pulsar ▲ hasta que se visualice P0003	P0003
3	Pulsar P para acceder al nivel de valor del parámetro	1
4	Pulsar ▲ o ▼ hasta el valor requerido	3
5	Pulsar P para confirmar y guardar el valor	P0003
6	El nivel de acceso 3 está ajustado. Se pueden seleccionar todos los parámetros de los niveles 1 a 3.	

6 Puesta en servicio

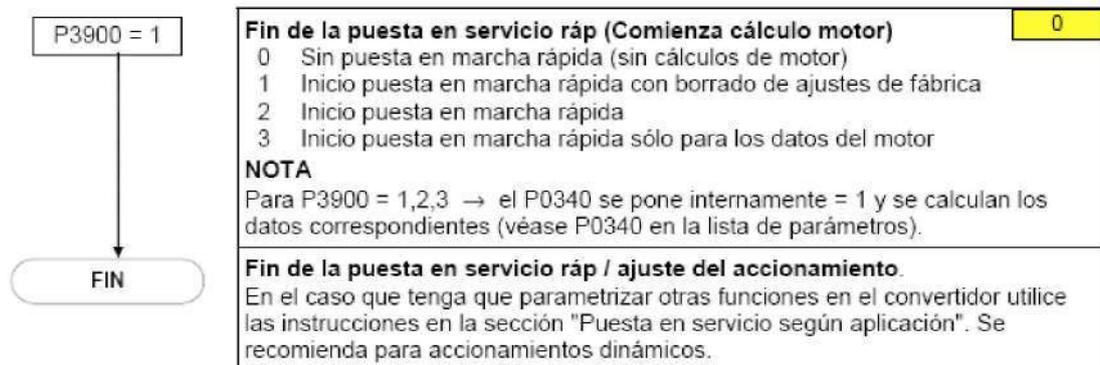
6.1 Puesta en servicio rápida

Con la puesta en servicio rápida se adapta el convertidor al motor y se ajustan parámetros importantes para las exigencias tecnológicas. La puesta en servicio rápida no es obligatoria si los datos del motor almacenados en el convertidor (motor de Siemens de 4 polos 1LA, conexión en estrella \triangle Espec. FU) coinciden con los de la placa de características.

Los parámetros marcados con * ofrecen más posibilidades de ajuste de las que se mencionan aquí. Para más detalles consulte la lista de parámetros.



P0310 =...	Frecuencia nominal del motor (Frecuencia nominal motor [Hz] de la placa de características) Se vuelve a calcular el número de pares de polos si se cambia el parámetro.	50.00 Hz
P0311 =...	Velocidad nominal del motor (Velocidad nominal motor [rpm] de la placa de características) El ajuste a 0 motiva el cálculo interno del valor. NOTA Para la compensación de deslizamiento es necesario dar un valor.	Espec. FU
P0335 =...	Refrigeración del motor (Selecciona el sistema de refrigeración utilizado) 0 Autoventilado 1 Ventilación forzada	0
P0640 =...	Factor sobrecarga motor (Define el límite de intensidad de sobrecarga del motor en [%] relativo a P0305) Determina en % el valor máx. de salida de la corriente nominal del motor (P0305).	150 %
P0700 =...	Selección fuente de ordenes 0 Ajuste por defecto de fábrica 1 BOP (teclado) 2 Terminal 4 USS en conexión BOP 5 USS en conexión COM 6 CB en conexión COM	2
P1000 =...	Selecc. consigna de frecuencia 1 Consigna MOP 2 Consigna analógica 3 Frecuencia fija 4 USS en conexión BOP 5 USS en conexión COM 6 CB en conexión COM	2
P1080 =...	Frecuencia mínima (En Hz) Ajusta la frecuencia mínima del motor a la cual el motor funcionará independientemente de la consigna de frecuencia. El ajuste de este valor es válido para ambos sentidos de rotación horaria y antihoraria.	0.00 Hz
P1082 =...	Frecuencia máx. (En Hz) Ajusta la frecuencia de motor máxima a la cual el motor funcionará independientemente de la consigna de frecuencia. El ajuste de este valor es válido para ambos sentidos de rotación horaria y antihoraria.	50.00 Hz
P1120 =...	Tiempo de aceleración (En s) Tiempo utilizado por el motor para acelerar desde el punto muerto hasta la frecuencia máxima del motor (P1082) cuando no se utiliza el redondeo.	10.00 s
P1121 =...	Tiempo de deceleración (En s) Tiempo utilizado por el motor para desacelerar desde la frecuencia máxima (P1082) hasta el punto muerto cuando no se utiliza el redondeo.	10.00 s
P1135 =...	Tiempo deceleración OFF3 (En s) Define el tiempo de deceleración desde la frecuencia máxima hasta el punto muerto para una orden OFF3.	5.00 s
P1300 =...	Modo de control (Entrada del modo de control deseado) 0 V/f con característ. lineal 1 V/f con FCC 2 V/f con característ. parabólica 3 V/f con característ. programable	0



6.2 Puesta en servicio según aplicación

Los parámetros marcados con * ofrecen más posibilidades de ajuste de las que se mencionan aquí. Para más detalles consulte la lista de parámetros.

El convertidor posee una serie de funciones que a veces no son necesarias. La puesta en servicio según aplicación sirve para optimizar la funcionalidad del sistema convertidor-motor a la aplicación requerida. Aquí se describen la mayoría de las funciones. Consulte la lista de parámetros para ver otras.



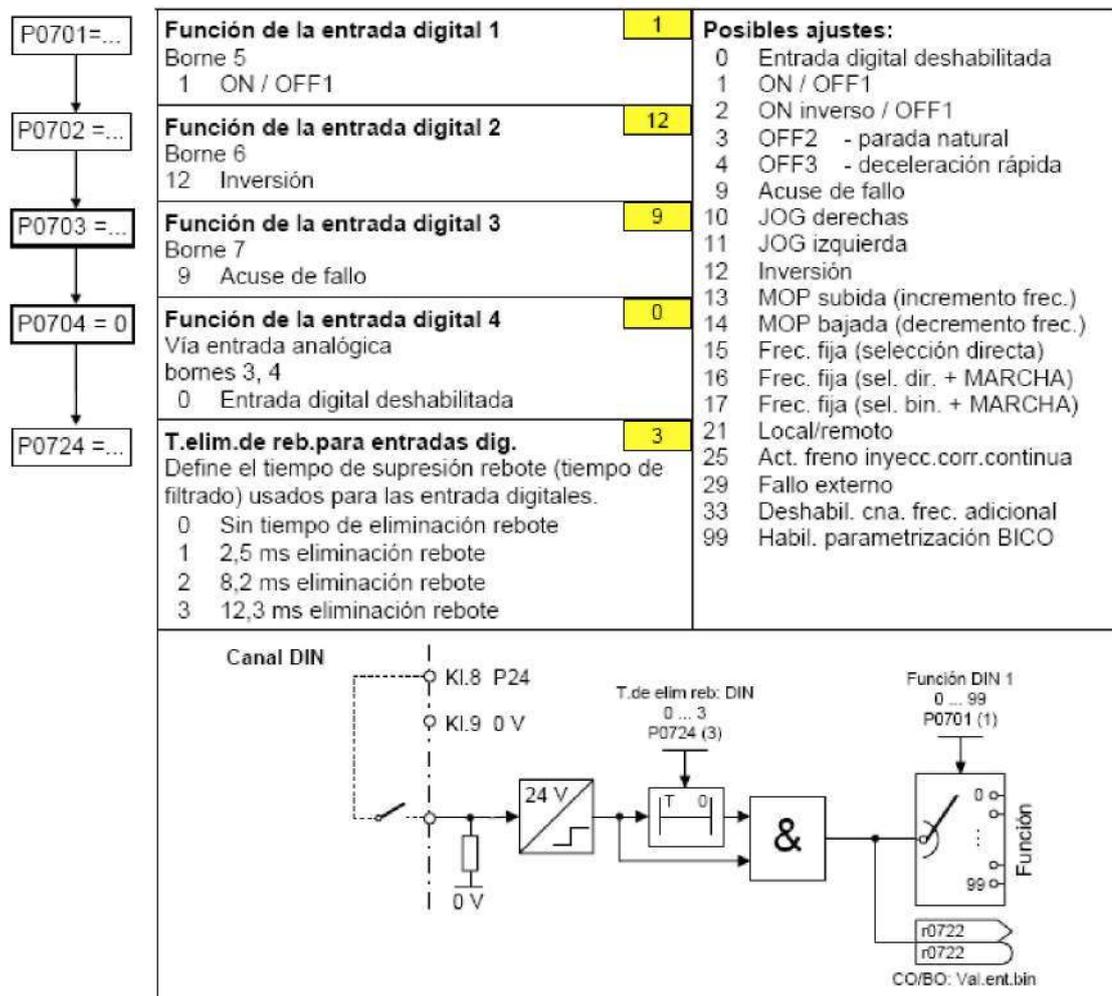
6.2.1 Interface en serie (USS)

P2010 =...	Velocidad transferencia USS 6 Ajuste de la velocidad de transmisión para la comunicación USS.	Posibles ajustes: 3 1200 baud 4 2400 baud 5 4800 baud 6 9600 baud 7 19200 baud 8 38400 baud 9 57600 baud
↓	Dirección USS 0 Ajuste de la dirección única para cada convertidor.	
P2011 =...	USS longitud PZD 2 Define el número de palabras de 16 bits en la parte PZD del telegrama USS.	
↓	USS longitud PKW 127 Define el número de palabras de 16-bit en la parte PKW del telegrama USS.	
P2012 =...		
↓		
P2013 =...		

6.2.2 Selección fuente de ordenes



6.2.3 Entrada digital (DIN)



6.2.4 Salida digital (DOUT)

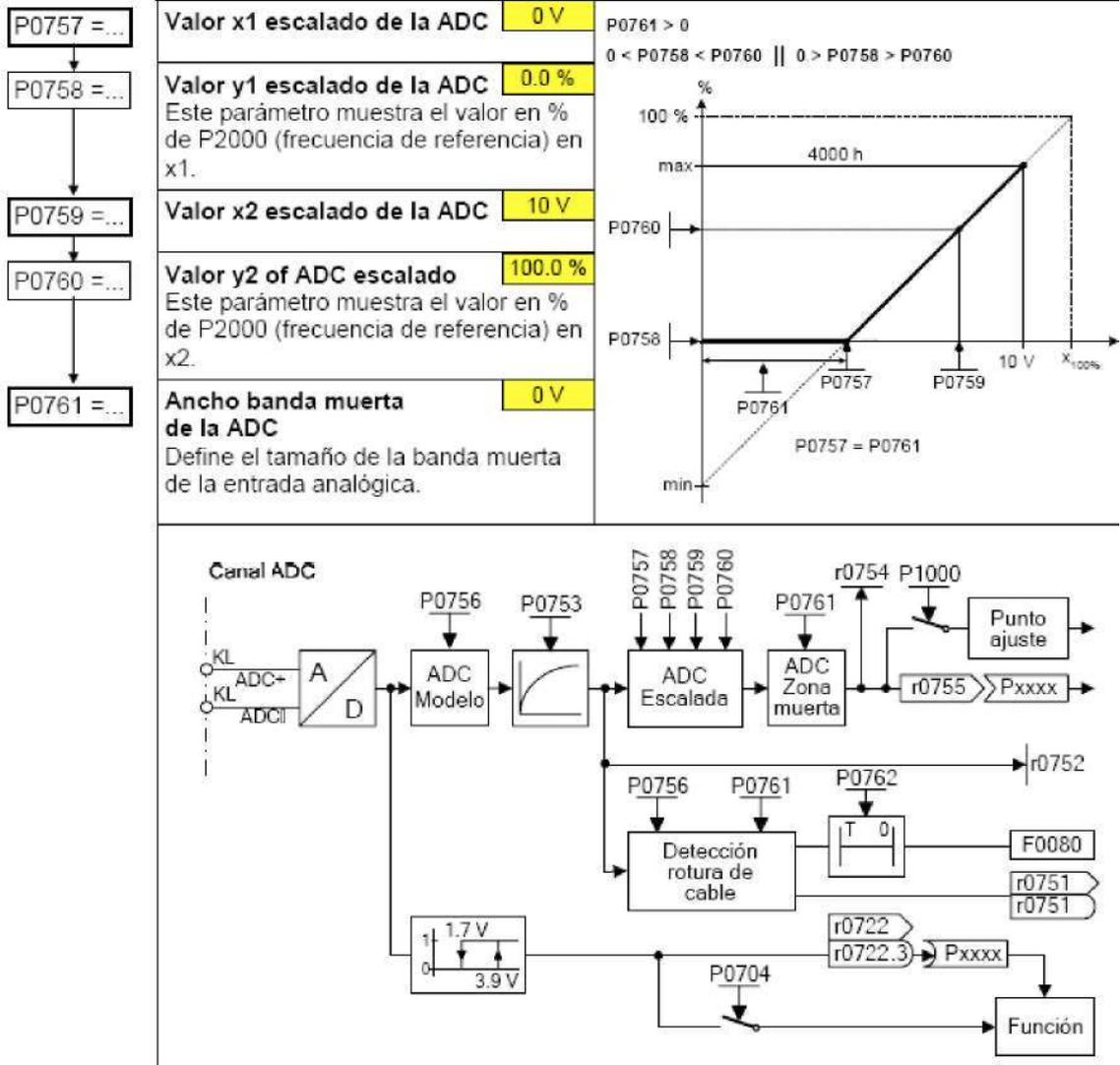
<p>P0731 = ...</p> <p>↓</p> <p>P0748 = 0</p>	<p>BI: Función de salida digital 1* 52.3</p> <p>Define la fuente de la salida digital 1.</p>	<p>Ajustes importantes / frecuentes</p> <table border="0"> <tr><td>52.0</td><td>Convertidor listo</td><td>0</td><td>Cerrado</td></tr> <tr><td>52.1</td><td>Convertidor listo para funcionar</td><td>0</td><td>Cerrado</td></tr> <tr><td>52.2</td><td>Convertidor funcionando</td><td>0</td><td>Cerrado</td></tr> <tr><td>52.3</td><td>Activación fallo convertidor</td><td>0</td><td>Cerrado</td></tr> <tr><td>52.4</td><td>OFF2 activo</td><td>1</td><td>Cerrado</td></tr> <tr><td>52.5</td><td>OFF3 activo</td><td>1</td><td>Cerrado</td></tr> <tr><td>52.6</td><td>Activación inhibición</td><td>0</td><td>Cerrado</td></tr> <tr><td>52.7</td><td>Aviso convertidor activo</td><td>0</td><td>Cerrado</td></tr> </table>	52.0	Convertidor listo	0	Cerrado	52.1	Convertidor listo para funcionar	0	Cerrado	52.2	Convertidor funcionando	0	Cerrado	52.3	Activación fallo convertidor	0	Cerrado	52.4	OFF2 activo	1	Cerrado	52.5	OFF3 activo	1	Cerrado	52.6	Activación inhibición	0	Cerrado	52.7	Aviso convertidor activo	0	Cerrado
52.0	Convertidor listo	0	Cerrado																															
52.1	Convertidor listo para funcionar	0	Cerrado																															
52.2	Convertidor funcionando	0	Cerrado																															
52.3	Activación fallo convertidor	0	Cerrado																															
52.4	OFF2 activo	1	Cerrado																															
52.5	OFF3 activo	1	Cerrado																															
52.6	Activación inhibición	0	Cerrado																															
52.7	Aviso convertidor activo	0	Cerrado																															
	<p>Invertir las salidas digitales 0</p> <p>Define los estados alto y bajo del relé par una función dada.</p>																																	

Canal DOUT

6.2.5 Selecc. consigna de frecuencia

<p>P1000 = ...</p>	<p>Selecc. consigna de frecuencia 2</p> <p>0 Sin consigna principal</p> <p>1 Consigna MOP</p> <p>2 Consigna analógica</p> <p>3 Frecuencia fija</p> <p>4 USS en conexión BOP</p> <p>5 USS en conexión COM</p> <p>6 CB en conexión COM</p>
--------------------	--

6.2.6 Entrada analógica (ADC)



6.2.7 Salida analógica (DAC)

P0771 = ...	CI: DAC Define la función de la salida analógica 0 - 20 mA.	21
P0773 = ...	Tiempo de filtrado DAC Define el tiempo de suavización [ms] para la señal de salida analógica. Este parámetro habilita la suavización de la DAC utilizando un filtro PT1.	2 ms
P0777 = ...	Valor x1 escalado de la DAC	0.0 %
P0778 = ...	Valor y1 escalado de la DAC	0
P0779 = ...	Valor x2 escalado de la DAC	100.0 %
P0780 = ...	Valor y2 escalado de la DAC	20
P0781 = ...	Ancho de la banda muerta de DAC Ajusta el ancho de la banda muerta en [mA] para la salida analógica.	0

The graph shows the DAC output in mA on the vertical axis and the input percentage on the horizontal axis. The output starts at 0 mA for 0% input and remains at 0 mA until the dead zone (P0781). After the dead zone, the output increases linearly from P0778 (y1) at input P0777 (x1) to P0780 (y2) at input P0779 (x2). The output reaches 20 mA at 100% input.

Canal DAC

The block diagram illustrates the signal flow in the DAC channel. It begins with a function block 'Función xxx' and a gain block 'r0xxx' (parameter P0771 = xxx). The signal then passes through a filter block 'P0773', a gain block 'P0775', a 'DAC Escalada' block (parameters P0777, P0778, P0779, P0780), a 'DAC Zona muerta' block (parameter P0781), and finally a 'D/A' converter block. The output is labeled 'r0774' and 'r0785'.

6.2.8 Potenciómetro motorizado (MOP)

<div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1031 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1032 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content;">P1040 =...</div>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 60%;">Memorización de consigna del MOP</td> <td style="width: 20%; text-align: center;">0</td> <td style="width: 20%;"></td> </tr> <tr> <td colspan="3">Almacena la última consigna del potenciómetro motorizado (MOP) activa previa a una orden OFF o a una desconexión.</td> </tr> <tr> <td>0</td> <td colspan="2">Cna. MOP no será guardada</td> </tr> <tr> <td>1</td> <td colspan="2">Cna. MOP será guardada (act. P1040)</td> </tr> <tr> <td>Inhibir consigna negativa-MOP</td> <td style="text-align: center;">1</td> <td></td> </tr> <tr> <td colspan="3">0 Consigna negativa del MOP habilitada</td> </tr> <tr> <td colspan="3">1 Consigna negativa del MOP inhabilitada</td> </tr> <tr> <td>Consigna del MOP</td> <td style="text-align: center;">5.00 Hz</td> <td></td> </tr> <tr> <td colspan="3">Determina la consigna el control del potenciómetro motorizado.</td> </tr> <tr> <td colspan="3">Los tiempos de aceleración y deceleración del potenciómetro motorizado se determinan con los parámetros P1120 y P1121.</td> </tr> <tr> <td colspan="3">Posibles ajustes de parámetro para el potenciómetro motorizado:</td> </tr> <tr> <td></td> <td style="text-align: center;">Selección</td> <td style="text-align: center;">aumentar MOP</td> <td style="text-align: center;">disminuir MOP</td> </tr> <tr> <td style="text-align: center;">DIN</td> <td>P0719 = 0, P0700 = 2, P1000 = 1 o P0719 = 1, P0700 = 2</td> <td style="text-align: center;">P0702 = 13 (DIN2)</td> <td style="text-align: center;">P0703 = 14 (DIN3)</td> </tr> <tr> <td style="text-align: center;">BOP</td> <td>P0719 = 0, P0700 = 1, P1000 = 1 o P0719 = 11</td> <td style="text-align: center;">UP button</td> <td style="text-align: center;">DOWN button</td> </tr> <tr> <td style="text-align: center;">USS en BOP</td> <td>P0719 = 0, P0700 = 4, P1000 = 1 o P0719 = 41</td> <td style="text-align: center;">Palabra mando USS r2032 Bit13</td> <td style="text-align: center;">Palabra mando USS r2032 Bit14</td> </tr> <tr> <td style="text-align: center;">USS en COM</td> <td>P0719 = 0, P0700 = 5, P1000 = 1 o P0719 = 51</td> <td style="text-align: center;">Palabra mando USS r2036 Bit13</td> <td style="text-align: center;">Palabra mando USS r2036 Bit14</td> </tr> <tr> <td style="text-align: center;">CB</td> <td>P0719 = 0, P0700 = 6, P1000 = 1 o P0719 = 61</td> <td style="text-align: center;">Palabra mando CB r2090 Bit13</td> <td style="text-align: center;">Palabra mando CB r2090 Bit14</td> </tr> </table>	Memorización de consigna del MOP	0		Almacena la última consigna del potenciómetro motorizado (MOP) activa previa a una orden OFF o a una desconexión.			0	Cna. MOP no será guardada		1	Cna. MOP será guardada (act. P1040)		Inhibir consigna negativa-MOP	1		0 Consigna negativa del MOP habilitada			1 Consigna negativa del MOP inhabilitada			Consigna del MOP	5.00 Hz		Determina la consigna el control del potenciómetro motorizado.			Los tiempos de aceleración y deceleración del potenciómetro motorizado se determinan con los parámetros P1120 y P1121.			Posibles ajustes de parámetro para el potenciómetro motorizado:				Selección	aumentar MOP	disminuir MOP	DIN	P0719 = 0, P0700 = 2, P1000 = 1 o P0719 = 1, P0700 = 2	P0702 = 13 (DIN2)	P0703 = 14 (DIN3)	BOP	P0719 = 0, P0700 = 1, P1000 = 1 o P0719 = 11	UP button	DOWN button	USS en BOP	P0719 = 0, P0700 = 4, P1000 = 1 o P0719 = 41	Palabra mando USS r2032 Bit13	Palabra mando USS r2032 Bit14	USS en COM	P0719 = 0, P0700 = 5, P1000 = 1 o P0719 = 51	Palabra mando USS r2036 Bit13	Palabra mando USS r2036 Bit14	CB	P0719 = 0, P0700 = 6, P1000 = 1 o P0719 = 61	Palabra mando CB r2090 Bit13	Palabra mando CB r2090 Bit14
Memorización de consigna del MOP	0																																																									
Almacena la última consigna del potenciómetro motorizado (MOP) activa previa a una orden OFF o a una desconexión.																																																										
0	Cna. MOP no será guardada																																																									
1	Cna. MOP será guardada (act. P1040)																																																									
Inhibir consigna negativa-MOP	1																																																									
0 Consigna negativa del MOP habilitada																																																										
1 Consigna negativa del MOP inhabilitada																																																										
Consigna del MOP	5.00 Hz																																																									
Determina la consigna el control del potenciómetro motorizado.																																																										
Los tiempos de aceleración y deceleración del potenciómetro motorizado se determinan con los parámetros P1120 y P1121.																																																										
Posibles ajustes de parámetro para el potenciómetro motorizado:																																																										
	Selección	aumentar MOP	disminuir MOP																																																							
DIN	P0719 = 0, P0700 = 2, P1000 = 1 o P0719 = 1, P0700 = 2	P0702 = 13 (DIN2)	P0703 = 14 (DIN3)																																																							
BOP	P0719 = 0, P0700 = 1, P1000 = 1 o P0719 = 11	UP button	DOWN button																																																							
USS en BOP	P0719 = 0, P0700 = 4, P1000 = 1 o P0719 = 41	Palabra mando USS r2032 Bit13	Palabra mando USS r2032 Bit14																																																							
USS en COM	P0719 = 0, P0700 = 5, P1000 = 1 o P0719 = 51	Palabra mando USS r2036 Bit13	Palabra mando USS r2036 Bit14																																																							
CB	P0719 = 0, P0700 = 6, P1000 = 1 o P0719 = 61	Palabra mando CB r2090 Bit13	Palabra mando CB r2090 Bit14																																																							

6.2.9 Frecuencia fija (FF)

<div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1001 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1002 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1003 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1004 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1005 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1006 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1007 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1016 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">P1017 =...</div> <div style="border: 1px solid black; padding: 2px; width: fit-content;">P1018 =...</div>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%;">Frecuencia fija 1</td> <td style="width: 15%; text-align: center;">0.00 Hz</td> <td rowspan="7" style="width: 45%; vertical-align: top;"> Al determinar la función de las entradas digitales (P0701 a P0703) se pudo seleccionar de tres formas diferentes la frecuencia fija: 15 = Selección directa (código binario) En este Modo la entrada digital selecciona la frecuencia fija correspondiente, p. ej.: Entrada digital 4 = Selección de la frecuencia fija 4. Si simultáneamente hay varias entradas activas se suman y se necesita además una orden ON. 16 = Selección directa + orden ON (código binario + ON / OFF1) En este Modo las frecuencias fijas se seleccionan igual que arriba y adicionalmente se combinan con una orden ON. 17 = Selección BCD + orden ON (código BCD + ON / OFF1) El modo con código BCD solo actúa con las entradas digitales 1 a 3. </td> </tr> <tr> <td>Frecuencia fija 2</td> <td style="text-align: center;">5.00 Hz</td> </tr> <tr> <td>Frecuencia fija 3</td> <td style="text-align: center;">10.00 Hz</td> </tr> <tr> <td>Frecuencia fija 4</td> <td style="text-align: center;">15.00 Hz</td> </tr> <tr> <td>Frecuencia fija 5</td> <td style="text-align: center;">20.00 Hz</td> </tr> <tr> <td>Frecuencia fija 6</td> <td style="text-align: center;">25.00 Hz</td> </tr> <tr> <td>Frecuencia fija 7</td> <td style="text-align: center;">30.00 Hz</td> </tr> <tr> <td>Modo Frecuencia fija - Bit 0</td> <td style="text-align: center;">1</td> <td>1 Selección directa 2 Selección directa + orden ON 3 Código de selección binario + orden ON</td> </tr> <tr> <td>Modo Frecuencia fija - Bit 1</td> <td style="text-align: center;">1</td> <td rowspan="2">NOTA En los ajustes 2 y 3, para que el convertidor acepte la orden ON, tienen que estar todos los parámetros P1016 a P1019 ajustados al mismo valor.</td> </tr> <tr> <td>Modo Frecuencia fija - Bit 2</td> <td style="text-align: center;">1</td> </tr> </table>	Frecuencia fija 1	0.00 Hz	Al determinar la función de las entradas digitales (P0701 a P0703) se pudo seleccionar de tres formas diferentes la frecuencia fija: 15 = Selección directa (código binario) En este Modo la entrada digital selecciona la frecuencia fija correspondiente, p. ej.: Entrada digital 4 = Selección de la frecuencia fija 4. Si simultáneamente hay varias entradas activas se suman y se necesita además una orden ON. 16 = Selección directa + orden ON (código binario + ON / OFF1) En este Modo las frecuencias fijas se seleccionan igual que arriba y adicionalmente se combinan con una orden ON. 17 = Selección BCD + orden ON (código BCD + ON / OFF1) El modo con código BCD solo actúa con las entradas digitales 1 a 3.	Frecuencia fija 2	5.00 Hz	Frecuencia fija 3	10.00 Hz	Frecuencia fija 4	15.00 Hz	Frecuencia fija 5	20.00 Hz	Frecuencia fija 6	25.00 Hz	Frecuencia fija 7	30.00 Hz	Modo Frecuencia fija - Bit 0	1	1 Selección directa 2 Selección directa + orden ON 3 Código de selección binario + orden ON	Modo Frecuencia fija - Bit 1	1	NOTA En los ajustes 2 y 3, para que el convertidor acepte la orden ON, tienen que estar todos los parámetros P1016 a P1019 ajustados al mismo valor.	Modo Frecuencia fija - Bit 2	1
Frecuencia fija 1	0.00 Hz	Al determinar la función de las entradas digitales (P0701 a P0703) se pudo seleccionar de tres formas diferentes la frecuencia fija: 15 = Selección directa (código binario) En este Modo la entrada digital selecciona la frecuencia fija correspondiente, p. ej.: Entrada digital 4 = Selección de la frecuencia fija 4. Si simultáneamente hay varias entradas activas se suman y se necesita además una orden ON. 16 = Selección directa + orden ON (código binario + ON / OFF1) En este Modo las frecuencias fijas se seleccionan igual que arriba y adicionalmente se combinan con una orden ON. 17 = Selección BCD + orden ON (código BCD + ON / OFF1) El modo con código BCD solo actúa con las entradas digitales 1 a 3.																						
Frecuencia fija 2	5.00 Hz																							
Frecuencia fija 3	10.00 Hz																							
Frecuencia fija 4	15.00 Hz																							
Frecuencia fija 5	20.00 Hz																							
Frecuencia fija 6	25.00 Hz																							
Frecuencia fija 7	30.00 Hz																							
Modo Frecuencia fija - Bit 0	1	1 Selección directa 2 Selección directa + orden ON 3 Código de selección binario + orden ON																						
Modo Frecuencia fija - Bit 1	1	NOTA En los ajustes 2 y 3, para que el convertidor acepte la orden ON, tienen que estar todos los parámetros P1016 a P1019 ajustados al mismo valor.																						
Modo Frecuencia fija - Bit 2	1																							

6.2.10 JOG

P1058 =...	Frecuencia JOG derecha 5.00 Hz	
P1059 =...	Frecuencia JOG izquierda 5.00 Hz	
P1060 =...	Tiempo de aceleración JOG 10.00 s	
P1061 =...	Tiempo de deceleración JOG 10.00 s	

6.2.11 Generador de rampas (RFG)

P1091 =...	Frecuencia inhibida 1 (En Hz) 0.00 Hz	
P1091 =...	Frecuencia inhibida 2 0.00 Hz	
P1091 =...	Frecuencia inhibida 3 0.00 Hz	
P1091 =...	Frecuencia inhibida 4 0.00 Hz	
P1101 =...	Ancho de banda para frecuencias (En Hz) 2.00 Hz	
P1120 =...	Tiempo de aceleración (en s) 10.00 s	
P1121 =...	Tiempo de deceleración (En s) 10.00 s	<p>Se recomienda utilizar los tiempos de redondeo como prevención ante reacciones bruscas y evitar así deterioros en las partes mecánicas. Los tiempos de aceleración y deceleración se alargan según las rampas de redondeo.</p>
P1130 =...	T. redondeo inicial aceleración (En s) 0.00 s	
P1131 =...	T. redondeo final aceleración (En s) 0.00 s	
P1132 =...	T. redondeo inicial deceleración (En s) 0.00 s	
P1133 =...	T. redondeo final deceleración (En s) 0.00 s	
P1134 =...	Tipo de redondeo 0	
P1135 =...	Tiempo deceleración OFF3 5.00 s	

6.2.12 Frecuencias límite y de referencia

P1080 = ...	Frecuencia mínima (En Hz)	0.00 Hz
↓	Ajusta la frecuencia mínima del motor a la cual el motor funcionará independientemente de la consigna de frecuencia. Si la consigna queda por debajo del valor de P1080, se pone la frecuencia de salida a P1080 tomando en cuenta el signo.	
P1082 = ...	Frecuencia máx. (En Hz)	50.00 Hz
↓	Ajusta la frecuencia de motor máxima a la cual el motor funcionará independientemente de la consigna de frecuencia. Si la consigna sobrepasa el valor de P1082, se limita la frecuencia de salida. El ajuste de este valor es válido para ambos sentidos de rotación horaria y antihoraria.	
P2000 = ...	Frecuencia de referencia (En Hz)	50.00 Hz
	La frecuencia de referencia en Hz corresponde al 100 %. Se puede cambiar el ajuste si se necesita una frecuencia máxima más alta de 50 Hz. Esta cambia automáticamente a 60 Hz, si se ha seleccionado con el interruptor DIP50/60 o con P0100 la frecuencia estándar de 60 Hz. NOTA Esta frecuencia de referencia actúa sobre la frecuencia de consigna, ya que tanto las consignas analógicas (100 % $\hat{=}$ P2000) como las de frecuencia vía USS (4000H $\hat{=}$ P2000) se basan en ese valor.	

6.2.13 Regulación del motor

P1300 = ...	Modo de control	0
↓	Con este parámetro se selecciona el modo de regulación. Con el Modo "característica V/f" se determina la relación entre la tensión de salida y la frecuencia de salida del convertidor. 0 V/f con característ. lineal 1 V/f con FCC 2 V/f con característ. parabólica 3 V/f con característ. programable (→ P1320 – P1325)	
P1310 = ...	Elevación continua (En %)	50.00 %
↓	Elevación de tensión en % relativo a P0305 (corriente nominal del motor) o P0350 (resistencia del estator). P1310 es válido para todas las variantes V/f (véase P1300). Si las frecuencias de salida son bajas no se pueden despreciar las resistencias efectivas de la bobina para mantener el flujo del motor.	
	<p>Lineal V/f</p> <p>Tensión de elevación</p> <p>Real V_{Boost}</p> <p>Normal V/f (P1300 = 0)</p> <p>Rango de validez</p> <p>ON OFF</p> <p>f</p> <p>P1310 activo</p> <p>1 0</p> <p>V_{max}</p> <p>V_n (P0304)</p> <p>$V_{ConBoost,100}$</p> <p>$V_{ConBoost,50}$</p> <p>0</p> <p>$f_{Boost, end}$ (P1316)</p> <p>f_n (P0310)</p> <p>f_{max} (P1082)</p> <p>f</p>	
P1311 = ...	Elevación para aceleración (En %)	0.0 %
↓	Elevación de tensión para acelerar/frenar en % relativo a P0305 (corriente nominal del motor) o P0350 (resistencia del estator). P1311 solo eleva la tensión durante la aceleración y la deceleración de la rampa y genera un par adicional para acelerar/frenar. Al contrario de P1312, que solo es activo durante el primer proceso de aceleración después de la orden ON, P1311 actúa en cada proceso de aceleración y frenado.	

P1312 =...	Elevación en arranque (En %)	0.0 %
	Elevación de tensión al arrancar (después de la orden ON) si se usa la característica V/f lineal o parabólica en % relativo a P0305 (corriente nominal del motor) o P0350 (resistencia del estator). La elevación de tensión permanece activa hasta que: 1) se alcanza por primera vez el valor de consigna o 2) la consiga se reduce a un valor menor que el valor actual en la salida del generador de rampas.	
P1320 =...	Coord.1 freq. program. curva V/F	0.0 Hz
	Ajusta las coordenadas V/f (P1320/1321 a P1324/1325) para definir la característica V/f.	
P1321 =...	Coord.1 tens. program. curva V/F	0.0 Hz
P1322 =...	Coord.2 freq. program. curva V/F	0.0 Hz
P1323 =...	Coord.2 tens. program. curva V/F	0.0 Hz
P1324 =...	Coord.3 freq. program. curva V/F	0.0 Hz
P1325 =...	Coord.3 tens. program. curva V/F	0.0 Hz
	<p> $P1310[V] = \frac{P1310[\%]}{100[\%]} \cdot \frac{r0395[\%]}{100[\%]} \cdot P0304[V]$ </p>	
P1335 =...	Compensación del deslizamiento (En %)	0.0 %
	Ajuste dinámico de la frecuencia de salida del convertidor a fin de mantener constante la velocidad del motor con independencia de la carga del mismo.	
P1338 =...	Amortiguam.resonanc.ganacia V/f	0.00
	Define ganancia de resonancia para V/f.	

6.2.14 Protección convertidor/motor

P0290 =...	Reacción convert. ante sobrec. Selecciona la reacción del convertidor ante una temperatura excesiva. 0 Reducción de frec. de salida 1 Fallo (F0004) 2 Pulso & reducción frec. sal. 3 Reducción frec. pulsos, fallo (F0004)	0
P0292 =...	Alarma de sobrecarga convertidor Establece la diferencia de temperatura (en °C) entre el umbral de desconexión por exceso de temperatura y el umbral de aviso por exceso de temperatura del convertidor. El correspondiente umbral de desconexión está consignado en el convertidor, por lo que el usuario no puede modificarlo	15 °C
P0335 =...	Refrigeración del motor (Sistema de refrigeración) 0 Autoventilado 1 Ventilación forzada	0
P0610 =...	Reacción I²t motor Define la reacción cuando se alcanza el umbral de aviso I ² t. 0 Aviso, sin reacción, sin fallo F0011 1 Aviso, reducción de I _{max} , fallo F0011 2 Aviso, sin reacción, fallo F0011	2
P0611 =...	Constante tiempo I²t del motor (En s) El parámetro P0611 se evalúa automáticamente a partir de los datos del motor durante la puesta en marcha rápida o durante el cálculo de los parámetros del motor P0340. Al término de la puesta en marcha rápida o del cálculo de los parámetros del motor, este valor se puede sustituir por el valor del fabricante del motor.	100 s
P0614 =...	Nivel al. p. sobrecarga I²t motor (En %) Define el valor al cual se genera el aviso A0511 (sobretensión del motor).	100.0 %
	<p>Umbral de desconexión 1.1·P0614</p> <p>Umbral de temperatura del motor P0614</p>	
P0640 =...	Factor sobrecarga motor Define el límite de intensidad de sobrecarga del motor en [%] relativo a P0305 (intensidad nominal del motor).	150.0 %

Funciones específicas del convertidor

Rearranque al vuelo

Rearranque al vuelo

0

Arranca el convertidor sobre un motor girando cambiando la frecuencia de salida del convertidor hasta que se encuentra la velocidad real del motor. Entonces, el motor subirá hasta la consigna utilizando el tiempo de rampa normal.

- 0 Rearranque volante deshabilitado
- 1 Rearranque volante activo siempre, arranque en la dirección de la consigna
- 2 Rearranque volante tras encendido, fallo. OFF2, arranque en la dirección de consigna
- 3 Rearranque volante activo tras fallo, OFF2, arranque en la dirección de consigna
- 4 Rearranque volante activo siempre, sólo en la dirección de consigna
- 5 Rearranque volante activo tras encendido, fallo. OFF2. sólo en la dirección de consigna
- 6 Rearranque volante activo tras fallo, OFF2, sólo en la dirección de consigna

Corriente-motor:Rearran.al vuelo (En %)

100%

Define la intensidad de búsqueda utilizada para el arranque al vuelo.

Búsqueda velocidad:Rear,al vuelo (En %)

100 %

Ajusta el factor por el cual la frecuencia de salida cambia durante el reارئانque al vuelo para sincronizarse con el motor que gira.

Rearranque automático

Rearranque automático

Habilita el reارئانque después de un fallo principal o después de un fallo.

- 0 Inhabilitado
- 1 Disp.re. tras ene.
- 2 Re.tras apagón pr.
- 3 Re.tras corte pr.o f.
- 4 Re.tras corte pr.
- 5 Retías ap.pr.yfa.
- 6 Re. t.co./ap. pr.of.

6.2.15.3 Freno de mantenimiento

P1215 =...	Habilitación del freno manten. 0 Habilita/deshabilita la función del mantenimiento del freno 0 Freno mantenim. motor deshabil. 1 Freno mantenim. motor habil. NOTA Para control del relé del freno vía salida digital: P0731 = 52.12 (= 52.C)
P1216 =...	Retardo apertura d.freno manten. (En s) 1.0 s Define el intervalo de tiempo durante el cual el convertidor funciona con la frecuencia mínima P1080 (después de la magnetización), antes de que comience la aceleración.
P1217 =...	Tiempo cierre tras deceleración (En s) 1.0 s Define el tiempo durante el cual el convertidor funciona a la frecuencia mínima (P1080) después de la deceleración en el punto 2.

6.2.15.4 Freno electrónico

P1232 =...	Corriente frenado c.continua (En %) 100 % Define el nivel de corriente continua en [%] relativo la intensidad nominal del motor (P0305).
P1233 =...	Duración del frenado c.continua (En s) 0 s Define cuanto dura la inyección de corriente en c.c. para frenar tras una orden OFF1 / OFF3.

6.2.15.5 Corriente frenado combinado

P1236 =...	Corriente frenado combinado (En %) 0 % Define el nivel en c. c. superpuesto a la forma de onda de corriente alterna. El valor es introducido en [%] relativo a la intensidad nominal del motor (P0305). Si P1254 = 0 : Umbral de activacion de freno combinado $U_{DC_Comp} = 1.13 \cdot \sqrt{2} \cdot V_{mains} = 1.13 \cdot \sqrt{2} \cdot P0210$ por los demás: Umbral de activacion de freno combinado $U_{DC_Comp} = 0.98 \cdot r1242$
------------	---

6.2.15.6 Regulador Vdc

P1240 =...	Configuración del regulador Vdc 1 Habilita / deshabilita el regulador Vdc. 0 Controlador Vdc deshabilitado 1 Controlador Vdc-máx habilitado	
P1254 =...	Autodetección niveles conex. Vdc 1 Activa y desactiva el ajuste automático de los umbrales de activación para la regulación del circuito intermedio. 0 Deshabilitado 1 Habilitado	

6.2.15.7 Regulador PID

P2200 = ...	BI: Habilitación regulador PID Modo PID Permite al usuario habilitar/deshabilitar el regulador PID. El ajuste a 1 habilita el regulador de lazo cerrado PID. El ajuste a 0 deshabilita automáticamente los tiempos de rampa normales ajustados en el P1120 y el P1121 y las consigna de frecuencia normales.	0.0
P2253 = ...	CI: Consigna PID Define la fuente de consigna para la entrada de consigna PID.	0.0
P2254 = ...	CI: Fuente compensación PID Selecciona la fuente de compensación para la consigna PID. Esta señal se multiplica por la ganancia de compensación y se añade a la consigna del PID.	0.0
P2257 = ...	Tiempo de aceleración cna. PID Ajusta el tiempo de aceleración para la consigna PID.	1.00 s
P2258 = ...	Tiempo de deceleración cna. PID Ajusta el tiempo de deceleración para la consigna PID.	1.00 s
P2264 = ...	CI: Realimentación PID Selecciona la fuente para la señal de realimentación del PID.	755.0
P2267 = ...	Valor máx. realimentación PID Ajusta el límite superior para el valor de la señal de realimentación en [%].	100.00 %
P2268 = ...	Valor mín. realimentación PID Ajusta el límite inferior para el valor de la señal de realimentación en [%].	0.00 %
P2280 = ...	Ganancia proporcional PID Permite al usuario ajustar la ganancia proporcional para el regulador PID.	3.000
P2285 = ...	Tiempo de integración PID Ajusta la constante de tiempo de integración para el regulador PID.	0.000 s
P2291 = ...	Límite superior salida PID Ajuste del límite superior para la salida del regulador PID en [%].	100.00 %
P2292 = ...	Límite inferior salida PID Ajuste del límite inferior de salida del regulador PID en [%].	0.00 %

Ejemplo:

Parámetro	Texto	Ejemplo
P2200	BI: Habilitación regulador PID	P2200 = 1.0 Regulador PID activo
P2253	CI: Consigna PID	P2253 = 2224 Consigna Fija PI
P2264	CI: Realimentación PID	P2264 = 755 ADC
P2267	Valor máx. realimentación PID	P2267 Adaptar a aplicación
P2268	Valor mín. realimentación PID	P2268 Adaptar a aplicación
P2280	Ganancia proporcional PID	P2280 Averiguado por optimización
P2285	Tiempo de integración PID	P2285 Averiguado por optimización
P2291	Límite superior salida PID	P2291 Adaptar a aplicación
P2292	Límite inferior salida PID	P2292 Adaptar a aplicación

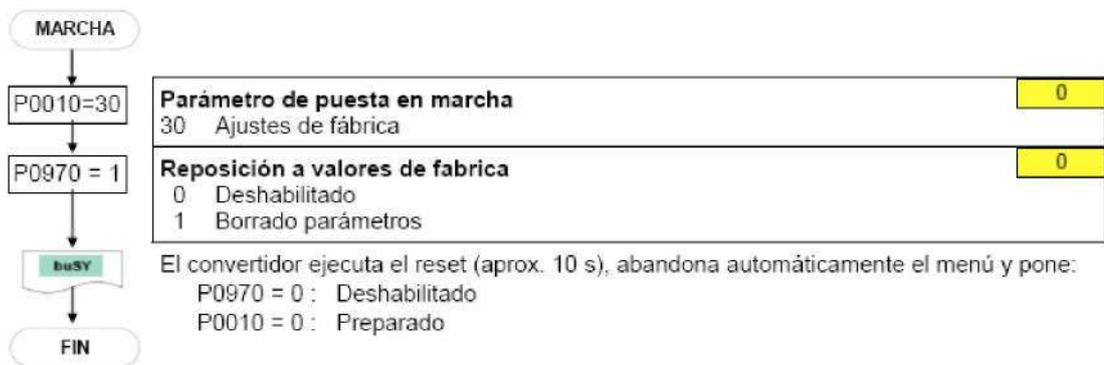
6.3 Puesta en servicio en serie

Si ya se dispone de un juego de parámetros se puede transmitir al convertidor MICROMASTER 420 con STARTER o bien con DriveMonitor (véase sección 4.1 "Estructura de la comunicación MICROMASTER 420 ⇔ STARTER").

Aplicaciones típicas donde efectuar la puesta en servicio en serie son:

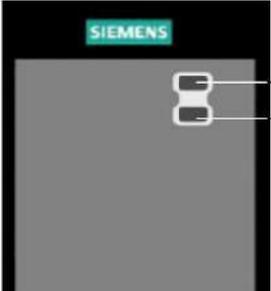
1. Poner en servicio varios accionamientos con la misma configuración y las mismas funciones. En el primer accionamiento se tiene que efectuar una puesta en servicio rápida o según la aplicación (primera puesta en servicio), cuyos valores de parámetros se transmitirán a los otros accionamientos.
2. Cambio del convertidor MICROMASTER 420.

6.4 Reset de parámetros al ajuste de fábrica



7 Visualizaciones y mensajes

7.1 Indicadores de estado LED

			
●	Red no presente	●	OFF
☀	Preparado para funcionar	☀	ON
●	Fallo en convertidor, uno de los listados abajo	⊙	ca. 0,3 s, Parpadeo
☀	Convertidor en marcha	⊙	ca. 1 s, Intermitencia
●	Fallo sobrecorriente	☀	Fallo sobretensión
⊙	Fallo sobretensión	☀	Fallo sobretensión convertidor
☀	Fallo sobretensión motor	⊙	Alarma límite corriente
		⊙	Ambos LEDs intermitentes al mismo tiempo
		⊙	Otras alarmas
		⊙	Ambos LEDs intermitentes alternativamente
		⊙	Disparo/alarma por mínima tensión
		⊙	Accionamiento no listo
		⊙	Fallo en ROM
		⊙	Ambos LEDs parpadean al mismo tiempo
		⊙	Fallo en RAM
		⊙	Ambos LEDs parpadean alternativamente

7.2 Fallos y alarmas

Fallo	Significado
F0001	Sobrecorriente
F0002	Sobretensión
F0003	Subtensión
F0004	Sobretensión convertidor
F0005	Convertidor I ² T
F0011	Sobretensión I ² T del motor
F0041	Fallo en la identificación de datos del motor
F0051	Fallo parámetro EEPROM
F0052	Fallo pila de energía
F0060	Timeout del ASIC
F0070	CB fallo consigna
F0071	USS (enlace-BOP) fallo consigna
F0072	USS (enlace COMM) fallo consigna
F0080	pérdida señal de entrada ADC
F0085	Fallo Externo
F0101	Desbordamiento de memoria
F0221	Realimentación PID por debajo del valor mínimo. valor
F0222	PID realimentación por encima del máximo. valor
F0450	Fallo en test BIST (sólo para modo de servicio técnico)

Alarma	Significado
A0501	Límite corriente
A0502	Límite por sobretensión
A0503	Límite de mínima tensión
A0504	Sobretensión del convertidor
A0505	I ² T del convertidor
A0506	Ciclo de carga del convertidor
A0511	Sobretensión I ² t
A0541	Identificación de datos de motor activo
A0600	Aviso RTOS
A0700 - A0709	CB alarma
A0710	Error comunicaciones CB
A0711	Error configuración CB
A0910	Regulador Vdc-max activo
A0911	Regulador Vdc-max activo
A0920	Los parámetros del ADC no están ajustados adecuadamente
A0921	Los parámetro de DAC no ajustados correctamente
A0922	No hay carga aplicada al convertidor
A0923	Señales JOG a derechas y JOG a izquierdas activas

También puede obtener información sobre el MICROMASTER 420 por los siguientes medios:

Representante regional

Contacte con el soporte técnico de su región para obtener información sobre servicios, precios y condiciones.

Soporte técnico central

Asesoramiento competente en cuestiones técnicas sobre nuestros productos y sistemas un amplio espectro de prestaciones.

Europa/ África

Tel: +49(0)180 5050 222
Fax: +49(0)180 5050 223
Email: adsupport@siemens.com

América

Tel: +1 423 262 2522
Fax: +1 423 262 2589
Email: simatic.hotlinefai.sea@siemens.com

Asia/ Pacífico

Tel: +86 1064 757 575
Fax: +86 1064 747 474
Email: adsupport.asia@siemens.com

Servicio Online & Support

Sistema de información vía internet amplio y con acceso las 24 h: soporte de productos, servicios y prestaciones incluido el soporte de herramientas de PC.

<http://www.siemens.com/automation/service&support>

Dirección en internet

Dirección donde puede obtener información técnica y general:
<http://www.siemens.com/micromaster>

ANEXO B MOTOR

Placa de características

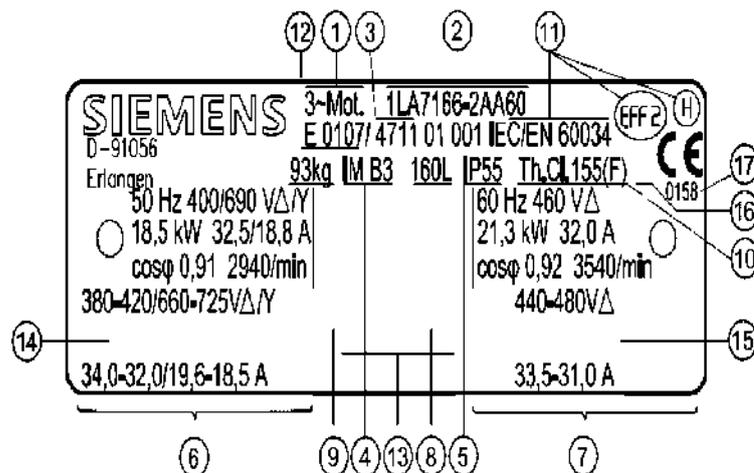
Datos técnicos

La placa de características de la máquina contiene los datos técnicos válidos para la máquina entregada.

La placa de características de la máquina incluye los datos técnicos siguientes:

Tabla 2-2 Placa de características de la máquina - Datos técnicos

Posición	Datos técnicos	Posición	Datos técnicos
1	Tipo de máquina: Máquina trifásica de baja tensión	10	Clase térmica
2	Referencia	11	Normas y reglamentos
3	N.º de fabricación (n.º de identificación, n.º de serie)	12	Fecha de fabricación AAMM
4	Forma constructiva	13	Homologaciones adicionales (opcionales)
5	Grado de protección	14	Altitud de instalación (sólo si supera los 1.000 m)
6	Datos a 50 Hz	15	Datos del cliente (opción)
7	Datos a 60 Hz	16	Gama de temperatura de empleo (sólo si se desvía de la norma)
8	Tamaño constructivo	17	Número de identificación del organismo de inspección técnica (opcional)
9	Peso de la máquina		



Diseño

Versión de la máquina

Las máquinas de esta gama son motores asíncronos trifásicos de baja tensión autoventilados natural con extremo de eje cilíndrico y chavetero. Están disponibles modelos de una velocidad y distintas clases de rendimiento y de polos conmutables para varias velocidades.

En el caso de máquinas con patas (modelo IM B3), éstas se encuentran atornilladas o integradas por fundición.

Es posible cambiar de posición de las patas atornilladas a la carcasa de la máquina (p. ej., para modificar la posición de la caja de conexiones), pero sólo podrá encomendarse dicha tarea a empresas autorizadas.

A continuación, deberán repasarse las superficies de apoyo de las patas para que queden niveladas y en paralelo a los ejes de la máquina y, si es preciso, deberán suplementarse. Las superficies pintadas que presenten daños deberán repararse adecuadamente.

Normas

Resumen

Las máquinas son conformes a las siguientes normas:

Tabla 2-3 Normas y prescripciones aplicables

Característica	Norma
Dimensionamiento y comportamiento en funcionamiento	IEC/EN 60034-1
Grado de protección	IEC/EN 60034-5
Refrigeración	IEC/EN 60034-6
Forma constructiva	IEC/EN 60034-7
Designación de conexiones y sentido de giro	IEC/EN 60034-8
Emisión de ruidos	IEC/EN 60034-9
Comportamiento de arranque, máquinas eléctricas giratorias	IEC/EN 60034-12
Niveles de vibraciones mecánicas	IEC/EN 60034-14
Tensiones normalizadas IEC	IEC 60038

Normas complementarias para máquinas con protección contra explosiones

Resumen

Tabla 2-4 Normas y prescripciones aplicables

Característica	Norma
Material eléctrico para atmósferas de gas explosivas. Parte 0: Requisitos generales	IEC/EN 60079-0
Material eléctrico para atmósferas de gas explosivas. Parte 1: Envoltentes antideflagrantes "d"	IEC/EN 60079-1
Material eléctrico para atmósferas de gas explosivas. Parte 7: Seguridad aumentada "e"	IEC/EN 60079-7
Material eléctrico para atmósferas de gas explosivas. Parte 14: Instalaciones eléctricas en emplazamientos peligrosos (a excepción de minas)	IEC/EN 60079-14
Material eléctrico para atmósferas de gas explosivas. Parte 15: Modo de protección "n"	IEC/EN 60079-15
Material eléctrico para atmósferas de gas explosivas. Parte 19: Reparación y revisión	IEC/EN 60079-19
Material eléctrico para uso en presencia de polvo combustible. Parte 0: Requisitos generales	IEC/EN 61241-0
Material eléctrico para uso en presencia de polvo combustible. Parte 1: Protección por carcasa "tD"	IEC/EN 61241-1
Material eléctrico para uso en presencia de polvo combustible. Parte 17: Inspección y mantenimiento de instalaciones eléctricas en emplazamientos peligrosos (con excepción de las minas)	IEC/EN 61241-17
Directiva relativa a la aproximación de las legislaciones de los Estados miembros sobre los aparatos y sistemas de protección para uso en atmósferas potencialmente explosivas	94/9/CE

Refrigeración, ventilación

Generalidades

Las máquinas de esta gama son máquinas asíncronas trifásicas con un circuito de refrigeración primario (interno) cerrado y un circuito de refrigeración por aire secundario abierto (refrigeración superficial). La refrigeración superficial puede variar según la versión:

Pasos previos a la instalación

Transporte

Utilizar cáncamos de elevación

PRECAUCIÓN

Durante el transporte deben apretarse bien todas los cáncamos de elevación enroscados presentes en la máquina. Estos cáncamos están diseñados para izar únicamente el peso de la máquina; no se debe agregar ningún peso adicional. Si es necesario se deben utilizar elementos auxiliares de elevación y transporte adecuados con la capacidad suficiente, como eslingas planas de levantamiento (EN 1492-1) y correas de amarre o estiba (EN 12195-2).

Transporte suspendido

ADVERTENCIA

Para el transporte y la instalación deben utilizarse eslingas y otros medios auxiliares adecuados. En el caso de elevación o transporte con varios auxiliares, dos cables o eslingas deben poder soportar todo el peso.
¡Los medios de transporte y elevación deberán asegurarse contra deslizamiento!

Retirar cualquier posible seguro de transporte antes de la puesta en marcha y guardarlo o anularlo. Si más adelante se realiza otro transporte, volver a utilizarlo o activarlo.

La máquina se embalará de acuerdo al tipo de transporte y a su tamaño. A menos que en el contrato se especifique lo contrario, el embalaje cumplirá las directrices de embalaje de las ISPM (normas internacionales para las medidas de protección de plantas).

Se deben observar los símbolos que aparezcan en el embalaje. Se deben observar los símbolos que aparezcan en el embalaje. Tienen el siguiente significado:



arriba



Frágil



Proteger de la
humedad



Proteger del calor

Almacenamiento

Almacenamiento al aire libre

A ser posible, se tiene que elegir un lugar de almacenamiento a prueba de inundaciones, sin vibraciones y seco. Antes de almacenar, reparar los daños en el embalaje si es necesario para el almacenamiento correcto. Colocar las máquinas, los aparatos y las cajas sobre paletas, vigas o cimientos que garanticen la protección contra la humedad del suelo. Se tiene que evitar el hundimiento en la tierra, y no debe quedar obstaculizada la circulación del aire debajo de los materiales almacenados.

Las cubiertas o toldos para la protección contra la intemperie no deben tocar la superficie del material almacenado. Mediante maderas distanciadoras insertadas se tiene que asegurar una suficiente circulación del aire.

PRECAUCIÓN
En caso de condiciones climáticas extremas, como atmósferas con sal o polvo, deben tomarse las precauciones adecuadas.

Almacenamiento en el interior

Los almacenes deberían estar secos, a prueba de heladas, libres de polvo, vibraciones y choques y bien ventilados. Asimismo, deben ofrecer protección contra condiciones meteorológicas extremas.

Superficies metálicas desnudas

Las piezas desnudas deben protegerse durante el transporte con un producto anticorrosión no permanente (< 6 meses). En caso de almacenamiento más prolongado, el cliente deberá tomar las oportunas medidas de protección anticorrosión para las superficies metálicas desnudas (extremo del eje, zona de unión de bridas, zonas de atornillado de patas).

Vida útil de los cojinetes

Tiempo de almacenamiento

En caso de almacenamiento prolongado se acorta la duración del uso de la grasa de los cojinetes. Si se almacena durante más de 12 meses, se tiene que revisar el estado de la grasa. En caso de que en la revisión se viera que ha desaparecido o se ha ensuciado la grasa (la penetración de agua de condensación modifica la consistencia de la grasa), deberá cambiarse la grasa.

Rodamientos

Los ejes deben girarse 1 vez al año para evitar marcas indelebles por permanecer parados. Los rodamientos deben renovarse si el período entre el suministro y la puesta en marcha de la máquina supera los 4 años. A medida que aumenta el tiempo de almacenaje aumenta la probabilidad de degradación de los rodamientos.

Compatibilidad electromagnética

Las máquinas en construcción cerrada (IP 55 y superior), si se usan para los fines especificados y se utilizan conectadas a la red de suministro eléctrico con las características estipuladas por la euronorma EN 50160, cumplen los requisitos de la Directiva de Compatibilidad Electromagnética 89/336/CEE.

ATENCIÓN

Si el par presenta fuertes irregularidades (p. ej., accionamiento de un compresor de émbolo), se fuerza en la máquina una corriente no senoidal cuyos armónicos podrían contaminar inadmisiblemente la red y, por consiguiente, provocar la propagación de perturbaciones no permitidas.

ATENCIÓN

Si el motor es alimentado por un convertidor, según la ejecución de éste (modelo, desparasitaje, fabricante) se emiten perturbaciones de diversa intensidad. Para que el sistema de accionamiento, compuesto de máquina y convertidor, no supere los límites especificados en la norma EN 50081 es imprescindible respetar las instrucciones relativas a la compatibilidad electromagnética dadas por el fabricante del convertidor. Si éste recomienda un cable de alimentación apantallado, el apantallamiento será más eficaz si se conecta eléctricamente sobre una superficie amplia a la caja de conexiones metálica de la máquina (con pasacables de metal).

En el caso de máquinas con sensores integrados (p. ej., termistores PTC), el convertidor puede inducir perturbaciones en el cable al sensor.

Inmunidad a perturbaciones

Las máquinas cumplen por principio los requisitos de inmunidad a perturbaciones según la norma EN 50082. En máquinas con sensores incorporados (p. ej., termistores PTC), el usuario debe procurar por su cuenta la suficiente inmunidad a perturbaciones eligiendo un cable al sensor (dado el caso, con apantallamiento, conexionado como en el cable de alimentación) y un relé de protección (aparato de evaluación) adecuados.

Si una máquina alimentada por convertidor funciona a una velocidad mayor a la asignada, deberán observarse las velocidades límite mecánicas (Safe Operating speed IEC/EN 60034-1).

Gestión de residuos

Para eliminar las máquinas se deben respetar las prescripciones nacionales y locales para un proceso de reciclaje normal o se debe realizar una devolución al fabricante,

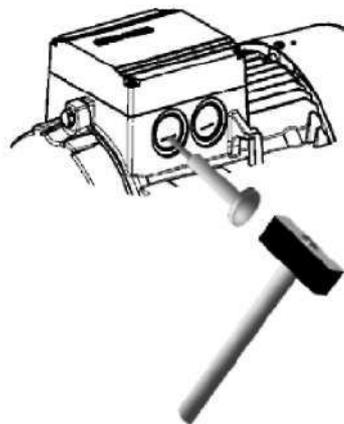
Para la gestión de residuos se ha de tener en cuenta lo siguiente:

- Aceites y grasas según la normativa sobre aceites usados
- No mezclar con disolventes, productos de limpieza en frío o restos de pintura
- Las piezas reutilizables se clasifican por:
 - residuos electrónicos (electrónica del captador)
 - hierro
 - aluminio
 - metal no ferroso (devanados, ruedas sin fin)
 - plástico

Entrada de los cables

Orificios pretrquelados rompibles

ATENCIÓN
Los orificios pretrquelados de la caja de conexiones deben romperse de la forma adecuada. La caja de conexiones, la placa de bornes, las conexiones de cable, etc. del interior de la caja de conexiones no deben dañarse.



Desmontaje

Instrucciones para el desarme

ATENCIÓN
Antes de comenzar el desmontaje, debe indicarse la asignación de los elementos de fijación y la disposición de las conexiones internas para el ensamblaje.

Ventilador

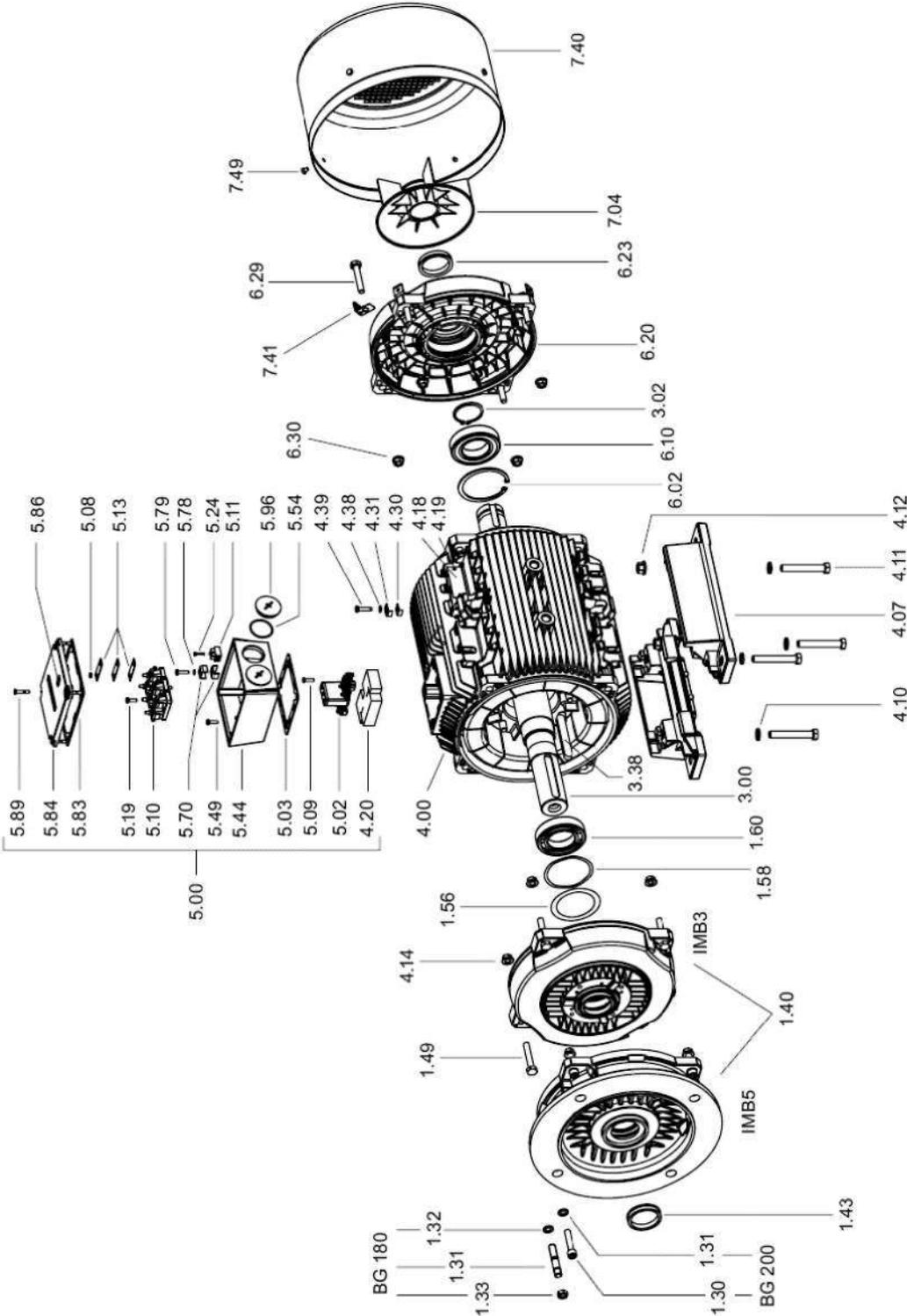
Si el ventilador tiene mecanismos de abroche, hay que cuidar de no dañarlos. En caso de daños, solicitar nuevas piezas.

Insertos de cojinete

Proteger los cojinetes contra la penetración de suciedad y humedad.

Conexiones desmontables

- Sustituir los tornillos que puedan estar corroídos.
- No dañar los aislamientos de las piezas que conduzcan tensión.
- Documentar la posición de las placas de características y placas adicionales que haya que desmontar.
- Evitar daños en los bordes de centrado.



ANEXO C REGLAS PARA MONTAR EL S7-200

Reglas para montar el S7-200

El S7-200 se puede montar en un armario eléctrico o en un rail normalizado (DIN), bien sea horizontal o verticalmente.

Alejar los equipos S7-200 de fuentes de calor, alta tensión e interferencias

Como regla general para la disposición de los equipos que conforman el sistema, aleje siempre los aparatos de alta tensión que generan interferencias de los equipos de baja tensión y de tipo lógico, tales como el S7-200.

Al configurar la disposición del S7-200 en el armario eléctrico, tenga en cuenta los aparatos que generan calor y disponga los equipos electrónicos en las zonas más frías del armario eléctrico. El funcionamiento de equipos electrónicos en entornos de alta temperatura acorta su vida útil.

Considere también la ruta del cableado de los equipos montados en el armario eléctrico. Evite colocar los conductores de señalización y los cables de comunicación en una misma canalización junto con los cables c.a. y los cables c.c. de alta tensión y de conmutación rápida.

Prever espacio suficiente para la ventilación y el cableado

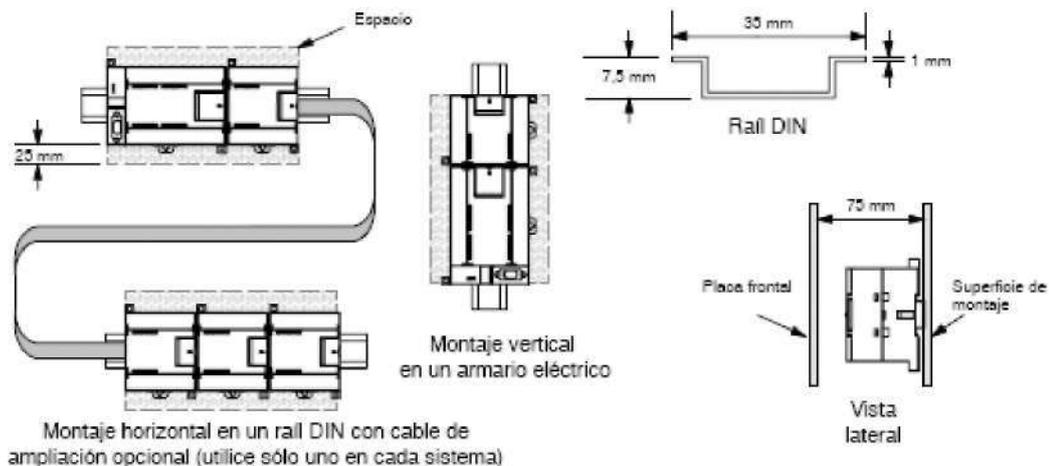
Para los equipos S7-200 se ha previsto la ventilación por convección natural. Por tanto, se deberá dejar un margen mínimo de 25 mm por encima y por debajo de los equipos. Asimismo, prevea 75 mm para la profundidad de montaje.



Consejo

En el montaje vertical, la temperatura ambiente máxima admisible se reduce en 10° C. Monte la CPU S7-200 debajo de los módulos de ampliación.

Al planificar la disposición del sistema S7-200, prevea espacio suficiente para el cableado y la conexión de los cables de comunicación. Para mayor flexibilidad al configurar la disposición del sistema S7-200, utilice un cable de conexión para los módulos de ampliación.



Alimentación

Las CPUs S7-200 tienen integrada una fuente de alimentación capaz de abastecer la CPU, los módulos de ampliación y otras cargas que precisen 24 V c.c.

La GPU 37-200 suministra la corriente continua de 5 V necesaria para los módulos de ampliación del sistema. Preste especial atención a la configuración del sistema para garantizar que la CPU pueda suministrarla corriente de 5V necesaria para los módulos de ampliación seleccionados. Si la configuración requiere más corriente de la que puede suministrar la CPU, deberá retirar un módulo o seleccionar una CPU de mayor capacidad. Para más información acerca de la corriente continua de 5 V c.c. que pueden aportar las diferentes CPUs S7-200 y la alimentación de 5 V c.c. que requieren los módulos de ampliación, consulte el anexo A. Consulte el anexo B para determinar cuánta energía (o corriente) puede suministrar la CPU a la configuración deseada.

Todas las CPUs S7-200 aportan también una alimentación para sensores de 24 V c.c. que puede suministrar corriente de 24 V c.c. a las entradas y a las bobinas de relés de los módulos de ampliación, así como a otros equipos. Si los requisitos de corriente exceden la capacidad de la alimentación para sensores, será preciso agregar una fuente de alimentación externa de 24 V c.c. al sistema. Para más información acerca de la capacidad de alimentación para sensores de 24 V c.c. que pueden aportar las diferentes CPUs S7-200, consulte el anexo A.

Si se precisa una fuente de alimentación externa de 24 V c.c. vigile que ésta no se conecte en paralelo con la alimentación para sensores de la CPU 37-200. Para aumentar la protección contra interferencias, se recomienda conectar los cables neutros (M) de las distintas fuentes de alimentación.

Precaución

Si se conecta una fuente de alimentación externa de 24 V c.c. en paralelo con la fuente de alimentación para sensores de 24 V c.c. del S7-200, podría surgir un conflicto entre ambas fuentes, ya que cada una intenta establecer su propio nivel de tensión de salida.

Este conflicto puede tener como consecuencia una reducción de la vida útil o la avería inmediata de una o ambas fuentes de alimentación y, en consecuencia, el funcionamiento imprevisible del sistema de automatización, lo que podría ocasionar la muerte o lesiones graves al personal, y/o daños al equipo.

La fuente de alimentación c.c. para sensores del S7-200 y la fuente de alimentación externa deben alimentar diferentes puntos.

ANEXO D MONTAR Y DESMONTAR EL S7-200

Montar y desmontar el S7-200

El S7-200 SQ puede montar fácilmente en un raíl DIN o en un armario eléctrico.

Requisitos previos

Antes **CQ** montar o desmontar cualquier aparato eléctrico, vigile que se haya desconectado la alimentación del mismo. Verifique también que se haya desconectado la alimentación de todos los equipos conectados.



Precaución

Si intenta montar o desmontar los módulos S7-200 y/o los equipos conectados a los mismos estando conectada la alimentación, puede producirse un choque eléctrico o fallos en bs equipos.

Si antes del montaje o desmontaje no se ha desconectado por completo la alimentación eléctrica del S7-200y de los equipos conectados a las mismas, ello podría causar la muerte o heridas graves al personal, y/o danos materiales.

Respete siempre las medidas de seguridad necesarias y verifique que la alimentación eléctrica del S7-200 y de los equipos conectados se haya desconectado antes del montaje o desmontaje.

Al sustituir o montar un sistema de automatización S7-200, vigile que se utilice siempre el módulo correcto o un equipo equivalente.

\ Precaución

f \ Si monta un módulo incorrecto, es posible que el programa contenido en el S7-200 funcione de forma impredecible.

Si un equipo S7-200 se sustituye con otro modelo o si no se monta con la orientación correcta y en el orden previsto, ello podría causar la muerte o lesiones graves al personal, y/o danos materiales.

Sustituya un equipo S7-200 con el mismo modelo y móntelo con la orientación correcta y en el orden previsto.

Dimensiones de montaje

Las CPUs S7-200 y los módulos de ampliación disponen de orificios para facilitar el montaje en armarios eléctricos. En la tabla 3-1 figuran las dimensiones de montaje.

Tabla 3-1 Dimensiones de montaje

Módulo S7-200	Ancho A	Ancho B
CPU 221 y CPU 222	90 mm	82 mm
CPU 224	120,5 mm	112,5 mm
CPU 224XP	140 mm	132 mm
CPU 226	196 mm	188 mm
Módulos de ampliación: 4 y 8 E/S DC y E/S de relé (8I, 4Q, 8Q, 4I/4Q) y salidas analógicas (2 AQ)	46 mm	38 mm
Módulos de ampliación: 16 E/S digitales (16I, 8I/8Q), E/S analógicas (4AI, 4AI/1AQ), RTD, Termopar, PROFIBUS, Ethernet, Internet, AS-Interface, 8 E/S AC (8I y 8Q), Posición y Módem	71,2 mm	63,2 mm
Módulos de ampliación: 32 E/S digitales (16I/16Q)	137,3 mm	129,3 mm

Montar una CPU o un módulo de ampliación

El montaje del S7-200 QS es muy sencillo. Proceda de la manera siguiente:

Montaje en un armario eléctrico

1. Posicione y taladre los orificios de montaje (M4 o estándar americano $n-8$) conforme a las dimensiones indicadas en la tabla 3-1.
2. Atornille la CPU o el módulo de ampliación al armario eléctrico utilizando los tornillos apropiados.
3. Si desea montar un módulo de ampliación, enchufe el cable plano en el conector del módulo (ubicado debajo de la tapa frontal).

Montaje en un raíl DIN

1. Atornille el raíl DIN al armario eléctrico dejando un espacio de 75 mm entre tornillo y tornillo.
2. Abra el gancho de retención (ubicado en el lado inferior de la CPU o del módulo) y enganche la parte posterior de la CPU o del módulo al raíl DIN.
3. Si desea montar un módulo de ampliación, enchufe el cable plano en el conector del módulo (ubicado debajo de la tapa frontal).
4. Gire la CPU o el módulo hacia el raíl DIN y cierre el gancho de retención. Vigile que la CPU o el módulo se hayan enganchado correctamente en el raíl. Para evitar détonos en la CPU o en el módulo, oprima la orejeta del orificio de montaje en vez de presionar directamente sobre el lado frontal de la CPU o del módulo.

Consejo

Si el S7-200 se monta en un raíl DIN en entornos donde se presenten vibraciones fuertes, o bien con orientación vertical, puede resultar necesario asegurarlo con topes.

Si el S7-200 se encuentra en un entorno donde se presenten vibraciones fuertes, es recomendable montarlo en un armario eléctrico, puesto que éste ofrece una mejor protección contra vibraciones.

Desmontar una CPU o un módulo de ampliación

Para desmontar una CPU o un módulo de ampliación S7-200, proceda de la manera siguiente:

1. Desconecte la alimentación del S7-200.
2. Desconecte todos los cables enchufados a la CPU o al módulo. La mayoría de las CPUs S7-200 y de los módulos de ampliación tienen conectores extraíbles que facilitan esta tarea.
3. Si hay módulos de ampliación conectados al equipo que desea desmontar, abra la tapa de acceso frontal y desconecte el cable plano de los módulos adyacentes.
4. Desatornille los tornillos de montaje o abra el gancho de retención.
5. Desmonte el módulo.

Extraer y reinsertar el bloque de terminales

La mayoría de las CPUs S7-200 y de los módulos de ampliación tienen bloques de terminales extraíbles incorporados que permiten montar y sustituir fácilmente el módulo. Consulte el anexo A para determinar si su módulo S7-200 incorpora un bloque de terminales extraíble. Para los módulos que no dispongan de un bloque de terminales extraíble se puede pedir un bloque de bornes opcional. Los números de referencia se indican en el anexo E.

Para extraer el bloque de terminales:

1. Abra la tapa del bloque de terminales para acceder al mismo.
2. Inserte un destornillador pequeño en la ranura central del bloque de terminales.
3. Extraiga el bloque de terminales haciendo palanca con el destornillador, alejándolo de la carcasa del S7-200 (v. fig. 3-2).

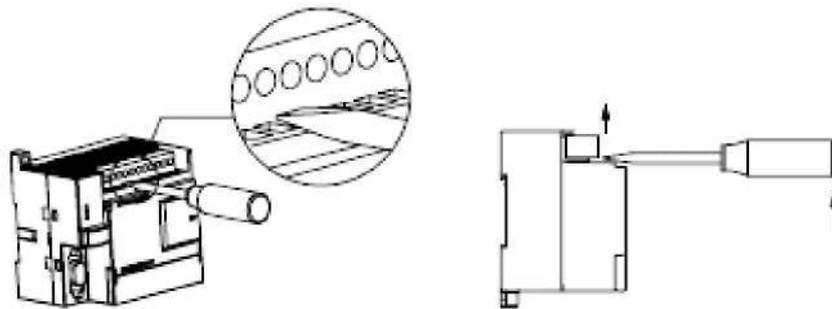


Figura 3-2 Extraer el bloque de terminales

Para reinsertar el bloque de terminales:

1. Abra la tapa del bloque de terminales.
2. Alinee el bloque de terminales con los pines de la CPU o del módulo de ampliación y alinee el borde del cableado con la base del bloque de terminales.
3. Empuje firmemente el bloque de terminales hacia abajo hasta que quede insertado correctamente. Compruebe si el bloque de terminales está bien alineado y acoplado por completo.

ANEXO E REGLAS DE PUESTA A TIERRA Y CABLEADO

Reglas de puesta a tierra y cableado

La puesta a tierra y el cableado de todos los equipos eléctricos es importante para garantizar el funcionamiento óptimo del sistema y para aumentar la protección contra interferencias en la aplicación y en el S7-200.

Requisitos previos

Antes de poner a tierra o cablear cualquier aparato eléctrico, vigile que se haya desconectado la alimentación del mismo. Verifique también que se haya desconectado la alimentación de todos los equipos conectados.

Al cablear un PLG S7-200 y los equipos conectados es necesario respetar todos los reglamentos, códigos y normas eléctricas vinculantes. Monte y utilice el equipo conforme a todas las normas nacionales y locales vigentes. Diríjase a las autoridades locales para informarse acerca de qué reglamentos, códigos o normas rigen en el lugar de montaje.



Precaución

Si intenta montar o cablear el S7-200 y/o los equipos conectados a los mismos estando conectada la alimentación, puede producirse un choque eléctrico o fallos en los equipos. Si antes del montaje o desmontaje no se ha desconectado por completo la alimentación eléctrica del S7-200 y de los equipos conectados a las mismas, ello podría causar la muerte o heridas graves al personal, y/o daños materiales.

Respete siempre las medidas de seguridad necesarias y vigile que la alimentación eléctrica del S7-200 y de los equipos conectados se haya desconectado antes del montaje o desmontaje.

Considere siempre los aspectos de seguridad al configurar la puesta a tierra y el cableado del sistema de automatización S7-200. Los aparatos electrónicos, tales como el S7-200, pueden fallar y causar un funcionamiento inesperado de los equipos conectados que se están controlando o vigilando. Foreste motivo, es recomendable que prevea medidas de seguridad independientes del S7-200 para evitar lesiones personales y/o daños al equipo.



Precaución

Un funcionamiento anormal de los equipos de control puede causar un funcionamiento inesperado del equipo controlado, lo que podría ocasionar la muerte o lesiones graves al personal, y/o daños al equipo.

Prevea dispositivos de parada de emergencia, dispositivos electromecánicos de mayor jerarquía y otras medidas redundantes de seguridad que sean independientes del S7-200.

Reglas de aislamiento

El aislamiento de la alimentación c.a. del S7-200 y de las E/S a los circuitos c.a. es de 1500 V c.a. Estos aislamientos han sido comprobados y aprobados, ofreciendo una separación segura entre el conductor c.a. y los circuitos de baja tensión.

Todos los circuitos de baja tensión conectados a un S7-200 (por ejemplo, la corriente de 24 V) deben ser alimentados por una fuente aprobada que proporcione un aislamiento seguro del conductor c.a. y de otros circuitos de alta tensión. Estas fuentes incorporan un aislamiento doble conforme a lo definido en las normas internacionales de seguridad eléctrica, teniendo salidas clasificadas como SELV, PELV, clase 2 o intensidad limitada (según la norma en cuestión).



Precaución

La utilización de fuentes de alimentación no aisladas o con aislamiento simple para abastecer los circuitos de baja tensión de un conductor c.a. 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.

Las altas tensiones inesperadas podrían ocasionar la muerte o lesiones graves al personal, y/o daños al equipo.

Utilice sólo convertidores de alta a baja tensión aprobados como fuentes de circuitos de tensión limitada seguros al tacto.

Reglas de puesta a tierra del S7-200

La mejor forma de poner a tierra la aplicación QS garantizar que todos los conductores neutros y de masa del S7-200 y de los equipos conectados se pongan a tierra en un mismo punto. Este punto se debería conectar directamente a la toma de tierra del sistema.

Para incrementar la protección contra interferencias es recomendable que todos los conductores de retorno c.c. neutros se conecten a un mismo punto de puesta a tierra. Conecte a tierra el conductor neutro (M) de la alimentación para sensores de 24 V c.c.

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

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

Reglas de cableado del S7-200

Al diseñar el cableado del sistema de automatización S7-200, incorpore un interruptor unipolar para cortar simultáneamente la alimentación de la CPJ S7-200, de todos los circuitos de entrada y de todos los circuitos de salida. Prevea dispositivos de protección contra sobreintensidad (por ejemplo, fusibles o cortacircuitos) para limitar las corrientes excesivas en el cableado de alimentación. Para mayor protección es posible instalar un fusible u otro limitador de sobreintensidad en todos los circuitos de salida.

Instale dispositivos de supresión de sobretensiones apropiados en el cableado susceptible de recibir sobretensiones causadas por rayos.

Evite colocar los conductores de señalización y los cables de comunicación en una misma canalización junto con los cables de corriente c.a. y los cables c.c. de alta tensión y de conmutación rápida. El cableado deberá efectuarse por pares; con el cable de neutro o común combinado con el cable de fase o de señal.

Utilice el cable más corto posible y vigile que tenga una sección suficiente para conducir la corriente necesaria. El cable debe aceptar cables con sección de 2 mm² a 0,30 mm^E (14 AWG a 22 AWG). Utilice cables apantallados para obtener el mayor nivel de inmunidad a interferencias. Por lo general, se obtienen los mejores resultados si la pantalla se pone a tierra en el S7-200.

Al cablear circuitos de entrada alimentados por una fuente externa, prevea dispositivos de protección contra sobreintensidad en esos circuitos. La protección externa no se requiere en los circuitos alimentados por la alimentación para sensores de 24 V c.c. del S7-200, puesto que la alimentación para sensores ya está protegida contra sobreintensidad.

La mayoría de los módulos S7-200 disponen de bloques de terminales extraíbles para el cableado de usuario. (Consulte el anexo A para determinar si su módulo S7-200 incorpora un bloque de terminales extraíble.) Para evitar conexiones "flojas, vigile que el bloque de terminales esté encajado correctamente y que el cable esté instalado de forma segura. No apriete excesivamente los tornillos para evitar que se deteriore el bloque de terminales. El par máximo de apriete de los tornillos del bloque de terminales es de 0.56 N-m.

El S7-200 incluye aislamientos en ciertos puntos para prevenir la circulación de corrientes indeseadas en la instalación. Tenga en cuenta estos elementos de aislamiento al planificar el cableado del sistema de automatización. Consulte el anexo A para obtener más información acerca de la ubicación de los puntos de aislamiento y la capacidad que ofrecen. Los aislamientos con valores nominales inferiores a 1.500 V c.a. no deberán tomarse para definir barreras de seguridad.

~* < Consejo

¡J En una red de comunicación, la longitud máxima del cable de comunicación debería ser de 50 metros sin utilizar un repetidor. El puerto de comunicación del S7-200 no está aislado. Para más información al respecto, consulte el capítulo 7.

Reglas relativas a las cargas inductivas

Las cargas inductivas deberán equiparse con circuitos de supresión destinados a limitar el incremento de tensión producido al desactivarse las salidas. Los circuitos de supresión protegen las salidas contra fallos prematuros debidos a elevadas corrientes de conmutación inductivas. Además, estos circuitos limitan las interferencias generadas al conmutar cargas inductivas.

- ~N Consejo
- *J La eficacia de un determinado circuito de supresión depende de la aplicación. Por tanto, deberá verificarse para cada caso particular. Vigile siempre que los componentes utilizados en el circuito de supresión se adecúen para la aplicación en cuestión.

Salidas c.c. y relés que controlan cargas c.c.

Las salidas c.c. tienen una protección interna adecuada para la mayoría de las aplicaciones. Puesto que los relés se pueden utilizar para cargas tanto c.c. como c.a., no proporcionan una protección interna.

La figura 3-3 muestra un circuito de supresión de ejemplo para una carga de corriente continua. En la mayoría de las aplicaciones es suficiente prever adicionalmente un diodo (A) en la carga inductiva. No obstante, si la aplicación requiere tiempos de desconexión más rápidos, se recomienda utilizar un diodo Zener (6). Vigile que el diodo Zener tenga suficiente capacidad para la cantidad de corriente en el circuito de salida.

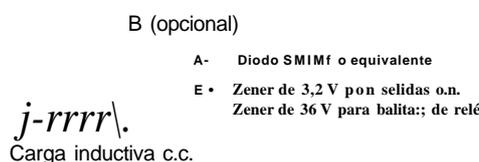


Figura 3-3 Circuito de supresión para una carga de corriente continua

Salidas de corriente alterna y relés que controlan cargas de corriente alterna
Las salidas de corriente alterna tienen una protección interna adecuada para la mayoría de las aplicaciones. Puesto que los relés se pueden utilizar para cargas tanto c.c. como c.a., no proporcionan una protección interna.

La figura 3-4 muestra un circuito de supresión de ejemplo para una carga de corriente alterna. Si utiliza un relé o una salida de corriente alterna para conmutar cargas de 115 W/230 V c.a., disponga redes de resistores/capacitores a lo largo de la carga de corriente alterna como se ilustra en la figura. También puede 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 nominal de fase.

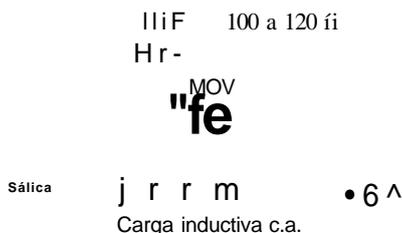


Figura 3-4 Circuito de supresión para una carga de corriente alterna

Nota

Si utiliza módulos de ampliación para conmutar cargas inductivas de 230 V c.a., el circuito externo de supresión de interferencias se deberá disponer a lo largo de la carga de corriente alterna como muestra la figura 3-4.

Reglas relativas a las cargas de lámparas

Las cargas de lámparas pueden averiar los contactos de relé, debido a la elevada intensidad momentánea al arrancar. Esta intensidad momentánea es 10 a 15 veces superior a la intensidad en modo estacionario de una lámpara de tungsteno. Se recomienda intercalar un relé sustituible o un limitador de sobretensión para las cargas de lámpara que se deban conmutar con frecuencia durante la vida útil de la aplicación.

ANEXO F EJECUTAR LA LÓGICA DE CONTROL EN EL S7-200

Ejecutar la lógica de control en el S7-200

El S7-200 ejecuta cíclicamente la lógica de control del programa, leyendo y escribiendo datos.

Relacionar el programa con las entradas y salidas físicas

El funcionamiento básico del S7-200 es muy sencillo:

- El S7-200 lee el estado de las entradas.
- El programa almacenado en el S7-200 utiliza las entradas para evaluar la lógica. Durante la ejecución del programa, el S7-200 actualiza los datos.
- El S7-200 escribe los datos en las salidas.

La figura 4-1 muestra cómo se procesa un esquema de circuitos simple en el S7-200. En este ejemplo, el estado del interruptor para arrancar el motor se combina con los estados de otras entradas. El resultado obtenido establece entonces el estado de la salida que corresponde al actuador que arranca el motor.

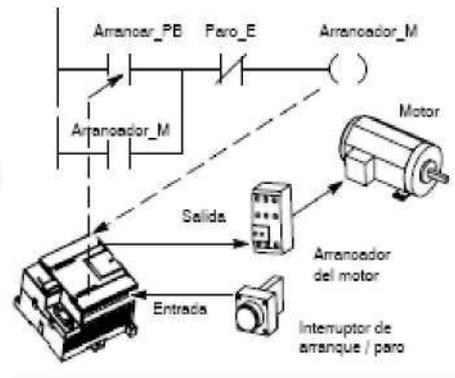


Figura 4-1 Controlar las entradas y salidas

Ejecutar las tareas en un ciclo

El S7-200 ejecuta una serie de tareas de forma repetitiva. Esta ejecución se denomina ciclo. Como muestra la figura 4-2, el S7-200 ejecuta la mayoría de las tareas siguientes (o todas ellas) durante un ciclo:

- Leer las entradas: el S7-200 copia el estado de las entradas físicas en la imagen del proceso de las entradas.
- Ejecutar la lógica de control en el programa: el S7-200 ejecuta las operaciones del programa y guarda los valores en las diversas áreas de memoria.
- Procesar las peticiones de comunicación: el S7-200 ejecuta las tareas necesarias para la comunicación.
- Efectuar el autodiagnóstico de la CPU: el S7-200 verifica si el firmware, la memoria del programa y los módulos de ampliación están trabajando correctamente.
- Escribir en las salidas: los valores almacenados en la imagen del proceso de las salidas se escriben en las salidas físicas.

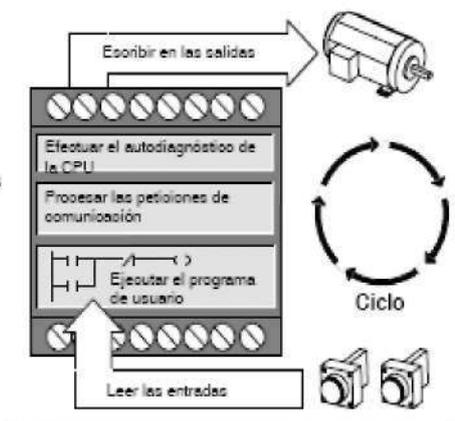


Figura 4-2 Ciclo del S7-200

La ejecución del programa de usuario depende de si el S7-200 está en modo STOP o RUN. El programa se ejecutará si el S7-200 está en modo RUN. En cambio, no se ejecutará en modo STOP.

Leer las entradas

Entradas digitales: Al principio de cada ciclo se leen los valores actuales de las entradas digitales y se escriben luego en la imagen del proceso de las entradas.

Entradas analógicas: El S7-200 no actualiza las entradas analógicas de los módulos de ampliación como parte del ciclo normal, a menos que se haya habilitado la filtración de las mismas. Existe un "Filtro analógico que permite disponer de una señal más estable. Este filtro se puede habilitar para cada una de las entradas analógicas.

Si se habilita la filtración de una entrada analógica, el S7-200 actualizará esa entrada una vez por ciclo, efectuará la "filtración y almacenará internamente el valor filtrado. El valor filtrado se suministrará cada vez que el programa accede a la entrada analógica.

Si no se habilita la filtración, el S7-200 leerá de los módulos de ampliación el valor de la entrada analógica cada vez que el programa de usuario acceda a esa entrada.

Las entradas analógicas AIW0 y AIW2 incorporadas en la CPU 224XP se actualizan en cada ciclo con el resultado más reciente del convertidor analógico/digital. Estos convertidores de tipo promedio (sigma-delta) y, por lo general, no es necesario filtrarlas en el software.

Consejo

La filtración de las entradas analógicas permite disponer de un valor analógico más estable. Utilice el filtro de entradas analógicas en aplicaciones donde la señal de entrada cambia lentamente. Si la señal es rápida, no es recomendable habilitar el filtro analógico.

No utilice el filtro analógico en módulos que transfieran informaciones digitales o indicaciones de alarma en las palabras analógicas. Desactive siempre el filtro analógico si utiliza módulos RTD, termopar o AS-Interface Máster.

Ejecutar el programa

Durante esta fase del ciclo, el S7-200 ejecuta el programa desde la primera operación hasta la última (= Finalizar programa). El control directo de las entradas y salidas permite acceder directamente a éstas mientras se ejecuta el programa o una rutina de interrupción.

Si se utilizan interrupciones, las rutinas asociadas a los eventos de interrupción se almacenarán como parte del programa. Las rutinas de interrupción no se ejecutan como parte del ciclo, sino sólo cuando ocurre el evento (en cualquier punto del ciclo).

Procesar las peticiones de comunicación.

Durante esta fase del ciclo, el S7-200 procesa los mensajes que haya recibido por el puerto de comunicación o de los módulos de ampliación inteligentes.

Efectuar el diagnóstico

Durante el autodiagnóstico, el S7-200 comprueba si la CPU está "funcionando correctamente, así como el estado de los módulos de ampliación.

Escribir las salidas digitales

Al final de cada ciclo, el S7-200 escribe los valores de la imagen del proceso de las salidas en las salidas digitales. (Las salidas analógicas se actualizan de inmediato, independientemente del ciclo.)

ANEXO G

SELECCIONAR EL MODO DE OPERACIÓN DEL S7-200

Seleccionar el modo de operación del S7-200

El S7-200 tiene dos modos de operación, a saber: STOP y RUN. Los diodos luminosos (LEDs) ubicados en la parte frontal de la CPU indican el modo de operación actual. En modo STOP, el S7-200 no ejecuta el programa. Entonces es posible cargar un programa o configurar la CPU. En modo RUN, el S7-200 ejecuta el programa.

- j El S7-200 incorpora un selector de modos que permite cambiar el modo de operación. El modo de operación se puede cambiar manualmente accionando el selector (ubicado debajo de la lámpara de acceso frontal del S7-200). Si el selector se pone en STOP, se detendrá la ejecución del programa. Si se pone en RUN, se iniciará la ejecución del programa. Si se pone en TERM, no cambiará el modo de operación.

Si se interrumpe la alimentación estando el selector en posición STOP o TERM, el S7-200 pasará a modo STOP cuando se le aplique tensión. Si se interrumpe la alimentación estando el selector en posición RUN, el S7-200 pasará a modo RUN cuando se le aplique tensión.

- j STEP 7-Micro/WIN permite cambiar el modo de operación del S7-200 conectado. Para que el modo de operación se pueda cambiar mediante el software, el selector del S7-200 deberá estar en posición TERM o RUN. Elija para ello el comando de menú CPU > STOP o CPU > RUN, respectivamente (o haga clic en los botones correspondientes de la barra de herramientas).
- j Para cambiar el S7-200 a modo STOP es posible introducir la correspondiente operación (STOP) en el programa. Ello permite detener la ejecución del programa en función de la lógica. Para más información sobre la operación STOP, consulte el capítulo 6.

ANEXO H

SIEMENS SIMATIC Manual del sistema de automatización S7-200

Juego de operaciones
del S7-200 de
Siemens

Convenciones utilizadas para describir las operaciones

La figura 6-1 muestra una descripción típica de una operación y señala las diferentes áreas utilizadas para describir la operación y su funcionamiento. La ilustración de la operación muestra el formato en KOP (LD), FUP (FBD) y AWL. Tenga en cuenta que la terminología IEC difiere considerablemente de la terminología SIMATIC (tanto en cuanto a los nombres de las operaciones como al de los lenguajes de programación). Por ejemplo, en SIMATIC existe la operación Incrementar contador (CTU), en tanto que en IEC se hace referencia al bloque funcional Contador ascendente (CTU). Además, en SIMATIC se habla del lenguaje KOP (Esquema de contactos) que equivale en IEC al lenguaje LD (Diagrama de escalera). Igualmente, el lenguaje FUP (Diagrama de funciones) de SIMATIC se denomina FBD (Diagrama de bloques funcionales) en IEC. En la tabla de operandos figuran los operandos de la operación, así como los tipos de datos válidos, las áreas de memoria y los tamaños de cada uno de los operandos. Los operandos y los tipos de datos EN/ENO no figuran en la tabla de operandos de la operación, puesto que son idénticos para todas las operaciones KOP y FUP.

- En KOP:** EN y ENO son conductores de corriente y su tipo de datos es BOOL.
- En FUP:** EN y ENO son I, Q, V, M, SM, S, T, C, L o conductores de corriente y su tipo de datos es BOOL.

Descripción de la operación y de los operandos

Operación AWL

Operaciones KOP (LD) y FUP (FBD)

Transmit and Receive Instructions

The Transmit instruction (XMT) is used in Freeport mode to transmit data by means of the communication port(s).

The Receive instruction (RCV) initiates or terminates the Receive Message service. You must specify a start and an end condition for the Receive box to operate. Messages received through the specified port (PORT) are stored in the data buffer (TBL). The first entry in the data buffer specifies the number of bytes received.

Error conditions that set ENO = 0

- SM0.6 or SM0.6.6 error bit set (RCV parameter error)
- R0M6 (address 24800)
- R0R6 (start address XMT/RCV is port 0)
- R0R6 (start address XMT/RCV is port 1)
- S7-200 CPU not in Freeport mode

Table 6-11 Valid Operands for the Transmit and Receive Instructions

Input/Output	Data Type	Operands
TBL	BYTE	V0, S0, Q0, M0, I0, SM0, *V0, *AG, *LD
PORT	BYTE	Constant for CPU 311, CPU 312, CPU 314: 0 for CPU 315 and CPU 314C-2: 0 or 1

Using Freeport Mode to Control the Serial Communication Port

You can select the Freeport mode to control the serial communication port of the S7-200 by means of the operating mode switch. When you select Freeport mode, your program controls the operation of the communication port. The use of the receive interrupts, the transmit interrupts, the Transmit instructions, and the Receive instruction. The communication protocol is entirely controlled by the ladder program while in Freeport mode. SMB100 (for port 0) and SMB130 (for port 1 if your S7-200 has two ports) are used to set rates and parity.

Tip
Entering Freeport mode is allowed if serial normal communication is reestablished (for example, programming the S7-200 communication while in Freeport mode).

Freeport communication is possible only when the S7-200 is in the RUN mode. Enable the Freeport mode by setting a value of 01 in the protocol select field of SMB100 (Port 0) or SMB130 (Port 1). While in Freeport mode, communication with the programming device is not possible.

Tip
Entering Freeport mode can be controlled using special memory bit SM0.7, which reflects the current position of the operating mode switch. When SM0.7 is equal to 0, the switch is in TEFM position; when SM0.7 = 1, the operating mode switch is in RUN position. If you enable Freeport mode only when the switch is in RUN position, you can use the programming device to monitor or control the S7-200 operation by changing the switch to any other position.

Diagramas: Se muestran diagramas de contactos (KOP) y bloques funcionales (FUP) para las instrucciones XMT y RCV, así como un diagrama de la estructura de la tabla de datos (TBL).

Condiciones de error que afectan a ENO y a las marcas especiales (SM)

Operandos de la operación

Tipos de datos válidos

Áreas de memoria válidas y tamaños de los operandos

Áreas de memoria y funciones del S7-200

Tabla 6-1 Áreas de memoria y funciones de las CPUs S7-200

Descripción	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Tamaño del programa de usuario	4096 bytes	4096 bytes	8192 bytes	12288 bytes	16384 bytes
Tamaño de los modos de imagen del	2048 bytes	2048 bytes	8192 bytes	10240 bytes	10240 bytes
Imagen del modo de las entradas	I0.0 a I15.7				
Imagen del modo de las salidas analógicas (solo de salida)	Q0.0 a Q15.7				
Memoria de	AIW0 a AIW30	AIW0 a AIW30	AIW0 a AIW62	AIW0 a AIW62	AIW0 a AIW62
Memoria de	AQW0 a AQW30	AQW0 a AQW30	AQW0 a AQW62	AQW0 a AQW62	AQW0 a AQW62
Memoria de	VB0 a VB0				
Memoria local (L) ¹	LB0 a LB63				
Área de marcas (M)	M0.0 a M0.0				
Marcas especiales (SM)	SM0.0 a SM179.7	SM0.0 a SM299.7	SM0.0 a SM549.7	SM0.0 a SM549.7	SM0.0 a SM549.7
Temporizadores	256 (T0 a T255)				
Retardo a la conexión con memoria	T0, T64				
1 ms	T1 a T4 y T65 a T68				
10 ms	T5 a T31 y T69 a T95				
100 ms	T32, T96				
Retardo a la conexión/desconexión	T33 a T36 y T97 a T100				
1 ms	T100	T100	T100	T100	T100
Contadores	C0 a C255				
Contadores rápidos	HC0 a HC5				
Relés de control secuencial (S)	S0.0 a S0.0				
Acumuladores	AC0 a AC3	AC0 a AC3	AC0 a AC3	AC0 a AC3	S31.7 AC0 a AC3
Salto a metas	0 a 255				
Llamadas a rutinas de detectar flanco	0 a 63	0 a 63	0 a 63	0 a 63	0 a 127
Lazos PID	0 a 127				
Puertos	256	256	256	256	256
	0a7	0a7	0a7	0a7	0a7
	Puerto 0	Puerto 0	Puerto 0	Puerto 0,	Puerto 0,

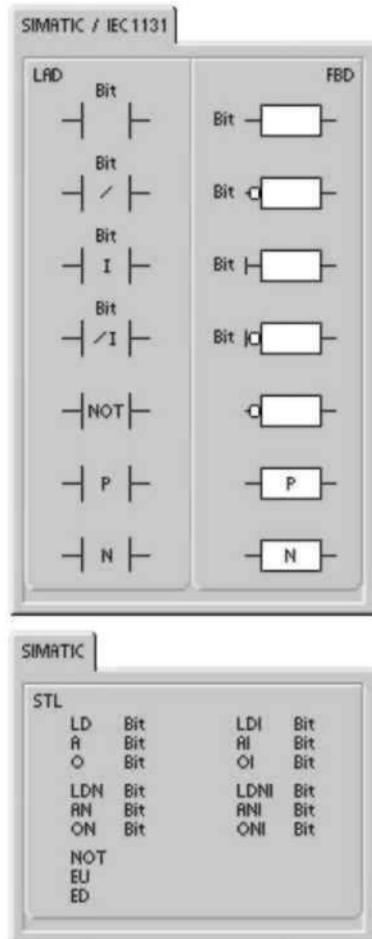
¹ STEP 7-Micro/WIN (versión 3.0 o posterior) reserva LB60 a LB63.

Operaciones lógicas con bits

Contactos

Contactos estándar

Las operaciones Contacto normalmente abierto (LD, A y O) y Contacto



normalmente cerrado (LDN, AN y ON) leen el valor direccionado de la memoria (o bien de la imagen del proceso, si el tipo de datos es I o Q).

El Contacto normalmente abierto se cierra (ON) si el bit es igual a 1, en tanto que el Contacto normalmente cerrado se cierra (ON) si el bit es igual a 0. En FUP, la cantidad de entradas de los cuadros AND y OR se puede incrementar a 32 como máximo. En AWL, el Contacto normalmente abierto carga, o bien combina con Y u O el valor binario del bit de dirección en el nivel superior de la pila. El Contacto normalmente cerrado carga, o bien combina con Y u O el valor negado del bit de dirección en el nivel superior de la pila.

Contactos directos

Los contactos directos no dependen del ciclo del S7-200 para actualizarse, sino que se actualizan inmediatamente. Las operaciones del Contacto abierto directo (LDI, AI y OI) y del Contacto cerrado directo (LDNI, ANI y ONI) leen el valor de la entrada física cuando se ejecuta la operación, pero la imagen del proceso no se actualiza.

El Contacto abierto directo se cierra (ON) si la entrada física (bit) es 1, en tanto que el Contacto cerrado directo se cierra (ON) si la entrada física (bit) es 0. El Contacto abierto directo carga, o bien combina con Y u O directamente el valor de la entrada física en el nivel superior de la pila. El Contacto cerrado directo carga, o bien combina con Y u O directamente el valor binario negado de la entrada física en el nivel superior de la pila.

NOT

La operación NOT cambia el estado de la entrada de circulación de corriente (es decir, modifica el valor del nivel superior de la pila de “0” a “1”, o bien de “1” a “0”).

Detectar flanco positivo y negativo

El contacto Detectar flanco positivo (EU) permite que la corriente circule durante un ciclo cada vez que se produce un cambio de “0” a “1” (de “off” a “on”). El contacto Detectar flanco negativo (ED) permite que la corriente circule durante un ciclo cada vez que se produce un cambio de “1” a “0” (de “on” a “off”). Cuando se detecta un cambio de señal de “0” a “1” en el primer valor de la pila, éste se pone a 1. En caso contrario, se pone a 0. Cuando se detecta un cambio de señal de “1” a “0” en el primer valor de la pila, éste se pone a 1. En caso contrario, se pone a 0.

Para poder editar el programa durante el tiempo de ejecución (es decir, en modo RUN) es preciso introducir un parámetro para las operaciones Detectar flanco positivo y Detectar flanco negativo. Para más información sobre la edición de programas en modo RUN, consulte el capítulo 5.

Tabla 6-3 Operandos válidos para las operaciones lógicas con bits de entrada

Entradas/salida	Tipos de	Operandos
Bit	BOO	I, Q, V, M, SM, S, T, C, L, circulación de
Bit (directo)	L	I

El S7-200 utiliza una pila lógica para resolver la lógica de control (v. fig. 6-2). En estos ejemplos, los valores iniciales de la pila se denominan “iv0” a “iv7”. Los nuevos valores se representan mediante “nv”, en tanto que “S0” es el valor calculado que se almacena en la pila lógica.



1 S0 identifica el valor calculado que se guarda en la pila lógica.

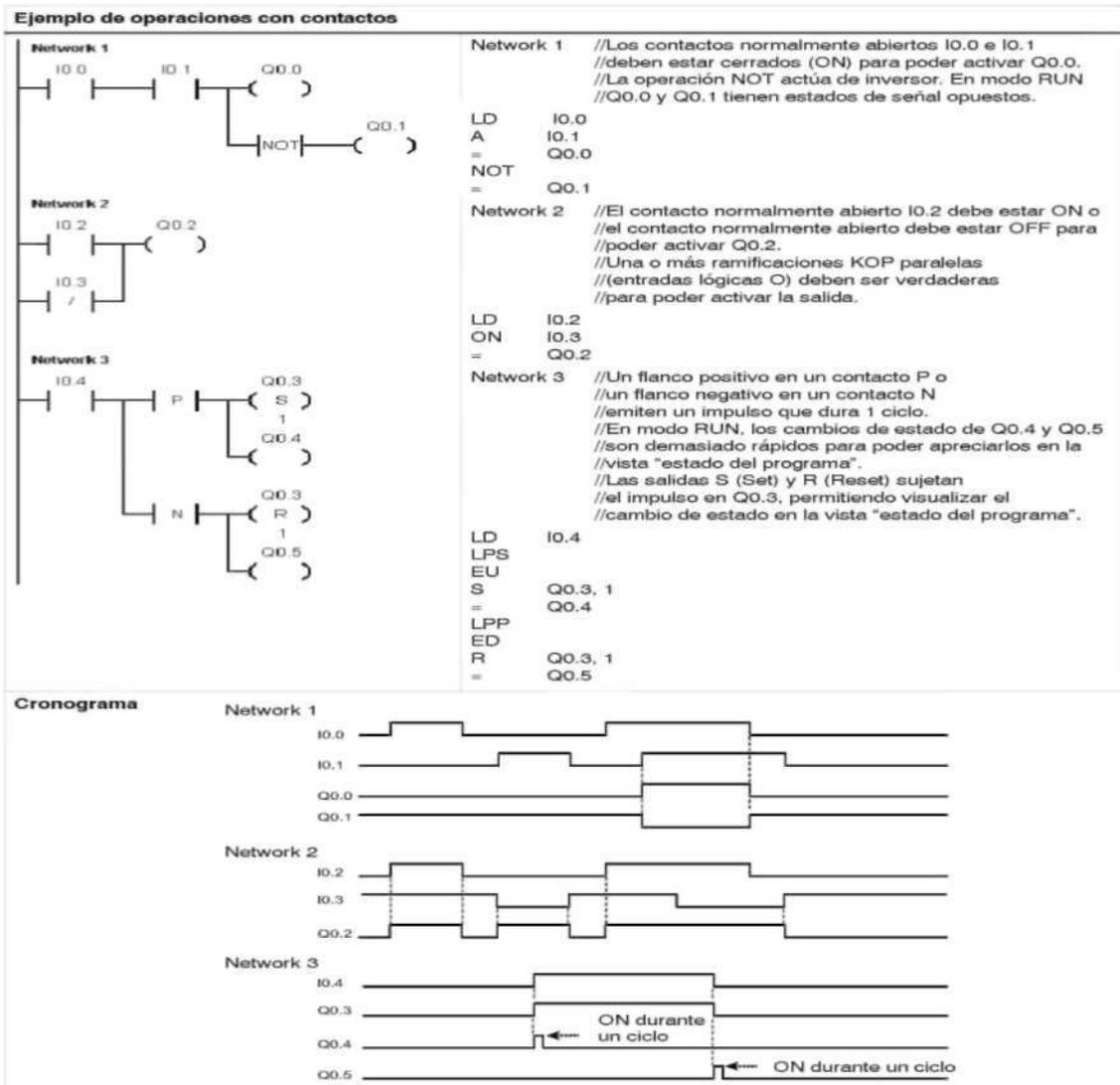
2 Tras ejecutarse una operación de carga, se pierde el valor iv8.

Figura 6-2 Funcionamiento de las operaciones con contactos

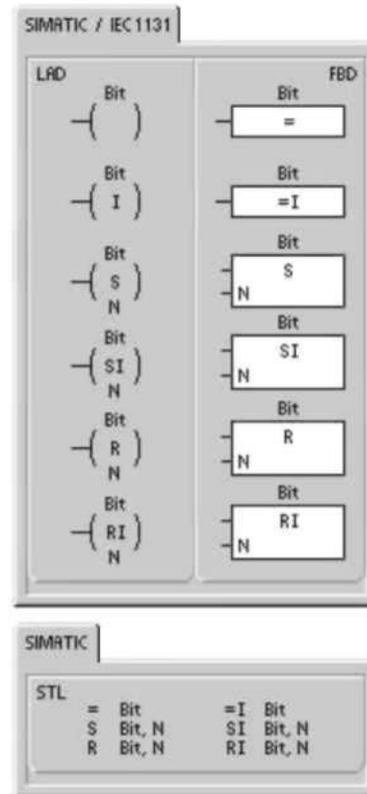
Consejo



Puesto que las operaciones Detectar flanco positivo y Detectar flanco negativo exigen una transición de "off" a "on" o de "on" a "off", respectivamente, no es posible detectar un flanco positivo o negativo en el primer ciclo. En el primer ciclo, el S7-200 ajusta el estado del bit indicado por estas operaciones. En los ciclos siguientes, las operaciones pueden detectar flancos en el bit indicado.



Bobinas



Asignar

La operación Asignar (=) escribe el nuevo valor del bit de salida en la imagen del proceso. Cuando se ejecuta la operación Asignar, el S7-200 activa o desactiva el bit de salida en la imagen del proceso. En KOP y FUP, el bit indicado se ajusta de forma equivalente a la circulación de la corriente. En AWL, el primer valor de la pila se copia en el bit indicado.

Asignar directamente

La operación Asignar directamente (=I) escribe el nuevo valor tanto en la salida física como en la correspondiente dirección de la imagen del proceso.

Cuando se ejecuta la operación Asignar directamente, la salida física (bit) se ajusta directamente de forma equivalente a la circulación de la corriente. En AWL, la operación copia el primer valor de la pila directamente en la salida física indicada (bit). La "I" indica que la operación se ejecuta directamente. El nuevo valor se escribe entonces tanto en la salida física como en la correspondiente dirección de la imagen del proceso. En cambio, en las operaciones no directas, el nuevo valor se escribe sólo en la imagen del proceso.

Poner a 1 y Poner a 0

Las operaciones Poner a 1 (S) y Poner a 0 (R) activan (ponen a 1) o desactivan (ponen a 0) el número indicado de E/S (N) a partir de la dirección indicada (bit). Es posible activar o desactivar un número de entradas y salidas (E/S) comprendido entre 1 y 255. Si la operación Poner a 0 indica un bit de temporización (T) o un bit de contaje (C), se desactivará el bit de temporización o de contaje y se borrará el valor actual del temporizador o del contador, respectivamente.

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)

Poner a 1 directamente y Poner a 0 directamente

Las operaciones Poner a 1 directamente (SI) y Poner a 0 directamente (RI) activan (ponen a 1) o desactivan (ponen a 0) directamente el número indicado de E/S (N) a partir de la dirección indicada (bit). Es posible activar o desactivar directamente un número de entradas y salidas (E/S) comprendido entre 1 y 128.

La “I” indica que la operación se ejecuta directamente. El nuevo valor se escribe tanto en la salida física como en la correspondiente dirección de la imagen del proceso. En cambio, en las operaciones no directas, el nuevo valor se escribe sólo en la imagen del proceso.

Condiciones de error que ponen ENO a 0:

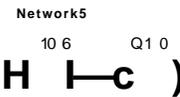
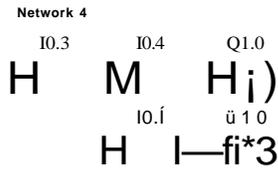
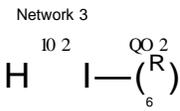
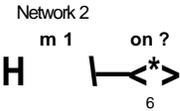
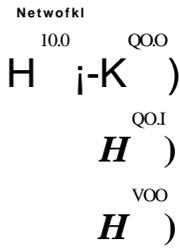
- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)

Tabla 6-4 Operandos válidos para las operaciones lógicas con bits de salida

Entradas/salida	Tipos de	Operandos
Bit	BOOL	I, Q, V, M, SM, S, T, C, L
Bit (directo)	BOOL	Q
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD,

*AC, constante

Ejemplo de operaciones con bobinas



Network 1 //Las operaciones de salida asignan valores //binarios a las E/S externas // (I, Q) y a la memoria interna (M, SM, T, C, V, S, L).

LD I0.0
Q0.0
Q0.1
V0.0

Network 2 //Poner a 1 un grupo de 6 bits consecutivos. Indicar una //dirección Inicial de bit y cuántos bits se deben activar, //El indicador de estado de Set (Poner a 1) está ON //cuando el valor (del primer bit (Q0.2) es 1.

LD I0.1
S Q0.2, 6

Network 3 //Poner a "0" un grupo de 6 bits consecutivos. //Indicar una dirección inicial de bit y cuántos bits se deben //desactivar. //El indicador de estado de Reset (Poner a 0) está ON //cuando el valor del primer bit (Q0.2) es 0.

LD I0.2
R Q0.2, 6

Network 4 //Activar y desactivar 8 bits de salida (Q1.0 a Q1.7) en grupo.

LD I0.3
LPS
A I0.4
S Q1.0, 8
LPP
A I0.5
R Q1.0, 8

Network 5 //Las operaciones Poner a 1 y Poner a 0 actúan de //cerrojo. Para aislar los bits de las operaciones Poner a 1 y //Poner a 0, vigile que no sean sobrescritos por ninguna //operación de asignar. //En este ejemplo, el segmento (network) 4 activa y //desactiva 8 bits (Q1.0 a Q1.7) en grupo. En modo RUN, //el segmento 5 puede sobrescribir el valor de Q1.0 controlar los //indicadores de estado de las operaciones //Poner a 1 y Poner a 0 en el segmento 4.

LD I0.6
Q1.0

Cronograma

Network 1
I0.0 _____
Q0.0, Q0.1.

Networks 2 and 3
I0.1 (Set) _____

ni.- r<ff.f.:



Poner a 0 sobrescribe Poner a 1 porque el ciclo del programa ejecuta (a operación Poner a 0 del segmento 3 tras la operación Poner a 1 del segmento 2.

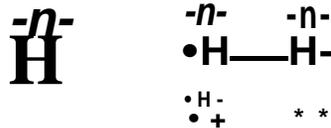
Networks 4 and 5
I0.3 _____

I0.4 (Set)



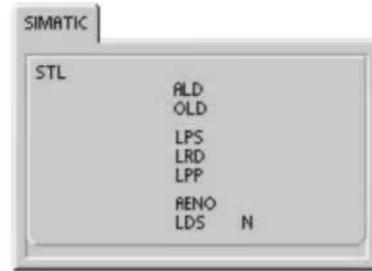
I0.5 (Resal)

Q1.0



La operación Asignar bit (=) del segmento 5 sobrescribe el primer bit (Q1.0) de Poner a 1 /Poner a 0 del segmento 4 porque el ciclo del programa ejecuta el segmento 5 de último.

Operaciones lógicas de pilas



Combinar primer y segundo valor mediante Y

La operación Combinar primer y segundo valor mediante Y (ALD) combina los valores del primer y segundo nivel de la pila mediante una operación lógica Y. El resultado se carga en el nivel superior de la pila. Una vez ejecutada la operación ALD, la profundidad de la pila tiene un nivel menos.

Combinar primer y segundo valor mediante O

La operación Combinar primer y segundo valor mediante O (OLD) combina los valores del primer y segundo nivel de la pila mediante una operación lógica O. El resultado se carga en el nivel superior de la pila. Una vez ejecutada la operación OLD, la profundidad de la pila tiene un nivel menos.

Duplicar primer valor

La operación Duplicar primer valor (LPS) duplica el primer valor de la pila y lo desplaza dentro de la misma. El último valor de la pila se expulsa y se pierde.

Copiar segundo valor

La operación Copiar segundo valor (LRD) copia el segundo valor de la pila en el nivel superior de la misma. En la pila no se carga ni se expulsa ningún valor. No obstante, el valor que se encontraba en el nivel superior se sobrescribe con el nuevo valor.

Sacar primer valor

La operación Sacar primer valor (LPP) desplaza el primer valor fuera de la pila. El segundo valor se convierte entonces en el primer nivel de la pila.

Y-ENO

La operación Y-ENO (AENO) combina mediante Y el bit ENO con el primer nivel de la pila para generar el mismo efecto que el bit ENO de un cuadro en KOP o FUP. El resultado de la operación de combinación mediante Y es el nuevo valor en el nivel superior de la pila.

ENO es una salida booleana para los cuadros KOP y FUP. Si la corriente fluye en un cuadro por la entrada EN y el cuadro se ejecuta sin error, la salida ENO conduce la corriente al siguiente elemento. ENO se puede utilizar como bit de habilitación para indicar que una operación se ha ejecutado correctamente. El bit ENO se utiliza en el primer nivel de la pila para influir en la circulación de la corriente cuando se ejecutan las operaciones posteriores. Las operaciones AWL carecen de una entrada de habilitación (EN). El primer nivel de la pila debe ser un 1 lógico para poder ejecutar las operaciones condicionadas. En AWL tampoco existe la salida de habilitación (ENO), pero las instrucciones AWL correspondientes a las operaciones KOP y FUP con salidas ENO activan un bit ENO especial. A este bit se puede acceder con la operación AENO.

Cargar pila

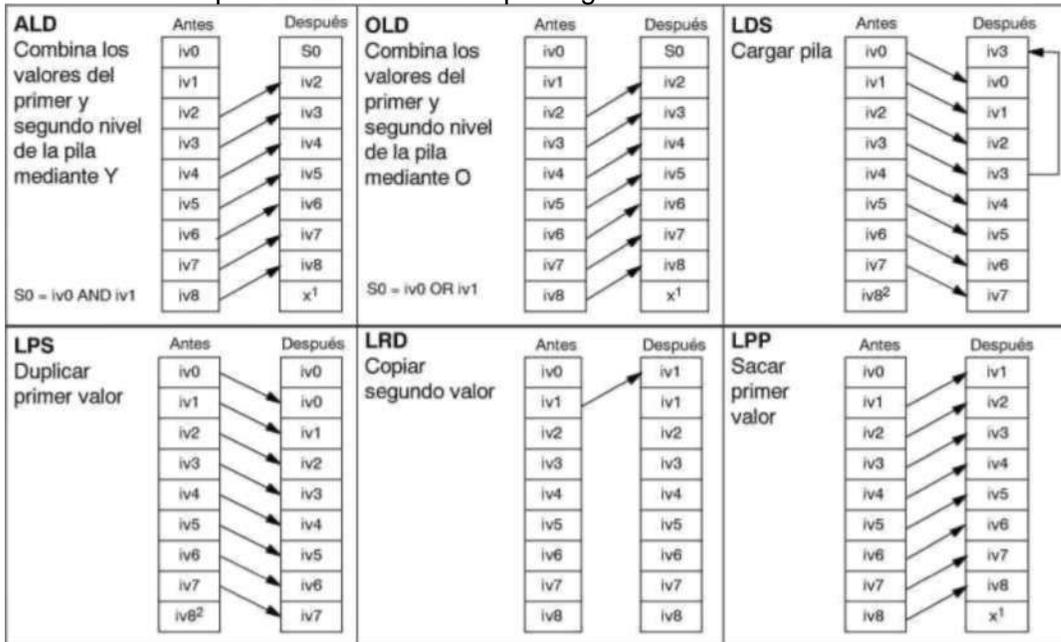
La operación Cargar pila (LDS) duplica el bit (N) la pila y lo deposita en el nivel

superior de la misma. El último valor de la pila se expulsa y se pierde.

Tabla 6-5 Operandos válidos para la operación Cargar pila

Entradas/salida	Tipos de	Operandos
N	BYTE	Constante (0 a 8)

El S7-200 utiliza una pila lógica para resolver la lógica de control (v. fig. 6-3). En estos ejemplos, los valores iniciales de la pila se denominan "iv0" a "iv7". Los nuevos valores se representan mediante "nv", en tanto que "S0" es el valor calculado que se almacena en la pila lógica.

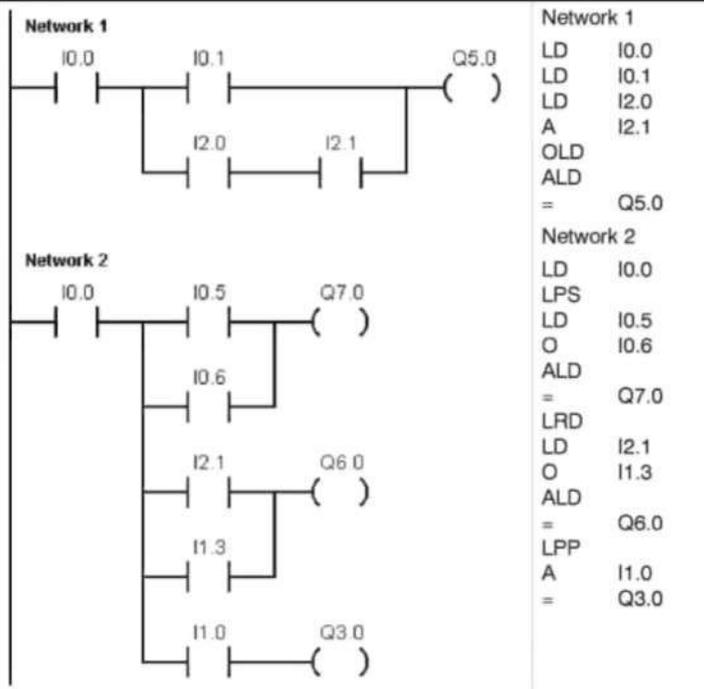


¹ La "x" significa que se desconoce el valor (puede ser 0 ó 1).

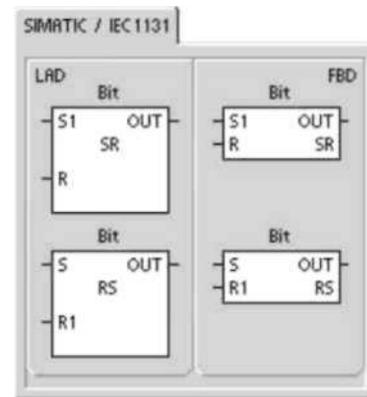
² Tras haberse ejecutado una operación Duplicar primer valor o Cargar pila, se pierde el valor iv8.

Figura 6-3 Funcionamiento de las operaciones lógicas de pilas

Ejemplo de operaciones lógicas de pilas



Posicionar y rearmar dominante biestable



La operación Posicionar dominante biestable es un flip-flop en el que domina la señal “posicionar”. Si tanto la señal “posicionar” (S1) como la señal “rearmar” (R) son verdaderas, la salida (OUT) será verdadera.

La operación Rearmar dominante biestable es un flip-flop en el que domina la señal “rearmar”. Si tanto la señal “posicionar” (S) como la señal “rearmar” (R1) son verdaderas, la salida (OUT) será falsa.

El parámetro “bit” indica el parámetro booleano que está activado (“posicionado”) o desactivado (“rearmado”). La salida opcional refleja el estado de señal del parámetro

“bit”.

La tabla 6-7 muestra las tablas de verdad del programa de ejemplo.

Tabla 6-6 Operandos válidos para las operaciones Posicionar dominante biestable y Rearmar dominante biestable

Entradas/salida	Tipos de	Operandos
S1, R	BOOL	I, Q, V, M, SM, S, T, C, circulación
S, R1, OUT	BOOL	I, Q, V, M, SM, S, T, C, L, circulación
Bit	BOOL	I, Q, V, M, S

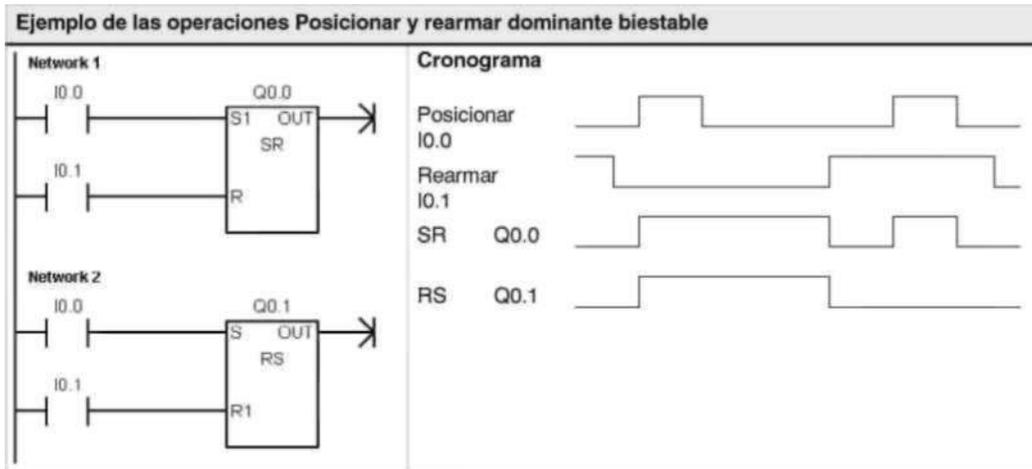
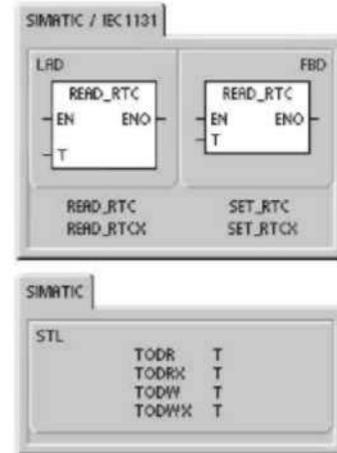


Tabla 6-7 Tabla de verdad de las operaciones Posicionar dominante biestable y Rearmar dominante biestable

Operación	S1	R	OUT (bit)
Posicionar dominante biestable (SR)	0	0	Estado anterior
	0	1	0
	1	0	1
	1	1	1
Operación	S	R1	OUT (bit)
Rearmar dominante biestable (RS)	0	0	Estado anterior
	0	1	0
	1	0	1
	1	1	0

Operaciones de reloj

Leer reloj de tiempo real y Ajustar reloj de tiempo real



La operación Leer reloj de tiempo real (TODR) lee la hora y fecha actuales del reloj del hardware y carga ambas en un búfer de tiempo de 8 bytes que comienza en la dirección T. La operación Ajustar reloj de tiempo real (TODW) escribe la hora y fecha actuales en el reloj del hardware, comenzando en la dirección del búfer de tiempo de 8 bytes indicada por T.

Todos los valores de la fecha y la hora se deben codificar en BCD (por ejemplo, 16#97 para el año 1997). La figura 6-4 muestra el formato del búfer de tiempo de 8 bytes (T).

El reloj de tiempo real se inicializa con la siguiente fecha y hora tras un corte de alimentación prolongado o cuando se produzca una pérdida de memoria:

Fecha: 01-Ene-90

Hora: 00:00:00

Día de la semana: Domingo

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0007 (error de datos TOD) (*sólo en el reloj de tiempo real*)
- 000C (falta reloj)

Tabla 6-8 Operandos válidos para las operaciones de reloj

Entradas/salida	Tipos de	Operandos
T	BYTE	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC

T	T+1	T+2	T+3	T+4	T+5	T+6	T+7
Año: 00 a 99	Mes: 01 a 12	Día: 01 a 31	Horas: 00 a 23	Minutos: 00 a 59	Segundos: 00a59	0	Día de la semana: 0 a 7

*T+7 1=Domingo,
7=Sábado

0 desactiva el día de la semana.

Figura 6-4 Formato del búfer de tiempo de 8 bytes (T)



Consejo

La CPU S7-200 no comprueba si el día de la semana coincide con la fecha. Así puede ocurrir que se acepten fechas no válidas, p. ej. el 30 de febrero. Asegúrese de que los datos introducidos sean correctos.

No utilice nunca las operaciones TODR y TODW en el programa principal y en una rutina de interrupción a la vez. Si se está procesando una operación TODR/TODW y se intenta ejecutar simultáneamente otra operación TODR/TODW en una rutina de interrupción, ésta no se ejecutará. SM4.3 se activa indicando que se han intentado dos accesos simultáneos al reloj (error no fatal 0007).

El reloj de tiempo real de la CPU S7-200 utiliza sólo los dos dígitos menos significativos para representar el año. Por tanto, el año 2000 se representa como "00". El S7-200 no utiliza la información relativa al año de ninguna forma. No obstante, si en los programas de usuario se utilizan operaciones aritméticas o de comparación con el valor del año, se deberá tener en cuenta la representación de dos dígitos y el cambio de siglo.

Los años bisiestos se tratan correctamente hasta el año 2096.

Leer reloj de tiempo real ampliado

Condiciones de error que ponen

ENO a 0:

La operación Leer reloj de tiempo real ampliado (TODRX) ^{a 0006} (direccionamiento indirecto) lee de la CPU la fecha y la hora actuales, así como la configuración del horario de verano y las carga en un búfer ^{• 000C} (falta el cartucho de reloj)

de 19 bytes que comienza en la dirección T.

- 0091 (error de rango)

Ajustar reloj de tiempo real ampliado

Condiciones de error que ponen

ENO a 0:

La operación Ajustar reloj de tiempo real ampliado ^{a 0006} (direccionamiento indirecto) (TODWX) escribe en la CPU la fecha y la hora actuales, así como la configuración del horario de verano, comenzando ^{• 0007} (error de daos TOD) en la dirección del búfer de 19 bytes indicada por T. ^{• 000C} (falta el cartucho de reloj)

Todos los valores de la fecha y la hora se deben codificar ^{• 0091} (error de rango) en BCD (por ejemplo, 16#02 para el año 2002). La tabla 6-9 muestra el formato del búfer de tiempo de 19 bytes (T).

El reloj de tiempo real se inicializa con la siguiente fecha y hora tras un corte de alimentación prolongado o una pérdida de memoria:

Fecha: 01--Ene--90
Hora: 00:00:00
Día de la semana: Domingo

Tabla 6-9 Formato del búfer de tiempo de 19 bytes (T)

Byte	Descripción	Datos de byte
0	Año (0-99)	Año actual (valor BCD)
1	Mes (1-12)	Mes actual (valor BCD)
2	Día (1-31)	Día actual (valor BCD)
3	Hora (0-23)	Hora actual (valor BCD)
4	Minuto (0-59)	Minuto actual (valor BCD)
5	Segundo (0-59)	Segundo actual (valor BCD)
6	00	Reservado - ajustado siempre a 00
7	Día de la semana (1 -	Día actual de la semana, 1=domingo (valor BCD)
8	Modo (00H-03H, 08H, 10H-13H, FFH)	Modo de corrección: 00H = corrección inhibida 01H = UE (diferencia horaria de UTC = 0 h) ¹ 02H = UE (diferencia horaria de UTC = +1 h) ¹ 03H = UE (diferencia horaria de UTC = +2 h) ¹ 04H-07H = reservados 08H = UE (diferencia horaria de UTC = -1 h) ¹ 09H-0FH = reservados 10H = EE UU ² 11H = Australia ³ 12H = Australia (Tasmania) ⁴
9	Horas de corrección	Cantidad de corrección, horas (valor BCD)
10	(0-23) Minutos de	Cantidad de corrección, minutos (valor BCD)
11	Mes inicial (1-12)	Mes inicial del horario de verano (valor BCD)
12	Día inicial (1-31)	Día inicial del horario de verano (valor BCD)
13	Hora inicial (0-23)	Hora inicial del horario de verano (valor BCD)
14	Minuto inicial (0-59)	Minuto inicial del horario de verano (valor BCD)
15	Mes final (1-12)	Mes final del horario de verano (valor BCD)
16	Día final (1-31)	Día final del horario de verano (valor BCD)
17	Hora final (0-23)	Hora final del horario de verano (valor BCD)
18	Minuto final (0-59)	Minuto final del horario de verano (valor BCD)

Convención de la Unión Europea (UE): Adelantar el reloj una hora el último domingo de marzo a las 01:00 h (UTC).

Retrasar el reloj una hora el último domingo de octubre a las 02:00 h (UTC). (La hora local a la que se efectúa

la corrección depende de la diferencia horaria frente a UTC).

Convención de los EE UU: Adelantar el reloj una hora el primer domingo de abril a las 02:00 h (hora local).

Retrasar el reloj una hora el último domingo de octubre a las 02:00 h (hora local).

Convención de Australia: Adelantar el reloj una hora el último domingo de octubre a las 02:00 h (hora local).

Retrasar el reloj una hora el último domingo de marzo a las 03:00 h (hora local).

Convención de Australia (Tasmania): Adelantar el reloj una hora el primer domingo de octubre a las 02:00 h (hora

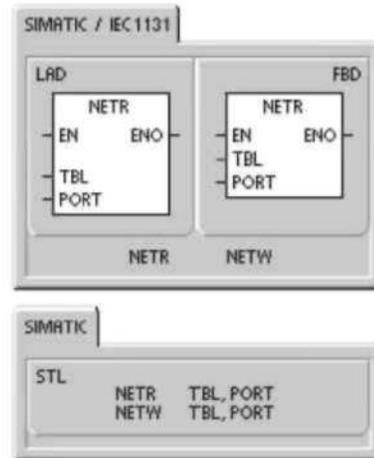
local). Retrasar el reloj una hora el último domingo de marzo a las 03:00 h (hora local).

Convención de Nueva Zelanda: Adelantar el reloj una hora el primer domingo de octubre a las 02:00 h (hora local).

Retrasar el reloj una hora el primer domingo o después del 15 de marzo a las 03:00 h (hora local).

Operaciones de comunicación Leer de la red y Escribir en la red

La operación Leer de la red (NETR) inicia una comunicación para leer datos de una estación remota a través del puerto indicado (PORT), según se define en la tabla (TBL). La operación Escribir en la red (NETW) inicia una comunicación para escribir



datos en una estación remota a través del puerto indicado (PORT), según se define en la tabla (TBL).

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- Si la función devuelve un error y activa el bit E del byte de la tabla de estado (v. fig. 6-5)

Con la operación Leer de la red (NETR) se pueden leer hasta 16 bytes de información de una estación remota, en tanto que con la operación Escribir en la red (NETW) se pueden escribir hasta 16 bytes de información en una estación remota.

El programa puede contener un número cualquiera de operaciones NETR y NETW, pero sólo ocho de ellas (en total) pueden estar activadas simultáneamente. Por ejemplo, en un sistema de automatización S7-200 pueden estar activadas cuatro operaciones NETR y cuatro NETW, o bien 2 operaciones NETR y 6 NETW.

El asistente de operaciones Leer de la red y Escribir en la red se puede utilizar para configurar el contador. Para iniciar el asistente, elija el comando de menú **Herramientas >**

Asistente de operaciones y, a continuación, seleccione el asistente NETR/NETW.

Tabla 6-10 Operandos válidos para las operaciones Leer de la red y Escribir en la red

Entradas/salida	Tipos de	Operandos	
TBL	BYTE	VB, MB, *VD, *LD, *AC	
PORT	BYTE	Constante para las CPUs 221, 222 y 224: para las CPUs 224XP y 226:	0 0 ó 1



Asistente de operaciones

La figura 6-5 muestra la tabla a la que hace referencia el parámetro TBL. En la tabla 6-11 se indican los códigos de error.

Offset de bytes	D	A	E	0	Código de error
0					
1	Dirección de la estación remota				
2	Puntero al área de datos				
3	en la				
4	estación remota				
5	(I. Q. M. o V)				
6	Longitud de datos				
7	Byte de datos 0				
8	Byte de datos 1				
	:				
22	Byte de datos 15				

D Concluida (operación ejecutada): 0 = no ejecutada 1 = ejecutada
 A Activa (operación en cola de espera): 0 = no activa 1 = activa
 E Error (operación incluye un error): 0 = sin error 1 = error

Dirección de la estación remota: dirección del PLC a cuyos datos se desea acceder.
 Puntero al área de datos en la estación remota: puntero indirecto a los datos que se desea acceder.
 Longitud de datos: número de bytes de datos a acceder en la estación remota (1 a 16 bytes).
 Área de datos a recibir o transferir. 1 a 16 bytes reservados para los datos.
 En una operación Leer de la red, esta área de datos almacena los valores leídos de la estación remota cuando se ejecutó la operación.
 En una operación Escribir en la red, esta área de datos almacena los valores a enviar a la estación remota cuando se ejecuta la operación.

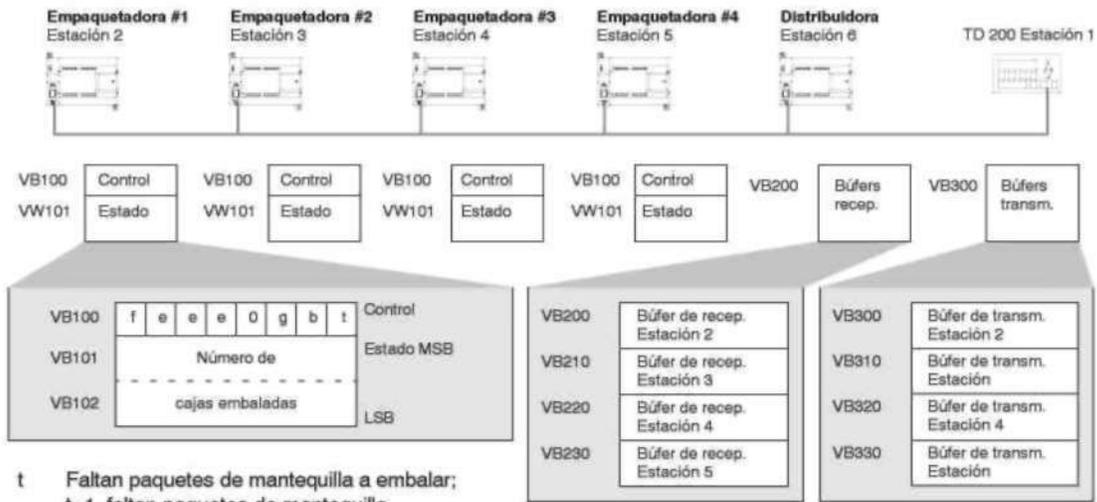
Figura 6-5 Parámetro TBL de las operaciones Leer de la red y Escribir en la red Tabla 6-11 Códigos de error del parámetro TBL

Código Definición

- 0 Sin error.
- 1 Error de timeout: no responde la estación remota.
- 2 Error de recepción: error de paridad, de ajuste o de suma de verificación
- 3 Error offline: colisiones causadas por direcciones de estación repetidas o
- 4 Error de desbordamiento en la cola de espera: se han activado más de 8 operaciones Leer de la red o Escribir en la red.
- 5 Violación de protocolo: se intentó ejecutar una operación Leer de la red o Escribir en la red sin habilitar el modo maestro PPI en SMB30
- 6 Parámetro no válido: el parámetro TBL contiene un valor no válido.
- 7 Sin recursos: la estación remota está ocupada. (Secuencia de cargar en PG o de cargar en CPU en curso.)
- 8 Error de capa 7: violación de protocolo de aplicación.
- 9 Error de mensaje: dirección de datos errónea o longitud de datos

AaF No utilizados (reservados).

La figura 6-6 muestra un ejemplo que ilustra la utilidad de las operaciones Leer de la red y Escribir en la red. Se trata de una línea de producción donde se están llenando paquetes de mantequilla que se envían a una de las cuatro máquinas empaquetadoras. La empaquetadora embala ocho paquetes de mantequilla en cada caja. Una máquina distribuidora controla el flujo de los paquetes de mantequilla hacia cada una de las empaquetadoras. Se utilizan cuatro PLCs S7-200 para controlar las empaquetadoras y un PLC S7-200 equipado con un visualizador de textos TD 200 para controlar a la distribuidora.



- t Faltan paquetes de mantequilla a embalar; t=1, faltan paquetes de mantequilla
- b Suministro bajo de cajas; b=1, hay que añadir cajas en 30 minutos
- g Suministro bajo de cola; g=1, hay que añadir cola en 30 minutos
- eee Código de error para identificar el tipo de fallo que se presentó
- f Indicador de fallos; f=1, la empaquetadora ha detectado un error

Figura 6-6 Ejemplo de las operaciones Leer de la red y Escribir en la red

La figura 6-7 muestra los búfers de recepción (VB200) y transmisión (VB300) para acceder a los datos de la estación 2. El S7-200 utiliza la operación Leer de la red para leer continuamente el control y el estado de cada una de las empaquetadoras. Cada vez que una empaquetadora ha embalado 100 cajas, la máquina distribidora lo registra y envía un mensaje para borrar la palabra de estado utilizando una operación Leer de la red.

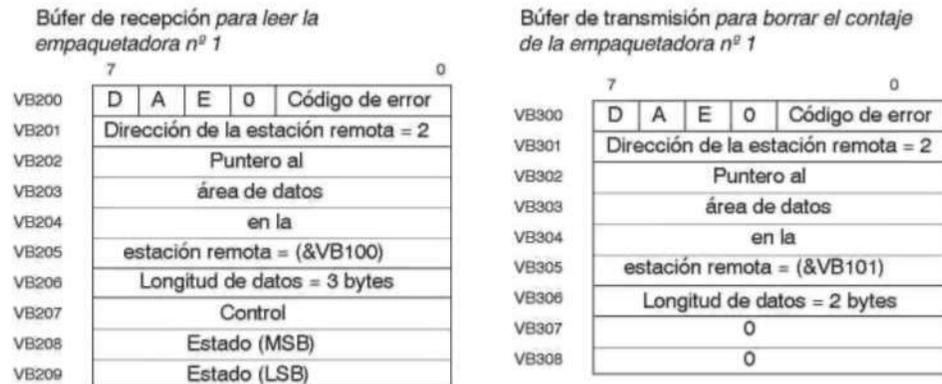
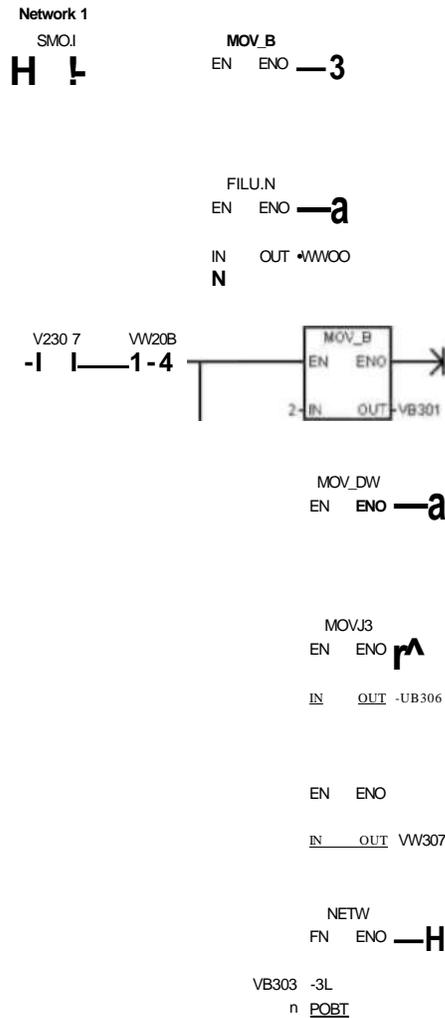


Figura 6-7 Ejemplo de TBL para las operaciones Leer de la red y Escribir en la red

Ejemplo de las operaciones Leer de la red y Escribir en la red



Network 1 //En el primer ciclo, habilitar el modo
//maestro PPI y borrar todos los
//bufers de recepción y transmisión.

LD SM0.1
MOVB 2, SMB30
FILL +0. VW200, 68

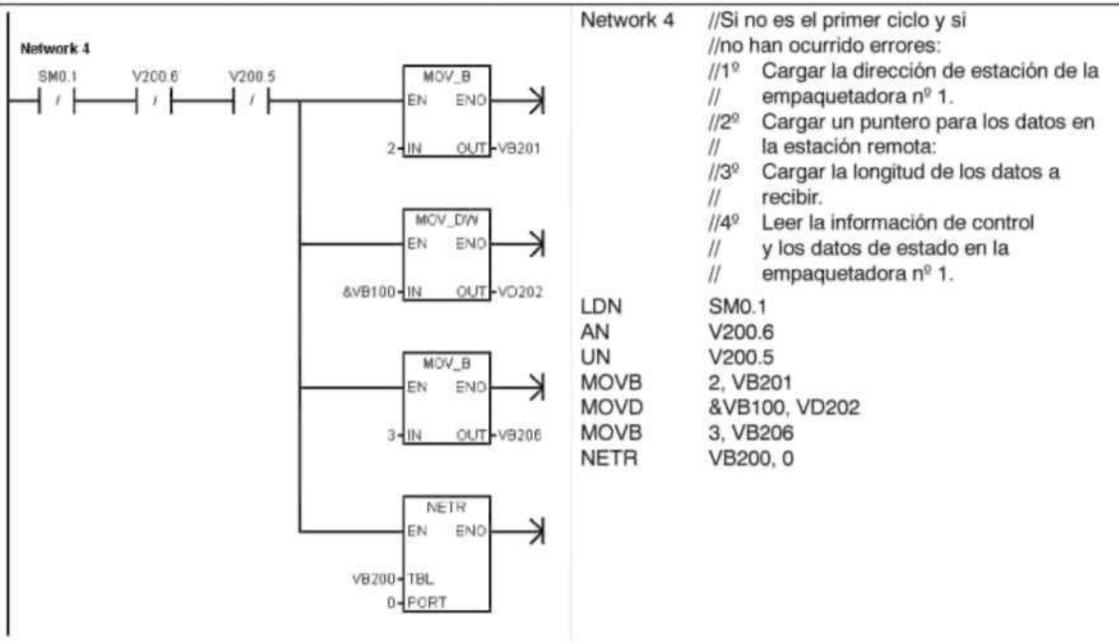
Network 2 //Cuando se active el bit Operación
//NETR (V200.7) ejecutada y se hayan
//empaquetado 100 cajas:
//1^o Cargar la dirección de estación de la
// empaquetadora n° 1.
m Cargar un puntero para los datos en
ii la estación remota:
//3^o Cargar la longitud de los datos a
// transmitir.
//4^o Cargar los datos a transmitir.
//5^o Poner a 0 el número de cajas
// embaladas por la empaquetadora n°1

LD V200.7
AW=
MOVB 2, VB301
MOVD &VB101, VD302
MOVB 2, VB306
MOVW +0. VW307
NETW VB300, 0

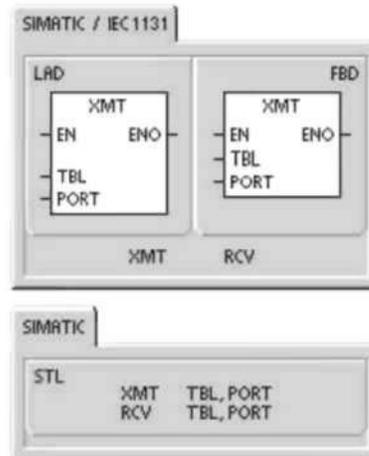
Network 3 //Cuando se active el bit Operación
//NETR ejecutada, guardar los datos de
//control de la empaquetadora n° 1

LD V200.7
MOVB VB207, VB400

Ejemplo de las operaciones Leer de la red y Escribir en la red, continuación



Transmitir mensaje y Recibir mensaje (Freeport)



La operación Transmitir mensaje (XMT) se utiliza en modo Freeport para transmitir datos por el (los) puerto(s) de comunicación.

La operación Recibir mensaje (RCV) inicia o finaliza la función Recibir mensaje. Para el cuadro Recibir mensaje es preciso indicar una condición inicial y final. Los mensajes que se hayan recibido a través del puerto indicado (PORT) se almacenan en el búfer de datos (TBL). La primera entrada del búfer indica el número de bytes que se han recibido.

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0009 (operaciones Transmitir mensaje/Recibir mensaje simultáneas en el puerto 0)
- 000B (operaciones Transmitir mensaje/Recibir mensaje simultáneas en el puerto 1)
- El error de parámetro de la operación Recibir mensaje activa SM86.6 ó SM186.6.
- La CPU S7-200 no está en modo Freeport.

Tabla 6-12 Operandos válidos para las operaciones Transmitir mensaje y Recibir mensaje

Entradas/salida	Tipos de	Operandos	
TBL	BYTE	IB, QB, VB, MB, SMB, SB, *VD, *LD,	
PORT	BYTE	Constante para las CPUs 221, 222 y 224: para las CPUs	0 0 ó 1

Para más información acerca de cómo utilizar el modo Freeport, consulte el apartado “Crear protocolos personalizados en modo Freeport” en la página 240 del capítulo 7.

Utilizar el modo Freeport para controlar el puerto serie de comunicación

El programa de usuario puede controlar el puerto serie del S7-200. La comunicación a través de este puerto se denomina modo Freeport (comunicación programable por el usuario). Eligiendo el modo Freeport, el programa de usuario controla el puerto de comunicación utilizando interrupciones de recepción y de transmisión, así como las operaciones Transmitir mensaje y Recibir mensaje. En modo Freeport, el programa KOP controla todo el protocolo de comunicación. Las marcas especiales SMB30 (para el puerto 0) y SMB130 (para el puerto 1, si el S7-200 dispone de dos puertos) se utilizan para elegir la velocidad de transferencia y la paridad.

Cuando el S7-200 pasa a modo STOP se inhibe el modo Freeport y se restablece la comunicación (por ejemplo, el acceso a través de la unidad de programación).

En el caso más simple se puede enviar un mensaje a la impresora o a la pantalla con sólo utilizar la operación Transmitir mensaje (XMT). Otros ejemplos incluyen la conexión a un lector de código de barras, una balanza o una soldadora. En todo caso, el programa deberá soportar el protocolo con el que el S7-200 se comunica en modo Freeport.

Para poder utilizar el modo Freeport, es preciso que el S7-200 esté en modo RUN. El modo Freeport se habilita ajustando el valor 01 en el campo de selección del protocolo de SMB30 (puerto 0) o de SMB130 (puerto 1). Estando en modo Freeport, la CPU no se puede comunicar con la unidad de programación.

Consejo

El modo Freeport se puede controlar con la marca especial SM0.7 que indica la posición actual del selector de modos de operación. Si SM0.7 = 0, el selector estará en posición TERM; si SM0.7 = 1, el selector estará en posición RUN. Si el modo Freeport se habilita sólo cuando el selector esté en RUN, la unidad de programación se podrá utilizar para vigilar o controlar el funcionamiento del S7-200, cambiando el selector a una posición diferente.

Cambiar la comunicación PPI a modo Freeport

SMB30 y SMB130 se utilizan para inicializar el modo Freeport en los puertos de comunicación 0 y 1, respectivamente, permitiendo elegir la velocidad de transferencia, la paridad y el número de bits por carácter. La figura 6-8 describe el byte de control Freeport. Se genera un bit de parada para todas las configuraciones.

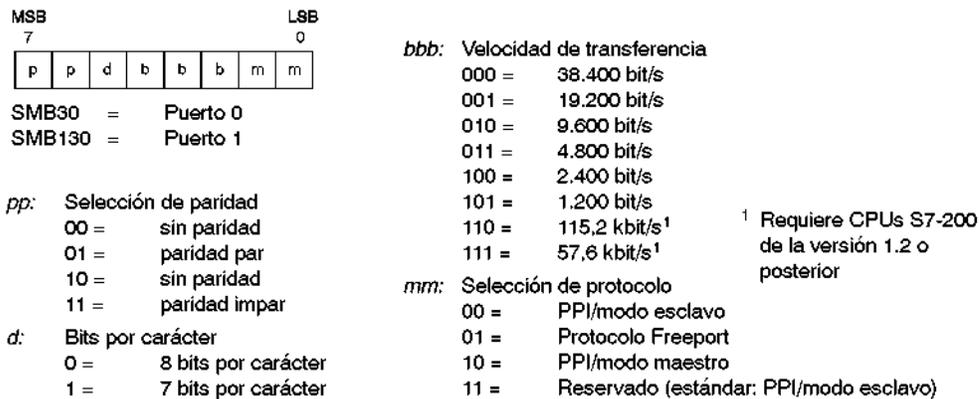


Figura 6-8 Byte de control para el modo Freeport (SMB30 ó SMB130)

Transmitir datos

Con la operación Transmitir mensaje se puede enviar un búfer de uno o más caracteres (255 como máximo).

La figura 6-9 muestra el formato del búfer de transmisión.

Una vez transmitido el último carácter del búfer, el S7-200 genera una interrupción (evento de interrupción 9 para el puerto 0 y evento de interrupción

26 para el puerto 1), si una rutina de interrupción se ha asociado al evento Transmisión finalizada.

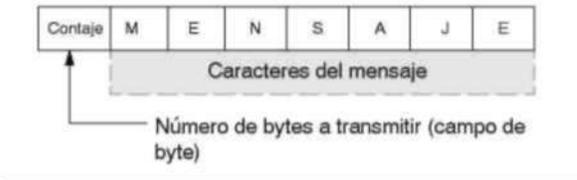


Figura 6-9 Formato del búfer de transmisión

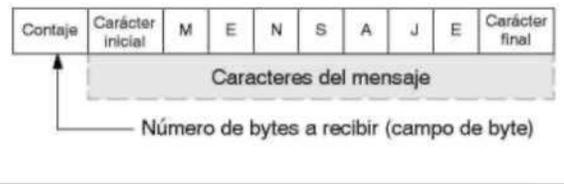
También es posible transmitir datos sin utilizar interrupciones (por ejemplo, enviar un mensaje a una impresora), vigilando SM4.5 ó SM4.6 hasta que finalice la transmisión. La operación Transmitir mensaje (XMT) sirve para generar una condición BREAK, poniendo el número de caracteres a cero y ejecutando luego la operación XMT. Así se genera una condición BREAK en la línea de temporizadores de 16 bits a la velocidad de transferencia actual. La transmisión de una condición BREAK se gestiona de la misma forma que la de cualquier otro mensaje. Una interrupción de transmisión se genera cuando se termina de transmitir la condición BREAK, indicando SM4.5 ó SM4.6 el estado actual de la transmisión.

Recibir datos

Con la operación Recibir mensaje se puede recibir un búfer de uno o más caracteres (255 como máximo).

La figura 6-10 muestra el formato del búfer de recepción.

Una vez recibido el último carácter del búfer, el S7-200 genera una interrupción (evento de interrupción 23 para el puerto 0 y evento de interrupción 24 para el



puerto 1), si una rutina de interrupción se

ha asociado al evento Recepción de mensajes finalizada.

Figura 6-10 Formato del búfer de recepción

También es posible recibir mensajes sin utilizar interrupciones, vigilando para ello SMB86 (puerto 0) o SMB186 (puerto 1). Este byte no será igual a cero si la operación Recibir mensaje está inactiva o si ha finalizado. En cambio, será igual a cero cuando se estén recibiendo datos.

Como muestra la tabla 6-13, la operación Recibir mensaje permite seleccionar las condiciones de comienzo y fin del mensaje, utilizando SMB86 a SMB94 para el puerto 0 y SMB186 a SMB194 para el puerto 1.

Consejo

La recepción de mensajes se finalizará automáticamente si se produce un desbordamiento o un error de paridad. Para la operación Recibir mensaje es preciso definir una condición inicial y una condición final (el número máximo de caracteres).

Tabla 6-13 Bytes del búfer de recepción (SMB86 a SMB94 y SMB186 a SMB194)

Puerto 0 Puerto 1 Descripción

SMB86 SMB186 Byte de estado de MSB
LSB
nr e00 t cp

n: 1 = Ha finalizado la recepción de mensajes: el usuario ha enviado un comando de inhibir.
r: 1 = Ha finalizado la recepción de mensajes: error en los parámetros de entrada o falta condición inicial o final.
e: 1 = Carácter final recibido.
t: 1 = Ha finalizado la recepción de mensajes: ha transcurrido la temporización.
c: 1 = Ha finalizado la recepción de mensajes: se

SMB87 SMB187 Byte de control de MSB
LSB 1 = Ha finalizado la recepción de mensajes: error de paridad.

en: 0 = Inhibida la función de recibir mensajes. 1 = Habilitada la función de recibir mensajes. El bit para habilitar/inhibir la recepción de mensajes se comprueba cada vez que se ejecuta la operación
sc: 0 = Ignorar SMB88 o SMB188.
1 = Utilizar el valor de SMB88 o de SMB188 para detectar el comienzo del mensaje.
ec: 0 = Ignorar SMB89 o SMB189.
1 = Utilizar el valor de SMB89 o de SMB189 para detectar el final del mensaje.
il: 0 = Ignorar SMB90 o SMB190.
1 = Utilizar el valor de SMW90 o SMW190 para detectar una condición de inactividad.
c/m: 0 = Utilizar el temporizador como temporizador entre caracteres. 1= Utilizar el temporizador como temporizador de mensajes.
tmr: 0 = Ignorar SMB92 o SMB192.

SMB88	SMB188	Carácter de comienzo del mensaje si se excede el período de en
SMB89	SMB189	Carácter de fin del mensaje. BREAK.
SMW90	SMW190	Tiempo de línea de inactividad en milisegundos. El primer carácter recibido una vez transcurrido el tiempo de línea de inactividad es el comienzo del nuevo mensaje.
SMW92	SMW192	Vigilancia de tiempo del temporizador entre caracteres/mensajes en milisegundos. Si se excede el tiempo, finalizará la función de recepción de mensajes.
SMB94	SMB194	Número máximo de caracteres a recibir (1 a 255 bytes). Este rango debe ajustarse al tamaño máximo esperado para el búfer, incluso si no se utiliza la finalización de mensajes por el contaje de caracteres.

Condiciones de comienzo y fin para la operación Transmitir mensaje

La operación Recibir mensaje utiliza los bits del byte de control de recepción de mensajes (SMB87 ó SMB187) para definir las condiciones de comienzo y fin del mensaje.



Consejo

Si en el puerto de comunicación hay tráfico con otros aparatos mientras se está ejecutando la operación Recibir mensaje, puede suceder que se comience a recibir un carácter en medio de ese carácter. Ello podría causar un error de paridad y la terminación de la operación Recibir mensaje. Si la paridad no está habilitada, el mensaje recibido podría contener caracteres incorrectos. Ésto puede ocurrir si se ha indicado que la condición inicial sea un carácter inicial en particular o un carácter cualquiera, conforme a lo descrito a continuación en los puntos 2. y 6.

La operación Recibir mensaje soporta varias condiciones iniciales. Definiendo una condición inicial que incluya un BREAK o una detección de inactividad es posible evitar este problema, puesto que se obliga a la operación Recibir mensaje a sincronizar el comienzo del mensaje con el comienzo de un carácter antes de depositar caracteres en el búfer de mensajes.

La operación Recibir mensaje soporta varias condiciones iniciales, a saber:

1. **Detección de inactividad:** La condición de inactividad representa el tiempo muerto (o tiempo de inactividad) en la línea de transmisión. La recepción se inicia cuando la línea de comunicación ha estado inactiva durante el número de milisegundos indicado en SMW90 ó SMW190. Cuando se ejecuta la operación Recibir mensaje en el programa, la función de recepción de mensajes inicia una búsqueda para comprobar si hay una condición de inactividad. Si se reciben caracteres antes de haber transcurrido el tiempo de inactividad, la operación Recibir mensaje ignorará esos caracteres y reiniciará el temporizador de inactividad con el tiempo indicado en SMW90 ó SMW190 (v. fig. 6-11). Una vez transcurrido el tiempo de inactividad, la operación Recibir mensaje almacenará en el búfer de mensajes todos los caracteres que se reciban posteriormente.

El tiempo de inactividad debería ser siempre superior al tiempo necesario para transmitir un carácter (bit de inicio, bits de datos, paridad y bits de parada) a la velocidad de transferencia indicada. Un valor típico del tiempo de inactividad equivale al tiempo necesario para transmitir tres caracteres a la velocidad de transferencia indicada.

La detección de inactividad se utiliza como condición inicial para los protocolos binarios, así como para los protocolos que carezcan de un carácter inicial determinado, o bien si el protocolo prevé un tiempo mínimo entre los mensajes.

Configuración: $il = 1, sc = 0, bk = 0,$

SMW90/SMW190 = timeout de inactividad en milisegundos

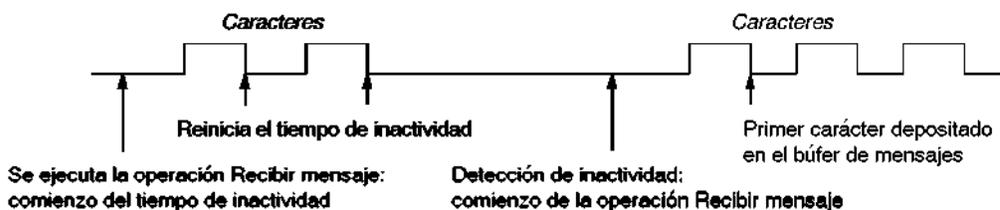


Figura 6-11 Utilizar la detección de inactividad para iniciar la operación Recibir mensaje

2. **Detección del carácter inicial:** El carácter inicial puede ser un carácter cualquiera utilizado como primer carácter de un mensaje. Un mensaje comienza cuando se

recibe el carácter inicial indicado en SMB88 ó SMB188. La operación Recibir mensaje almacena el carácter inicial en el búfer de recepción como primer carácter del mensaje e ignora los caracteres que se hayan recibido antes del carácter inicial. El carácter inicial y todos los demás caracteres recibidos posteriormente se almacenan en el búfer de mensajes.

La detección del carácter inicial se utiliza generalmente con los protocolos ASCII en los que todos los mensajes comienzan con un mismo carácter.

Configuración: il = 0, sc = 1, bk = 0, SMW90/SMW190 = no es relevante,
SMB88/SMB188 = carácter inicial

3. Inactividad y carácter inicial: La operación Recibir mensaje puede iniciar un mensaje combinando las condiciones de inactividad y de carácter inicial. Cuando se ejecuta la operación Recibir mensaje, ésta inicia una búsqueda para comprobar si hay una condición

de inactividad. Tras haber detectado la condición de inactividad, la operación Recibir mensaje buscará el carácter inicial indicado. Si se ha recibido algún carácter que no sea el

carácter inicial, esta operación buscará de nuevo una condición de inactividad, ignorando

todos los caracteres recibidos antes de cumplirse la condición de inactividad y antes de

haberse recibido el carácter inicial. El carácter inicial se deposita en el búfer de mensajes

junto con los caracteres posteriores.

El tiempo de inactividad debería ser siempre superior al tiempo necesario para transmitir un carácter (bit de inicio, bits de datos, paridad y bits de parada) a la velocidad de transferencia indicada. Un valor típico del tiempo de inactividad equivale al tiempo necesario para transmitir tres caracteres a la velocidad de transferencia indicada.

Por lo general, este tipo de condición inicial se utiliza si el protocolo indica un tiempo mínimo entre los mensajes y si el primer carácter del mensaje es una dirección o algo que haga referencia a un aparato en particular. Ello se adecúa especialmente si se debe implementar un protocolo que comprenda varios aparatos en el enlace de comunicación. En este caso, la operación Recibir mensaje disparará una interrupción sólo si se ha recibido un mensaje para la dirección indicada o para los aparatos especificados por el carácter inicial.

Configuración: il = 1, sc = 1, bk = 0, SMW90/SMW190 > 0,
SMB88/SMB188 = carácter inicial

4. Detección de BREAK: Cuando los datos recibidos se conservan a un valor cero durante un tiempo superior al necesario para transmitir un carácter entero, se indicará una condición BREAK. El tiempo de transferencia de un carácter entero equivale al tiempo total para transmitir

los bits de inicio, de datos, de paridad y de parada. Si la operación Recibir mensaje se configura de manera que inicie un mensaje cuando se reciba una condición BREAK, todos los

caracteres recibidos después de esa condición se depositarán en el búfer de mensajes. Todos

los caracteres recibidos antes de la condición BREAK se ignorarán.

Por lo general, la detección de BREAK se utiliza como condición inicial si lo exige el

protocolo en cuestión.

Configuración: il = 0, sc = 0, bk = 1, SMW90/SMW190 = no es relevante,
SMB88/SMB188 = no es relevante

5. BREAK y carácter inicial: La operación Recibir mensaje se puede configurar de manera que comience a recibir caracteres tras haberse detectado una condición BREAK y, posteriormente, un carácter inicial en particular (en ese orden). Tras haberse detectado la condición BREAK, la función de recepción de mensajes buscará el carácter inicial indicado.

Si se ha recibido algún carácter que no sea el carácter inicial, esta operación buscará de

nuevo una condición BREAK, ignorando todos los caracteres recibidos antes de cumplirse

la condición de BREAK y antes de haberse recibido el carácter inicial. El carácter inicial se

deposita en el búfer de mensajes junto con los caracteres posteriores.

Configuración: il = 0, sc = 1, bk = 1, SMW90/SMW190 = no es relevante,
SMB88/SMB188 = carácter inicial

6. Carácter cualquiera: La operación Recibir mensaje se puede configurar de manera que comience a recibir inmediatamente un carácter cualquiera o todos los caracteres, depositándolos luego en el búfer de mensajes. Este es un caso especial de la detección de inactividad. En este caso, el tiempo de inactividad (SMW90 ó SMW190) se pone a cero. Ello obliga a la operación Recibir mensaje a comenzar a recibir caracteres inmediatamente después de su ejecución.

Configuración: il = 1, sc = 0, bk = 0, SMW90/SMW190 = 0,
SMB88/SMB188 = no es relevante

Si un mensaje comienza con un carácter cualquiera, el temporizador de mensajes se puede utilizar para medir el tiempo de espera (timeout) durante la recepción de mensajes. Ello es especialmente útil cuando el modo Freeport se utiliza para implementar el maestro o la sección "host" de un protocolo, siendo necesario utilizar un timeout si no se obtiene respuesta de un esclavo en un período de tiempo determinado. El temporizador de mensajes arrancará cuando se ejecute la operación Recibir mensaje, puesto que el tiempo de inactividad se había ajustado a cero. El temporizador de mensajes se detendrá tras transcurrir el tiempo de espera y finalizará la operación de recepción si no se ha cumplido ninguna otra condición final.

Configuración: il = 1, sc = 0, bk = 0, SMW90/SMW190 = 0,

SMB88/SMB188 = no es relevante

c/m = 1, tmr = 1, SMW92 = tiempo de espera del mensaje en milisegundos

La operación Recibir mensaje soporta varias formas de finalizar un mensaje. El mensaje puede terminar cuando se cumplan una o varias de las condiciones siguientes:

1. Detección del carácter final: El carácter final puede ser un carácter cualquiera utilizado para señalar el fin del mensaje. Tras encontrar la condición inicial, la operación Recibir mensaje comprobará cada carácter recibido para determinar si concuerda con el carácter final. Una vez recibido el carácter final, éste se deposita en el búfer de mensajes y se finaliza la

recepción.

La detección del carácter final se utiliza generalmente con los protocolos ASCII en los que todos los mensajes terminan con un determinado carácter. La detección del carácter final se puede combinar con el temporizador entre caracteres, con el temporizador de mensajes, o bien con el contaje máximo de caracteres para terminar un mensaje.

Configuración: $ec = 1$, SMB89/SMB189 = carácter final

2. Temporizador entre caracteres: El tiempo entre caracteres es el período medido desde el final de un carácter (el bit de parada) y el final del carácter siguiente (el bit de parada). Si el tiempo entre caracteres (incluyendo el segundo carácter) excede el número de milisegundos indicado en SMW92 ó SMW192, se finalizará la operación Recibir mensaje.

El temporizador entre caracteres se inicializa cada vez que se recibe un carácter (v. fig. 6-12).

El temporizador entre caracteres se puede utilizar para terminar un mensaje en el caso de los protocolos que carezcan de un carácter final específico. Este temporizador se debe ajustar a un valor superior al tiempo necesario para transmitir un carácter a la velocidad de transferencia seleccionada, toda vez que el temporizador comprende siempre el tiempo para recibir un carácter entero (bit de inicio, bits de datos, paridad y bits de parada).

El temporizador entre caracteres se puede combinar con la detección del carácter final y el contaje máximo de caracteres para terminar un mensaje.

Configuración: $c/m = 0$, $tmr = 1$,

SMW92/SMW192 = tiempo de espera en milisegundos

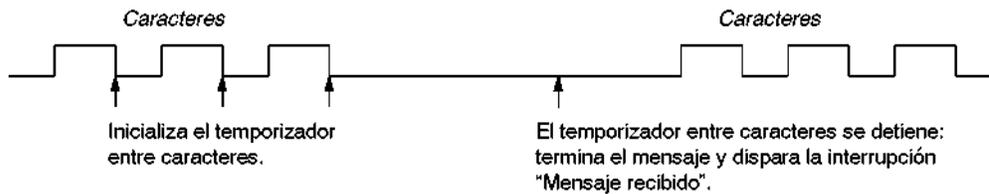


Figura 6-12 Utilizar el temporizador entre caracteres para finalizar la operación Recibir mensaje

3. Temporizador de mensajes: El temporizador de mensajes termina un mensaje tras haber transcurrido un tiempo determinado desde el comienzo del mismo. El temporizador de mensajes arrancará una vez que se hayan cumplido la condición o las condiciones iniciales para la recepción de mensajes y se detendrá tras haber transcurrido el número de milisegundos indicado en SMW92 ó SMW192 (v. fig. 6-13). Por lo general, el temporizador de mensajes se utiliza si los aparatos que intervienen en la comunicación no pueden garantizar que no habrá intervalos de tiempo entre los caracteres, o bien si se utilizan módems. En este último caso, es posible utilizar un temporizador de mensajes para determinar el tiempo máximo admisible para recibir un mensaje una vez que este haya comenzado. Un valor típico para un temporizador de mensajes sería aproximadamente 1,5 veces mayor que el tiempo necesario para recibir el mensaje más largo posible a la velocidad de transferencia seleccionada. El temporizador de mensajes se puede combinar con la detección del carácter final y el conteo máximo de caracteres para terminar un mensaje.

Configuración: $c/m = 1$, $tmr = 1$, SMW92/SMW192 = tiempo de espera en milisegundos

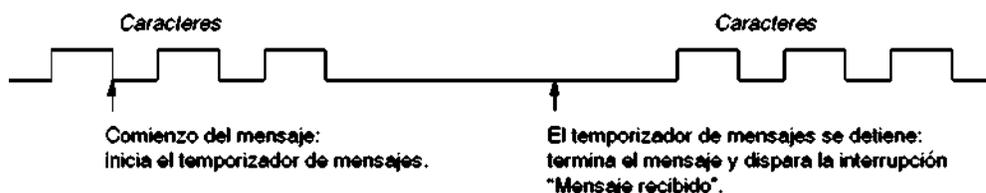


Figura 6-13 Utilizar el temporizador de mensajes para finalizar la operación Recibir mensaje

4. Número máximo de caracteres: Al configurar la operación Recibir mensaje es preciso indicar el número máximo de caracteres a recibir (SMB94 ó SMB194). Una vez alcanzado o excedido este valor, finalizará la recepción de mensajes. La operación Recibir mensaje exige que el usuario indique un conteo máximo de caracteres, aunque éste no se utilice como condición final. Ello se debe a que esta operación necesita conocer el tamaño máximo del mensaje recibido, de manera que no se sobrescriban los datos de usuario depositados después del búfer de mensajes. El conteo máximo de caracteres se puede utilizar para terminar mensajes en el caso de los protocolos en los que se conozca la longitud del mensaje y donde dicha longitud sea siempre igual. El conteo máximo de caracteres se utiliza siempre en combinación con la detección del carácter final, el temporizador entre caracteres, o bien el temporizador de mensajes.
5. Errores de paridad: La operación Recibir mensaje se finalizará automáticamente si el hardware detecta un error de paridad en un carácter recibido. Los errores de paridad se detectan únicamente si la paridad se ha habilitado en SMB30 ó SMB130. Esta función no se puede inhibir.
6. Terminación por el programa de usuario: El programa de usuario puede terminar la función de recepción de mensajes ejecutando otra operación Recibir mensaje con el bit de habilitación (EN) puesto a cero en SMB87 ó SMB187. Ello termina inmediatamente la recepción de mensajes.

Utilizar interrupciones de caracteres para recibir datos

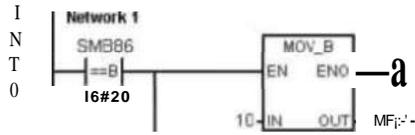
Para disponer de una mayor flexibilidad en los protocolos soportados, los datos se pueden recibir también de forma controlada por interrupciones de caracteres. Cada carácter recibido genera una interrupción. El carácter recibido se deposita en SMB2 y el estado de la paridad (si se ha habilitado) se deposita en SM3.0. Ello sucede inmediatamente antes de ejecutarse la rutina de interrupción asociada al evento Recibir carácter. SMB2 es el búfer de recepción de caracteres en modo Freeport. Cada carácter recibido en modo Freeport se deposita en esa dirección para que el programa de usuario pueda acceder rápidamente a los valores. SMB3 se utiliza para el modo Freeport y contiene un bit de error de paridad que se activa si se detecta un error de ese tipo en un carácter recibido. Todos los demás bits del byte se reservan. Utilice el bit de paridad para rechazar el mensaje o para generar un acuse negativo del mensaje.

Si la interrupción de caracteres se utiliza a velocidades de transferencia elevadas (38,4 kbit/s a 115,2 kbit/s), el tiempo entre las interrupciones será muy breve. Por ejemplo, la interrupción de caracteres es de 260 microsegundos a 38,4 kbit/s, de 173 microsegundos a 57,6 kbit/s y de 86 microsegundos a 115,2 kbit/s. Vigile que las rutinas de interrupción sean muy cortas para evitar que se pierdan caracteres, o bien utilice la operación Recibir mensaje.

Consejo

SMB2 y SMB3 son compartidos por los puertos 0 y 1. Si, debido a la recepción de un carácter por el puerto 0, se ejecuta la rutina de interrupción asociada a ese evento (evento de interrupción 8), SMB2 contendrá el carácter recibido por el puerto 0, en tanto que SMB3 contendrá la paridad de ese carácter. Si, debido a la recepción de un carácter por el puerto 1, se ejecuta la rutina de interrupción asociada a ese evento (evento de interrupción 25), SMB2 contendrá el carácter recibido por el puerto 1, en tanto que SMB3 contendrá la paridad de ese carácter.

Ejemplo de operaciones de transmisión y recepción, continuación



```

Network 1 //Interrupción "Recepción finalizada":
//1 - Si el estado muestra la recepción del
// carácter final, asociar un temporizador
// de 10 ms para disparar una transmisión;
// luego retornar.
//2º Si la recepción se ha finalizado por otro motivo,
// iniciar una nueva recepción.

```

```

          ATCH
          EN ENO -51
INT_1 INT
1C EVNT

-(RETÍ)

```

```

LDB SMB86, 16#20
MOVB 10, SMB34
ATCH INT 1,10
CRETI
NOT
RCV VB100, 0

```

```

---|NOTJ|---EN ENO---|)

```

VB10C-TBL
C-PORT

Network 1

SVOXJ
H h

```

          DTCH
          EN ENO -51
10- EVNT

```

```

Network 1 //Interrupción temporizada de 10 ms:
//1 - Desasociar la interrupción temporizada.
//2ª Transmitir el mensaje de vuelta al usuario en
// el puerto.

```

```

LD SMOO
DTCH 10
XMT VB100. 0

```

```

          XMT
          EN ENO -a

```

vBIOD TBL
0 PORT

Network 1
SMD 0
H I-

```

          RCV
          EN ENO
vBinr-TFH
G-PORT

```

```

Network 1 //Interrupción "Transmisión finalizada":
//Habilitar otra recepción.

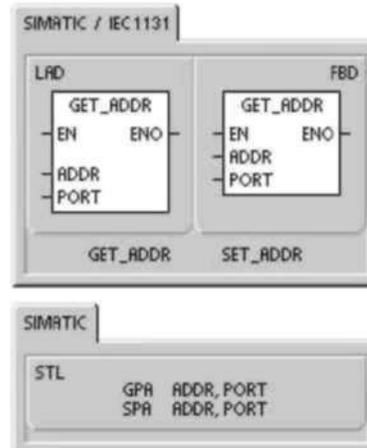
```

```

LD SMOO
RCV VB100. 0

```

Leer dirección de puerto y Ajustar dirección de puerto



La operación Leer dirección de puerto (GPA) lee la dirección de estación del puerto de la CPU S7-200 indicado en PORT y deposita el valor en la dirección indicada en ADDR. La operación Ajustar dirección de puerto (SPA) ajusta la dirección de estación del puerto (PORT) al valor indicado en ADDR. La nueva dirección no se almacena de forma permanente. Tras desconectar y conectar de nuevo la alimentación de la CPU, el puerto afectado recuperará su dirección antigua (la que se había cargado junto con el bloque de sistema).

Condiciones de error que ponen ENO a 0:

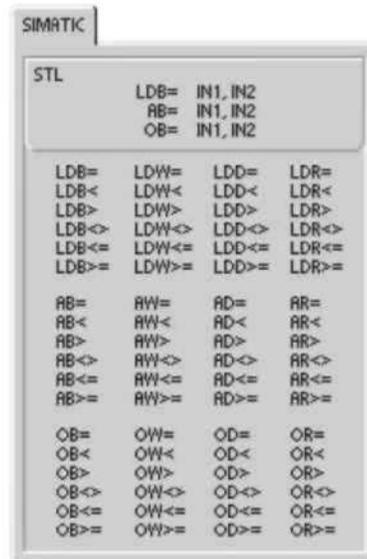
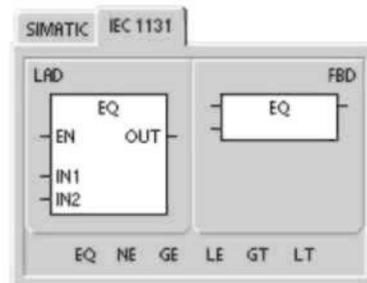
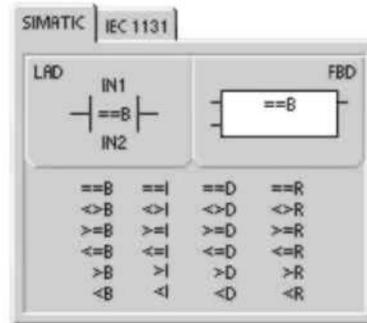
- 0006 (direccionamiento indirecto)
- 0004 (intento de ejecutar una operación Ajustar dirección de puerto en una rutina de interrupción)

Tabla 6-14 Operandos válidos para las operaciones Leer dirección de puerto y Ajustar dirección de puerto

Entradas/salida	Tipos de	Operandos
ADDR	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante <u>(Los valores de constante son aplicables sólo</u>
PORT	BYTE	Constante para las CPUs 221, 222y 224: 0 para las CPUs 224XPy226: 061

Operaciones de comparación Comparar valores numéricos

Las operaciones de comparación se utilizan para comparar dos valores:



IN1 = IN2 IN1 >= IN2 IN1 <= IN2

IN1 > IN2 IN1 < IN2 IN1 <> IN2

Las comparaciones de bytes no llevan signo. Las comparaciones de enteros llevan signo. Las comparaciones de palabras dobles llevan signo. Las comparaciones de números reales llevan signo.

En KOP y FUP: Si la comparación es verdadera, la operación de comparación activa el contacto (KOP) o la salida (FUP).

En AWL: Si la comparación es verdadera, la operación de comparación carga un 1 en el nivel superior de la pila, o bien lo combina con Y u O.

Si se utilizan las operaciones de comparación IEC, es posible utilizar diversos

tipos de datos para las entradas. No obstante, el tipo de datos de los dos valores de entrada deberá ser idéntico.

Nota

Las siguientes condiciones son errores fatales que detendrán inmediatamente la ejecución del programa en el S7-200:

- Detección de una dirección indirecta no válida (en todas las operaciones de comparación)
- Detección de un número real no válido (por ejemplo, NAN) (en la operación Comparar reales)

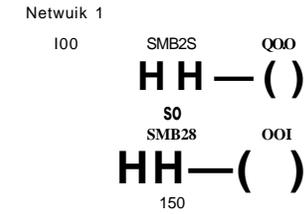
Para evitar estas condiciones de error, inicialice correctamente los punteros y los valores que contengan números reales antes de ejecutar las operaciones de comparación que utilicen estos valores.

Las operaciones de comparación se ejecutan sin tener en cuenta el estado de señal.

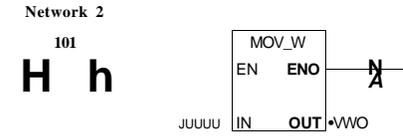
Tabla 6-15 Operandos válidos para las operaciones de comparación

Entradas/salidas	Tipos	Operandos
IN1, IN2	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC,
	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante
	DINT REAL	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,
OUT	BOOL	I, Q, V, M, SM, S, T, C, L, circulación de corriente

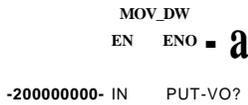
Ejemplo de operaciones de comparación



Network 1 //Activar el potenciómetro analógico 0 para
 //cambiar el valor de byte de SMB28.
 //Q0.0 está activada si el valor de SMB28
 //es menor que o igual a 50.
 //Q0.1 está activada si el valor de SMB28
 //es mayor que o igual a 150.
 //El indicador de estado se activa si
 //la comparación es verdadera.

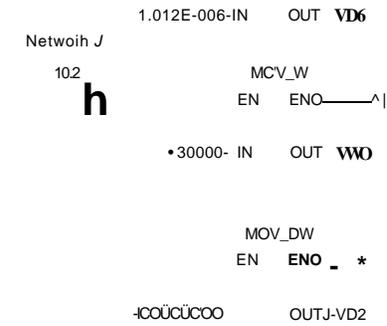


LD I0.0
 LPS
 AB<= SMB28, 50
 Q00
 LPP
 AB>= SMB28, 150
 Q0.1



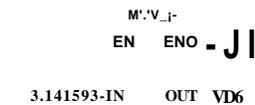
Network 2 //Cargar las direcciones de la memoria V con
 //valores bajos para que las comparaciones
 //sean falsas y para que se apaguen
 //los indicadores de estado.

LD 10.1
 MOVW -30000, VWO
 MOVD -200000000, VD2
 MOVR 1.012E-006, VD6



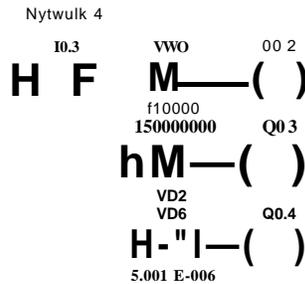
Network 3 //Cargar las direcciones de la memoria V con
 //valores altos para que las comparaciones
 //sean verdaderas y para que se enciendan
 //los indicadores de estado.

LD 10.2
 MOVW +30000, VWO
 MOVD -100000000, VD2
 MOVR 3.141593, VD6



Network 4 //La comparación de enteros de 16 bits
 //comprueba si VWO > +10000 es verdadero.
 //Utiliza constantes del programa para
 //mostrar los diferentes tipos de datos.
 //También se pueden comparar dos valores
 //guardados en la memoria programable
 //p. ej.: VWO>VW100

LD I03
 LPS
 AW> VWO, +10000
 = Q0.2
 LRD
 AD< -150000000, VD2
 = Q0.3
 LPP
 AR> VD6, 5.001 E-006
 = Q0.4



Comparar cadenas

La operación Comparar cadenas compara dos cadenas de caracteres ASCII:

IN1 = IN2 IN1 <> IN2

Si la comparación es verdadera, la operación de comparación activa el contacto (KOP) o la salida (FUP), o bien carga un 1 en el nivel superior de la pila, o lo combina con YuO(AWL).

Nota

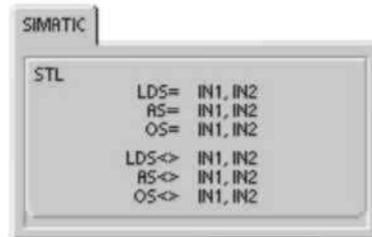
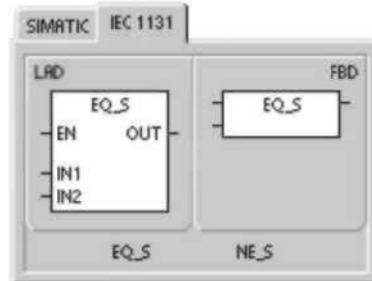
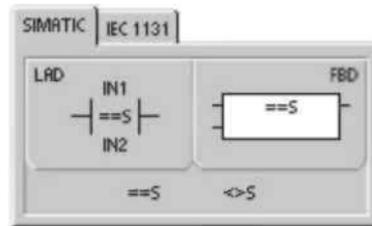
Las siguientes condiciones son errores fatales que detendrán inmediatamente la ejecución del programa en el S7-200:

- Detección de una dirección indirecta no válida (en todas las operaciones de comparación)
- Detección de una cadena que comprenda más de 254 caracteres (en la operación Comparar cadenas)
- Detección de una cadena cuya dirección inicial y longitud no quepan en el área de memoria indicada (en la operación Comparar cadenas)

Para evitar estas condiciones de error, inicialice correctamente los punteros y las direcciones que deban acoger cadenas ASCII antes de ejecutar las operaciones de comparación que utilicen estos valores. Vigile que el búfer reservado para una cadena ASCII se pueda depositar por completo en el área de memoria indicada.

Las operaciones de comparación se ejecutan sin tener en cuenta el estado de señal.

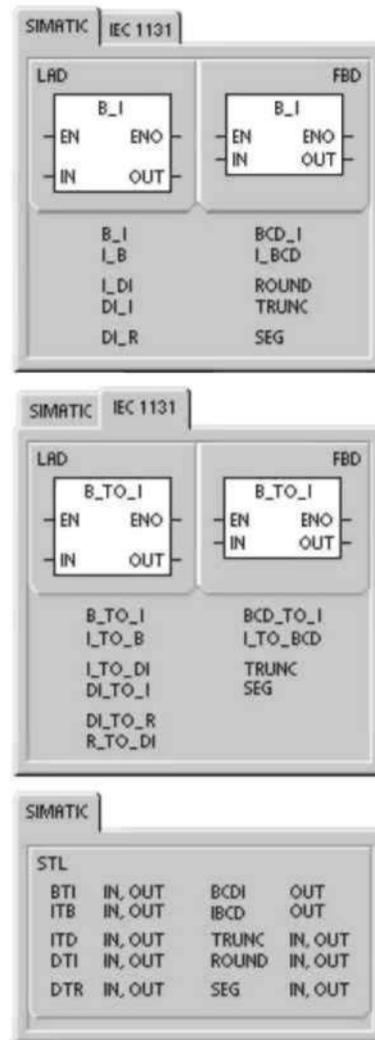
Tabla 6-16 Operandos válidos para la operación Comparar cadenas



Entradas/sali	Tipos de	Operandos
IN1	STRING	VB, LB, *VD, *LD, *AC, constante
IN2	STRING	VB, LB, *VD, *LD, *AC
OUT	BOOL	I, Q, V, M, SM, S, T, C, L, circulación de corriente

Operaciones de conversión

Operaciones de conversión normalizadas



Conversiones numéricas

Las operaciones Convertir byte en entero (BTI), Convertir entero en byte (ITB), Convertir entero en entero doble (ITD), Convertir entero doble en entero (DTI), Convertir entero doble en real (DTR), Convertir BCD en entero (BCDI) y Convertir entero en BCD (IBCD) convierten un valor de entrada IN en el formato indicado y almacenan el valor de salida en la dirección especificada por OUT. Por ejemplo, es posible convertir un valor de entero doble en un número real. También es posible convertir un entero en un número BCD y viceversa.

Redondear a entero doble y Truncar

La operación Redondear (ROUND) convierte un valor real (IN) en un valor de entero doble y deposita el resultado redondeado en la variable indicada por OUT.

La operación Truncar (TRUNC) convierte un número real (IN) en un entero doble y carga la parte del número entero del resultado en la variable indicada por OUT.

Segmento

La operación Segmento (SEG) sirve para generar una configuración binaria (OUT)

que ilumina los segmentos de un indicador de siete segmentos.

Tabla 6-17 Operandos válidos para las operaciones de conversión normalizadas

Entradas/salidas	Tipos	Operandos
IN	BYTE WORD, INT	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, IW, QW, VW, MW, SMW, SW, T, C, LW, AIW, AC, *VD, *LD, *AC, constante
	DINT REAL	ID, QD, VD, MD, SMD, SD, LD, HC, AC, *VD, *LD, ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,
OUT	BYTE WORD, INT	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD,
	DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,

Funcionamiento de las operaciones Convertir BCD en entero y Convertir entero en BCD

La operación Convertir BCD en entero (BCDI) convierte el **Condiciones de error que ponen**

valor decimal codificado en binario IN en un valor de entero y carga el resultado en la variable indicada por OUT. El rango válido de IN está comprendido entre 0 y 9999 BCD.

ENO a 0:

- SM1.6 (BCD no válido)
- 0006 (direccionamiento indirecto)

La operación Convertir entero en BCD (IBCD) convierte el valor entero de entrada IN en un valor BCD y carga el resultado en la variable indicada por OUT. El rango válido (válido) de IN está comprendido entre 0 y 9999 enteros.

Marcas especiales afectadas:

- SM1.6 (BCD no

Funcionamiento de la operación Convertir entero doble en real

La operación Convertir entero doble en real (DTR) convierte **Condiciones de error que ponen**

un entero de 32 bits con signo IN en un número real de 32 bits y deposita el resultado en la variable indicada por OUT.

ENO a 0:

- 0006 (direccionamiento indirecto)

Funcionamiento de la operación Convertir entero doble en entero

La operación Convertir entero doble en entero (DTI) **Condiciones de error que ponen**

convierte el valor de entero doble IN en un valor de entero y deposita el resultado en la variable indicada por OUT.

ENO a 0:

- SM1.1 (desbordamiento)
- " 0006 (direccionamiento indirecto)

Si el valor a convertir es demasiado grande para ser

representado en la salida, la marca de desbordamiento se **Marcas especiales afectadas:**

activará y la salida no se verá afectada.

a SM1 1 (desbordamiento)

Funcionamiento de la operación Convertir entero en entero doble

La operación Convertir entero en entero doble (ITD) **Condiciones de error que ponen**

convierte el valor de entero IN en un valor de entero doble y deposita el resultado en la variable indicada por OUT. El signo se amplía.

ENO a 0:

- 0006 (direccionamiento indirecto)

Funcionamiento de la operación Convertir byte en entero

La operación Convertir byte en entero (BTI) convierte el **Condiciones de error que**

ponen

valor de byte IN en un valor de entero y deposita el ^{ENO a 0:} resultado en la variable indicada por OUT. El byte no tiene • 0006 (direccionamiento indirecto) signo. Por tanto, no hay ampliación de signo.

Funcionamiento de la operación Convertir entero en byte

La operación Convertir entero en byte (ITB) convierte el **Condiciones de error que ponen**

valor de entero IN en un valor de byte y deposita el ^{ENO a 0:} resultado en la variable indicada por OUT. Se convierten los • SM1.1 (desbordamiento)

valores comprendidos entre 0 y 255. Todos los demás • 0006 (direccionamiento indirecto)

valores producen un desbordamiento y la salida no se ve

afectada.

Marcas especiales afectadas:

- SM1.1 (desbordamiento)

Consejo

Para convertir un entero en un número real, utilice la operación Convertir entero en entero doble y luego la operación Convertir entero doble en real.

Funcionamiento de las operaciones Redondear a entero doble y Truncar

La operación Redondear (ROUND) convierte un número **Condiciones de error que ponen** real (IN) en un valor de entero doble y deposita el resultado ^{ENO a 0:}

en la variable indicada por OUT. Si la fracción es 0,5 o • SM1.1 (desbordamiento) superior, el número se redondeará al próximo entero superior. „ 0006 (direccionamiento indirecto)

La operación Truncar (TRUNC) convierte un número real **Marcas especiales**

afectadas: (IN) en un entero doble y carga el resultado en la variable • SM1.1 (desbordamiento) indicada por OUT. Sólo se convierte la parte entera del número real. La fracción se pierde.

Si el valor a convertir no es un número real válido o si es demasiado grande para ser representado en la salida, la marca de desbordamiento se activará y la salida no se verá afectada.

Ejemplo Operaciones de conversión normalizadas

<p>Network 1</p> <p>I0.0</p> <p>H I-</p> <p>I_DI EN EN←</p> <p>C1Q IN OUT-AC1</p> <p>D1_R EN ENC 3</p> <p>AC 1 - IN OUT -VDO</p> <p>M1_R EN ENO - ^</p> <p>VDO IN1 OUT-VD8 VD4 IN2</p> <p>ROUND EN ENO</p> <p>VD8 OUT-VD12</p> <p>NftwniK;'</p> <p>I0.3</p> <p>H I-</p> <p>BCD. EN ENC</p> <p>ACÓ- IN OUT-ACG</p>	<p>Network 1 //Cambiar de pulgadas a centímetros: //1^a Cargar el valor del contador // (pulgadas) en AC1. 112- Convertir el valor en un número real. 1/3- Multiplicar por 2,54 (cambiar a centímetros). //4^o Reconvertir el valor en un número entero.</p> <p>LD 10.0 ITD C10. AC1 DTR AC1, VDO MOVR VDO. VD8 *R VD4. VD8 ROUND VD8. VD12</p> <p>Network 2 //Convertir un valor BCD en un número entero.</p> <p>LD I0.3 BCDI ACÓ</p>
---	---

Convertir entero de palabra doble a real y redondear

C10	101	Contar = 101 pulgadas
VDO	<u>101.0</u>	Contar (como número real)
VD4	<u>2.54</u>	Factor 2.54 (cambiar de pulgadas a centímetros)
VD8	256.54	256,54 centímetros es un número real.
VD12	257	257 centímetros es un entero doble

BCD a entero

ACÓ	<u>1234</u>
	BCDI
ACÓ	<u>104D2</u>

Funcionamiento de la operación Segmento

Con objeto de iluminar los segmentos de un indicador de siete segmentos, la operación Segmento (SEG) convierte el carácter (byte) indicado por IN para generar una configuración binaria (byte) en la dirección indicada por OUT.

Los segmentos iluminados representan el carácter **Condiciones de error que ponen** depositado en el dígito menos significativo del byte de ^{ENO a 0:} entrada.

La figura 6-14 muestra la codificación del indicador de siete segmentos utilizado por la operación Segmento.

(IN) LSD	Indicador	(OUT) -gfedcba	(IN) LSD	Indicador	(OUT) -gfedcba
0	0	0011 1111	8	8	0111 1111
1	1	0000 0110	9	9	0110 0111
2	2	0101 1011	A	A	0111 0111
3	3	0100 1111	B	B	0111 1100
4	4	0110 0110	C	C	0011 1001
5	5	0110 1101	D	D	0101 1110
6	6	0111 1101	E	E	0111 1001
7	7	0000 0111	F	F	0111 0001

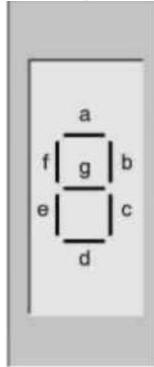
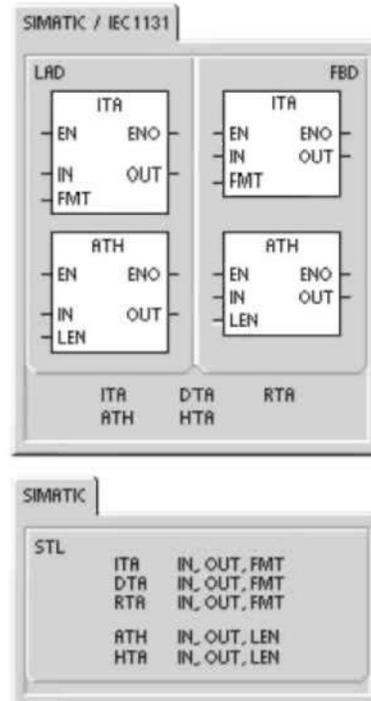


Figura 6-14 Codificación de un indicador de siete segmentos
Ejemplo de la operación Segmento

Networkkl		Network 1		05	SEG	6D
11 0	SEO	LD	11.0	VB48		AC1
H	H ^{EN}	SEG	VB48, AC1			
VB48	i ; --f				(carácter Indicado)	

Operaciones de conversión ASCII

Los caracteres ASCII admisibles son los valores hexadecimales 30 a 39 y 41 a 46.



Convertir ASCII en hexadecimal y viceversa

La operación Convertir ASCII en hexadecimal (ATH) convierte un número LEN de caracteres ASCII, a partir del carácter IN, en dígitos hexadecimales, comenzando en OUT. La operación Convertir hexadecimal en ASCII (HTA) convierte los dígitos hexadecimales a partir del byte de entrada IN en caracteres ASCII, comenzando en OUT. El número de dígitos hexadecimales a convertir viene indicado por la longitud LEN. Es posible convertir como máximo 255 caracteres ASCII, o bien dígitos hexadecimales. Introducción ASCII válida

Los caracteres ASCII válidos son los caracteres alfanuméricos 0 a 9 con un valor de código hexadecimal de 30 a 39, así como las mayúsculas A a F con un valor de código hexadecimal de 41 a 46.

Condiciones de error que ponen ENO a 0:

- SM1.7 (ASCII no válido) *Sólo al convertir ASCII en hexadecimal*
- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)

Marcas especiales afectadas:

- SM1.7 (ASCII no válido)

Convertir valores numéricos en ASCII

Las operaciones Convertir entero en ASCII (ITA), Convertir entero doble en ASCII (DTA) y Convertir real en ASCII (RTA) convierten enteros, enteros dobles o números reales en caracteres ASCII.

Tabla 6-18 Operandos válidos para las operaciones de conversión ASCII

Entradas/salidas	Tipos	Operandos
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante
	INT	
LEN, FMT	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,
	REAL	
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC

Funcionamiento de la operación Convertir entero en ASCII

La operación Convertir entero en ASCII (ITA) convierte un entero (IN) en un array de caracteres ASCII. El formato FMT indica la precisión de la conversión a la derecha del separador decimal, así como si éste debe aparecer en forma de coma o de punto. La conversión resultante se deposita en 8 bytes consecutivos comenzando en OUT. El array de caracteres ASCII comprende siempre 8 caracteres.

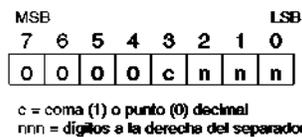
- Condiciones de error que ponen ENO a 0:**
- 0006 (direccionamiento indirecto)
 - $nnn > 5$

La figura 6-15 describe el operando de formato (FMT) para la operación Convertir entero en ASCII. El tamaño del búfer de salida es siempre de 8 bytes. El campo *nnn* indica el número de dígitos a la derecha del separador decimal en el búfer de salida. El rango válido para el campo *nnn* está comprendido entre 0 y 5. Si se indican 0 dígitos a la derecha del separador decimal, el valor se visualizará sin separador. Si el valor *nnn* es mayor que 5, el búfer de salida se llenará con espacios ASCII. El bit *c* indica si se utiliza una coma (*c* = 1) o un punto decimal (*c* = 0) como separador entre el número entero y la fracción. Los 4 bits superiores deben ser "0".

La figura 6-15 muestra ejemplos de valores que se formatean utilizando un punto decimal (*c*=0) con tres dígitos a la derecha del mismo (*nnn* = 011). El búfer de salida se formatea conforme a las siguientes reglas:

- Los valores positivos se escriben sin signo en el búfer de salida.
- Los valores negativos se escriben precedidos de un signo menos (-) en el búfer de salida.
- Los ceros a la izquierda del separador decimal (con excepción del dígito adyacente al mismo) se suprimen.
- Los valores se justifican a la derecha en el búfer de salida.

FMT



	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7
in=12			0	.	0	1	2
in=-123		-	0	.	1	2	3
in=1234			1	.	2	3	4
in = -12345	-	1	2	.	3	4	5

Figura 6-15 Operando FMT para la operación Convertir entero en ASCII (ITA)

Funcionamiento de la operación Convertir entero doble en ASCII

La operación Convertir entero doble en ASCII (DTA) **Condiciones de error que ponen**

convierte un entero de 32 bits (IN) en un array de caracteres ASCII. El operando de formato FMT indica la precisión de **• 0006** (direccionamiento indirecto)

conversión a la derecha del separador decimal. La **• F t áld**
 conversión resultante se deposita en 12 bytes consecutivos comenzando en OUT. **• nnn > 5**

El tamaño del búfer de salida es siempre de 12 bytes.

La figura 6-16 describe el operando de formato (FMT) para la operación Convertir entero doble en ASCII. El campo *nnn* indica el número de dígitos a la derecha del separador decimal en el búfer de salida. El rango válido para el campo *nnn* está comprendido entre 0 y 5. Si se indican 0 dígitos a la derecha del separador decimal, el valor se visualizará sin separador. Si el valor *nnn* es mayor que 5, el búfer de salida se llenará con espacios ASCII. El bit *c* indica si se utiliza una coma (*c* = 1) o un punto decimal (*c* = 0) como separador entre el número entero y la fracción. Los 4 bits superiores deben ser "0".

La figura 6-16 muestra ejemplos de valores que se formatean utilizando un punto decimal (*c* = 0) con cuatro dígitos a la derecha del mismo (*nnn* = 100). El búfer de salida se formatea conforme a las siguientes reglas:

- Los valores positivos se escriben sin signo en el búfer de salida.
- Los valores negativos se escriben precedidos de un signo menos (-) en el búfer de salida.
- Los ceros a la izquierda del separador decimal (con excepción del dígito adyacente al mismo) se suprimen.
- Los valores se justifican a la derecha en el búfer de salida.

FMT

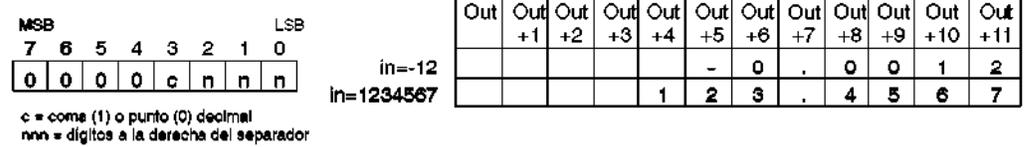


Figura 6-16 Operando FMT para la operación Convertir entero doble en ASCII (DTA)

Funcionamiento de la operación Convertir real en ASCII

La operación Convertir real en ASCII (RTA) convierte un **Condición de error que ponen**

número real IN en caracteres ASCII. El formato FMT indica **ENO a 0:**
 la precisión de la conversión a la derecha del separador **• 0006** (direccionamiento indirecto)

decimal, así como si éste debe aparecer en forma de coma **• nnn > 5**
 o de punto, y también el tamaño del búfer de salida.

La conversión resultante se deposita en un búfer de salida **• ssss < 3**

que comienza en OUT. **• ssss < número de caracteres en OUT**

El número (o longitud) de los caracteres ASCII resultantes corresponde al tamaño del

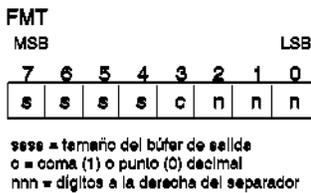
búfer de salida, pudiendo indicarse en un rango comprendido entre 3 y 15 bytes o caracteres.

El formato de número real utilizado por la CPU S7-200 soporta 7 dígitos significativos como máximo. Si se intenta visualizar más de estos 7 dígitos, se producirá un error de redondeo.

La figura 6-17 describe el operando de formato (FMT) para la operación Convertir real en ASCII (RTA). El campo ssss indica el tamaño del búfer de salida. No es válido un tamaño de 0, 1 ó 2 bytes. El campo nnn indica el número de dígitos a la derecha del separador decimal en el búfer de salida. El rango válido para el campo nnn está comprendido entre 0 y 5. Si se indican 0 dígitos a la derecha del separador decimal, el valor se visualizará sin separador. El búfer de salida se rellena con espacios ASCII si los valores nnn son mayores que 5 o si el búfer es demasiado pequeño para almacenar el valor convertido. El bit c indica si se utiliza una coma (c = 1) o un punto decimal (c = 0) como separador entre el número entero y la fracción.

La figura 6-17 muestra ejemplos de valores que se formatean utilizando un punto decimal (c = 0) con un dígito a la derecha del mismo (nnn = 001) y un tamaño de búfer de seis bytes (ssss = 0110). El búfer de salida se formatea conforme a las siguientes reglas:

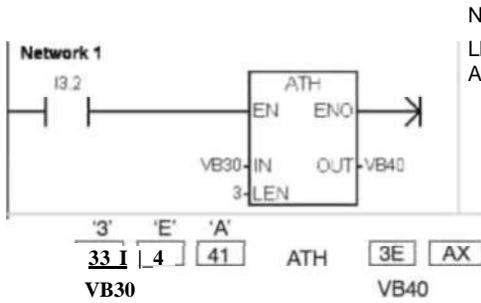
- Los valores positivos se escriben sin signo en el búfer de salida.
- Los valores negativos se escriben precedidos de un signo menos (-) en el búfer de salida.
- Los ceros a la izquierda del separador decimal (con excepción del dígito adyacente al mismo) se suprimen.
- Los valores a la derecha del separador decimal se redondean para que correspondan al número de dígitos indicado.
- El búfer de salida deberá ser por lo menos tres bytes más grande que el número de dígitos a la derecha del separador decimal.
- Los valores se justifican a la derecha en el búfer de salida.



	Out +1	Out +2	Out +3	Out +4	Out +5
in = 1234.5	1	2	3	4	. 5
in = -0.0004				0	. 0
in = -3.67526			-	3	. 7
in = 1.95				2	. 0

Figura 6-17 Operando FMT para la operación Convertir real en ASCII (RTA)

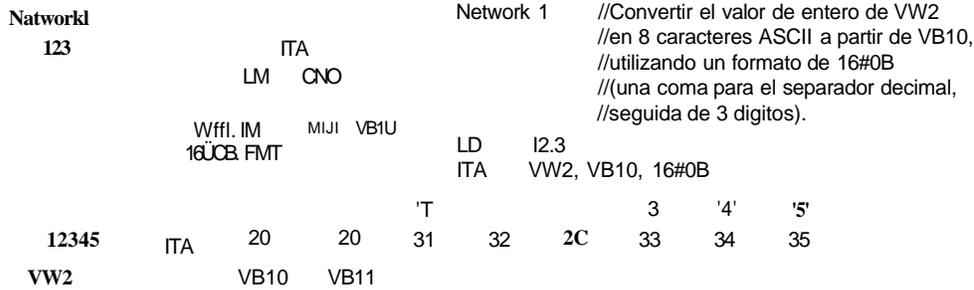
Ejemplo de la operación Convertir ASCII en hexadecimal



Network 1
 LD I3.2
 ATH VB30, VB40, 3

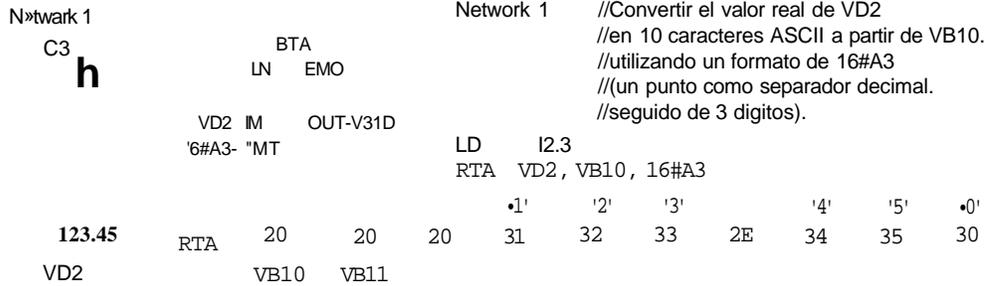
Nota: La "X" indica que el medio byte no se ha modificado.

Ejemplo de la operación Convertir entero en ASCII



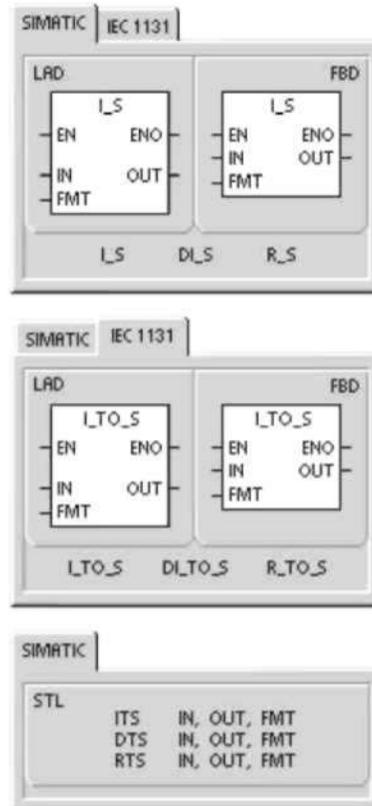
Network 1 //Convertir el valor de entero de VW2
 //en 8 caracteres ASCII a partir de VB10,
 //utilizando un formato de 16#0B
 //(una coma para el separador decimal,
 //seguida de 3 digitos).
 LD I23
 ITA VW2, VB10, 16#0B

Ejemplo de la operación Convertir real en ASCII



Network 1 //Convertir el valor real de VD2
 //en 10 caracteres ASCII a partir de VB10.
 //utilizando un formato de 16#A3
 //(un punto como separador decimal.
 //seguido de 3 digitos).
 LD I23
 RTA VD2, VB10, 16#A3

Operaciones de conversión de cadenas



Convertir valores numéricos en cadenas

Las operaciones Convertir entero en cadena (ITS), Convertir entero doble en cadena (DTS) y Convertir real en cadena (RTS) convierten valores de enteros, enteros dobles o números reales (IN) en una cadena ASCII (OUT).

Funcionamiento de la operación Convertir entero en cadena

La operación Convertir entero en cadena (ITS) convierte un entero (IN) en una cadena ASCII de 8 caracteres de longitud. El formato (FMT) indica la precisión de la conversión a la derecha del separador decimal, así como si éste debe aparecer en forma de coma o de punto. La cadena resultante se escribe en 9 bytes consecutivos a partir de OUT. Para más información sobre el formato de cadenas, consulte el capítulo 4.

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)
- Formato no válido ($nnn > 5$)

La figura 6-18 describe el operando de formato de la operación Convertir entero en cadena. La cadena resultante tiene siempre una longitud de 8 caracteres. El campo nnn indica el número de dígitos a la derecha del separador decimal en el búfer de salida. El rango válido para el campo nnn está comprendido entre 0 y 5. Si se indican 0 dígitos a la derecha del separador decimal, el valor se visualizará sin separador. Si el valor nnn es mayor que 5, resultará una cadena de 8 caracteres de espacios ASCII. El bit c indica si se utiliza una coma ($c = 1$) o un punto decimal ($c = 0$) como separador

entre el número entero y la fracción. Los 4 bits superiores del formato deben ser "0".

La figura 6-18 muestra ejemplos de valores que se formatean utilizando un punto decimal (c = 0) con tres dígitos a la derecha del mismo (nnn = 011). El valor de OUT representa la longitud de la cadena.

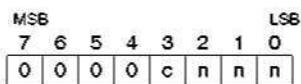
La cadena resultante se formatea conforme a las siguientes reglas:

- Los valores positivos se escriben sin signo en el búfer de salida.
- Los valores negativos se escriben precedidos de un signo menos (-) en el búfer de salida.
- Los ceros a la izquierda del separador decimal (con excepción del dígito adyacente al mismo) se suprimen.
- Los valores se justifican a la derecha en la cadena resultante.

Tabla 6-19 Operandos válidos de las operaciones que convierten valores numéricos en cadenas

Entradas/salidas	Tipos de datos	Operandos
IN	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AIW, *VD, *LD, *AC, constante
	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, constante
	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, constante
FMT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante
OUT	STRING	VB, LB, *VD, *LD, *AC

FMT



c = coma (1) o punto (0) decimal
 nnn = dígitos a la derecha del separador

	Out +1	Out +2	Out +3	Out +4	Out +5	Out +6	Out +7	Out +8
in=12	8			0	.	0	1	2
in=-123	8			0	.	1	2	3
in=1234	8			1	.	2	3	4
in = -12345	8		-	1	.	2	3	4

Figura 6-18 Operando FMT para la operación Convertir entero en cadena

Funcionamiento de la operación Convertir entero doble en cadena

La operación Convertir entero doble en cadena (DTS) **Condiciones de error que ponen**

convierte un entero doble (IN) en una cadena ASCII de 12 caracteres de longitud. El formato (FMT) indica la precisión (direccionamiento indirecto) de la conversión a la derecha del separador decimal, así

ENO a 0:

- 0006

• 0091 (operando fuera de rango)

como si éste debe aparecer en forma de coma o de punto.

La cadena resultante se escribe en 13 bytes consecutivos a partir de OUT. Para más información sobre el formato de cadenas, consulte el capítulo 4.

La figura 6-19 describe el operando de formato de la operación Convertir entero en cadena. La cadena resultante tiene siempre una longitud de 8 caracteres. El campo nnn indica el número de dígitos a la derecha del separador decimal en el búfer de salida. El rango válido para el campo nnn está comprendido entre 0 y 5. Si se indican 0 dígitos a la derecha del

La figura 6-19 describe el operando de formato de la operación Convertir entero en cadena. La cadena resultante tiene siempre una longitud de 8 caracteres. El campo nnn indica el número de dígitos a la derecha del separador decimal en el búfer de salida. El rango válido para el campo nnn está comprendido entre 0 y 5. Si se indican 0 dígitos a la derecha del

separador decimal, el valor se visualizará sin separador. Si el valor nnn es mayor que 5, resultará una cadena de 12 caracteres de espacios ASCII. El bit c indica si se utiliza una coma (c = 1) o un punto decimal (c = 0) como separador entre el número entero y la fracción. Los 4 bits superiores del formato deben ser "0".

La figura 6-19 muestra también ejemplos de valores que se formatean utilizando un punto decimal (c = 0) con cuatro dígitos a la derecha del mismo (nnn = 100). El valor de OUT representa la longitud de la cadena. La cadena resultante se formatea conforme a las siguientes reglas:

- Los valores positivos se escriben sin signo en el búfer de salida.
- Los valores negativos se escriben precedidos de un signo menos (-) en el búfer de salida.
- Los ceros a la izquierda del separador decimal (con excepción del dígito adyacente al mismo) se suprimen.
- Los valores se justifican a la derecha en la cadena resultante.

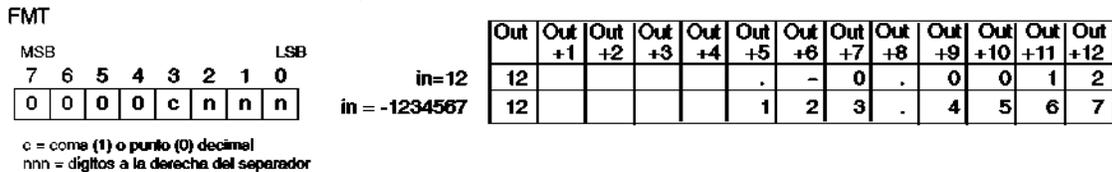


Figura 6-19 Operando FMT para la operación Convertir entero doble en cadena

Funcionamiento de la operación Convertir real en cadena

La operación Convertir real en cadena (RTS) convierte un **Condicionen de error que ponen**

número real (IN) en una cadena ASCII. El formato (FMT) ENO a 0:
indica la precisión de la conversión a la derecha del separador decimal, así como si éste debe aparecer en forma de coma o de punto, y también la longitud de la cadena resultante.

- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)
- Formato no válido: nnn > 5

La conversión resultante se deposita en una cadena que comienza en OUT. El formato indica la longitud de la cadena resultante, que puede comprender entre 3 y 15 caracteres. ssss < número de caracteres necesarios

Para más información sobre el formato de cadenas, consulte el capítulo 4.

El formato de número real utilizado por la CPU S7-200 soporta 7 dígitos significativos como máximo. Si se intenta visualizar más de estos 7 dígitos, se producirá un error de redondeo.

La figura 6-20 describe el operando de formato de la operación Convertir real en cadena. El campo ssss indica la longitud de la cadena resultante. No es válido un tamaño de 0, 1 ó 2 bytes. El campo nnn indica el número de dígitos a la derecha del separador decimal en el búfer de salida. El rango válido para el campo nnn está comprendido entre 0 y 5. Si se indican 0 dígitos a la derecha del separador decimal, el valor se visualizará sin separador. La cadena resultante se rellena con caracteres de espacios ASCII si nnn es mayor que 5

o si la cadena resultante es demasiado pequeña para almacenar el valor convertido. El bit c indica si se utiliza una coma (c = 1) o un punto decimal (c = 0) como separador entre el número entero y la fracción.

La figura 6-20 muestra ejemplos de valores que se formatean utilizando un punto decimal (c = 0) con un dígito a la derecha del mismo (nnn = 001) y una cadena de 6 caracteres de longitud (ssss = 0110). El valor de OUT representa la longitud de la cadena. La cadena resultante se formatea conforme a las siguientes reglas:

- Los valores positivos se escriben sin signo en el búfer de salida.
- Los valores negativos se escriben precedidos de un signo menos (-) en el búfer de salida.
- Los ceros a la izquierda del separador decimal (con excepción del dígito adyacente al mismo) se suprimen.
- Los valores a la derecha del separador decimal se redondean para que correspondan al número de dígitos indicado.
- La cadena resultante deberá ser por lo menos tres bytes más grande que el número de dígitos a la derecha del separador decimal.
- Los valores se justifican a la derecha en la cadena resultante.

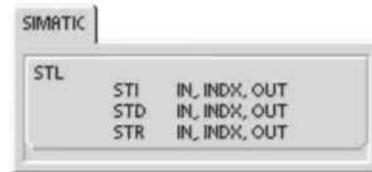
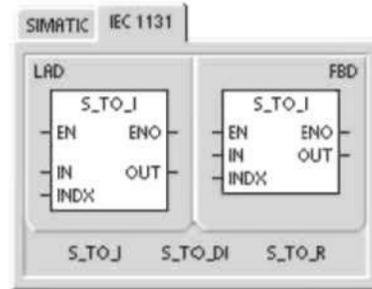
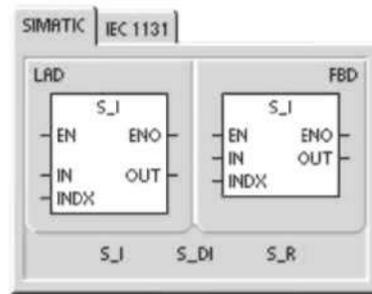
FMT

MSB							LSB
7	6	5	4	3	2	1	0
s	s	s	s	c	n	n	n

ssss = longitud de la cadena resultante
 c = coma (1) o punto (0) decimal
 nnn = dígitos a la derecha del separador

	Out						
	+1	+2	+3	+4	+5	+6	
in = 1234.5	6	1	2	3	4	.	5
in = -0.0004	6				0	.	0
in = -3.67526	6			-	3	.	7
in = 1.95	6				2	.	0

Figura 6-20 Operando FMT para la operación Convertir real en cadena



Convertir subcadenas en valores numéricos

Las operaciones Convertir subcadena en entero (STI), Convertir subcadena en entero doble (STD) y Convertir subcadena en real (STR) convierten un valor de cadena (IN), a partir del offset INDX, en un entero, un entero doble o un número real (OUT).

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)
- 009B (índice = 0)
- SM1.1 (desbordamiento)

Las operaciones Convertir subcadena en entero y Convertir subcadena en entero doble convierten cadenas que tengan el formato siguiente: [espacios] [+ ó -] [dígitos 0 - 9]

La operación Convertir subcadena en real convierten cadenas que tengan el formato siguiente: [espacios] [+ ó -] [dígitos 0 - 9] [. ó ,] [dígitos 0 - 9]

Por lo general, el valor de INDX se pone a "1", lo que inicia la conversión del primer carácter de la cadena. INDX se puede ajustar a otros valores para iniciar la conversión en diferentes puntos de la cadena. Esto se puede utilizar si la cadena de entrada contiene texto que no pertenezca al número a convertir. Por ejemplo, si la cadena de entrada es "Temperatura: 77,8", el valor de INDX se debe ajustar a 13 para omitir la palabra "Temperatura:" al comienzo de la cadena.

La operación Convertir subcadena en real no convierte cadenas que utilicen una notación científica o formas exponenciales de números reales. La operación no causa un error de rebase del límite superior (SM1.1), sino que convierte la cadena en un número real hasta el exponencial y termina luego la conversión. Por ejemplo, la cadena „1.234E6 se convierte sin errores en el valor real 1.234.

La conversión finaliza al alcanzarse el final de la cadena o cuando se detecte el primer carácter no válido. Los caracteres no válidos son todos aquellos que no sean dígitos (0 - 9).

La marca especial de rebase del límite superior (SM1.1) se activará si de la conversión resulta un número entero que sea demasiado grande para el valor de salida. Por ejemplo, la operación Convertir subcadena en entero activará esa marca de error si la cadena de entrada produce un valor que sea mayor que 32767 o menor que -32768.

La marca especial de rebase del límite superior (SM1.1) se activará también si no es posible efectuar una conversión, debido a que la cadena de entrada no contiene un valor válido. Por ejemplo, si la cadena de entrada es „A123 , la operación de conversión activará la marca especial SM1.1 (rebase del límite superior) y el valor de salida permanecerá inalterado.

Tabla 6-20 Operandos válidos de las operaciones que convierten subcadenas en valores numéricos

Entradas/sali	Tipos de	Operandos
IN	STRING	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC,
INDX	BYTE	VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC,
OUT	INT DINT, REAL	VW, IW, QW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Cadenas de entrada válidas para enteros y enteros dobles

Cadena de entrada	Entero resultante
'123'	123
'-00456'	-456
'123.45'	123
'+2345'	2345
'000000123ABCD'	123

Cadenas de entrada válidas para números reales

Cadena de entrada	Real resultante
'123'	123.0
'-00456'	-456.0
'123.45'	123.45
'+2345'	2345.0
'00.000000123'	0.000000123

Cadenas de entrada no válidas

Cadena de entrada
'A123'
' '
'++123'
'+-123'
'+ 123'

Figura 6-21 Ejemplos de cadenas de entrada válidas y no válidas

Ejemplo de operaciones de conversión de cadenas: Convertir subcadena en entero, entero doble y real

```

I      Network      1      Network 1 //Convertir la cadena numérica en un
entero.
      IO.0          I  s~j  I          //Convertir la cadena numérica en un
--- 1  I 1-----EN ENO --- ^|          //entero doble.
                                          //Convertir la cadena numérica en un
                                          //número real.

```

```

VBOJIN  OUTLVWIOO
      7-yNDX --- 1  LD  IO.0
                      STI  VB0,7,VW100
                      v. STD  VB0,7,VD200
                      s* STR  VB0,7,VD300

```

```

VBüjIN
  QUTLVD2ÜÜ
  7-IINDX |

```

```

  EN ÉNO - ^|

```

```

VBojIN
  OUTLVD300
  7-IINDX |

```

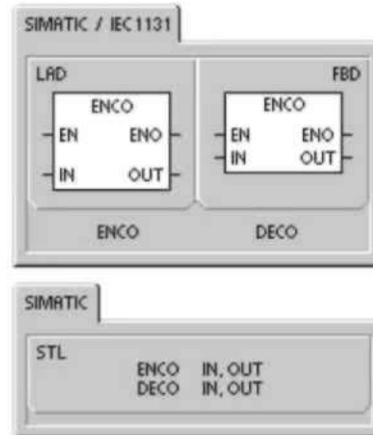
```

VB0      VB11
11 T e m p          9 8 . 6 F

```

Tras ejecutar el segmento: VW100
 (entero) = 98 VD200
 (entero doble) = 98
 VD300 (real) = 98.6

Codificar y Decodificar



Codificar

La operación Codificar (ENCO) escribe el número del bit menos significativo de la palabra de entrada IN en el medio byte menos significativo (4 bits) del byte de salida OUT.

Decodificar

La operación Decodificar (DECO) activa el bit de la palabra de salida OUT. Este bit corresponde al número de bit representado por el medio byte menos significativo (4 bits) del byte de entrada IN. Todos los demás bits de la palabra de salida se ponen a 0.

Marcas especiales y ENO

Las siguientes condiciones afectan a ENO en las operaciones Codificar y Decodificar.

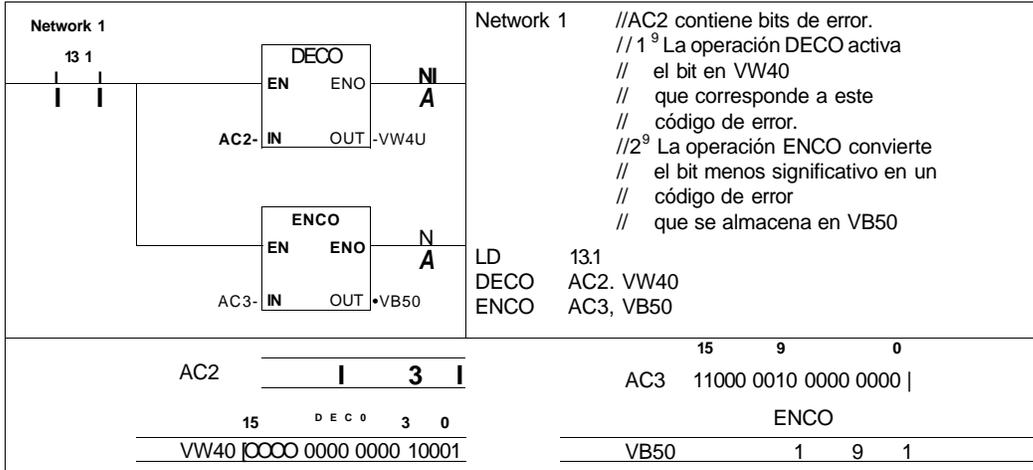
Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)

Tabla 6-21 Operandos válidos para las operaciones Codificar y Decodificar

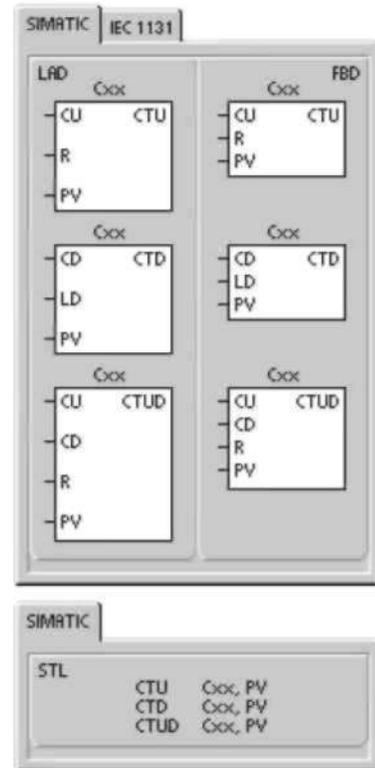
Entradas/salidas	Tipos de datos	Operandos
IN	BYTE WORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante
OUT	BYTE WORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC

Ejemplo de las operaciones Codificar y Decodificar



Operaciones de contaje

Operaciones de contaje (SIMATIC)



Incrementar contador

La operación Incrementar contador (CTU) empieza a contar adelante a partir del valor actual cuando se produce un flanco positivo en la entrada de contaje adelante (CU). Si el valor actual (Cxx) es mayor o igual al valor de preselección PV, se activa el bit de contaje Cxx. El contador se inicializa cuando se activa la entrada de desactivación (R) o al ejecutarse la operación Poner a 0. El contador se detiene cuando el valor de contaje alcance el valor límite superior (32.767).

Funcionamiento en AWL:

- Entrada de desactivación: valor superior de la pila
- Entrada de contaje adelante: valor cargado en el segundo nivel de la pila

Decrementar contador

La operación Decrementar contador (CTD) empieza a contar atrás a partir del valor actual cuando se produce un flanco negativo en la entrada de contaje atrás (CD). Si el valor actual Cxx es igual a 0, se activa el bit de contaje Cxx. El contador desactiva el bit de contaje Cxx y carga el valor actual con el valor de preselección (PV) cuando se activa la entrada de carga LD. El contador se detiene al alcanzar el valor cero y el bit de contaje Cxx se activa.

Funcionamiento en AWL:

- Entrada de carga: valor superior de la pila
- Entrada de contaje atrás: valor cargado en el segundo nivel de la pila

Incrementar/decrementar contador

La operación Incrementar/decrementar contador (CTUD) empieza a contar adelante cuando se produce un flanco positivo en la entrada de contaje adelante (CU), y empieza a contar atrás cuando se produce un flanco positivo en la entrada de contaje atrás (CD). El valor actual Cxx del contador conserva el contaje actual. El valor de preselección PV se compara con el valor actual cada vez que se ejecuta la operación de contaje.

Cuando se alcanza el valor máximo (32.767), el siguiente flanco positivo en la entrada de contaje adelante invertirá el contaje hasta alcanzar el valor mínimo (-32.768). Igualmente, cuando se alcanza el valor mínimo (-32.768), el siguiente flanco positivo en la entrada de contaje atrás invertirá el contaje hasta alcanzar el valor máximo (32.767).

Si el valor actual (Cxx) es mayor o igual al valor de preselección PV, se activa el bit de contaje Cxx. En caso contrario, se desactiva el bit. El contador se inicializa cuando se activa la entrada de desactivación (R) o al ejecutarse la operación Poner a 0. El contador adelante/atrás se detiene al alcanzar el valor de preselección (PV).

Funcionamiento en AWL:

- Entrada de desactivación: valor superior de la pila
- Entrada de contaje atrás: valor cargado en el segundo nivel de la pila
- Entrada de contaje adelante: valor cargado en el tercer nivel de la pila

Tabla 6-22 Operandos válidos para las operaciones de contaje SIMATIC

Entradas/salida	Tipos de	Operandos
Cxx	WORD	Constante (C0 a C255)
CU , CD , LD , R	BOOL	I, Q, V, M, SM, S, T, C, L, circulación de
PV	INT	IW, QW, VW, MW, SMW, SW, LW, T, C, AC, AIW, *VD, *LD, *AC, constante

Consejo

Puesto que cada contador dispone sólo de un valor actual, no se puede asignar un mismo número a varios contadores. (Las operaciones Incrementar contador, Incrementar/decrementar contador y Decrementar contador acceden a un mismo valor actual).

Cuando se inicializa un contador con la operación Poner a 0, se desactivan tanto el bit de contaje como el valor actual del contador. El número del contador se debe utilizar para direccionar tanto el valor actual como el bit de ese contador.

Tabla 6-23 Funcionamiento de las operaciones de contaje

Tipos de datos	Funcionamiento	Bit de	Alimentación/primer
CTU	CU incrementa el valor actual. El valor actual se incrementando hasta alcanzar	El bit de contaje se activa si el valor actual >= valor preselección	El bit de contaje está desactivado. El valor actual se conserva.
CTUD	CD decreuenta el valor CU incrementa el valor	El bit de contaje se activa si el valor actual >= valor preselección	El bit de contaje está desactivado. El valor actual se puede

	actual. El valor actual se sigue incrementando o decrementando		conservar. ¹
CTD	CD decrementa el valor actual hasta que alcance 0.	El bit de conteo se activa si el valor actual ≥ 0	El bit de conteo está desactivado. El valor actual se puede

¹ Es posible ajustar que se memorice el valor actual del contador. Para más información sobre el respaldo de la memoria de la CPU S7-200, consulte el capítulo 4.

Ejemplo de la operación Decrementar contador (SIMATIC)

Network 1

Network 2

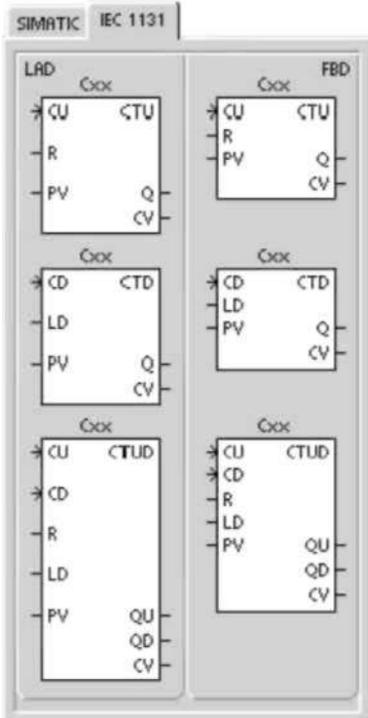
Network 1 //El contador atrás C1 cuenta
//de 3 a 0
//estando I0.1 OFF,
//I0.0 OFF-ON decrementa el valor actual de C1
//I0.1 ON carga el valor de preselección 3 de
//contaje atrás

LD I0.0
LD I0.1
CTD C1, +3

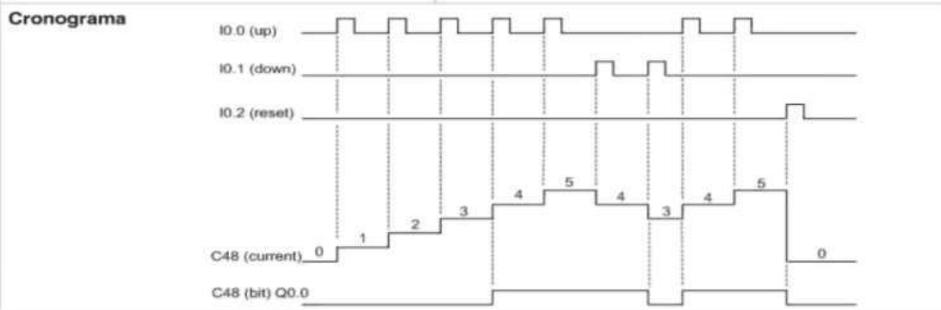
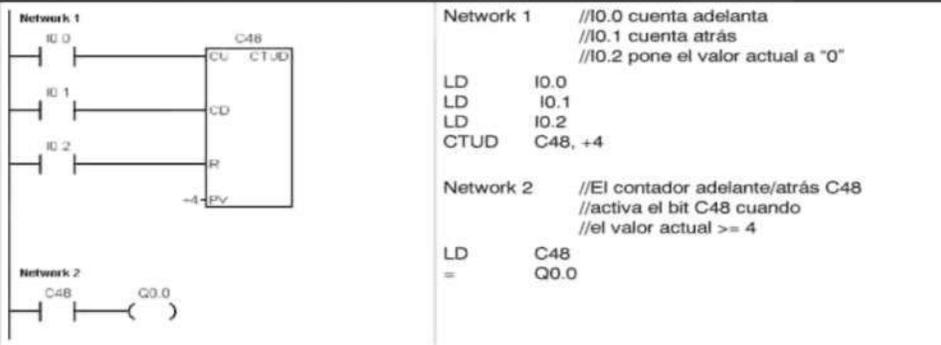
Network 2 //El bit C1 está ON cuando el valor actual de C1 = 0

LD C1
= Q0.0

Cronograma



Ejemplo de la operación Incrementar/decrementar contador (SIMATIC)



Operaciones de contaje (IEC)

Contador ascendente

La operación Contador ascendente (CTU) cuenta adelante desde el valor actual hasta el valor prefijado (PV) al producirse un flanco positivo en la entrada de contaje adelante (CU). Si el valor actual (CV) es mayor o igual al valor prefijado, se activa el bit de contaje (Q). El contador se inicializa al activarse la entrada de desactivación (R). El contador ascendente se detiene al alcanzar el valor prefijado.

Contador descendente

La operación Contador descendente (CTD) empieza a contar atrás desde el valor prefijado (PV) cuando se produce un flanco positivo en la entrada de contaje atrás (CD). Si el valor actual (CV) es igual a cero, se activa el bit de salida del contador (Q). El contador se inicializa y carga el valor actual en el valor prefijado cuando se habilita la entrada de carga (LD). El contador descendente se detiene al alcanzar el valor cero.

Contador ascendente/descendente

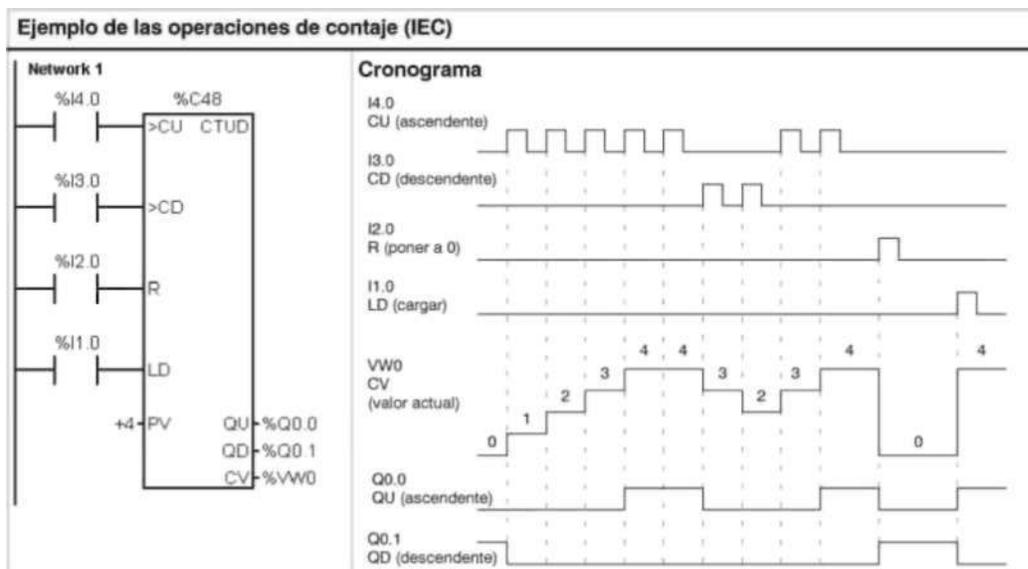
La operación Contador ascendente/descendente (CTUD) cuenta adelante o atrás desde el valor actual (CV) al producirse un flanco positivo en la entrada de contaje adelante (CU) o de contaje atrás (CD), respectivamente. La salida ascendente (QU) se activa cuando el valor actual (CV) es igual al valor prefijado. La salida descendente (QD) se activa cuando el valor actual (CV) es igual a cero. El contador carga el valor actual en el valor prefijado (PV) cuando se habilita la entrada de carga (LD). De forma similar, el contador se inicializa y carga el valor actual con 0 cuando se habilita la desactivación (R). El contador se detiene cuando alcanza el valor prefijado, o bien 0.

Tabla 6-24 Operandos válidos para las operaciones de contaje IEC

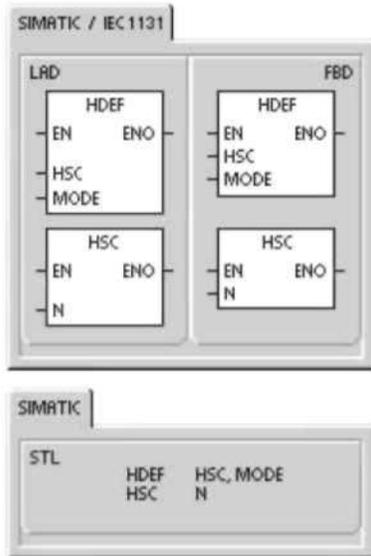
Entradas/salid	Tipos de datos	Operandos
Cxx		Constante (C0 a C255)
CU , CD , LD ,	BOOL	I, Q, V, M, SM, S, T, C, L, circulación
PV	INT	IW, QW, VW, MW, SMW, SW, LW, AC, *AC, AIW, *VD, *LD, constante
Q, QU, QD	BOOL	I, Q, V, M, SM, S, L
CV	INT	IW, QW, VW, MW, SW, LW, AC, *VD,

Consejo

Puesto que cada contador dispone sólo de un valor actual, no se puede asignar un mismo número a varios contadores. (Los contadores ascendentes, descendentes y ascendentes-descendentes acceden a un mismo valor actual).



Operaciones con contadores rápidos Definir modo para contador rápido



La operación Definir modo para contador rápido (HDEF) selecciona el modo de operación de un contador rápido en particular (HSCx). La selección de modo define el reloj, el sentido de contaje, así como las funciones de arranque y puesta a 0 del contador rápido.

Por cada contador sólo se puede ejecutar una operación Definir modo para contador rápido (HDEF).

Condiciones de error que ponen ENO a 0:

- 0003 (conflicto de entradas)
- 0004 (operación no válida en una interrupción)
- 000A (redefinición de HSC)

Activar contador rápido

La operación Activar contador rápido (HSC) configura y controla el funcionamiento del contador rápido direccionado, basándose en el estado de las marcas especiales del mismo. El parámetro N indica el número del contador rápido. Los contadores rápidos se pueden configurar para 12 modos de operación diferentes (v. tabla 6-26).

Todos los contadores disponen de entradas que soportan funciones tales como relojes, control del sentido, puesta a 0 y arranque. Para los contadores de dos fases, ambos relojes pueden funcionar a máxima frecuencia. Los contadores A/B permiten elegir una velocidad simple (1x) o cuádruple (4x) para el contaje. Todos los contadores funcionan a velocidades máximas sin interferir entre sí.

Condiciones de error que ponen ENO a 0:

- 0001 (HSC antes de HDEF)
- 0005 (operaciones HSC/PLS simultáneas)

Tabla 6-25 Operandos válidos para las operaciones con contadores rápidos



Ejemplos de programación

Entradas/salid	Tipos de	Operandos
HSC, MODE	BYTE	Constante
N	WORD	Constante

Los ejemplos de programación del CD de documentación contienen programas que utilizan contadores rápidos (consulte los ejemplos 4 y 29).

Los contadores rápidos cuentan eventos rápidos que no se pueden controlar durante el ciclo del S7-200. La frecuencia máxima de un contador rápido depende del modelo de CPU S7-200. Para más información al respecto, consulte el anexo A.

Consejo



Las CPUs 221 y 222 soportan los contadores de velocidad cuádruple. HSC0, HSC3, HSC4 y HSC5. Estas CPUs no soportan los contadores HSC1 y HSC2.

Las CPUs 224, 224XP y 226 soportan seis contadores rápidos: HSC0 a HSC5.

Los contadores rápidos se utilizan habitualmente como accionamiento para temporizadores que funcionan impulsados por un árbol que gira a un régimen constante y provisto de un encoder incremental. Éste último provee un número determinado de valores de contaje por giro, así como un impulso de puesta a "0" una vez por giro. El reloj (o relojes) y el impulso de puesta a "0" del encoder suministran las entradas para el contador rápido.

El primero de los valores predeterminados se carga en el contador y las salidas deseadas se activan para el intervalo de tiempo en que el valor actual del contador es menor que el valor predeterminado. El contador se ajusta para que una interrupción se active cuando el contaje actual sea igual al predeterminado o cuando el contador se ponga a 0.

Cuando el valor actual es igual al predeterminado y se presenta un evento de interrupción, entonces se carga un nuevo valor predeterminado y se activa el siguiente estado de señal para las salidas. Si se produce un evento de interrupción porque el contador se ha inicializado, entonces se ajusta el primer valor predeterminado y los primeros estados de las salidas, repitiéndose el ciclo. Puesto que las interrupciones se producen a una velocidad muy inferior a la de los contadores rápidos, es posible implementar un control preciso de las operaciones rápidas con un impacto relativamente bajo en el ciclo total del sistema de automatización. La posibilidad de asociar interrupciones a rutinas de interrupción permite cargar nuevos valores predeterminados en una rutina de interrupción separada, lo cual simplifica el control del estado. (Alternativamente, todos los eventos de interrupción se pueden ejecutar en una sola rutina de interrupción.)

Descripción de las operaciones con contadores rápidos

Todos los contadores funcionan de la misma manera en el mismo modo de operación. Hay cuatro tipos básicos de contadores, a saber: contadores de fase simple con control interno del sentido de contaje, contadores de fase simple con control externo del sentido de contaje, contadores de dos fases con 2 entradas de reloj, así como contadores A/B. Es preciso tener en cuenta que no todos los contadores soportan todos los modos. Todos los contadores se pueden utilizar sin entrada de puesta a 0 y sin entrada de arranque, con entrada de puesta a 0 pero sin entrada de arranque, o bien, con entrada de puesta a 0 y de arranque.

- Activando la entrada de puesta a 0 se borra el valor actual del contador hasta que esa entrada se desactive de nuevo.
- Cuando se activa la entrada de arranque, se habilita el contador. Si se desactiva esa entrada se mantiene el valor actual del contador, ignorándose los eventos de reloj.
- Si se activa la entrada de puesta a 0 mientras está desactivada la entrada del arranque, se ignorará la activación de la entrada de puesta a 0, con lo que no se modificará el valor actual. Si la entrada de arranque se activa mientras está activada la entrada de puesta a 0, el valor actual se borrará.

Antes de utilizar un contador rápido es preciso usar la operación HDEF (Definir modo para contador rápido) con objeto de seleccionar el modo del contador. Utilice la marca del primer ciclo SM0.1 (este bit se activa sólo en el primer ciclo y se desactiva posteriormente) para llamar a la subrutina que contiene la operación HDEF.

Programar un contador rápido



Asistente de operaciones

El asistente de operaciones HSC se puede utilizar para configurar el contador. El asistente utiliza las informaciones siguientes: tipo y modo del contador, valor predeterminado y valor actual del contador, así como sentido de contaje inicial. Para iniciar el asistente, elija el comando de menú **Herramientas > Asistente de operaciones** y, a continuación, seleccione el asistente HSC.

Para programar un contador rápido es preciso realizar las siguientes tareas básicas:

- Definir el contador y el modo.
- Ajustar el byte de control.
- Ajustar el valor actual (valor inicial).
- Ajustar el valor predeterminado (valor de destino).
- Asociar y habilitar la rutina de interrupción.
- Activar el contador rápido.

Definir el modo del contador y las entradas

Utilice la operación Definir modo para contador rápido para seleccionar el modo del contador y las entradas.

La tabla 6-26 muestra las entradas correspondientes al reloj, el control del sentido, la puesta a 0 y las funciones de arranque de los contadores rápidos. 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 contador rápido se puede usar para otro fin. Por ejemplo, si HSC0 se está utilizando en modo 1 (que utiliza las entradas I0.0 e I0.2), I0.1 se podrá utilizar para interrupciones de flanco o para HSC3.



Consejo

Tenga en cuenta que todos los modos de HSC0 (excepto el modo 12) utilizan siempre I0.0 y que todos los modos de HSC4 usan siempre I0.3. Por tanto, estas entradas nunca estarán disponibles para otros fines cuando se estén utilizando esos contadores.

Tabla 6-26 Entradas de los contadores rápidos

Mode	Descripción	Entradas			
	HSC0	I0.0	I0.1	I0.2	
	HSC1	I0.6	I0.7	I1.0	I1.1
	HSC2	I1.2	I1.3	I1.4	I1.5
	HSC3	I0.1			
	HSC4	I0.3	I0.4	I0.5	
	HSC5	I0.4			
0	Contador de fase simple con control interno del sentido de contaje	Reloj			
1	Contador de fase simple con control externo del sentido de contaje	Reloj		Puesta a	Arranque
2		Reloj		Puesta a	
3	Contador de dos fases con 2 entradas de reloj	Reloj	Sentido		Arranque
4		Reloj	Sentido	Puesta a	
5		Reloj	Sentido	Puesta a	
6	Contador A/B	Reloj	Reloj		Arranque
7		adelante	Reloj	Puesta a	
8		Reloj	atrás	0 Puesta	Arranque
9	Contador A/B	Reloj A	Reloj B		Arranque
10		Reloj A	Reloj B	Puesta a	
11		Reloj A	Reloj B	Puesta a	

12 HSC0 y HSC3 son los únicos que soportan el modo 12.
 HSC0 cuenta el número de impulsos que salen de Q0.0.
 HSC3 cuenta el número de impulsos que salen de Q0.1.

Modos de los contadores rápidos (ejemplos)

Los siguientes cronogramas (figuras 6-22 a 6-26) muestran cómo cada contador funciona conforme a su categoría.

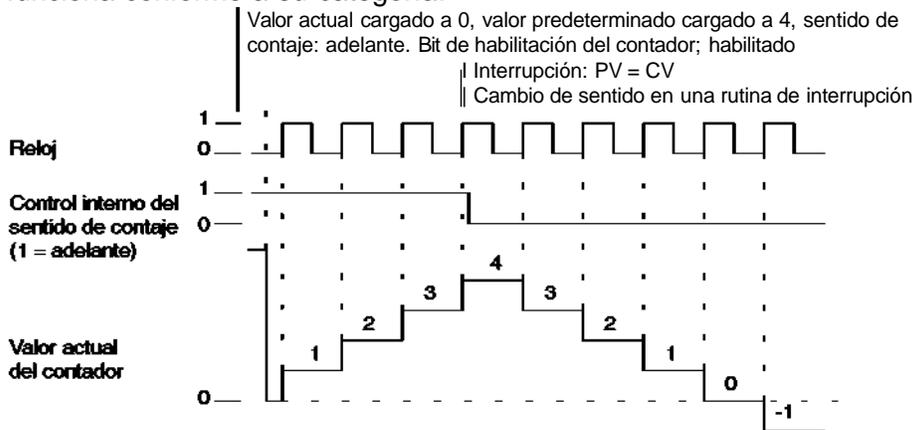


Figura 6-22 Ejemplo del funcionamiento de los modos 0, 1 ó 2

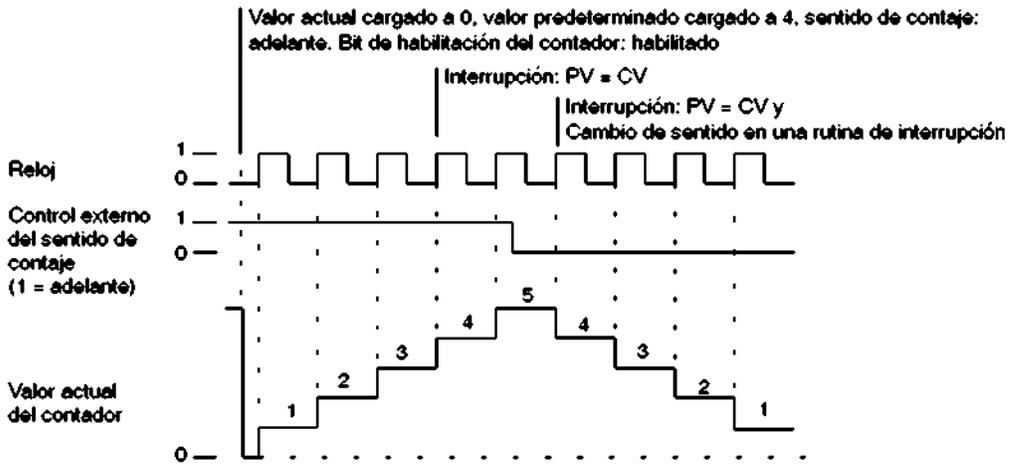


Figura 6-23 Ejemplo del funcionamiento de los modos 3, 4 ó 5

Si se utilizan los modos de conteo 6, 7 u 8 y ocurre un flanco positivo tanto en la entrada de reloj de conteo adelante como en la de conteo atrás en menos de 0,3 microsegundos, puede ser que el contador rápido considere simultáneos ambos eventos. En este caso, el valor actual permanecerá inalterado y tampoco cambiará el sentido de conteo. Si entre el flanco positivo de la entrada de conteo adelante y el flanco positivo de la entrada de conteo atrás transcurren más de 0,3 microsegundos, el contador rápido recibirá ambos eventos por separado. En ninguno de ambos casos se produce un error.

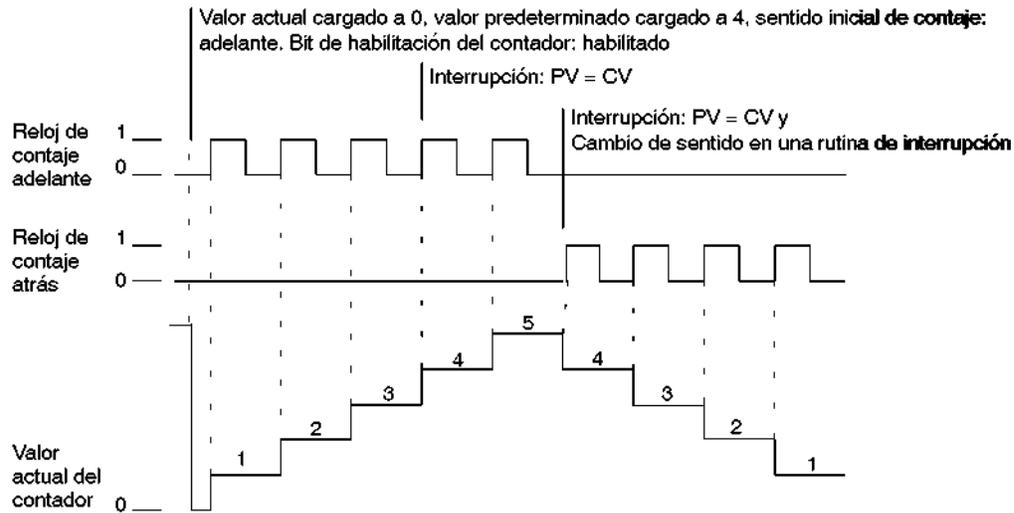


Figura 6-24 Ejemplo del funcionamiento de los modos 6, 7 u 8
 Valor actual cargado a 0, valor predeterminado cargado a 3, sentido inicial de conteo: adelante. Bit de habilitación del contador: habilitado

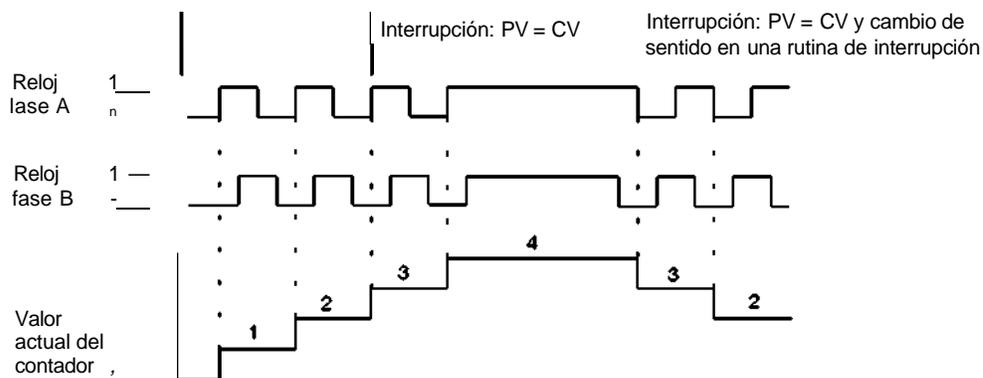
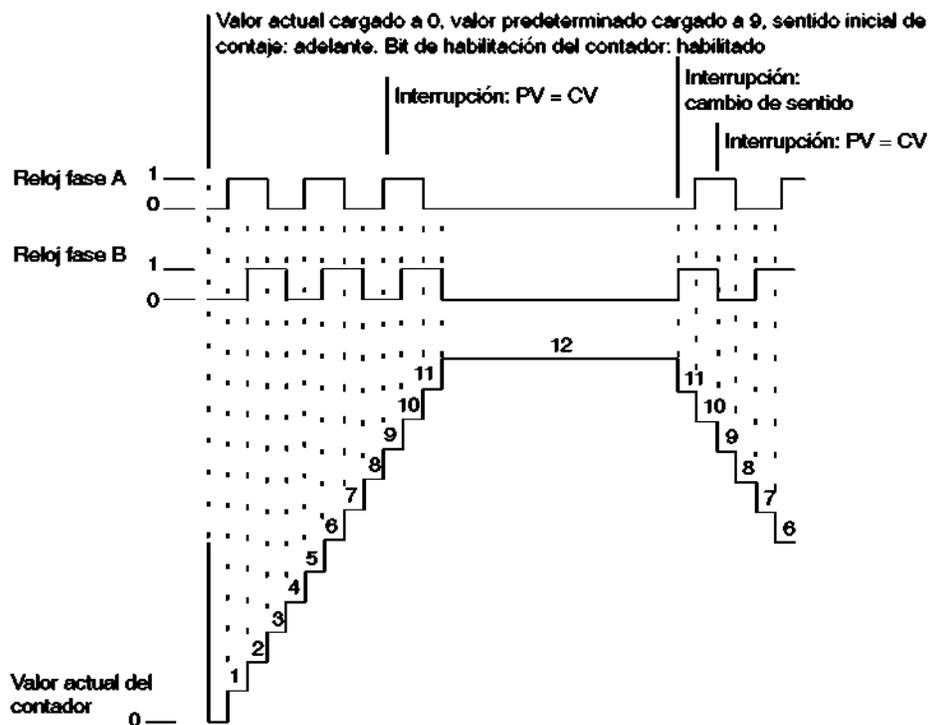


Figura 6-25 Ejemplo del funcionamiento de los modos 9, 10 u 11 (contador A/B, velocidad simple)



Cuatro contadores tienen tres marcas de control que se utilizan para configurar el estado activo de las entradas de puesta a 0 y de arranque, así como para seleccionar la velocidad simple o cuádruple (ésto sólo en los contadores A/B). Estas marcas están depositadas en el byte de control del respectivo contador y se emplean solamente cuando se ejecuta la operación HDEF. La tabla 6-27 muestra esas marcas.

Consejo

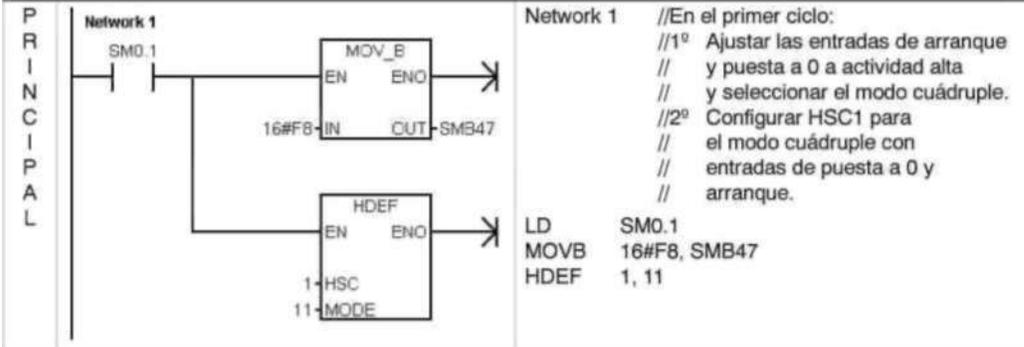
Antes de poder ejecutar la operación HDEF es preciso ajustar las tres marcas de control al estado deseado. De lo contrario, el contador adoptará la configuración predeterminada del modo de conteaje elegido.

Una vez ejecutada la operación HDEF, ya no se podrá modificar el ajuste de los contadores, a menos que el S7-200 cambie a modo STOP.

Tabla 6-27 Nivel de actividad de las entradas de puesta a 0 y de arranque, marcas para elegir la velocidad simple o cuádruple

HSC0	HSC2	HSC4	HSC147	Descripción (sólo cuando se ejecuta HDEF)
SM37.	SM47.	SM57.	SM147	Bit de control de nivel de actividad para puesta a 0:
0	0	0	0	0 = entrada de puesta a 0 (actividad alta)
---	SM47.	SM57.	---	Bit de control de nivel de actividad para arranque:
	1	1		1 = entrada
SM37.	SM47.	SM57.	SM147	Velocidad de conteaje de los contadores A/B:
2	2	2	2	0 = <u>velocidad cuádruple</u> 1 = <u>velocidad simple</u>
1	El ajuste estándar de las entradas de puesta a 0 y de arranque es de actividad alta, y la velocidad de conteaje es la cuádruple (es decir, la frecuencia del reloj de entrada multiplicada por cuatro).			

Ejemplo de la operación Definir modo para contador rápido



Ajustar el byte de control

Una vez definido el contador y el modo de contaje es preciso programar los parámetros dinámicos del mismo. Cada contador rápido tiene un byte de control que permite realizar las siguientes acciones:

- Habilitar o inhibir el contador.
- Controlar el sentido (sólo en los modos 0, 1 y 2), o bien el sentido de contaje inicial (en los demás modos).
- Cargar el valor actual.
- Cargar el valor predeterminado.

El byte de control, los valores actuales asignados y los valores predeterminados se comprueban al ejecutarse la operación HSC. La tabla 6-28 describe cada una de las marcas del byte de control.

Tabla 6-28 Bits de estado para HSC0, HSC1, HSC2, HSC3, HSC4 y HSC5

HSC0		HSC2		HSC3		HSC4		Descripción	
SM37.3		SM57		Bit de control para el sentido de contaje: SM137.3					
SM47.3		.3		SM147.3		SM157.3			
SM37	SM47	SM57	SM137	SM147	SM157	Escribir el sentido de contaje en			
.4	.4	.4	.4	.4	.4	el contador rápido:			
						0 = no actualizar 1 =			
SM37	SM47	SM57	SM137	SM147	SM157	Escribir el nuevo valor			
.5	.5	.5	.5	.5	.5	predeterminado en el contador			
						rápido:			
SM37	SM47	SM57	SM137	SM147	SM157	Escribir el nuevo valor actual en			
.6	.6	.6	.6	.6	.6	el contador rápido:			
						0 = no actualizar 1 =			
SM37	SM47	SM57	SM137	SM147	SM157	Habilitar el contador rápido:			
.7	.7	.7	.7	.7	.7	0 = Inhibir el contador rápido			
						1 = Habilitar el contador rápido			

Ajustar los valores actuales y predeterminados

Todos los contadores rápidos disponen de un valor actual y de un valor predeterminado de 32 bits. Ambos son valores enteros con signo. Para cargar un nuevo valor actual o predeterminado en el contador rápido es preciso ajustar el byte de control y los bytes de marcas que contienen los valores actuales y/o predeterminados, así como ejecutar la operación HSC (Activar contador rápido) para transferir los nuevos valores al contador rápido. En la tabla 6-29 figuran los bytes de marcas especiales que contienen los nuevos valores y los valores predeterminados.

Además de los bytes de control y de los bytes que contienen los nuevos valores predeterminados, también se puede leer el valor actual de cada uno de los contadores rápidos, utilizando el tipo de datos HC (valor actual del contador rápido) seguido del número de contador (0, 1, 2, 3, 4 ó 5), como muestra la tabla 6-29. Ello permite acceder directamente al valor actual para operaciones de lectura. Por el contrario, este valor sólo se puede escribir utilizando la operación HSC.

Tabla 6-29 Nuevos valores actuales y predeterminados de los contadores HSC0, HSC1, HSC2, HSC3, HSC4 y HSC5

Valor a cargar	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Nuevo valor actual	SMD38	SMD48	SMD58	SMD138	SMD14	SMD15
Nuevo valor	SMD42	SMD52	SMD62	SMD142	SMD15	SMD16

Tabla 6-30 Valores actuales de los contadores HSC0, HSC1, HSC2, HSC3, HSC4 y HSC5

Valor	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Valor actual	HC0	HC1	HC2	HC3	HC4	HC5

SM36	SM46.	SM56.	SM136	SM146	SM156	El valor actual es igual al bit de estado del valor predeterminado:
.6	6	6	.6	.6	.6	0 = diferente
SM36	SM46.	SM56.	SM136	SM146	SM156	El valor actual es mayor que el bit de estado del valor predeterminado:
.7	7	7	.7	.7	.7	0 = menor o igual
						1 = mayor que

Ejemplos de secuencias de inicialización de los contadores rápidos

HSC1 se utiliza como contador modelo en las descripciones siguientes de las secuencias de inicialización y funcionamiento. En las explicaciones acerca de la inicialización se supone que el sistema de automatización S7-200 se encuentra en modo RUN y que, por consiguiente, la marca del primer ciclo es verdadera. En otro caso se deberá tener en cuenta que la operación HDEF sólo puede ejecutarse una vez por cada contador rápido, tras haber cambiado a modo RUN. Si la operación HDEF se ejecuta por segunda vez en un contador rápido, se producirá un error en el tiempo de ejecución. Los ajustes del contador permanecerán entonces tal y como se configuraron con la primera operación HDEF que se ejecutó para el contador en cuestión.



"~" \ Consejo

¡ij A continuación se describe cómo modificar individualmente el sentido de contaje, el valor actual o el valor predeterminado de los contadores rápidos. No obstante, todos o sólo algunos de estos ajustes se pueden cambiar también en ese mismo orden, definiendo el valor de SMB47 de forma apropiada y ejecutando luego la operación HSC.

Modos de inicialización 0, 1 ó 2

Para inicializar HSC1 como contador adelante/atrás de fase simple con control interno del sentido de contaje (modos 0, 1 ó 2), proceda de la manera siguiente:

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada. Por ejemplo,
SMB47 = 16#F8 *Resultados:*
Se habilita el contador.
Se escribe un nuevo valor actual.
Se escribe un nuevo valor predeterminado.
Se ajusta el sentido de contaje adelante.
Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.
3. Ejecute la operación HDEF con la entrada HSC puesta a "1" y la entrada MODE con uno de los ajustes siguientes: "0" para puesta a 0 o arranque no externos, "1" para puesta a 0 externa y sin arranque, o bien "2" para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).

6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción. En el apartado "Operaciones de interrupción" se describe detalladamente cómo procesar las interrupciones.
7. Para poder detectar una puesta a "0" externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
8. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
9. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
10. Finalice la subrutina.

Modos de inicialización 3, 4 ó 5

Para inicializar HSC1 como contador adelante/atrás de fase simple con control externo del sentido de conteo (modos 3, 4 ó 5), proceda de la manera siguiente:

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada. Por ejemplo,
SMB47 = 16#F8 Resultados:
 Se habilita el contador.
 Se escribe un nuevo valor actual.
 Se escribe un nuevo valor predeterminado.
 Se ajusta el sentido inicial de conteo adelante.
 Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.
3. Ejecute la operación HDEF con la entrada HSC puesta a "1" y la entrada MODE con uno de los ajustes siguientes: "3" para puesta a 0 o arranque no externos, "4" para puesta a 0 externa y sin arranque, o bien "5" para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción. En el apartado "Operaciones de interrupción" se describe detalladamente cómo procesar las interrupciones.
7. Para poder detectar un cambio del sentido de conteo, programe una interrupción asociando el evento de interrupción de cambio de sentido (evento 14) a una rutina de interrupción.
8. Para poder detectar una puesta a "0" externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
9. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
10. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
11. Finalice la subrutina.

Modos de inicialización 6, 7 u 8

Para inicializar HSC1 como contador adelante/atrás de dos fases con relojes adelante/atrás (modos 6, 7 u 8), proceda de la manera siguiente:

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada. Por ejemplo,
SMB47 = 16#F8 Resultados:
 Se habilita el contador.
 Se escribe un nuevo valor actual.
 Se escribe un nuevo valor predeterminado.
 Se ajusta el sentido inicial de contaje adelante.
 Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.
3. Ejecute la operación HDEF con la entrada HSC puesta a "1" y la entrada MODE con uno de los ajustes siguientes: "6" para puesta a 0 o arranque no externos, "7" para puesta a 0 externa y sin arranque, o bien "8" para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción. En el apartado "Operaciones de interrupción" se describe detalladamente cómo procesar las interrupciones.
7. Para poder detectar un cambio del sentido de contaje, programe una interrupción asociando el evento de interrupción de cambio de sentido (evento 14) a una rutina de interrupción.
8. Para poder detectar una puesta a "0" externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
9. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
10. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
11. Finalice la subrutina.

Modos de inicialización 9, 10 u 11

Para inicializar HSC1 como contador A/B (modos 9, 10 u 11), proceda de la manera siguiente:

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB47 en la subrutina de inicialización conforme a la operación de control deseada.

Ejemplo (frecuencia simple):

SMB47 = 16#FC

Resultados:

Se habilita el contador.

Se escribe un nuevo valor actual.

Se escribe un nuevo valor predeterminado.

Se ajusta el sentido inicial de contaje adelante.

Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.

Ejemplo (frecuencia cuádruple):

SMB47 = 16#F8

Resultados:

Se habilita el contador.

Se escribe un nuevo valor actual.

Se escribe un nuevo valor predeterminado.

Se ajusta el sentido inicial de contaje adelante.

Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.

3. Ejecute la operación HDEF con la entrada HSC puesta a "1" y la entrada MODE con uno de los ajustes siguientes: "9" para puesta a 0 o arranque no externos, "10" para puesta a 0 externa y sin arranque, o bien "11" para puesta a 0 y arranque externos.
4. Cargue el valor actual deseado en SMD48 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD52 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción. En el apartado "Operaciones de interrupción" se describe detalladamente cómo procesar las interrupciones.
7. Para poder detectar un cambio del sentido de contaje, programe una interrupción asociando el evento de interrupción de cambio de sentido (evento 14) a una rutina de interrupción.
8. Para poder detectar una puesta a "0" externa, programe una interrupción asociando el evento de interrupción Puesta a 0 externa (evento 15) a una rutina de interrupción.
9. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
10. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.
11. Finalice la subrutina.

Modo de inicialización 12

Para inicializar HSC0 con objeto de contar impulsos generados por PTO0 (modo 12), proceda de la manera siguiente:

1. Con la marca del primer ciclo, llame a una subrutina para ejecutar la inicialización. Puesto que se utiliza una llamada a subrutina, los siguientes ciclos ya no llaman a la misma, con lo cual se acorta el tiempo de ciclo y el programa queda mejor estructurado.
2. Cargue la marca SMB37 en la subrutina de inicialización conforme a la operación de control deseada. Por ejemplo,
SMB37 = 16#F8 Resultados:
 Se habilita el contador.
 Se escribe un nuevo valor actual.
 Se escribe un nuevo valor predeterminado.
 Se ajusta el sentido de contaje adelante.
 Se ajusta la actividad alta de las entradas de arranque y de puesta a 0.
3. Ejecute la operación HDEF con la entrada HSC puesta a "0" y la entrada MODE puesta a "12".
4. Cargue el valor actual deseado en SMD38 (valor de palabra doble) (cargue 0 para borrar la marca).
5. Cargue el valor predeterminado deseado en SMD42 (valor de palabra doble).
6. Para averiguar si el valor actual es igual al predeterminado, programe una interrupción asociando el evento de interrupción CV = PV (evento 13) a una rutina de interrupción. En el apartado "Operaciones de interrupción" se describe detalladamente cómo procesar las interrupciones.
7. Ejecute la operación Habilitar todos los eventos de interrupción (ENI) para habilitar las interrupciones.
8. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC0.
9. Finalice la subrutina.

Cambiar el sentido en los modos 0, 1, 2 ó 12

Para configurar el cambio de sentido de HSC1 como contador de fase simple con control interno del sentido de contaje (modos 0, 1, 2 ó 12), proceda de la manera siguiente:

1. Cargue SMB47 para escribir la dirección deseada:
SMB47 = 16#90 Habilita el contador.
 Ajusta el sentido de contaje atrás.
SMB47 = 16#98 Habilita el contador.
 Ajusta el sentido de contaje adelante.
2. Ejecute la operación HSC para que el sistema de automatización S7-200 programe el contador HSC1.

Ejemplo de la operación Activar contador rápido

```

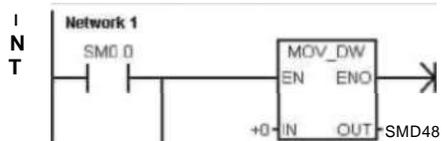
P
R
I
N
C
I
P
A
L
S
B
R

```

Network 1	SMB 0	SBR 0	Network 1	//Llamar a SBRO en el primer ciclo.
			LD	SMB.1
			CALL	SBR 0
Nettmrk 1		MOV_B	Network 1	//En el primer ciclo, configurar HSC1:
				//1 - Habilitar el contador.
		IN		// - Escribir un nuevo valor actual.
				// - Escribir un nuevo valor predeterminado.
		EM		// - Ajustar el sentido inicial de contaje adelante.
				// - Ajustar las entradas de arranque y puesta a 0 a actividad alta.
				// - Seleccionar el modo cuádruple.
		1-HSC		//2º Configurar HSC1 para el modo cuádruple
		11 MOQE		// con entradas de puesta a 0 y arranque.
				//3º Borrar el valor actual de HSC1.
		MOV_DW		//4º Ajustar a 50 el valor predeterminado de HSC1.
		EKI ENC		//5º Si el valor actual de HSC1 = valor de
				// preselección, asociar el evento 13 a la rutina de
				// interrupción INT_0.
		OUT 3MDdB		//6º Habilitar todos los eventos de interrupción.
				//7º Programar HSC 1.
		MOV.DW	LD	SMB.1
		EN EMO ^	MOVB	16#F8, SMB47
			HDEF	1, 11
		160 UU-3MJ52	MOVD	+0, SMD48
			MOVD	+50. SMD52
			ATCH	INT_0, 13
		AT;H	ENI	
		fcN ENa	HSC	1
		INT 0- INT		

ENI)

HSC
EN ENO



```

Network 1 //Programar HSC1:
//1' Borrar el valor actual de HSC1.
//2º Seleccionar que se escriba sólo un nuevo
// valor actual y dejar HSC1 habilitado.

```

```

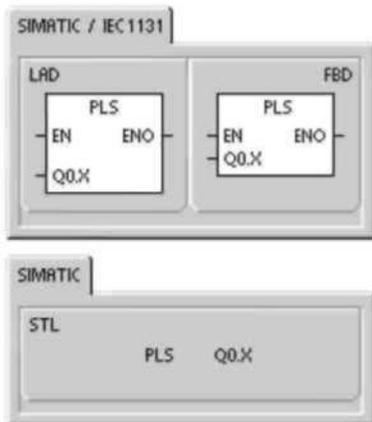
LD SMB.0
MOVD +0, SMD48
MOVB 16#C0, SMB47
HSC 1

```

MOV_E
EN ENO

16#C0-IN QIIT -5MB47

HSC
EN ENO _ ^



Salida de impulsos

La operación Salida de impulsos (PLS) se utiliza para controlar las funciones Tren de impulsos (PTO) y Modulación por ancho de impulsos (PWM) disponibles en las salidas rápidas (Q0.0 y Q0.1).

**1 El asistente de control de posición mejorado ofrece la

Sj\$b posibilidad de crear operaciones personalizadas para la

^^^ aplicación que simplifican las tareas de programación,

te ••Control permitiendo aprovechar las nuevas funciones de las posición

CPUs S7-200. Para más información acerca del asistente de control de posición, consulte el capítulo 9.

La operación PLS antigua se puede seguir utilizando para crear aplicaciones de movimiento. No obstante, sólo las operaciones creadas con el asistente de control de posición mejorado soportan la rampa lineal del PTO.

PTO ofrece una onda cuadrada (con un ciclo de trabajo de 50%), pudiendo el usuario controlar el tiempo de ciclo y el número de impulsos.

PWM ofrece una salida continua con un ciclo de trabajo variable, pudiendo el usuario controlar el tiempo de ciclo y el ancho de impulsos.

El S7-200 dispone de dos generadores PTO/PWM que crean trenes de impulsos rápidos, o bien formas de onda moduladas por ancho de impulsos. Uno de estos generadores está asignado a las salida digital Q0.0 y, el otro, a la salida digital Q0.1. Una marca especial (SM) almacena los siguientes datos para cada uno de los generadores: un byte de control (8 bits), un valor de conteo de impulsos (valor de 32 bits sin signo), así como un valor de tiempo de ciclo y un valor de ancho de impulsos (valores de 16 bits sin signo).

Los generadores PTO/PWM y la imagen del proceso comparten el uso de las salidas Q0.0 y Q0.1. Cuando está activa una función PTO o PWM en Q0.0 o Q0.1, el generador PTO/PWM controla la salida e inhibe el uso normal de la misma. La forma de onda de la salida no se ve afectada por el estado de la imagen del proceso, ni por el valor forzado de la salida o la ejecución de las operaciones directas de salida. Si el generador PTO/PWM está desactivado, el control de la salida retornará a la imagen del proceso. Ésta determina los estados inicial y final de la forma de onda de la salida, para que la forma de onda comience y termine en un nivel alto o bajo.

Tabla 6-32 Operandos válidos para la operación Salida de impulsos

Entradas/salida	Tipos de	Operandos
Q0.X	WORD	Constante: 0 (= o bien, 1 (= Q0.1)

Consejo



Antes de habilitar las operaciones PTO o PWM, ajuste a "0" el valor de la imagen del proceso de Q0.0 y Q0.1.

El ajuste estándar de los bits de control, del tiempo de ciclo, del ancho de impulsos y de los valores de conteo es 0.

Las salidas PTO/PWM requieren una carga mínima de 10% de la carga nominal para poder garantizar transiciones idóneas de "off" a "on", y viceversa.

Los ejemplos de programación del CD de documentación contienen programas que utilizan la operación PLS para las funciones PTO/PWM (consulte los ejemplos 7, 22, 23, 30 y 50).



Ejemplos de programación

Tren de impulsos (PTO)

PTO ofrece una salida en cuadratura (con un ciclo de trabajo de 50%) para un número determinado de impulsos y un tiempo de ciclo determinado (v. fig. 6-29) La función Tren de impulsos (PTO) puede producir uno o varios trenes de impulsos (utilizando un perfil de impulsos). El número de impulsos y el tiempo de ciclo pueden indicarse en incrementos en microsegundos o milisegundos.

- Número de impulsos: 1 a 4.294.967.295
- Tiempo de ciclo: 10 μ s a 65.535 μ s, o bien 2 ms a 65.535 ms.



Un número impar de microsegundos o milisegundos del tiempo de ciclo (por ejemplo, **75 ms**) **causa una distorsión del ciclo de trabajo.**

Figura 6-29 Tren de impulsos (PTO)

En la tabla 6-33 figuran los límites del conteo de impulsos y del tiempo de ciclo. Tabla 6-33 Contaje de impulsos y tiempo de ciclo en la función PTO

Contaje de	Reacción
Tiempo de ciclo < 2 unidades de tiempo	El tiempo de ciclo se ajusta a 2 unidades de tiempo.
Contaje de impulsos = 0	El contaje de impulsos se ajusta a 1 impulso.

La operación PTO permite efectuar un “pipelining”, es decir, concatenar o canalizar los trenes de impulsos. Tras haber finalizado el tren de impulsos activo, comenzará inmediatamente la salida de un nuevo tren de impulsos. Así se garantiza la continuidad de los trenes de impulsos subsiguientes.

Utilizar el asistente de control de posición

El asistente de control de posición procesa automáticamente el pipelining monosegmento y multisegmento de los impulsos PTO, la modulación por ancho de impulsos y la configuración de las marcas especiales. Además, crea una tabla de perfiles. La información contenida aquí se suministra a título de referencia. Es recomendable que utilice el asistente de control de posición. Para más información acerca de este asistente, consulte el capítulo 9.

Pipelining monosegmento de impulsos PTO

En este modo, el usuario debe actualizar las direcciones de las marcas especiales para el siguiente tren de impulsos. Tras haber arrancado el segmento PTO inicial, el usuario debe modificar inmediatamente las direcciones de las marcas especiales conforme a lo requerido para la segunda forma de onda, y ejecutar nuevamente la operación PLS. Los atributos del segundo tren de impulsos se conservan en una “pipeline” hasta finalizar el primer tren de impulsos. En la “pipeline” sólo se puede almacenar un registro. Tras haber finalizado el primer tren de impulsos, comenzará la salida de la segunda forma de onda y la “pipeline” quedará libre para acoger un nuevo tren de impulsos. Este proceso se puede repetir entonces para ajustar las características del siguiente tren de impulsos.

Entre los trenes de impulsos se producen transiciones suaves, a menos que cambie la base de tiempo o si el tren de impulsos activo finaliza antes de que un nuevo tren de impulsos sea capturado al ejecutarse la operación PLS.

Pipelining multisegmento de impulsos PTO

En este modo, el S7-200 lee automáticamente las características de todos los trenes de impulsos en una tabla de perfiles almacenada en la memoria de variables (memoria V). Las marcas especiales utilizadas en este modo son el byte de control, el byte de estado y el offset inicial de la tabla de perfiles en la memoria V (SMW168 ó SMW178). La base de tiempo se puede indicar en microsegundos o milisegundos, pero la selección será aplicable a todos los valores de tiempo de ciclo en la tabla de perfiles, no pudiendo modificarse durante la ejecución del perfil. Al ejecutarse la operación PLS se inicia el modo multisegmento.

Todos los registros de segmentos tienen una longitud de 8 bytes, comprendiendo un valor de tiempo de ciclo de 16 bits, un valor delta de tiempo de ciclo de 16 bits y un valor de conteo de impulsos de 32 bits. La tabla 6-34 muestra el formato de la tabla de perfiles. El tiempo de ciclo se puede incrementar o decrementar automáticamente programando una cantidad determinada para cada impulso. Este tiempo se prolonga o se acorta programando en el campo delta un valor positivo o negativo, respectivamente, y permanece inalterado si se programa el valor "0".

Mientras se está ejecutando el perfil PTO, el número del segmento activo actualmente se indica en SMB166 (ó SMB176).

Tabla 6-34 Formato de la tabla de perfiles para la función PTO multisegmento

Offset de	Segmen	Descripción de los registros de la tabla
0		Número de segmentos: 1 a 255 ¹
1	#1	Tiempo de ciclo inicial (2 a 65.535 unidades de la base
3		Tiempo de ciclo delta por impulso (valor con signo) (-32.768 a 32.767 unidades de la base de tiempo)
5		Contaje de impulsos (1 a 4.294.967.295)
9	#2	Tiempo de ciclo inicial (2 a 65.535 unidades de la base
11		Tiempo de ciclo delta por impulso (valor con signo) (-32.768 a 32.767 unidades de la base de tiempo)
13		Contaje de impulsos (1 a 4.294.967.295)
(Continúa)	#3	(Continúa)

¹ Si se introduce el valor 0 para el número de segmentos, se producirá un error no fatal y no se generará una salida PTO.

Modulación por ancho de impulsos (PWM)

La función PWM ofrece un tiempo de ciclo fijo con un ciclo de trabajo variable (v. fig. 6-30). El tiempo de ciclo y el ancho de impulsos pueden indicarse en incrementos de microsegundos o milisegundos.

- Tiempo de ciclo: 10 μ s a 65.535 μ s, o bien 2 ms a 65.535 ms

Tiempo de ciclo J

Tiempo de ancho de impulsos Tiempo de ancho de impulsos

Figura 6-30 Modulación por ancho de impulsos (PWM)

- Tiempo de ancho de impulsos: 0 μ s a 65.535 μ s, o bien
0 ms a 65.535 ms

Como muestra la tabla 6-35, si el ancho de impulsos se ajusta a un valor igual al tiempo de ciclo (con lo que el ciclo de trabajo es de 100 por ciento), la salida permanecerá activada continuamente. Si el ancho de impulsos es cero, el ciclo de trabajo (relación impulso-pausa) será de 0 por ciento y se desactivará la salida.

Tabla 6-35 Tiempo de ancho de impulsos y tiempo de ciclo y su efecto en la función PWM

Tiempo de ancho de impulsos / tiempo de	Reacción
Tiempo de ancho de impulsos \geq valor de tiempo de ciclo	Ciclo de trabajo = 100%: la salida permanece activada continuamente.
Tiempo de ancho de impulsos = 0	Ciclo de trabajo = 0%: se desactiva la salida.
Tiempo de ciclo $<$ 2 unidades de tiempo	El tiempo de ciclo se ajusta a 2 unidades de tiempo

Hay dos maneras diferentes de cambiar las características de una forma de onda PWM:

- Actualización síncrona: Si no es necesario cambiar la base de tiempo, se utiliza la actualización síncrona. En este caso, el cambio de la forma de onda se efectúa en el límite de un ciclo, ofreciendo una transición suave.
- Actualización asíncrona: Generalmente, el ancho de impulsos cambia en tanto que el tiempo de ciclo permanece constante en la función PWM, por lo que no es necesario cambiar la base de tiempo. No obstante, si es preciso modificar la base de tiempo del generador PTO/PWM, se utiliza una actualización asíncrona. Ésta inhibe el generador PTO/PWM instantáneamente, de forma asíncrona a la forma de onda PWM. Ello puede provocar fluctuaciones no deseadas en los equipos controlados. Por tanto, se recomienda la actualización síncrona de las formas de ondas PWM. Elija una base de tiempo que se adecúe para todos los valores previstos para el tiempo de ciclo.

Consejo

El bit de actualización PWM (SM67.4 ó SM77.4) en el byte de control indica el

tipo de actualización utilizado cuando la operación PLS se ejecuta para invocar cambios.

Si modifica la base de tiempo se efectuará en todo caso una actualización asíncrona, sin importar el estado de esta marca especial.

Utilizar marcas especiales para configurar y controlar las operaciones PTO/PWM

La operación PLS lee los datos almacenados en las marcas especiales indicadas y

programa el generador PTO/PWM conforme a ello. SMB67 controla PTO 0 ó PWM 0, en tanto que SMB77 controla PTO 1 ó PWM 1. En la tabla 6-36 se describen los registros utilizados para controlar las funciones PTO/PWM. Mediante la tabla 6-37 es posible determinar rápidamente el valor que debe depositarse en el registro de control PTO/PWM para solicitar la operación deseada.

Para cambiar las características de una forma de onda PTO o PWM es preciso modificar las direcciones en el área de marcas especiales (incluyendo el byte de control) y ejecutar luego la operación PLS. La generación de una forma de onda PTO o PWM se puede inhibir en cualquier momento, poniendo a "0" el bit de habilitación PTO/PWM del byte de control (SM67.7 ó SM77.7) y ejecutando luego la operación PLS.

El bit de inactividad PTO (SM66.7 ó SM76.7) en el byte de estado indica si se ha completado el tren de impulsos programado. Además, tras finalizarse un tren de impulsos se puede llamar a una rutina de interrupción. (Consulte a este respecto las descripciones de las operaciones de comunicación e interrupción.) Si se utiliza el pipelining multisegmento, la rutina de interrupción se invoca tras completarse la tabla de perfiles.

Las siguientes condiciones activan SM66.4 (ó SM76.4) y SM66.5 (ó SM76.5):

- Si indica un valor delta del tiempo de ciclo que produzca un tiempo de ciclo no válido al cabo de un determinado número de impulsos, se generará un desbordamiento aritmético que finalizará la función PTO, con lo cual la marca de error de cálculo delta (SM66.4 ó SM76.4) se pondrá a 1. La salida retornará al control de la imagen del proceso.
- Si un perfil PTO que se esté ejecutando se interrumpe manualmente, la marca de interrupción anormal por el usuario (SM66.5 ó SM76.5) se pondrá a 1.
- Si intenta cargar la "pipeline" mientras está llena, la marca de desbordamiento PTO (SM66.6 ó SM76.6) se pondrá a 1. Para poder detectar los desbordamientos siguientes, esta marca se deberá poner a "0" manualmente tras haberse detectado un desbordamiento. Al pasar a RUN, la marca se vuelve a poner a 0.

Consejo

Si desea cargar un nuevo valor de contaje (SMD72 ó SMD82), de ancho de impulsos

(SMW70 ó SMW80), o bien de tiempo de ciclo (SMW68 ó SMW78), deberá cargar tanto estos valores como el registro de control antes de ejecutar la operación PLS. Si desea utilizar la función PTO multisegmento, también deberá cargar el offset inicial (SMW168 ó SMW178) de la tabla de perfiles y los valores de ésta antes de ejecutar la operación PLS.

Tabla 6-36 Marcas especiales de los registros de control PTO / PWM

Q0.0	Q0.1	Bits de estado	
SM66.4	SM76.4	Interrupción anormal del perfil PTO (error de cálculo delta): 0=sin error	
SM66.5	SM76.5	Interrupción anormal del perfil PTO causada por el usuario: 0 = sin	
SM66.6	SM76.6	Desbordamiento positivo/negativo del "pipeline" PTO: 0 = sin desbordamiento 1 = desbordamiento	
SM66.7	SM76.7	PTO inactivo:	0=PTO activo 1=PTO
Q0.0	Q0.1	Bits de control	
SM67.0	SM77.0	Actualizar el tiempo de ciclo PTO/PWM:	0 = no actualizar
SM67.1	SM77.1	Actualizar el tiempo de ancho de impulsos PWM: 0=noactualizar	
SM67.2	SM77.2	Actualizar el valor de conteo de impulsos de PTO: 0 = no actualizar	
SM67.3	SM77.3	Elegir la base de tiempo PTO/PWM:	0 = 1 µs/ciclo;
SM67.4	SM77.4	Actualización PWM:	0=asíncrona
SM67.5	SM77.5	Función PTO monosegmento/multisegmento:	
SM67.6	SM77.6	Elegir el modo PTO/PWM 0 = PTO	1 = PWM
SM67.7	SM77.7	Habilitar PTO/PWM: 0 = inhibir	1 = habilitar
Q0.0	Q0.1	Otros registros PTO/PWM	
SMW68	SMW7	Tiempo de ciclo PTO/PWM	rango: 2a65.535
SMW70	SMW8	Valor del ancho de impulsos PWM	rango: 0 a
SMD72	SMD82	Valor de conteo de impulsos PTO	rango: 1 a
SMB16 6	SMB17 6	Número del segmento que se está ejecutando	Sólo en el modo
SMW16 8	SMW17 8	Dirección inicial de la tabla de perfiles multisegmento PTO (offset de bytes a partir de V0)	Sólo en el modo
SMB17	SMB18	Byte de estado del perfil lineal	
SMB17	SMB18	Byte de resultado del perfil lineal	
SMD17	SMD18	Registro de frecuencia en modo manual	

Tabla 6-37 Referencias del byte de control PTO/PWM

Registr control (valor 16#)	Resultado de la operación				Base de	Contaje de impulsos	Ancho de impulsos	Tiempo de Carga
	Habilit	Segme	Actualiza					
16#81	Sí	☒	Monosegm					
16#84	Sí	☒	Monosegm					
16#85	Sí	☒	Monosegm					Carga
16#89	Sí	PTO	Monosegm					Carga
16#8C	Sí	PTO	Monosegm			Cargar		
16#8D	Sí	PTO	Monosegm			Cargar		Carga
16#A0	Sí	PTO	Multisegme					
16#A8	Sí	PTO	Multisegme					
16#D1	Sí	PW		Síncrona				Carga

M

r

16#D2	Sí	PW		Síncrona	1		Cargar	
16#D3	Sí	PW		Síncrona	1		Cargar	Carga
16#D9	Sí	PW		Síncrona	1			Carga
16#DA	Sí	PW		Síncrona	1		Cargar	
16#DB	Sí	PW		Síncrona	1		Cargar	Carga

Calcular los valores de la tabla de perfiles

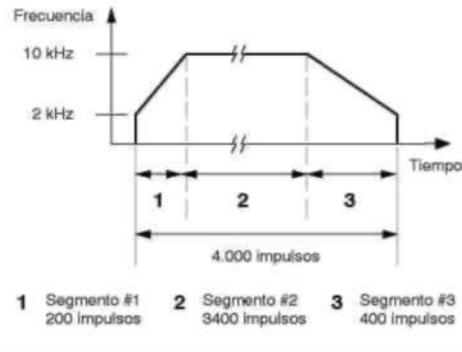


Figura 6-31 Diagrama frecuencia/tiempo

El pipelining multisegmento que ofrecen los generadores PTO/PWM se puede utilizar para numerosas aplicaciones, en particular para el control de motores paso a paso. Por ejemplo, la función PTO se puede usar con un perfil de impulsos para controlar un motor paso a paso, de manera que éste acelere, marche a una velocidad constante y desacelere luego. También se puede emplear en secuencias más complicadas, definiendo un perfil de impulsos que puede comprender hasta 255 segmentos, correspondiendo cada uno de ellos a una función **de aceleración, marcha constante o** desaceleración.

La figura 6-31 muestra los valores de la tabla de perfiles necesarios para generar una forma de onda de salida con objeto de acelerar un motor paso a paso (segmento #1), permitir que funcione a una velocidad constante (segmento #2) y desacelerarlo luego (segmento #3).

En el presente ejemplo, la frecuencia de impulsos inicial y final es de 2 kHz, la frecuencia de impulsos máxima es de 10 kHz y se requieren 4000 impulsos para alcanzar el número deseado de revoluciones del motor. Puesto que los valores de la tabla de perfiles se expresan en términos de período (tiempo de ciclo) y no de frecuencia, es preciso convertir los valores de frecuencia dados en valores de tiempo de ciclo. Por tanto, el tiempo de ciclo inicial y final es de 500 μ s, en tanto que el tiempo de ciclo correspondiente a la frecuencia máxima es de 100 μ s. Durante la etapa de aceleración del perfil de salida, la frecuencia máxima de impulsos se debe alcanzar en aproximadamente 200 impulsos. La etapa de desaceleración del perfil se debe finalizar en aproximadamente 400 impulsos.

La siguiente fórmula sencilla se puede utilizar para calcular el valor delta del tiempo de ciclo de un segmento dado que utiliza el generador PTO/PWM para ajustar el tiempo de ciclo de cada impulso:

$$\text{Tiempo de ciclo delta de un segmento} = | \text{End_CT}_{\text{seg}} - \text{Init_CT}_{\text{seg}} | / \text{Cantidad}_{\text{seg}}$$

donde: $\text{End_CT}_{\text{seg}}$ = tiempo de ciclo final de este segmento
 $\text{Init_CT}_{\text{seg}}$ = tiempo de ciclo inicial de este segmento
 $\text{Cantidad}_{\text{seg}}$ = cantidad de impulsos de este segmento

Utilizando esta fórmula se calculan los valores del tiempo de ciclo para la aplicación de ejemplo:

Segmento 1 (aceleración):

Tiempo de ciclo delta = -2

Segmento 2 (velocidad constante):

Tiempo de ciclo delta = 0

Segmento 3 (desaceleración):

Tiempo de ciclo delta = 1

Suponiendo que la tabla de perfiles se encuentra almacenada en la memoria V (comenzando en V500), los valores utilizados para generar la forma de onda deseada figuran en la tabla 6-38. Es posible incluir operaciones en el programa, con objeto de cargar estos valores en la memoria V, o bien definir los valores del perfil en el bloque de datos.

Tabla 6-38 Valores de la tabla de perfiles

Direcci	Valo	Descripción	
VB500	3	Número total de	
VW501	500	Tiempo de ciclo	Segment o 1
VW503	-2	Tiempo de ciclo delta inicial	
VD505	200	Número de	
VW509	100	Tiempo de ciclo	
VW511	0	Tiempo de ciclo	Segment
VD513	3400	Número de	
VW517	100	Tiempo de ciclo	Segment o 3
VW519	1	Tiempo de ciclo	
VD521	400	Número de impulsos	

Para determinar si son aceptables las transiciones entre los segmentos de formas de ondas, es preciso definir el tiempo de ciclo del último impulso de un segmento. A menos que el tiempo de ciclo delta sea 0, es preciso calcular dicho tiempo de ciclo, puesto que este valor no se indica en el perfil. Utilice la fórmula siguiente para calcular el tiempo de ciclo del último impulso:

Tiempo de ciclo del último impulso de un segmento = $\text{Init_CT}_{\text{seg}} + (\text{delta}_{\text{seg}} * (\text{cantidad}_{\text{seg}} - 1))$

donde: **$\text{Init_CT}_{\text{seg}}$** = tiempo de ciclo inicial de este segmento

$\text{Delta}_{\text{seg}}$ = valor delta del tiempo de ciclo de este segmento

$\text{Cantidad}_{\text{seg}}$ = cantidad de impulsos de este segmento

Aunque el ejemplo simplificado descrito arriba sirve de introducción a la materia, es posible que en una aplicación real se necesiten perfiles de formas de ondas más complejos. Recuerde que el tiempo de ciclo delta sólo se puede indicar en forma de número entero en microsegundos o milisegundos. Asimismo, considere que en cada impulso se modifica el tiempo de ciclo.

El efecto de estos dos puntos es que el cálculo del valor delta del tiempo de ciclo para un determinado segmento podría exigir una aproximación iterativa. Puede ser necesaria cierta flexibilidad en el valor del tiempo de ciclo final o en el número de impulsos para un determinado segmento.

La duración de un segmento de perfil puede ser útil a la hora de determinar los valores correctos para la tabla de perfiles. Utilice la fórmula siguiente para calcular la duración de un segmento dado:

Duración del segmento = $\text{cantidad}_{\text{seg}} * (\text{Init_CT} + ((\text{Delta}_{\text{seg}}/2) * (\text{cantidad}_{\text{seg}} - 1)))$

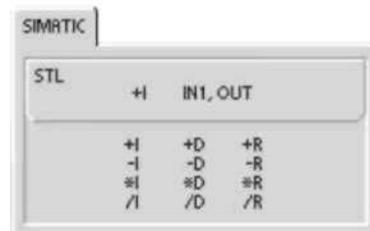
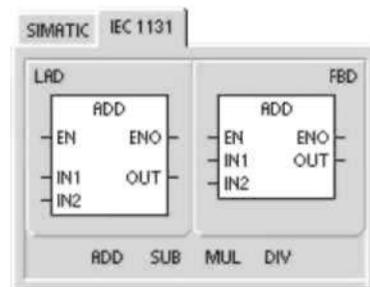
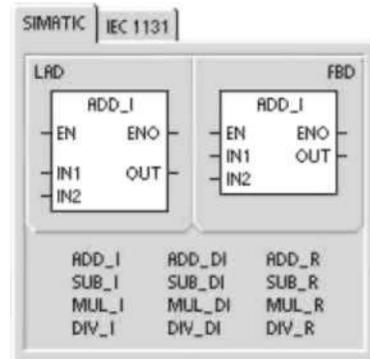
donde: **$\text{Cantidad}_{\text{seg}}$** = cantidad de impulsos de este segmento

$\text{Init_CT}_{\text{seg}}$ = tiempo de ciclo inicial de este segmento

$\text{Delta}_{\text{seg}}$ = valor delta del tiempo de ciclo de este segmento

Operaciones aritméticas

Operaciones de sumar, restar, multiplicar y dividir



Sumar

$IN1+IN2=OUT$

FUP

$IN1+OUT=OUT$

Restar

$IN1-IN2=OUT$

KOP y

$OUT-IN1=OUT$

AWL

Las operaciones Sumar enteros (+I) y Restar enteros (-I) suman/restan dos enteros de 16 bits, arrojando un resultado de 16 bits. Las operaciones Sumar enteros dobles (+D) y Restar enteros dobles (-D) suman/restan dos enteros de 32 bits, arrojando un resultado de 32 bits. Las operaciones Sumar reales (+R) y Restar reales (-R) suman/restan dos números reales de 32 bits, dando como resultado un número real de 32 bits.

Multiplicar

$IN1*IN2=OUT$

FUP

$IN1*OUT=OUT$

Dividir

$IN1/IN2=OUT$

KOP y

$OUT/IN1=OUT$

AWL

Las operaciones Multiplicar enteros (*I) y Dividir enteros (/I) multiplican o dividen dos enteros de 16 bits, respectivamente, arrojando un resultado de 16 bits. (En la división no se conserva un resto.) Las operaciones Multiplicar enteros dobles (*D) y Dividir enteros dobles (/D) multiplican o dividen dos enteros de 32 bits, respectivamente, arrojando un resultado de 32 bits. (En la división no se conserva

un resto.) Las operaciones Multiplicar reales (*R) y Dividir reales (/R) multiplican o dividen dos números reales de 32 bits, respectivamente, dando como resultado un número real de 32 bits.

Marcas especiales y ENO

SM1.1 indica errores de desbordamiento y valores no válidos. Si se activa SM1.1, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 y SM1.3 no se activan, la operación aritmética habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido. Si se activa SM1.3 durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado.

Condiciones de error que Marcas especiales afectadas

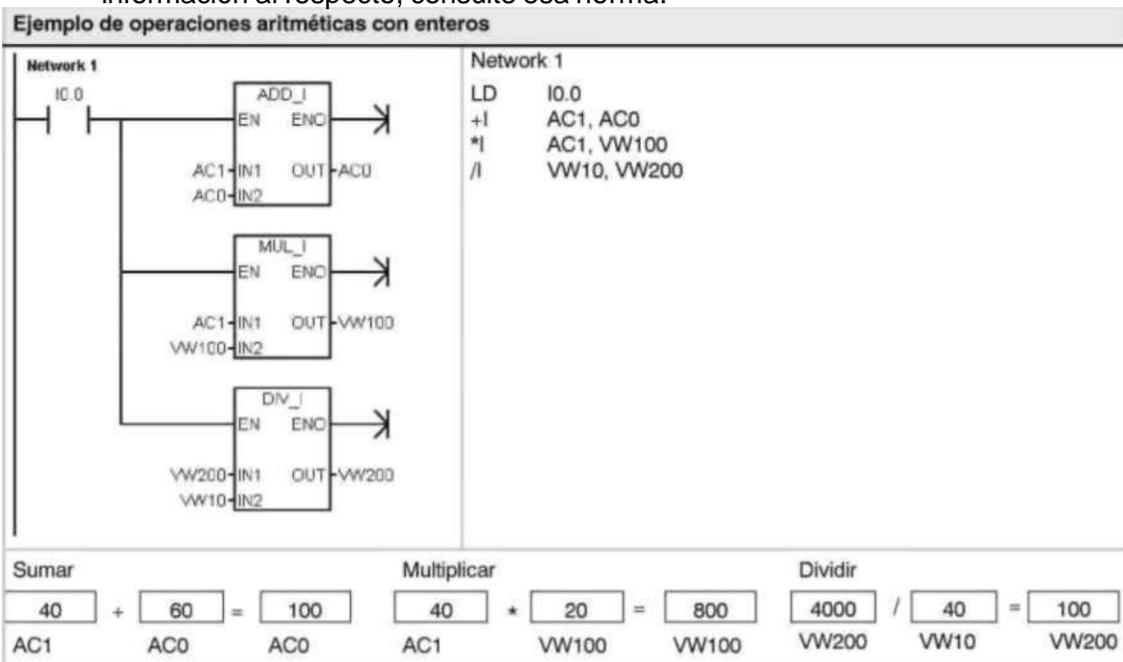
ponen ENO a 0:

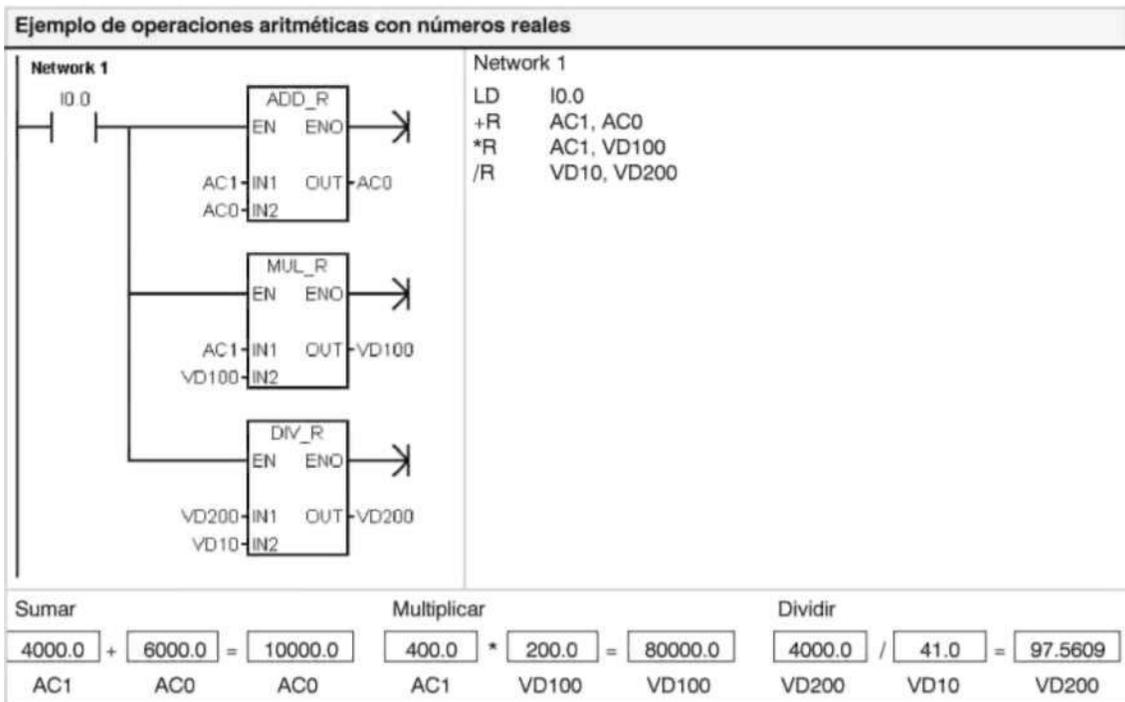
- SM1.0(cero)
- SM1.1 (desbordamiento) " SM1.1 (desbordamiento, valor no válido generado durante la operación o parámetro de entrada no válido)
- SM1.3 (división por o) • SM 1.2 (negativo)
- 0006 (direccionamiento por cero) " SM1.3 (división indirecto)

Tabla 6-39 Operandos válidos para las operaciones de sumar, restar, multiplicar y dividir

Entradas/salidas	Tipos	Operandos
IN1, IN2	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC,
	DINT	AIW, *VD, *AC, *LD, constante
	REA	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD,
	L	<u>*AC, constante</u>
OUT	INT	IW, QW, VW, MW, SMW, SW, LW, T, C,
	DINT, REAL	AC, *VD, *AC, *LD ID, QD, VD, MD, SMD,

Los números reales (o números en coma flotante) se representan en el formato descrito en la norma ANSI/IEEE 754-1985 (precisión sencilla). Para obtener más información al respecto, consulte esa norma.





Multiplicar enteros a enteros dobles y Dividir enteros con resto

Multiplicar enteros a enteros dobles

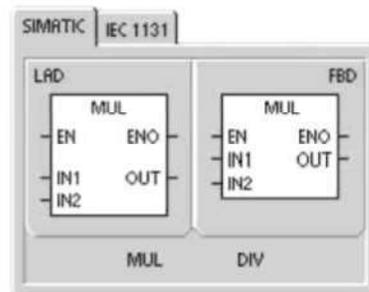
$IN1 * IN2 = OUT$ KOP y FUP
 $IN1 * OUT = OUT$ AWL

La operación Multiplicar enteros a enteros dobles (MUL) multiplica dos números enteros de 16 bits, arrojando un producto de 32 bits. En la operación AWL de multiplicación, la palabra menos significativa (16 bits) del OUT de 32 bits se utiliza como uno de los factores.

Dividir enteros con resto

$IN1 / IN2 = OUT$ KOP y FUP
 $OUT / IN1 = OUT$ AWL

La operación Dividir enteros con resto (DIV) divide dos números enteros de 16 bits, arrojando un resultado de 32 bits, compuesto por un resto de 16 bits (la palabra más significativa) y un cociente de 16 bits (la palabra menos significativa).



En la operación AWL de división, la palabra menos significativa (16 bits) del OUT de 32 bits se utiliza como dividendo.

Marcas especiales y ENO

En las operaciones Multiplicar enteros a enteros dobles y Dividir enteros con resto, las marcas especiales indican errores y valores no válidos. Si se activa SM1.3 (división por cero) durante una operación de división, permanecerán inalterados los demás bits aritméticos de estado. En otro caso, todos los bits aritméticos de estado soportados

contendrán el estado válido al finalizar la operación aritmética.

Condiciones de error que ponen ENO a 0: Marcas especiales afectadas

- SM 1.1 (desbordamiento)
- SM 1.3 (división por cero)
- 0006 (direccionamiento indirecto)
- SM 1.0 (cero)
- SM 1.1 (desbordamiento)
- SM 1.2 (negativo)
- SM1.3 (división por cero)

Tabla 6-40 Operandos válidos para las operaciones Multiplicar enteros a enteros dobles y Dividir enteros con resto

Entradas/sali	Tipos de	Operandos
IN1, IN2	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante
OUT	DINT	ID, QD, VD, MD, SMD, SD, LD, AC, *VD,

Ejemplo de las operaciones Multiplicar enteros a enteros dobles y Dividir enteros con resto

Network 1

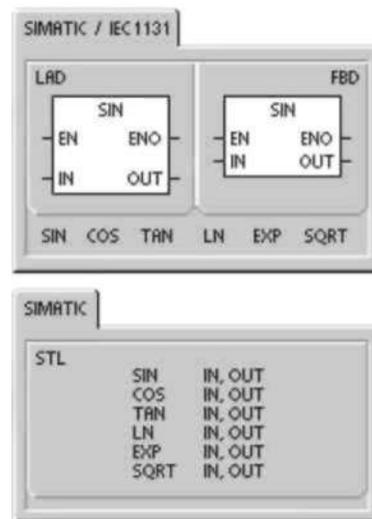
Network 1

```
LD 10.0
MUL AC1.VD100
DIV VW10.VD200
```

Multiplicar enteros a enteros dobles	403	*	200	=	80000
	AUI		VW10		VUIüO
				resto	cociente
Dividir enteros	4000	;	41	=	23 97
	VW202		VW10		VW200 VW20 VD200

Nota: VD100 contiene VW100 y VW102. en tanto que VD200 contiene VW200 y VW202

Operaciones con funciones numéricas



Seno, Coseno y Tangente

Las operaciones Seno (SIN), Coseno (COS) y Tangente (TAN) evalúan la función trigonométrica del valor del ángulo IN y depositan el resultado en OUT. El valor del ángulo de entrada se indica en radianes.

SIN (IN) = OUT COS (IN)= OUT TAN (IN)= OUT

Para convertir un ángulo de grados a radianes, utilice la operación Multiplicar reales (MUL_R (*R)), con objeto de multiplicar el ángulo en grados por 1,745329E-2 (aproximadamente por $\pi/180$).

Logaritmo natural y Exponente natural

La operación Logaritmo natural (LN) calcula el logaritmo natural del valor de IN y deposita el resultado en OUT.

La operación Exponente natural (EXP) ejecuta el cálculo exponencial de la constante “e” elevada a la potencia del valor de IN y deposita el resultado en OUT.

LN (IN) = OUT EXP (IN)= OUT

Para obtener el logaritmo con base 10 del logaritmo natural, divida el logaritmo natural por 2,302585 (que es aproximadamente el logaritmo natural de 10).

*Para elevar un número real cualquiera a la potencia de otro número real, incluyendo exponentes fraccionarios, combine la operación Exponente natural con la operación Logaritmo natural. Por ejemplo, para elevar X a la potencia de Y, introduzca la operación siguiente: EXP (Y * LN (X)).*

Raíz cuadrada

La operación Raíz cuadrada (SQRT) extrae la raíz cuadrada de un número real de 32 bits (IN), dando como resultado un número real de 32 bits (OUT), como muestra la ecuación siguiente:

SQRT (IN)= OUT

Para obtener otras raíces: 5 elevado al cubo = $5^3 = \text{EXP}(3*\text{LN}(5)) = 125$

Raíz cúbica de 125 = $125^{(1/3)} = \text{EXP}((1/3)*\text{LN}(125))= 5$

Raíz cúbica de 5 elevado al cubo = $5^{(3/2)} =$

$\text{EXP}(3/2*\text{LN}(5)) =$

11,18034

Marcas especiales y ENO para las operaciones con funciones numéricas

En todas las operaciones descritas en esta página, SM1.1 se utiliza para indicar errores de rebase y valores no válidos. Si se activa SM1.1, el estado de SM1.0 y de SM1.2 no será válido y no se alterarán los operandos de entrada originales. Si SM1.1 y SM1.2 no se activan durante una operación de división, la operación aritmética habrá finalizado con un resultado válido, y tanto SM1.0 como SM1.2 contendrán un estado válido.

Condiciones de error que Marcas especiales afectadas

ponen ENO a 0: . SM1.0(cero)

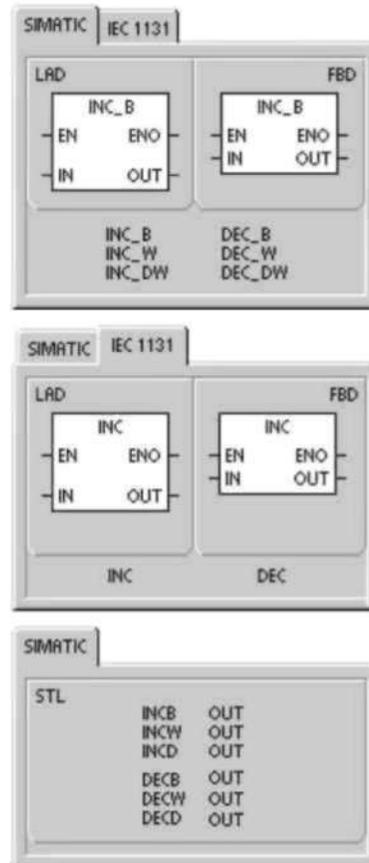
- SM1.1
(desbordamiento) " SM11 (desbordamiento)
- 0006 (direccionamiento " SM 1.2
(negativo)
indirecto)

Tabla 6-41 Operandos válidos para las funciones numéricas

Entradas/salida	Tipos de datos	Operandos
IN	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,
OUT	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,

Los números reales (o números en coma flotante) se representan en el formato descrito en la norma ANSI/IEEE 754-1985 (precisión sencilla). Para obtener más

información al respecto, consulte esa norma.
Incrementar y decrementar



Incrementar

IN +1=OUT *KOP y FUP*

OUT + 1 = OUT *AWL*

Decrementar

IN - 1 = OUT *KOP y FUP*

OUT - 1 = OUT *AWL*

Las operaciones Incrementar y Decrementar suman/restan 1 al valor de la entrada IN y depositan el resultado en OUT

Las operaciones Incrementar byte (INCB) y Decrementar byte (DECB) no llevan signo.

Las operaciones Incrementar palabra (INCW) y Decrementar palabra (DECW)

llevan signo.

Las operaciones Incrementar palabra doble (INCD) y Decrementar palabra doble (DECD) llevan signo.

Condiciones de error que ponen ENO a 0:

- SM1.1 (desbordamiento)
- 0006 (direccionamiento indirecto)

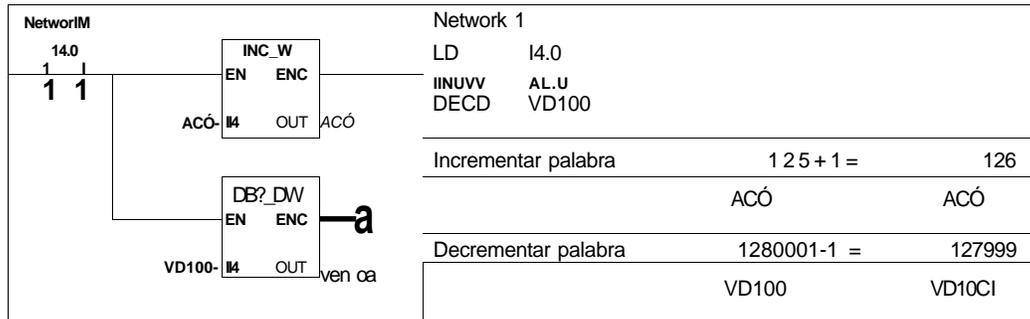
Marcas especiales afectadas:

- SM 1.0 (cero)
- SM1.1 (desbordamiento)

- SM1.2 (negativo) para operaciones con palabras y palabras dobles
- Tabla 6-42 Operandos válidos para las operaciones Incrementar y Decrementar

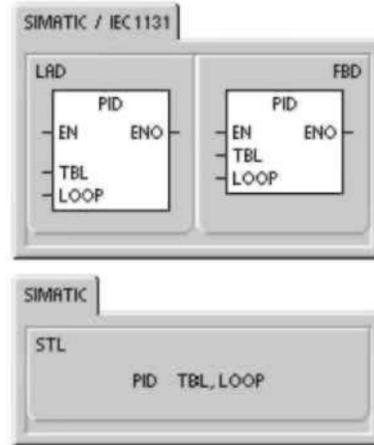
Entradas/sali	Tipos de	Operandos
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC,
	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC,
OUT	DINT	AIW, *VD, *LD, *AC, constante
	BYTE	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD,
	INT	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
	DINT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,

Ejemplo Incrementar y decrementar



Regulación PID proporcional/integral/derivativa

La operación Regulación PID (PID) ejecuta el cálculo de un lazo de regulación PID en el



LOOP referenciado, conforme a las informaciones de entrada y configuración definidas en TABLE (TBL).

Condiciones de error que ponen ENO a 0:

- SM1.1 (desbordamiento)
- 0006 (direccionamiento indirecto)

Marcas especiales afectadas:

- SM1.1 (desbordamiento)

La operación PID (lazo de regulación con acción proporcional, integral, derivativa) sirve para ejecutar el cálculo PID. Para habilitar el cálculo PID, el primer nivel de la pila lógica (TOS) deberá estar a ON (circulación de corriente). Esta operación tiene dos operandos:

una dirección TABLE que constituye la dirección inicial de la tabla del lazo y un número LOOP que es una constante comprendida entre 0 y 7.

Un programa sólo admite ocho operaciones PID. Si se utilizan dos o más operaciones PID con el mismo número de lazo (aunque tengan diferentes direcciones de tabla), los dos cálculos PID se interferirán mutuamente siendo impredecible el resultado.

La tabla del lazo almacena nueve parámetros que sirven para controlar y supervisar la operación del mismo. Incluye el valor actual y previo de la variable del proceso (valor real), la consigna, la salida o magnitud manipulada, la ganancia, el tiempo de muestreo, el tiempo de acción integral, el tiempo de acción derivativa y la suma integral (bias).

Para poder realizar el cálculo PID con el intervalo de muestreo deseado, la operación PID deberá ejecutarse bien dentro de una rutina de interrupción temporizada, o bien desde el programa principal, a intervalos controlados por un temporizador. El tiempo de muestreo debe definirse en calidad de entrada para la operación PID a través de la tabla del lazo.

La función de autosintonía se ha agregado a la operación PID. Para obtener una descripción detallada acerca de esta función, consulte el capítulo 15. El Panel de sintonía PID sólo funciona con los lazos PID creados con el asistente PID.

Tabla 6-43 Operandos válidos para la operación Regulación PID

Entradas/salida	Tipos de	Operandos
TBL	BYTE	VB
LOOP	BYTE	Constante (0 a 7)

STEP 7-Micro/WIN incorpora el asistente PID que ayuda a definir un algoritmo PID para un proceso de control de bucle cerrado. Elija el comando de menú **Herramientas > Asistente de operaciones** y seleccione **PID** en la ventana del asistente.

Consejo

La consigna de los límites inferior y superior debería corresponder a los límites inferior y superior de la variable del proceso.



Asistente de
operaciones



Algoritmo PID

En modo estacionario, un regulador PID varía el valor de su salida para llevar a cero el error de regulación (e). El error es la diferencia entre el valor de consigna (SP) (el punto de trabajo deseado) y la variable del proceso (PV) (el punto de trabajo real). El principio de una regulación PID se basa en la ecuación que se indica a continuación y que expresa la salida $M(t)$ como una función de un término proporcional, uno integral y uno derivativo:

Salida	=	término proporcional	+	término integral	+	término derivativo
$M(t)$	=	$K_C * e$	+	$K_C \int_0^t e dt - M_{inicial}$	+	$K_C * de/dt$
donde:	$M(t)$	es la salida del lazo en función del tiempo				
	K_C	es la ganancia del lazo				
	e	es el error de regulación (diferencia entre la consigna y la variable de proceso)				
	$M_{inicial}$	es el valor inicial de la salida del lazo				

Para poder implementar esta función de regulación en un sistema digital, la función continua deberá cuantificarse mediante muestreos periódicos del valor del error, calculándose seguidamente el valor de la salida. La ecuación que constituye la base de la solución en un sistema digital es:

M_n	=	$K_C * e_n$	+	$K_I * \sum_1^n e_x + M_{inicial}$	+	$K_D * (e_n - e_{n-1})$
Salida	=	término proporcional	+	término integral	+	término derivativo
donde:	M_n	es el valor de salida del lazo calculado en el muestreo n-ésimo				
	K_C	es la ganancia del lazo				
	e_n	es el valor del error del lazo en el muestreo n-ésimo				
	e_{n-1}	es el valor previo del error de regulación (en el muestreo (n-1)-ésimo)				
	e_x	es el valor del error de regulación en el muestreo x				
	K_I	es la constante proporcional del término integral				
	$M_{inicial}$	es el valor inicial de la salida del lazo				
	K_D	es la constante proporcional del término derivativo				

Para esta ecuación, el término integral se muestra en función de todos los términos del error, desde el primer muestreo hasta el muestreo actual. El término derivativo es una función del muestreo actual y del muestreo previo; mientras que el término proporcional sólo es función del muestreo actual. En un sistema digital no es práctico almacenar todos los muestreos del término del error, además de no ser necesario.

Puesto que un sistema digital debe calcular el valor de salida cada vez que se muestre el error, comenzando en el primer muestreo, basta con almacenar el valor previo del error y el valor previo del término integral. Debido a la naturaleza repetitiva de la solución basada en un sistema digital, es posible simplificar la ecuación a resolver en cada muestreo. La ecuación simplificada es:

M_n	=	$K_C * e_n$	+	$K_I * e_n + MX$	+	$K_D * (e_n - e_{n-1})$
Salida	=	término proporcional	+	término integral	+	término derivativo
donde:	M_n	es el valor de salida del lazo calculado en el muestreo n-ésimo				
	K_C	es la ganancia del lazo				
	e_n	es el valor del error del lazo en el muestreo n-ésimo				
	e_{n-1}	es el valor previo del error de regulación (en el muestreo (n-1)-ésimo)				
	K_I	es la constante proporcional del término integral				
	MX	es el valor previo del término integral (en el muestreo (n-1)-ésimo)				
	K_D	es la constante proporcional del término derivativo				

Para calcular el valor de salida del lazo, el S7-200 utiliza una forma modificada de la ecuación simplificada anterior. Esta ecuación modificada equivale a:

$$M_n = MP_n + MI_n + MD_n$$

Salida término proporcional término integral término derivativo

donde: M_n es el valor de salida del lazo calculado en el muestreo n-ésimo
 MP_n es el valor del término proporcional de salida del lazo en el muestreo n-ésimo
 MI_n es el valor del término integral de salida del lazo en el muestreo n-ésimo
 MD_n es el valor del término derivativo de salida del lazo en el muestreo n-ésimo

Término proporcional de la ecuación PID

El término proporcional MP es el producto de la ganancia (K_C), la cual controla la sensibilidad del cálculo de la salida, y del error (e), que es la diferencia entre el valor de consigna (SP) y el valor real o de la variable del proceso (PV) para un instante de muestreo determinado. La ecuación que representa el término proporcional según la resuelve el S7-200 es la siguiente:

$$MP_n = K_C * (SP_n - PV_n)$$

donde: MP_n es el valor del término proporcional de salida del lazo en el muestreo n-ésimo
 K_C es la ganancia del lazo
 SP_n es el valor de la consigna en el muestreo n-ésimo
 PV_n es el valor de la variable del proceso en el muestreo n-ésimo

Término integral de la ecuación PID

El término integral MI es proporcional a la suma del error a lo largo del tiempo. La ecuación que representa el término integral según la resuelve el S7-200 es la siguiente:

$$MI_n = K_C * \left(\frac{T_s}{T_I} * (SP_n - PV_n) + MX \right)$$

donde: MI_n es el valor del término integral de salida del lazo en el muestreo n-ésimo
 K_C es la ganancia del lazo
 T_s es el tiempo de muestreo del lazo
 T_I es el período de integración del lazo (también llamado tiempo de acción integral)
 SP_n es el valor de la consigna en el muestreo n-ésimo
 PV_n es el valor de la variable del proceso en el muestreo n-ésimo
 MX es el valor del término integral en el muestreo (n-1)-ésimo (también llamado suma integral o "bias")

La suma integral o bias (MX) es la suma acumulada de todos los valores previos del término integral. Después de cada cálculo de MI_n se actualiza la suma integral con el valor de MI_n que puede ajustarse o limitarse (para más información, consulte la sección "Variables y rangos"). Por regla general, el valor inicial de la suma integral se ajusta al valor de salida ($M_{inicial}$) justo antes de calcular la primera salida del lazo. El término integral incluye también varias constantes tales como la ganancia (K_C), el tiempo de muestreo (T_s), que define el intervalo con que se recalcula periódicamente el valor de salida del lazo PID, y el tiempo de acción integral (T_I), que es un tiempo utilizado para controlar la influencia del término integral en el cálculo de la salida.

Término derivativo de la ecuación PID

El término derivativo MD es proporcional a la tasa de cambio del error. El S7-200 utiliza la ecuación siguiente para el término derivativo:

$$MD_n = K_C * T_D / T_S * ((SP_n - PV_n) - (SP_{n-1} - PV_{n-1}))$$

Para evitar cambios o saltos bruscos de la salida debidos a cambios de la acción derivativa o de la consigna, se ha modificado esta ecuación bajo la hipótesis de que la consigna es constante ($SP_n = SP_{n-1}$). En consecuencia, se calcula el cambio en la variable del proceso en lugar del cambio en el error, como puede verse a continuación:

$$MD_n = K_C * T_D / T_S * (SP_n - PV_n - SP_{n-1} + PV_{n-1})$$

o simplificando:

$$MD_n = K_C * T_D / T_S * (PV_{n-1} - PV_n)$$

donde:

MD_n	es el valor del término derivativo de la salida del lazo en el muestreo n-ésimo
K_C	es la ganancia del lazo
T_S	es el tiempo de muestreo del lazo
T_D	es el período de diferenciación de lazo (también llamado tiempo de acción derivativa)
SP_n	es el valor de la consigna en el muestreo n-ésimo
PV_{n-1}	es el valor de la consigna en el muestreo n-1
PV_n	es el valor de la variable del proceso en el muestreo n-ésimo
PV_{n-1}	es el valor de la variable del proceso en el muestreo (n-1)-ésimo

En lugar del error es necesario guardar la variable del proceso para usarla en el próximo cálculo del término derivativo. En el instante del primer muestreo, el valor de PV_{n-1} se inicializa a un valor igual a PV_n .

Seleccionar el tipo de regulación

En numerosos sistemas de regulación basta con utilizar una o dos acciones de regulación. Así, por ejemplo, puede requerirse únicamente regulación proporcional o regulación proporcional e integral. El tipo de regulación se selecciona ajustando correspondientemente los valores de los parámetros constantes.

Por tanto, si no se desea acción integral (sin "I" en el cálculo PID), entonces el tiempo de acción integral deberá ajustarse a infinito ("INF"). Incluso sin acción integral, es posible que el valor del término integral no sea "0", debido a que la suma integral MX puede tener un valor inicial.

Si no se desea acción derivativa (sin "D" en el cálculo PID), entonces el tiempo de acción derivativa deberá ajustarse a 0.0.

Si no se desea acción proporcional (sin "P" en el cálculo PID) y se desea regulación I o ID, entonces la ganancia deberá ajustarse a 0.0. Puesto que la ganancia interviene en las ecuaciones para calcular los términos integral y derivativo, si se ajusta a 0.0 resulta un valor de 1.0, que es el utilizado para calcular los términos integral y derivativo.

Convertir y normalizar las entradas del lazo

El lazo tiene dos variables o magnitudes de entrada, a saber: la consigna y la variable del proceso. La consigna es generalmente un valor fijo (por ejemplo, el ajuste de velocidad en el ordenador de abordo de un automóvil). La variable del proceso es una magnitud relacionada con la salida del lazo y que mide por ello el efecto que tiene la misma sobre el sistema regulado. En el ejemplo del ordenador de abordo, la variable del proceso sería la entrada al tacómetro que es una señal proporcional a la velocidad de giro de las ruedas.

Tanto la consigna como la variable del proceso son valores físicos que pueden tener diferente magnitud, rango y unidades de ingeniería. Para que la operación PID pueda utilizar esos valores físicos, éstos deberán convertirse en representaciones normalizadas en coma flotante.

El primer paso es convertir el valor físico de un valor entero de 16 bits en un valor en coma flotante o real. La siguiente secuencia de operaciones muestra cómo convertir un valor entero en un número real.

```
ITD    AIW0, AC0    //Convertir un valor de entrada en una palabra doble.
DTR    AC0, AC0    //Convertir un entero doble en un número real.
```

El próximo paso consiste en convertir el número real representativo del valor físico en un valor normalizado entre 0.0 y 1.0. La ecuación siguiente se utiliza para normalizar tanto la consigna como el valor de la variable del proceso.

$$RN_{\text{Norm}} = ((RNO_{\text{norm}} / \text{Alcance}) + \text{Offset})$$

- donde:
- RN_{Norm} es la representación como número real normalizado del valor físico
 - RNO_{norm} es la representación como número real no normalizado del valor físico
 - Offset vale 0,0 para valores unipolares
vale 0,5 para valores bipolares
 - Alcance es la diferencia entre el máximo valor posible menos el mínimo valor posible:
= 32.000 para valores unipolares (típico)
= 64.000 para valores bipolares (típico)

La siguiente secuencia de operaciones muestra la forma de normalizar el valor bipolar contenido en AC0 (cuyo alcance vale 64.000), continuando la secuencia previa:

```
/R    64000.0, AC0    // Normalizar el valor en el acumulador.
+R    0,5, AC0        //Desplazar el valor al rango entre 0.0 y 1.0.
MOVR  AC0, VD100     // Almacenar el valor normalizado en la tabla del lazo.
```

Convertir la salida del lazo en un valor entero escalado

La salida del lazo constituye la variable manipulada; en el caso del automóvil, la posición de la mariposa en el carburador. La salida del lazo es un valor real normalizado comprendido entre 0.0 y 1.0. Antes de que la salida del lazo pueda utilizarse para excitar una salida analógica, deberá convertirse en un valor escalado de 16 bits. Esta operación constituye el proceso inverso de convertir PV y SP en un valor normalizado. El primer paso es convertir la salida del lazo en un valor real escalado usando la fórmula siguiente:

$$R_{\text{Scal}} = (M_n - \text{Offset}) \cdot \text{Alcance}$$

- donde:
- R_{Scal} es el valor real escalado de la salida del lazo
 - M_n es el valor real normalizado de la salida del lazo
 - Offset vale 0,0 para valores unipolares
vale 0,5 para valores bipolares
 - Alcance es la diferencia entre el máximo valor posible menos el mínimo valor posible:
= 32.000 para valores unipolares (típico)
= 64.000 para valores bipolares (típico)

La siguiente secuencia de operaciones muestra la forma de escalar la salida del lazo:

```

MOVR  VD108, ACÓ      //Transferir la salida del lazo al acumulador.
-R     0,5. ACÓ       //Incluir esta operación sólo si el valor es bipolar.
*R     64000.0, ACÓ   //Escalar el valor en el acumulador.

```

Seguidamente es necesario convertir el valor real escalado representativo de la salida del lazo en un entero de 16 bits La secuencia siguiente muestra cómo

```

efectuar esta operación : Convertir el número real en un entero doble.
DTI   AC0, LW0        //Convertir el valor en un entero de 16 bits.
MOVW  LW0, AQW0       //Escribir el valor en la salida analógica.

```

Lazos con acción positiva o negativa

El lazo tiene acción positiva si la ganancia es positiva y acción negativa si la ganancia es negativa. (En regulación I o ID, donde la ganancia vale 0.0, si se indica un valor positivo para el tiempo de acción integral y derivativa resulta un lazo de acción positiva y de acción negativa al indicarse valores negativos).

Variables y rangos

La variable del proceso y la consigna son magnitudes de entrada para el cálculo PID. Por tanto, la operación PID lee los campos definidos para estas variables en la tabla del lazo, pero no los modifica.

El valor de salida se genera al realizar el cálculo PID. Debido a ello, el campo en la tabla del lazo que contiene el valor de salida se actualiza cada vez que se termina un cálculo PID. El valor de salida está limitado entre 0.0 y 1.0. El usuario puede utilizar el campo de valor de salida en calidad de campo de entrada para indicar un valor de salida inicial cuando se conmuta de control manual a automático (consulte también la sección "Modos").

Si se utiliza regulación integral, la suma integral es actualizada por el cálculo PID y el valor actualizado se utiliza como entrada para el siguiente cálculo PID. Si el valor de salida calculado se sale de rango (salida inferior a 0.0 o superior a 1.0), la suma integral se ajustará conforme a las fórmulas siguientes:

```

MX          1.0          (MP,, + MD,,)          si la salida calculada, Mn > 1.0
o bien,
MX          -(MP,, + MD,,)          si la salida calculada, Mn < 0.0

```

donde: MX es el valor de la suma integral ajustada
MP_r es el valor del término proporcional de salida del lazo en el muestreo n-ésimo
MD_n es el valor del término derivativo de la salida del lazo en el muestreo n-ésimo
es el valor de la salida del lazo en el muestreo n-ésimo

Si la suma integral se calcula de la forma descrita, mejorará la respuesta del sistema cuando la salida calculada retorne al rango adecuado. Es decir, la suma integral calculada se limita entre 0.0 y 1.0 y luego se escribe en el campo reservado para ella en la tabla del lazo cada vez que se finaliza un cálculo PID. El valor almacenado en la tabla del lazo se utiliza para el próximo cálculo PID.

A fin de evitar problemas con valores de la suma integral en determinadas aplicaciones, el usuario puede modificar el valor de la suma integral en la tabla del lazo antes de ejecutar la operación PID. Cualquier modificación manual de la suma integral deberá realizarse con gran precaución. En todo caso, el valor de la suma integral escrito en la tabla del lazo deberá ser un número real comprendido entre 0,0 y 1,0.

En la tabla del lazo se mantiene un valor de comparación de la variable del proceso para su uso en la parte de acción derivativa del cálculo PID. El usuario no deberá

modificar este valor.

Modos

Los lazos PID del S7-200 no incorporan el control del modo de operación. El cálculo PID sólo se ejecuta si circula corriente hacia el cuadro PID. Por ello resulta el modo “automático” cuando se ejecuta cíclicamente el cálculo PID. Resulta el modo “manual” cuando no se ejecuta el cálculo PID.

La operación PID tiene un bit de historial de circulación de corriente similar a una operación de contador. La operación utiliza este bit para detectar una transición de “0” a “1” de la circulación de la corriente. Cuando se detecta la transición, la operación ejecutará una serie de acciones a fin de asegurar un cambio sin choques entre control manual y automático. Para evitar choques en la transición al modo automático, el valor de la salida ajustado manualmente deberá entregarse en calidad de entrada a la operación PID (escrita en la entrada para M_n en la tabla del lazo) antes de conmutar a modo automático. La operación PID ejecuta las siguientes acciones con los valores de la tabla del lazo a fin de asegurar un cambio sin choques entre control manual y automático cuando se detecta una transición de la circulación de corriente de “0” a “1”:

- Ajustar la consigna (SP_n) = variable de proceso (PV_n)
- Ajustar la variable del proceso antigua (PV_{n-1}) = variable del proceso (PV_n)
- Ajustar la suma integral (MX) = valor de salida (M_n)

El estado por defecto de los bits de historial PID es “activado”. Este estado se establece en el arranque o cada vez que haya una transición de modo STOP a RUN en el sistema de automatización. Si circula corriente hacia el cuadro PID la primera vez que se ejecuta tras entrar en el modo RUN, entonces no se detectará ninguna transición de la circulación de corriente y, por consecuencia, no se ejecutarán las acciones destinadas a evitar choques en el cambio de modo.

Alarmas y operaciones especiales

La operación PID es simple, pero ofrece grandes prestaciones para ejecutar cálculos PID. Si se precisan funciones de postprocesamiento (tales como funciones de alarma o cálculos especiales en base a las variables de lazo), ello deberá implementarse utilizando las operaciones básicas soportadas por el S7-200 en cuestión.

Condiciones de error

A la hora de compilar, la CPU generará un error de compilación (error de rango) y la compilación fallará si los operandos correspondientes a la dirección inicial o al número de lazo PID en la tabla del lazo están fuera de rango.

La operación PID no comprueba si todos los valores de entrada en la tabla del lazo respetan los límites de rango. Es decir, el usuario deberá vigilar que la variable del proceso y la consigna (al igual que la suma integral y la variable del proceso previa, si se utilizan como entradas) sean números reales comprendidos entre 0.0 y 1.0.

Si se detecta algún error al ejecutar las operaciones aritméticas del cálculo PID se activará la marca SM1.1 (desbordamiento o valor no válido) y se finalizará la ejecución de la operación PID. (La actualización de los valores de salida en la tabla del lazo podría ser incompleta, por lo que se deberán descartar estos valores y corregir el valor de entrada que ha provocado el error matemático antes de volver a ejecutar la operación de regulación PID.)

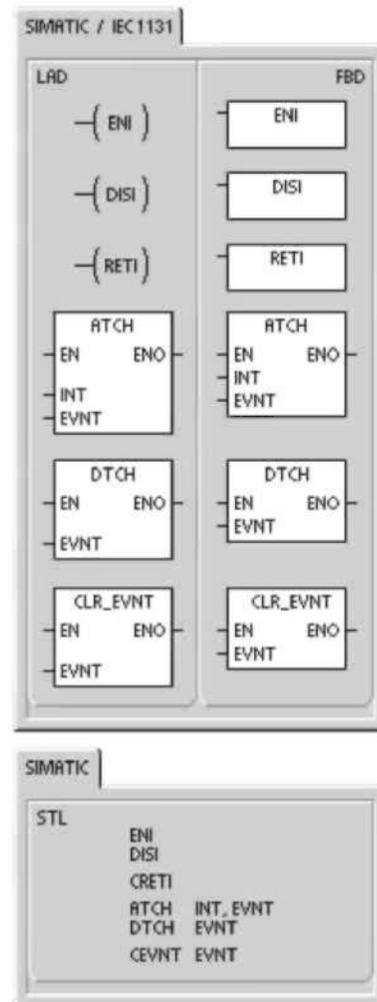
Tabla del lazo

La tabla del lazo tiene 80 bytes de longitud y el formato que muestra la tabla 6-44:
Tabla 6-44 Tabla del lazo

Offset	Campo	Format	Tipos de	Descripción
0	Variable del proceso (PV_n)	REAL	IN	Contiene la variable del proceso que debe estar
4	Consigna	REAL	IN	Contiene la consigna que <u>debe estar escalada entre</u>
8	Salida	REAL	IN/OUT	Contiene la salida calculada, <u>escalada entre 0.0 y 1.0.</u>
12	Ganancia (K_C)	REAL	IN	Contiene la ganancia, que es una constante proporcional. Puede ser un número positivo.
16	Tiempo de muestreo (T_s)	REAL	IN	Contiene el tiempo de muestreo en segundos.
20	Tiempo de acción integral	REAL	IN	Contiene el tiempo de acción integral en minutos. Tiene que ser un número positivo.
24	Tiempo de acción derivativa (T_D)	REAL	IN	Contiene el tiempo de acción derivativa en minutos. Tiene que ser un número positivo.
28	Suma integral (MX)	REAL	IN/OUT	Contiene el valor de la suma integral entre 0.0 y
32	Variable del proceso previa	REAL	IN/OUT	Contiene el valor almacenado de la variable de proceso al ejecutar por
36 a 79	Reservados para variable	s de autosin	tonía. Para más det	alles, consulte la tabla 15-1. PID.

Operaciones de interrupción

Habilitar todos los eventos de interrupción e Inhibir todos los eventos de



interrupción

La operación Habilitar todos los eventos de interrupción (ENI) habilita la ejecución de todos los eventos asociados. La operación Inhibir todos los eventos de interrupción (DISI) inhibe la ejecución de todos los eventos asociados.

Las interrupciones se inhiben cuando la CPU pasa a modo RUN. En modo RUN es posible habilitar el procesamiento de las interrupciones con la operación Habilitar todos los eventos de interrupción. Ejecutando la operación Inhibir todos los eventos de interrupción se inhibe el procesamiento de las interrupciones. No obstante, los eventos de interrupción activos se siguen poniendo en la cola de espera.

Condiciones de error que ponen ENO a 0:

- 0004 (intento de ejecutar una operación ENI, DISI o HDEF en una rutina de interrupción)

Retorno condicionado desde rutina de interrupción

La operación Retorno condicionado desde rutina de interrupción (CRETI) finaliza una rutina en función de la combinación lógica precedente.

Asociar interrupción

La operación Asociar interrupción (ATCH) asocia el número de una rutina de interrupción INT a un evento de interrupción EVNT, habilitando así éste último.

Condiciones de error que ponen ENO a 0:

- 0002 (conflicto de asignación de entradas a un contador rápido)

Desasociar interrupción

La operación Desasociar interrupción (DTCH) desasocia un evento de interrupción EVNT de todas las rutinas de interrupción, deshabilitando así el evento.

Borrar evento de interrupción

La operación Borrar evento de interrupción elimina todos los eventos de interrupción del tipo EVNT de la cola de interrupciones. Utilice esta operación para eliminar de la cola de interrupciones los eventos de interrupción indeseados. Si esta operación se utiliza para borrar eventos de interrupción falsos, es recomendable desasociar el evento en cuestión antes de borrar los eventos de la cola de interrupciones. De lo contrario, los nuevos eventos se agregarán a la cola tras haberse ejecutado la operación.

El ejemplo muestra un contador rápido A/B que utiliza la operación CLR_EVNT para borrar interrupciones. Si un sensor paso a paso troceador de luz se ha detenido en una posición límite entre un paso de claro a oscuro, las vibraciones leves de la máquina podrían generar interrupciones indeseadas antes de que se pueda cargar el nuevo valor de preselección.

Tabla 6-45 Operandos válidos para las operaciones de interrupción

Entradas/sali	Tipos de	Operandos
INT	BYTE	Constante (0 a 127)
EVNT	BYTE	Constante CPU 221 y 0 a 12, 19 a 23 y CPU 222: CPU 224: CPU 27 a 33 0 a 23 y 27 224XP y CPU 226: a 33 0 a 33

Funcionamiento de las operaciones

Asociar interrupción y Desasociar interrupción

Antes de poder llamar a una rutina de interrupción es preciso establecer un enlace entre el evento de interrupción y la parte del programa que se desee ejecutar cuando se presente el evento. La operación Asociar interrupción sirve para asignar el evento de interrupción (indicado por el número de evento) a una parte del programa (indicada por el número de la rutina de interrupción). También es posible asociar varios eventos de interrupción a una única rutina de interrupción. Por el contrario, no se puede asociar un sólo evento a distintas rutinas.

Cuando se asocia un evento a una rutina de interrupción, se habilita automáticamente el evento. Si se inhiben todos los eventos de interrupción, cada vez que se presente la interrupción se pondrá en cola de espera hasta que las interrupciones se habiliten de nuevo (utilizando para ello la operación Habilitar todos los eventos de interrupción), o bien hasta que se desborde la cola de espera de interrupciones.

También es posible inhibir ciertos eventos de interrupción, eliminando la asociación entre el evento y la correspondiente rutina mediante la operación Desasociar interrupción. Esta operación retorna la interrupción a un estado inactivo o ignorado. La tabla 6-46 muestra los diferentes tipos de eventos de interrupción.

Tabla 6-46 Eventos de interrupción

Evento	Descripción	CPU 221	CPU 224	CPU 224XP
0	I0.0 Flanco positivo	Sí	Sí	Sí
1	I0.0 Flanco negativo	Sí	Sí	Sí
2	I0.1 Flanco positivo	Sí	Sí	Sí
3	I0.1 Flanco negativo	Sí	Sí	Sí
4	I0.2 Flanco positivo	Sí	Sí	Sí
5	I0.2 Flanco negativo	Sí	Sí	Sí
6	I0.3 Flanco positivo	Sí	Sí	Sí
7	I0.3 Flanco negativo	Sí	Sí	Sí
8	Puerto0 Recibir carácter	Sí	Sí	Sí
9	Puerto 0 Transmisión	Sí	Sí	Sí
10	Interrupción temporizada 0	Sí	Sí	Sí
11	Interrupción temporizada 1	Sí	Sí	Sí
12	HSC0 CV=PV (valor actual = valor predeterminado)	Sí	Sí	Sí
13	HSC1 CV=PV (valor actual = valor predeterminado)		Sí	Sí
14	HSC1 Cambiodesentido		Sí	Sí
15	HSC1		Sí	Sí
16	HSC2 CV=PV (valor actual = valor predeterminado)		Sí	Sí
17	HSC2 Cambiodesentido		Sí	Sí
18	HSC2		Sí	Sí
19	PLS0 Interrupción Valordecontaje de impulsos PTO	Sí	Sí	Sí
20	PLS1 Interrupción Valordecontaje de impulsos PTO	Sí	Sí	Sí
21	Interrupción temporizador T32 CT=PT	Sí	Sí	Sí

Tabla 6-46 Eventos de interrupción, continuación

Evento	Descripci	CPU	CPU	CPU
22	Interrupción temporizador T96	Sí	Sí	Sí
23	Puerto 0 Recepción de mensajes	Sí	Sí	Sí
24	Puerto 1 Recepción de mensajes			Sí
25	Puerto 1 Recibir carácter			Sí
26	Puerto 1 Transmisión finalizada			Sí
27	HSC0 Cambio de sentido	Sí	Sí	Sí
28	HSC0 Puesto a 0 externamente	Sí	Sí	Sí
29	HSC4 CV=PV (valor actual = valor predeterminado)	Sí	Sí	Sí

30	HSC4	Cambio de sentido	Sí	Sí	Sí
31	HSC4	Puesto a 0 externamente	Sí	Sí	Sí
32	HSC3	CV=PV (valor actual = valor)	Sí	Sí	Sí
33	HSC5	CV=PV (valor actual = valor)	Sí	Sí	Sí

Procesar rutinas de interrupción en el S7-200

Las rutinas de interrupción se ejecutan como respuesta a un evento interno o externo asociado. Tras haberse ejecutado la última operación de la rutina de interrupción, el control retorna al programa principal. Para salir de la rutina se puede ejecutar una operación Retorno condicionado desde rutina de interrupción (CRETI). En la tabla 6-47 se indican algunas reglas y restricciones para utilizar rutinas de interrupción en el programa.

Tabla 6-47 Reglas y restricciones para utilizar rutinas de interrupción

Reglas

El procesamiento de interrupciones permite reaccionar rápidamente ante determinados eventos internos o externos. Las rutinas de interrupción se deben estructurar de forma que - una vez ejecutadas determinadas tareas - devuelvan el control al programa principal.

Para ello es conveniente crear rutinas de interrupción cortas con indicaciones precisas, de manera que se puedan ejecutar rápidamente sin interrumpir otros procesos durante períodos demasiado largos. Si no se observan estas medidas,

Restricciones produzcan estados imprevistos que podrían afectar a los procesos. Las operaciones Inhibir todos los eventos de interrupción (DISI), Habilitar todos los eventos de interrupción (ENI), Definir modo para contador rápido (HDEF) y utilizar en las

Soporte del sistema durante las interrupciones

nas de interrupción. Puesto que las interrupciones pueden afectar a la lógica de los contactos, las bobinas y los acumuladores, el sistema almacena la pila lógica, los acumuladores y las marcas especiales (SM) que indican el estado de los acumuladores y las operaciones, volviéndolos a cargar posteriormente. De este modo se evitan perturbaciones en el programa principal causadas por derivaciones a rutinas de interrupción o desde ellas.

Compartir datos entre el programa principal y las rutinas de interrupción

El programa principal y una o varias rutinas de interrupción pueden compartir datos. Puesto que no es posible saber con anterioridad cuándo el S7-200 generará una interrupción, es recomendable limitar el número de variables utilizadas tanto por la rutina de interrupción como en otra parte del programa. Los problemas de coherencia de los datos compartidos pueden ser ocasionados por las acciones de las rutinas de interrupción, al interrumpir éstas la ejecución de las operaciones del programa principal. Utilice la tabla de variables locales de la rutina de interrupción para garantizar que ésta utilice únicamente la memoria temporal, de manera que no se sobrescriban los datos utilizados en ninguna otra parte del programa.

Hay diversas técnicas de programación que se pueden utilizar para garantizar que el programa principal y las rutinas de interrupción compartan los datos correctamente. Estas técnicas restringen la forma de acceder a las direcciones compartidas en la memoria o evitan que se interrumpan las secuencias de operaciones que utilicen direcciones compartidas.

- En un programa AWL que comparta sólo una variable: Si los datos compartidos representan una sola variable en formato de byte, palabra o palabra doble, y el programa se ha escrito en AWL, los resultados intermedios de operaciones con datos compartidos sólo se podrán almacenar en direcciones o en acumuladores que no se compartan.
- En un programa KOP que comparta sólo una variable: Si los datos compartidos representan una sola variable en formato de byte, palabra o palabra doble, y el programa se ha escrito en KOP, es preciso acceder a las direcciones compartidas utilizando las operaciones de transferencia (MOVB, MOVW, MOVDW, MOVR). En tanto que numerosas operaciones KOP comprenden secuencias de instrucciones AWL que se pueden interrumpir, estas operaciones de transferencia equivalen a una sola instrucción AWL, cuya ejecución no se ve afectada por los eventos de interrupción.
- En un programa AWL o KOP que comparta varias variables: Si los datos compartidos representan varios bytes, palabras o palabras dobles contiguas, la ejecución de la rutina de interrupción se puede controlar con las operaciones Habilitar todos los eventos de interrupción (ENI) e Inhibir todos los eventos de interrupción (DISI). Las interrupciones se deben inhibir en aquel punto del programa principal donde figuran las operaciones que acceden a las direcciones compartidas. Una vez ejecutadas todas las operaciones que utilicen las direcciones compartidas, se deberán habilitar de nuevo las interrupciones. Mientras esté inhibida la interrupción no se podrá ejecutar la rutina correspondiente. Por tanto, no será posible acceder entonces a las direcciones compartidas. Sin embargo, esta técnica de programación puede causar que se ignoren los eventos de interrupción.

Llamar a subrutinas desde rutinas de interrupción

Desde una rutina de interrupción se puede llamar a un nivel de anidado de subrutinas. Los acumuladores y la pila lógica son compartidos por la rutina de interrupción y por la subrutina invocada.

Interrupciones soportadas por el S7-200

El S7-200 soporta los siguientes tipos de rutinas de interrupción:

- Interrupciones del puerto de comunicación: El S7-200 genera eventos que le permiten al programa controlar el puerto de comunicación.
- Interrupciones de E/S: El S7-200 genera eventos para los diferentes cambios de estado de diversas entradas y salidas. Estos eventos le permiten al programa reaccionar a los contadores rápidos, a las salidas de impulsos, o bien a los flancos positivos o negativos en las entradas.
- Interrupciones temporizadas: El S7-200 genera eventos que le permiten al programa reaccionar a determinados intervalos.

Interrupciones del puerto de comunicación:

El programa puede controlar el puerto serie de comunicación del S7-200. La comunicación a través de este puerto se denomina modo Freeport (comunicación programable por el usuario). En modo Freeport, el programa define la velocidad de transferencia, los bits por carácter, la paridad y el protocolo. Las interrupciones de

transmisión y recepción permiten controlar la comunicación mediante el programa. Para obtener más información al respecto, consulte la descripción de las operaciones Transmitir mensaje y Recibir mensaje.

Interrupciones de E/S

Las interrupciones de E/S abarcan interrupciones al producirse flancos positivos y negativos, interrupciones de los contadores rápidos, así como interrupciones de salidas de impulsos. El S7-200 puede generar una interrupción en los flancos positivos y/o negativos de una entrada (bien sea I0.0, I0.1, I0.2, o bien I0.3). Los eventos Flanco positivo y Flanco negativo se pueden capturar para cada una de dichas entradas. Estos eventos también sirven para indicar una condición que requiera atención inmediata en cuanto se produzca el evento.

Las interrupciones de los contadores rápidos permiten responder rápidamente a condiciones tales como: a) el valor actual ha alcanzado el valor predeterminado, b) el sentido de contaje ha cambiado de forma inversa al sentido de giro del árbol de accionamiento y c) el contador se ha puesto a "0" externamente. Todos estos eventos de los contadores rápidos permiten reaccionar ante eventos que no se puedan controlar durante el tiempo de ciclo del sistema de automatización.

Las interrupciones de salida de impulsos avisan inmediatamente cuándo ha finalizado la salida del número indicado de impulsos. Por lo general, las salidas de impulsos se utilizan para controlar motores paso a paso.

Todas estas interrupciones se habilitan asociando una rutina de interrupción al evento de E/S en cuestión.

Interrupciones temporizadas

Las interrupciones temporizadas comprenden también las de los temporizadores T32/T96. Estas interrupciones se utilizan para indicar tareas que deban ejecutarse cíclicamente. El tiempo de ciclo se incrementa en intervalos de 1 ms, abarcando desde 1 ms hasta 255 ms. El tiempo de ciclo de la interrupción temporizada 0 se debe escribir en SMB34, y el de la interrupción temporizada 1, en SMB35.

Cada vez que termina la temporización, el evento de interrupción temporizado transfiere el control a la rutina de interrupción correspondiente. Por lo general, las interrupciones temporizadas se utilizan para controlar el muestreo de las entradas analógicas o para ejecutar un bucle PID en intervalos regulares.

Asociando un evento de interrupción temporizado a una rutina de interrupción, se habilita el evento e inmediatamente se empieza a temporizar. Durante ese proceso, el sistema captura el valor del tiempo de ciclo, de forma que los cambios siguientes en SMB34 y SMB35 no lo pueden alterar. Para poder modificar el tiempo de ciclo se deberá cambiar el valor del mismo y reasociar luego la rutina de interrupción al evento de la interrupción temporizada. Al reasociar la rutina de interrupción, la función borra los tiempos acumulados de la asociación anterior, con lo cual se vuelve a temporizar a partir del nuevo valor.

Una vez habilitada, la interrupción temporizada funciona de forma continua, ejecutando la rutina asociada cada vez que transcurre el intervalo de tiempo indicado. La interrupción temporizada se inhibe saliendo del modo RUN o desasociándola de la rutina correspondiente. Si se ejecuta la operación Inhibir todos los eventos de interrupción, se siguen generando interrupciones temporizadas, pero se ponen en cola de espera (hasta que se habiliten nuevamente o hasta llenarse la cola).

Las interrupciones de los temporizadores T32 y T96 permiten reaccionar una vez transcurrido un determinado intervalo de tiempo. Estas interrupciones se soportan

únicamente en T32 y T96, siendo éstos temporizadores como retardo a la conexión (TON) con resolución de 1 ms. Por lo demás, T32 y T96 disponen de las funciones habituales. Una vez habilitada la interrupción, la rutina asociada se ejecutará cuando el valor actual del temporizador activo sea igual a su valor de preselección al actualizar el S7-200 el temporizador de 1 ms. Estas interrupciones se habilitan asociando una rutina de interrupción a los eventos de interrupción T32/T96.

Prioridades de las interrupciones y colas de espera

El S7-200 procesa las interrupciones según su prioridad dentro de su respectivo grupo de prioridad. Sólo se ejecuta una rutina de interrupción en cada caso. Cuando se comienza a ejecutar una rutina de interrupción, ésta se procesa hasta el final. No puede ser interrumpida por otra, ni siquiera por una rutina de mayor prioridad. Las interrupciones que aparezcan mientras se esté ejecutando otra interrupción se ponen en cola de espera para ser procesadas posteriormente.

La tabla 6-48 muestra las tres colas de espera y el número máximo de interrupciones que pueden acoger.

Tabla 6-48 Número máximo de interrupciones en las colas de espera

Cola de espera	CPU 221, CPU 222,	CPU 224XP y CPU 226
Interrupciones de	4	8
Interrupciones de E/S	16	16
Interrupciones temporizadas	8	8

Es posible que se presenten más interrupciones de las que puede acoger la cola de espera. Por esta razón, el sistema dispone de marcas de desbordamiento que indican qué eventos de interrupción no se han podido acoger en la cola de espera. La tabla 6-49 muestra las marcas de desbordamiento correspondientes. Éstas sólo se pueden utilizar en una rutina de interrupción, porque se desactivan tras vaciarse la cola de espera y al reanudar la ejecución del programa principal.

En la tabla 6-50 figuran todas las interrupciones, sus prioridades y los números de los eventos asociados.

Tabla 6-49 Marcas de desbordamiento de la cola de espera

Descripción (0 = sin desbordamiento, 1 =	Marca especial
Interrupciones de comunicación	SM4.0
Interrupciones de E/S	SM4.1
Interrupciones temporizadas	SM4.2

Tabla 6-50 Prioridades de los eventos de interrupción

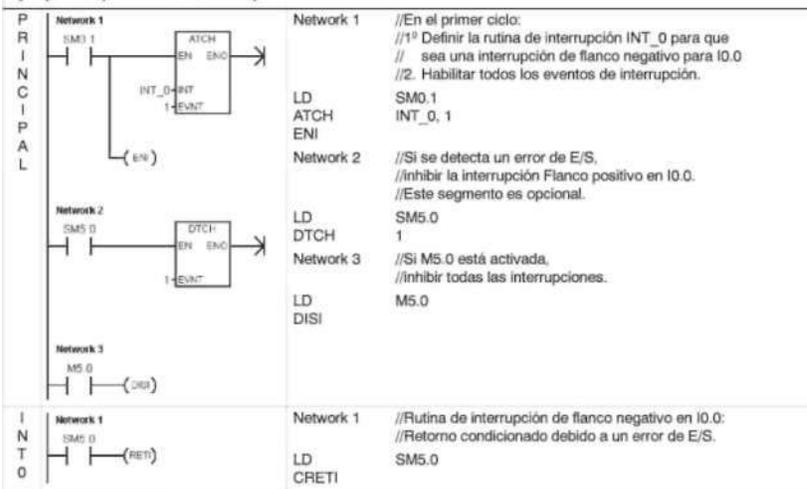
Evento	Descripción	Prioridad	Grupo de
8	Puerto 0 Recibir carácter	Comunicación Prioridad más alta	0
9	Puerto 0 Transmisión finalizada		0
23	Puerto 0 Recepción de mensajes finalizada		0
24	Puerto 1 Recepción de mensajes finalizada		1
25	Puerto 1 Recibir carácter		1
26	Puerto 1 Transmisión finalizada		1
19	PLS0 Interrupción Valor de conteo de	Digital	0
20	PLS1 Interrupción Valor de conteo de		1

impulsos PTO

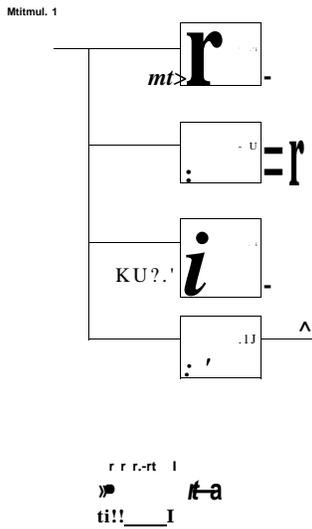
0	I0.0	Flanco positivo	Prioridad media	2
2	I0.1	Flanco positivo		3
4	I0.2	Flanco positivo		4
6	I0.3	Flanco positivo		5
1	I0.0	Flanco negativo		6
3	I0.1	Flanco negativo		7
5	I0.2	Flanco negativo		8
7	I0.3	Flanco negativo		9
12	HSC0	CV=PV (valor actual = valor		10
27	HSC0	Cambio de sentido		11
28	HSC0	Puesto a 0 externamente		12
13	HSC1	CV=PV (valor actual = valor		13
14	HSC1	Cambio de sentido		14
15	HSC1	Puesto a 0 externamente		15
16	HSC2	CV=PV (valor actual = valor		16
17	HSC2	Cambio de sentido		17
18	HSC2	Puesto a 0 externamente		18
32	HSC3	CV=PV (valor actual = valor		19
29	HSC4	CV=PV (valor actual = valor		20
30	HSC4	Cambio de sentido		21
31	HSC4	Puesto a 0 externamente		22
33	HSC5	CV=PV (valor actual = valor		23
10	Interrupción temporizada 0	SMB34		Temporización Prioridad más baja
11	Interrupción temporizada 1	SMB35	1	
21	Interrupción temporizador T32	CT=PT	2	
22	Interrupción temporizador T96	CT=PT	3	

baja

Ejemplo de operaciones de interrupción



Ejemplo Borrar evento de interrupción



```

Network 1 // Asistente HSC
LD SMO.O
MOVB 16#A0, SMB47
//Activar los bits de control:
//escribir el valor de preselección;

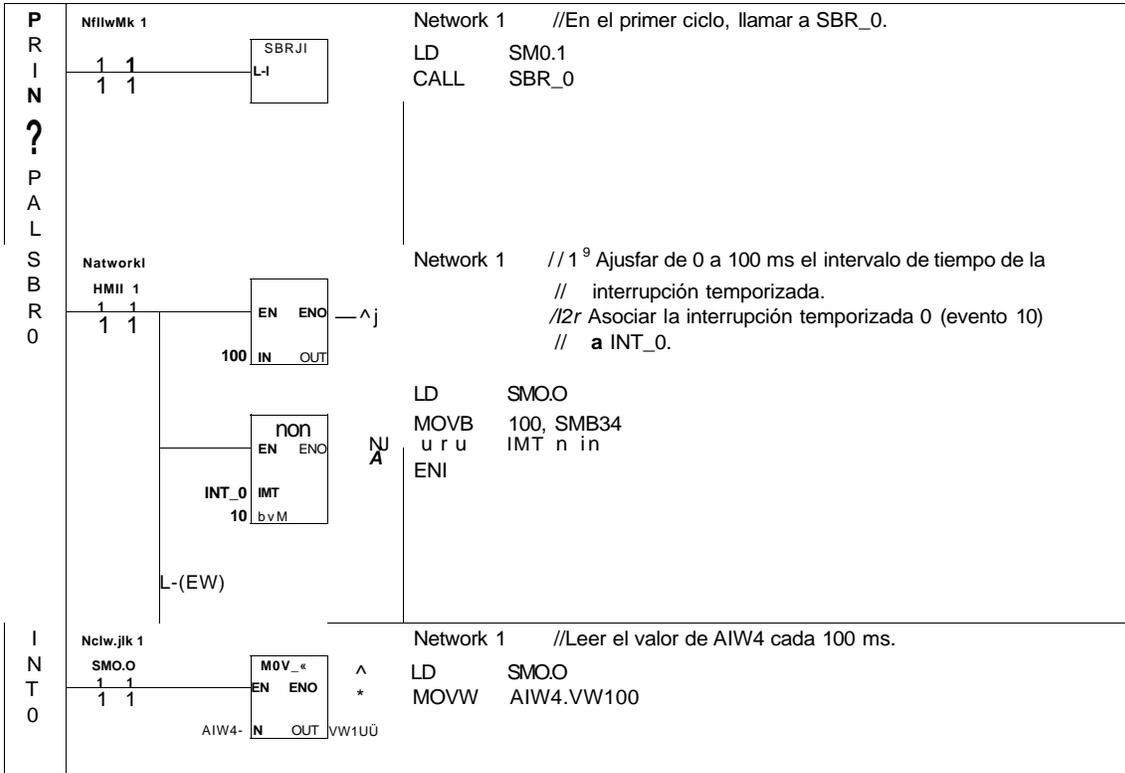
MOVD +6. SMD52
//PV = 6;

ATCH HSC1_STEP1, 13
//Interrupción HSC1_STEP1: CV = PV
//para HC1

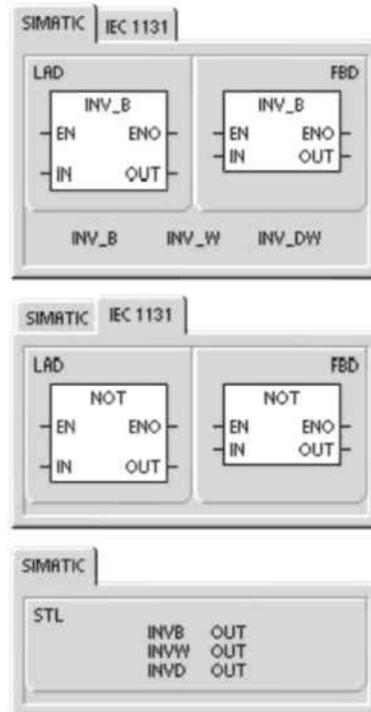
Network 2 //Borrar todas las interrupciones indeseadas
//causadas por la vibración de la máquina.

LD SMO.O
CEVNT 13
    
```

Ejemplo de una interrupción temporizada para leer el valor de una entrada analógica



Operaciones lógicas



Operaciones de invertir

Invertir byte, Invertir palabra e Invertir palabra doble

Las operaciones Invertir byte (INVB), Invertir palabra (IN VW) e Invertir palabra doble (INVD) forman el complemento a 1 de la entrada IN y cargan el resultado en la dirección de la memoria OUT.

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)

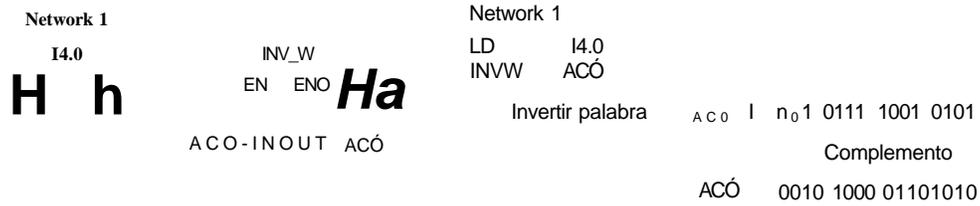
Marcas especiales afectadas:

- SM1.0 (cero)

Tabla 6-51 Operandos válidos para las operaciones de invertir

Entradas/salidas	Tipos	Operandos
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC,
	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC,
OUT	DWORD	AIW, *VD, *LD, *AC, constante
	BYTE	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD,
	WORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	DWORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD,
	DWORD	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,

Ejemplo de la operación Invertir



Operaciones de combinación con Y, O y O-exclusiva

Combinación Y con bytes, con palabras y con palabras dobles

Las operaciones Combinación Y con bytes (ANDB), Combinación Y con palabras (ANDW) y Combinación Y con palabras dobles (ANDD) combina los bits correspondientes de dos valores de entrada IN1 e IN2 mediante Y, y cargan el resultado en una dirección de la memoria OUT.

Combinación O con bytes, con palabras y con palabras dobles

Las operaciones Combinación O con bytes (ORB), Combinación O con palabras (ORW) y Combinación O con palabras dobles (ORD) combinan los bits correspondientes de dos valores de entrada IN1 e IN2 mediante O y cargan el resultado en una dirección de la memoria OUT.

Combinación O-exclusiva con bytes, con palabras o con palabras dobles

Las operaciones Combinación O-exclusiva con bytes (XORB), Combinación O-exclusiva con palabras (XORW) y Combinación O-exclusiva con palabras dobles (XORD) combinan los bits correspondientes de dos valores de entrada (IN1 e IN2) mediante O-exclusiva y cargan el resultado en una dirección de la memoria OUT.

Marcas especiales y ENO

En todas las operaciones descritas en esta página, las condiciones siguientes afectan a las marcas especiales y a ENO:

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)

Marcas especiales afectadas:

- SM1.0 (cero)

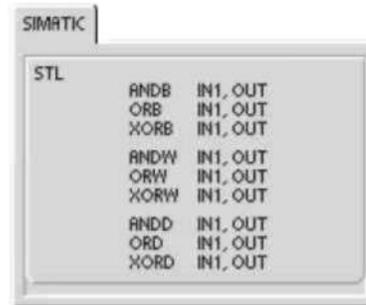
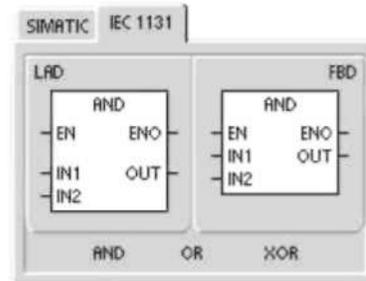
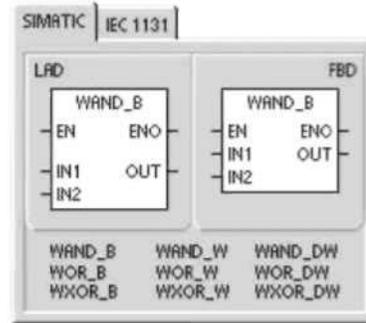
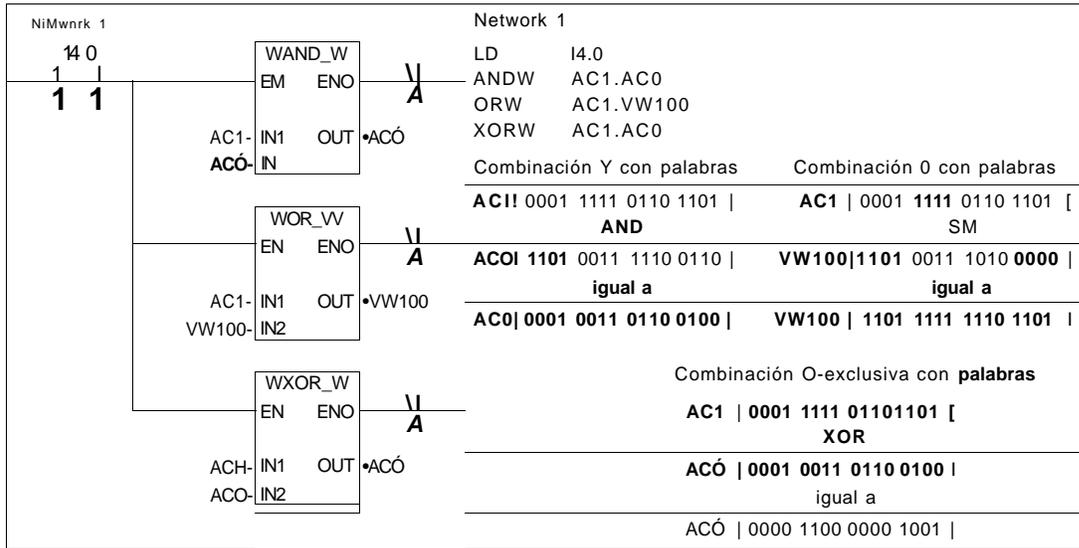


Tabla 6-52 Operandos válidos para las operaciones de combinación con Y, O y O-exclusiva

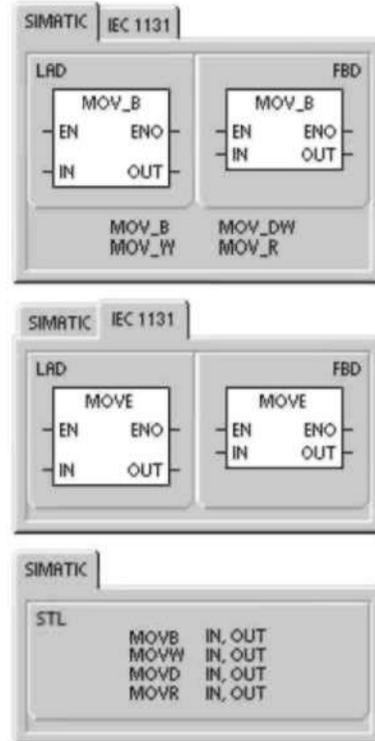
Entradas/salidas	Tipos	Operandos
IN1, IN2	BYTE WORD DWORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante
OUT	BYTE WORD DWORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *AC, *LD ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

Ejemplo Operaciones de combinación con Y, O y O-exclusiva



Operaciones de transferencia

Transferir bytes, palabras, palabras dobles y números reales



Las operaciones Transferir byte (MOVB), Transferir palabra (MOVW), Transferir palabra doble (MOVD) y Transferir real (MOVR) transfieren un valor de una dirección (IN) a una nueva dirección (OUT) sin modificar el valor original. Si desea crear un puntero, utilice la operación Transferir palabra doble. Para más información, consulte la descripción de los punteros y el direccionamiento indirecto en el capítulo 4.

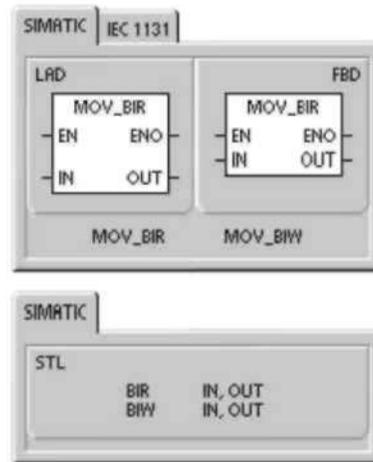
En el caso de la operación IEC Transferir (MOVE), los tipos de los datos de entrada y salida pueden ser diferentes, pero su tamaño debe ser igual.

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)

Tabla 6-53 Operandos válidos para las operaciones de transferencia

Entradas/sali	Tipos de datos	Operandos
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD,
	WORD, INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *AC, *LD, constante
	DWORD, DINT	ID, QD, VD, MD, SMD, SD, LD, HC, &VB, &IB, &QB, &MB, &SB, &T, &C, &SMB, &AIW, &AQW, AC, *VD, *LD, *AC,
OUT	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,
	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD,
	WORD, INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC
	DWORD, DINT, REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC



Transferir bytes directamente (lectura y escritura)

Las operaciones de transferencia directa de bytes permiten transferir directamente un byte entre la E/S física y una dirección de la memoria.

La operación Lectura y transferencia directa de bytes (BIR) lee la entrada física (IN) y escribe el resultado en la dirección de la memoria (OUT), sin actualizar la imagen del proceso.

La operación Escritura y transferencia directa de bytes (BIW) lee los datos de la dirección de la memoria (IN) y los escribe en una salida física (OUT), así como en la correspondiente dirección de la imagen del proceso.

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- Imposible acceder al módulo de ampliación

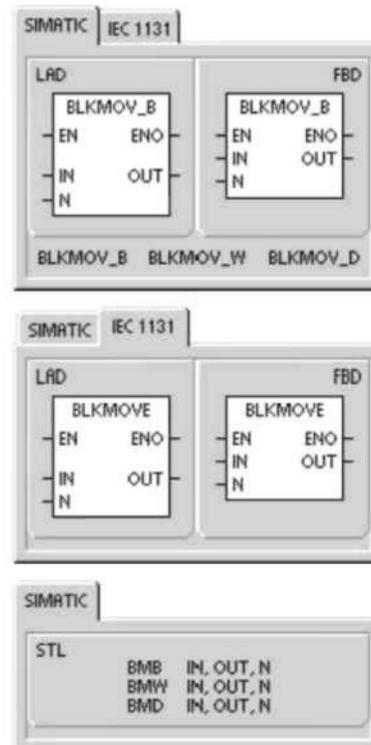
Tabla 6-54 Operandos válidos para la operación Lectura y transferencia directa de bytes

Entradas/salidas	Tipos de	Operandos
IN	BYTE	IB, *VD, *LD, *AC
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD,

Tabla 6-55 Operandos válidos para la operación Escritura y transferencia directa de bytes

Entradas/sali	Tipos de datos	Operandos
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD,
OUT	BYTE	QB, *VD, *LD, *AC

Operaciones de transferencia en bloque



Transferir bytes, palabras y palabras dobles en bloque

Las operaciones Transferir bytes en bloque (BMB), Transferir palabras en bloque (BMW) y Transferir palabras dobles en bloque (BMD) transfieren una determinada cantidad de datos a una nueva dirección de la memoria, transfiriendo el número de bytes, palabras o palabras dobles N a partir de la dirección de entrada IN hasta un nuevo bloque que comienza en la dirección de salida OUT

N puede estar comprendido entre 1 y 255.

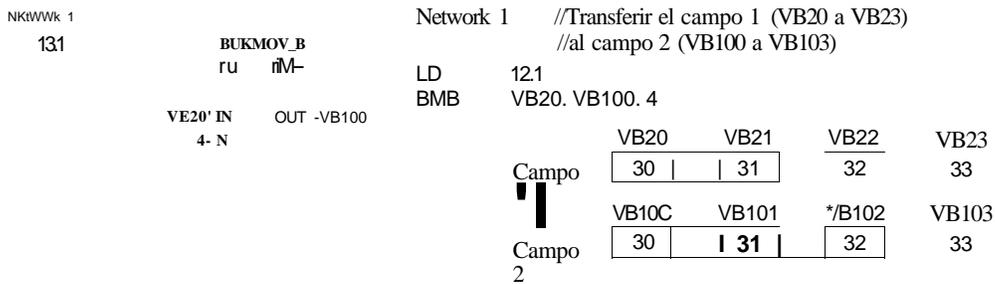
Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)

Tabla 6-56 Operandos válidos para las operaciones de transferencia en bloque

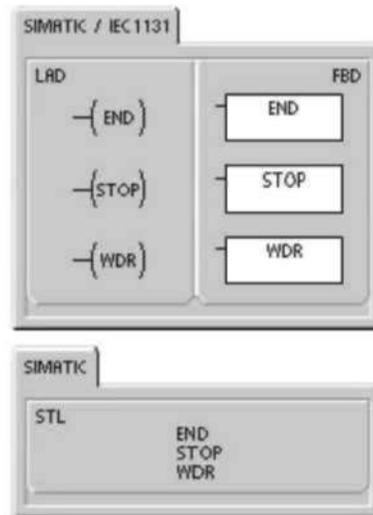
Entradas/salidas	Tipos	Operandos
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, *VD, *LD, *AC
	WORD, INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AIW, *VD, *LD, *AC
	DWORD,	*LD, *AC
OUT	BYTE	IB, QB, VB, MB, SMB , SB, LB, *VD, *LD, *AC
	WORD, INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AQW, *VD, *LD, *AC
	DWORD,	*VD, *LD, *AC
N	BYTE	IB, QB, VB, MB , SMB, SB, LB, AC, constante,

Ejemplo de la operación de Transferir bytes en bloque



Operaciones de control del programa Fin condicionado

La operación Fin condicionado (END) finaliza el ciclo actual en función de la



combinación lógica precedente. Esta operación se puede utilizar en el programa principal, mas no en subrutinas ni en rutinas de interrupción.

STOP

La operación STOP finaliza la ejecución del programa haciendo que la CPU S7-200 cambie de RUN a STOP.

Si la operación STOP se ejecuta en una rutina de interrupción, ésta se finalizará inmediatamente ignorando las interrupciones pendientes. Las demás acciones en el ciclo actual se completan, incluyendo la ejecución del programa principal. El cambio de RUN a STOP se produce al final del ciclo actual.

Borrar temporizador de vigilancia

La operación Borrar temporizador de vigilancia (WDR) redispara el temporizador de vigilancia de la CPU S7-200 para prolongar el tiempo de ciclo sin que se indique un error de vigilancia.

Esta operación se debe utilizar con gran precaución. Si se utilizan bucles para que no finalice el ciclo o para prolongarlo excesivamente, es posible que no se ejecuten los procesos siguientes hasta completar el ciclo:

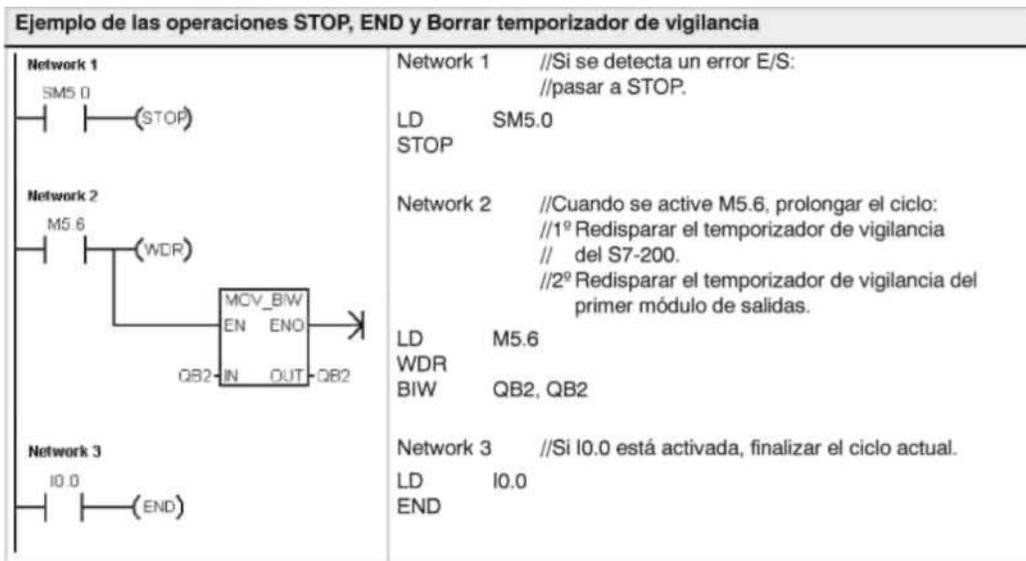
- Comunicación (excepto modo Freeport)
- Actualización de las entradas y salidas (exceptuando el control directo de las E/S)
- Actualización de los valores forzados
- Actualización de las marcas especiales (no se actualizan las marcas SM0 y SM5 a SM29)
- Tareas de diagnóstico en el tiempo de ejecución
- Los temporizadores con resolución de 10 ms y 100 ms no contarán correctamente los ciclos que excedan los 25 segundos.
- Operación STOP si se utiliza en una rutina de interrupción
- Los módulos de ampliación dotados con salidas digitales también incorporan un temporizador de vigilancia que desactiva las salidas si el S7-200 no escribe en el módulo. Utilice una operación de escritura directa en cada módulo de ampliación dotado con salidas digitales para conservar las salidas correctas en los tiempos de ciclo largos. Consulte el ejemplo que le sigue a esta descripción.



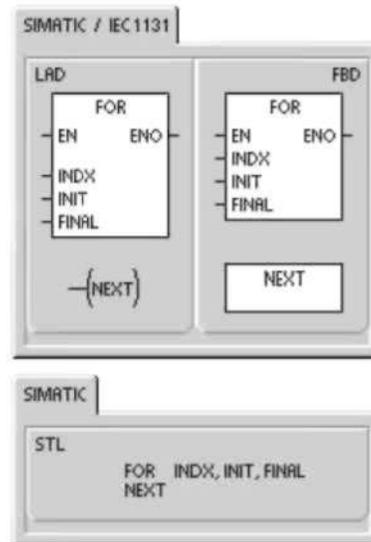
Consejo

Si es previsible que el tiempo de ciclo durará más de 500 ms o que la actividad de interrupción aumentará de manera que el ciclo principal quede interrumpido más de 500 ms, es preciso utilizar la operación WDR para redisparar el temporizador de vigilancia. Cuando utilice la operación Borrar temporizador de vigilancia (WDR), es recomendable que programe también una operación de escritura directa en un byte de salida (QB) de todos los módulos de ampliación para poner a "0" los temporizadores de vigilancia de todos ellos.

Si utiliza la operación Borrar temporizador de vigilancia para poder ejecutar un programa que tenga un tiempo de ciclo prolongado y el selector de modos de operación se pone en posición STOP, el S7-200 cambiará a modo STOP al cabo de 1,4 segundos.



FOR y NEXT



Las operaciones FOR y NEXT repiten un bucle del programa un número determinado de veces. Toda operación FOR exige una operación NEXT. Los bucles FOR/NEXT pueden anidarse (insertar un bucle FOR/NEXT dentro de otro) hasta una profundidad de ocho niveles.

La operación FOR ejecuta las operaciones que se encuentren entre FOR y NEXT. Es preciso indicar el valor del índice o el conteo actual del bucle (INDX), el valor inicial (INIT) y el valor final (FINAL).

La operación NEXT marca el final del bucle FOR.

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)

Al habilitar el bucle FOR/NEXT, éste se ejecutará hasta finalizar las iteraciones, a menos que el usuario cambie el valor final dentro del bucle. Los valores se pueden modificar mientras se ejecute el bucle FOR/NEXT. Si se vuelve a habilitar el bucle, éste

copiará el valor inicial (INIT) en el valor actual de contaje del bucle (IDX).

La operación FOR/NEXT se desactivará automáticamente la próxima vez que se habilite.

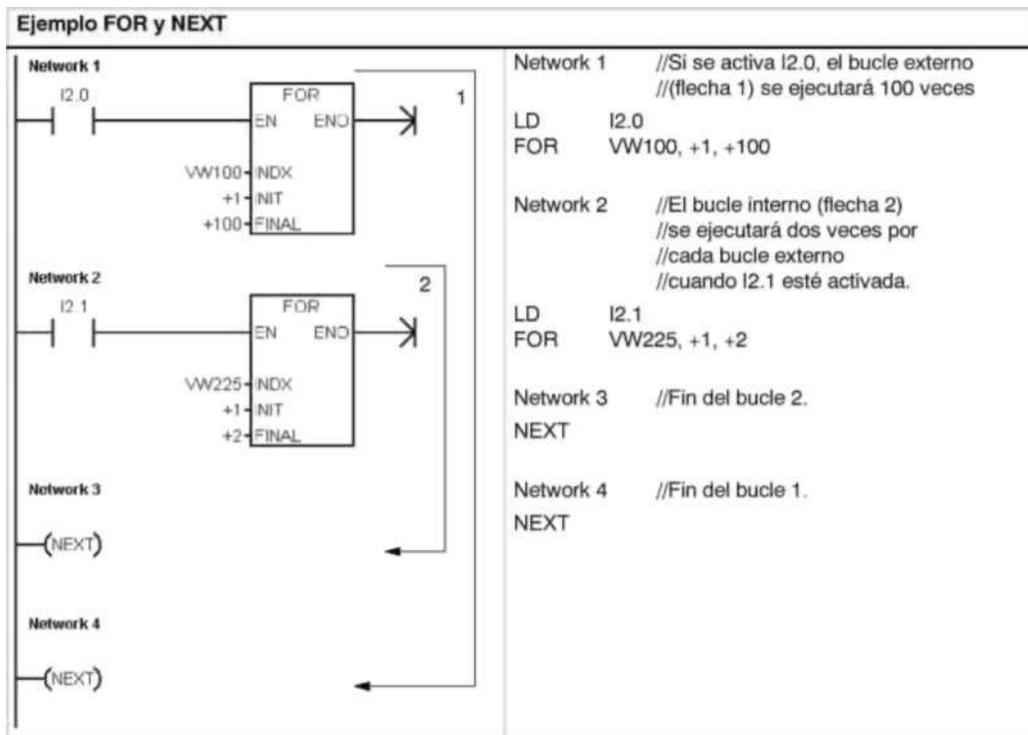
Ejemplo: si el valor INIT = 1 y si el valor FINAL = 10, las operaciones que se encuentren entre FOR y NEXT se ejecutarán 10 veces, incrementando el valor de contaje INDX en 1, 2, 3, ...10.

Si el valor inicial es mayor que el valor final, no se ejecutará el bucle. Tras ejecutarse las operaciones que se encuentran entre FOR y NEXT, se incrementará el valor de INDX y el resultado se comparará con el valor final. Si INDX es mayor que el valor final, finalizará el bucle.

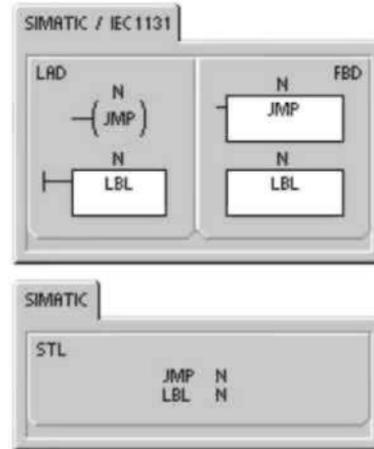
Si el primer nivel de la pila es 1 cuando el programa comienza el bucle FOR/NEXT, seguirá siendo 1 cuando el programa salga del bucle.

Tabla 6-57 Operandos válidos para las operaciones FOR y NEXT

Entradas/salida	Tipos de	Operandos
INDX	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC,
INIT, FINAL	INT	VW, IW, QW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante



Operaciones de salto



La operación Saltar a meta (JMP) deriva la ejecución del programa a la meta indicada N.

La operación Definir meta (LBL) indica la dirección de la meta de salto N.

La operación Saltar a meta se puede utilizar en el programa principal, en las subrutinas o en las rutinas de interrupción. La operación de salto y la meta correspondiente deben encontrarse siempre en el mismo segmento lógico (es decir, bien sea en el programa principal, en la subrutina, o bien en la rutina de interrupción).

Desde el programa principal no se puede saltar a una meta que se encuentre en una subrutina o en una rutina de interrupción. Tampoco es posible saltar desde una subrutina o una rutina de interrupción a una meta que se encuentre fuera de ella.

La operación Saltar a meta se puede utilizar en un segmento SCR, pero la correspondiente operación Definir meta debe encontrarse en ese mismo segmento SCR.

Tabla 6-58 Operandos válidos para las operaciones de salto

Entradas/sal	Tipos de	Operandos
N	WORD	Constante (0 a

Ejemplo de la operación Saltar a meta

Network 1		Network 1	//Si no se han perdido datos remanentes, //saltar a LBL4
SM0.2	4	LDN	SM0.2
/ I-	-(JMP)	JMP	4
		Network 2	
Network 2		LBL	4
		LBL	

L

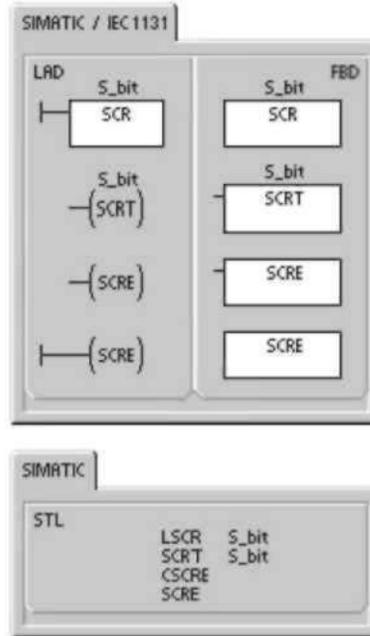
Operaciones del relé de control secuencial (SCR)

Las operaciones del relé de control secuencial (SCR) ofrecen un método de control de estado sencillo pero eficaz que se adapta naturalmente a los programas KOP, FUP y AWL.

Siempre que la aplicación comprenda una secuencia de operaciones que se deban repetir una y otra vez, los relés de control secuencial (SCRs) se pueden utilizar para estructurar el programa, de manera que éste corresponda directamente a la aplicación. Ello permite programar y comprobar la aplicación de forma más rápida y sencilla.

La operación Cargar relé de control secuencial (LSCR) carga el valor del bit S indicado por la operación N en la pila del relé de control secuencial (pila SCR) y en la pila lógica.

El segmento SCR se activa o se desactiva en función del resultado de la pila SCR. El valor de la pila SCR se copia en el nivel superior de la pila lógica, pudiéndose conectar directamente los cuadros y las bobinas a la barra de alimentación izquierda sin necesidad de interconectar un contacto.



Restricciones

Al utilizar los relés de control secuencial es preciso tener en cuenta las restricciones siguientes:

- Un mismo bit S no se puede utilizar en más de una rutina. Por ejemplo, si S0.1 se utiliza en el programa principal, no se podrá utilizar además en una subrutina.
- No es posible saltar hacia adentro ni hacia afuera de un segmento SCR. No obstante, las operaciones Saltar a meta y Definir meta se pueden utilizar para saltar segmentos SCR, o bien en un segmento SCR.
- En un segmento SCR no se puede utilizar la operación END. Tabla 6-59 Operandos válidos para las operaciones del relé de control secuencial

Entradas/salid Tipos de datos

S_BIT	BOOL	S
-------	------	---

La figura 6-32 muestra la pila SCR y la pila lógica, así como los efectos de la operación Cargar relé de control secuencial. Tenga en cuenta lo siguiente al utilizar las operaciones del relé de control secuencial:

- La operación Cargar relé de control secuencial (LSCR) indica el comienzo de un segmento SCR, en tanto que la operación Fin del relé secuencial (SCRE) señala el fin de un segmento SCR. La ejecución de todas las operaciones que se encuentren entre la operación LSCR y la operación SCRE depende del valor de la pila SCR. Las operaciones que se encuentren entre la operación SCRE y la siguiente operación LSCR no dependen del valor de la pila SCR.

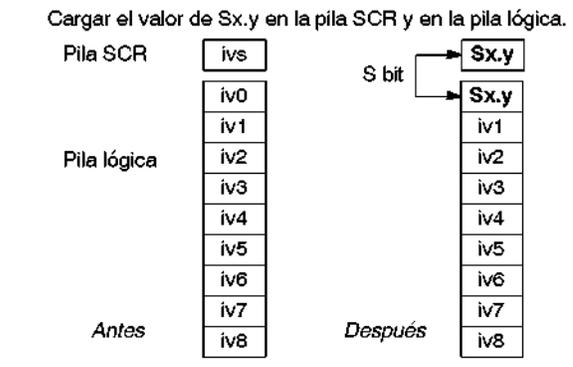


Figura 6-32 Efectos de la operación LSCR en la pila lógica

- La operación Transición del relé secuencial (SCRT) permite transferir el control de un segmento SCR activo a otro segmento SCR.

Si la operación SCRT se ejecuta con circulación de corriente, se desactiva el bit S del segmento activo actualmente y se activa el bit S del segmento direccionado. Al desactivarse el bit S del segmento activo, ello no tiene

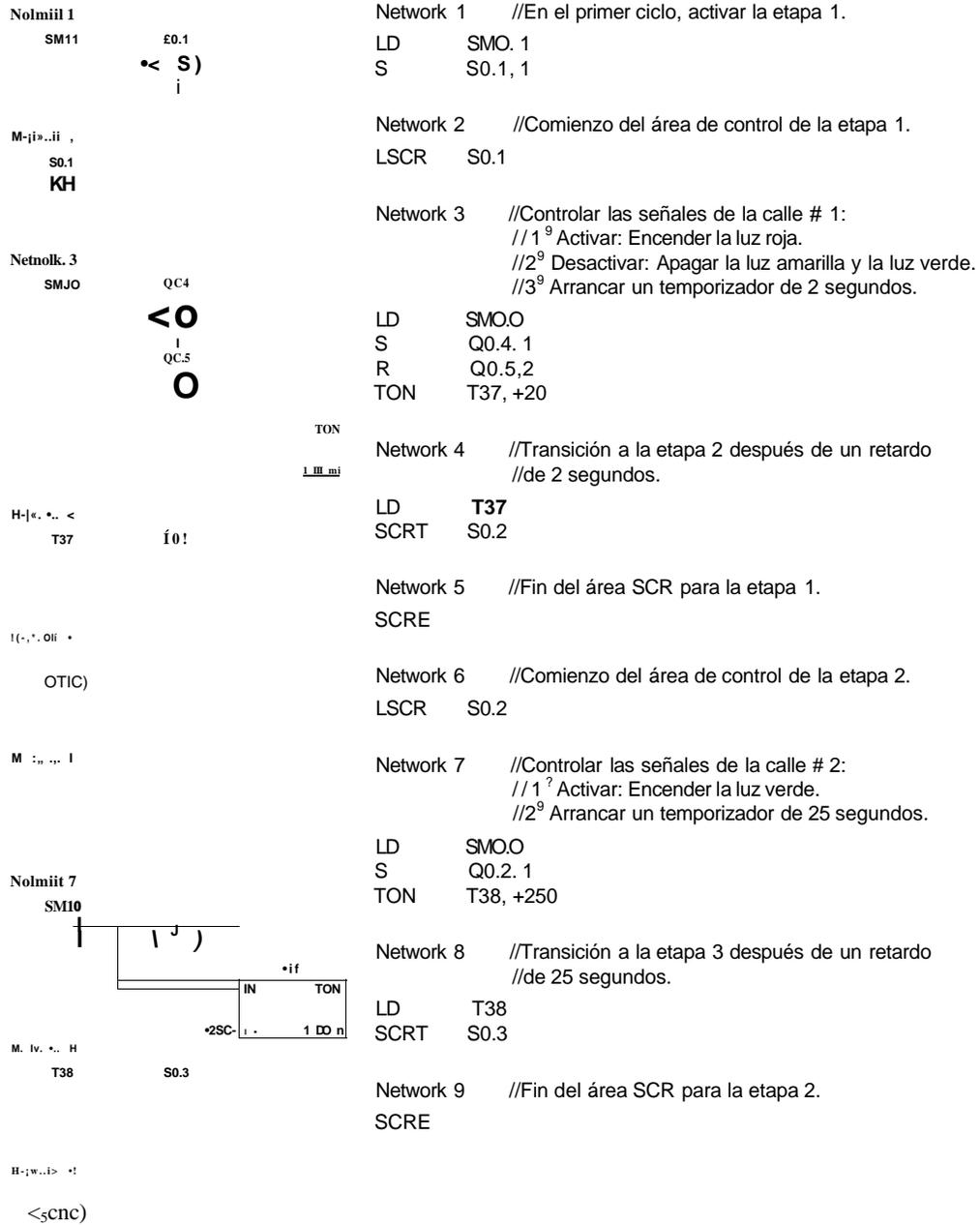
efecto alguno en la pila SCR

cuando se ejecuta la operación Transición del relé secuencial. Por consiguiente, el segmento SCR permanece excitado hasta que se finaliza.

- La operación Fin condicionado del relé secuencial (CSCRE) permite salir de un segmento SCR activo sin ejecutar las operaciones que se encuentran entre las operaciones Fin condicionado del relé secuencial y Fin del relé secuencial. La operación CSCRE no afecta ningún bit S ni tampoco la pila SCR.

En el ejemplo siguiente, S0.1 se activa con la marca especial SM0.1 (marca del primer ciclo). S0.1 será entonces la etapa 1 activa en el primer ciclo. Una vez transcurrido un retardo de 2 segundos, T37 provoca una transición a la etapa 2. Esta transición desactiva el segmento SCR (S0.1) de la primera etapa y activa el segmento SCR (S0.2) de la segunda etapa.

Ejemplo de operaciones de control secuencial



Dividir cadenas secuenciales

En numerosas aplicaciones es necesario dividir una cadena secuencial en dos o más cadenas. Si una cadena secuencial se divide en varias cadenas, es preciso activar simultáneamente todas las nuevas cadenas secuenciales, como muestra la figura 6-33.

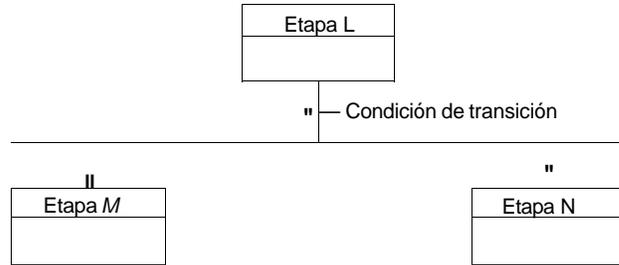
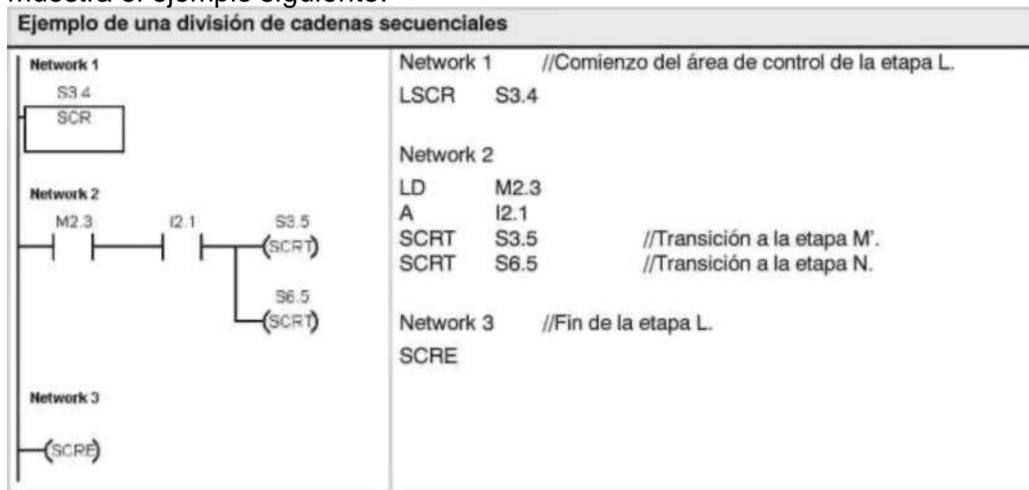


Figura 6-33 División de una cadena secuencial

La división de cadenas secuenciales se puede implementar en un programa SCR, activando varias operaciones SCRT con una misma condición de transición, como muestra el ejemplo siguiente.



Convergir cadenas secuenciales

Algo similar a la división de cadenas secuenciales ocurre cuando dos o varias cadenas secuenciales deban convergir en una sola. Todas las cadenas secuenciales se deben terminar antes de poder ejecutar la siguiente etapa. La figura 6-34 muestra la convergencia de dos cadenas secuenciales.

La convergencia de cadenas secuenciales se puede implementar en un programa SCR creando una transición de la etapa L a la etapa L, y de la etapa M a la etapa M. Si los bits SCR que representan L y M son verdaderos, se podrá habilitar la etapa N como lo muestra el siguiente ejemplo.

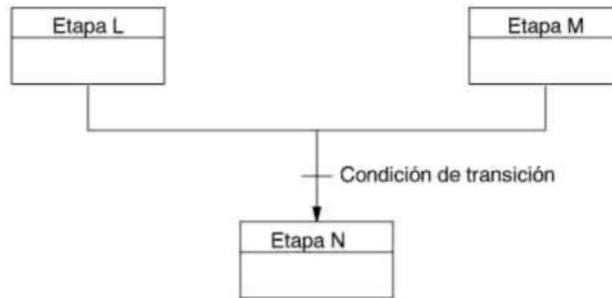
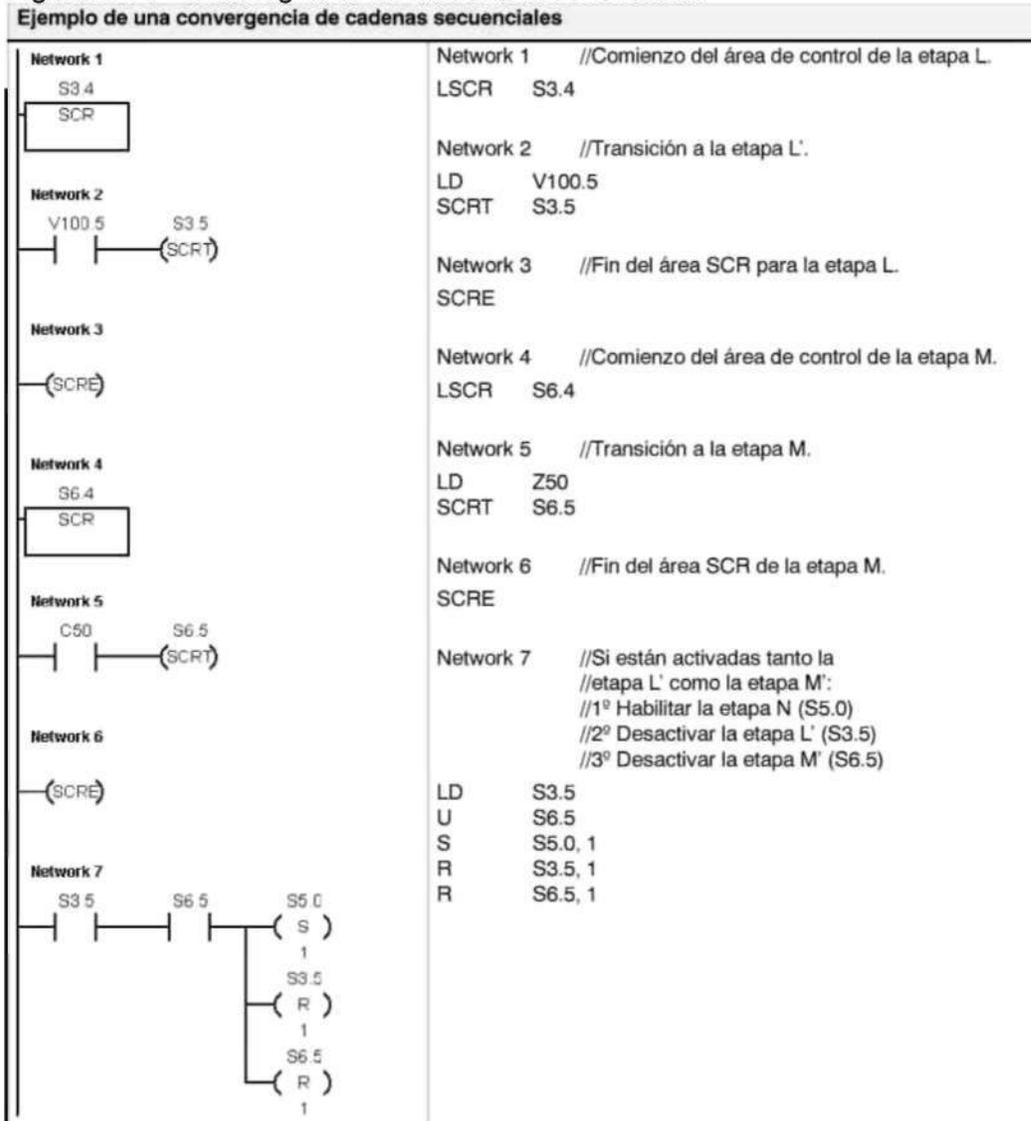


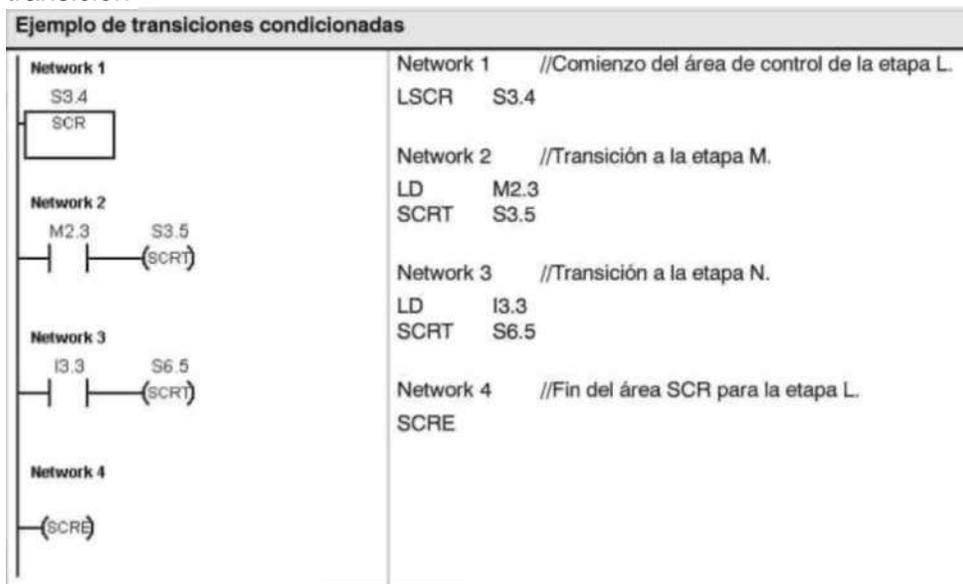
Figura 6-34 Convergencia de una cadena secuencial



En otras situaciones, una cadena secuencial se puede dirigir a una de varias cadenas secuenciales posibles, dependiendo de la primera condición de transición que sea verdadera. La figura 6-35 muestra esa situación en un programa SCR equivalente.



Figura 6-35 Dirigir una cadena secuencial a otra, dependiendo de la condición de transición



LED de diagnóstico

El LED de diagnóstico está apagado si el valor del parámetro de entrada IN es "0". Este LED se enciende (en color amarillo) si el valor del parámetro de entrada IN es mayor que "0".

El diodo luminoso (LED) "SF/ DIAG" se puede configurar para que se encienda en color amarillo cuando las condiciones especificadas en el bloque de sistema sean verdaderas o cuando la operación DIAG_LED se ejecute con un parámetro IN que no sea "0".

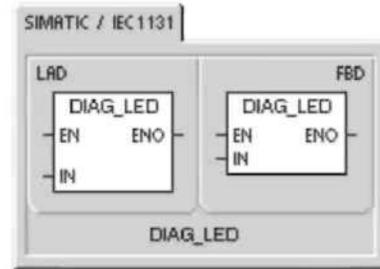
Opciones en el bloque de sistema para configurar el LED:

- El LED "SF/ DIAG" se debe encender (en amarillo) cuando un elemento se fuerce en la CPU.
- El LED "SF/ DIAG" se debe encender (en amarillo) cuando ocurra un error de E/S en un módulo.

Desactive ambas opciones si desea que la operación DIAG_LED sea la única que controle la iluminación del LED "SF/ DIAG". Si se produce un fallo del sistema (SF) en la CPU, el LED se encenderá en color rojo.

Tabla 6-60 Operandos válidos para la operación LED de diagnóstico

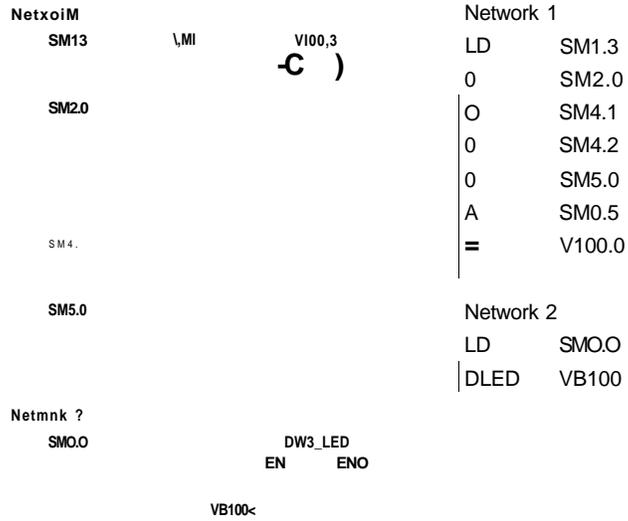
Entradas/salidas	Tipos	Operandos
IN	BYTE	VB, IB, QB, MB, SB, SMB, LB, AC, constante, *VD, *LD, *AC



Ejemplo 1 de la operación "LED de diagnóstico"

El LED de diagnóstico debe parpadear si se detecta un error.

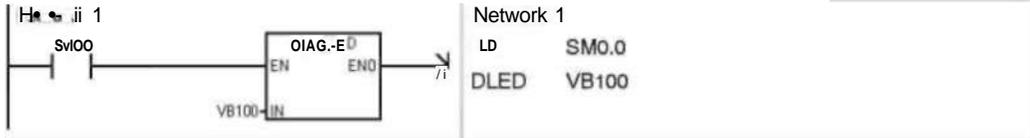
El LED de diagnóstico debe parpadear cada vez que se detecte cualquiera de las 5 condiciones de error.



Ejemplo 2 de la operación "LED de diagnóstico"

El LED de diagnóstico se debe encender cuando se devuelva un error.

El LED de diagnóstico se debe encender cuando se indique un código de error en VB10.



Operaciones de desplazamiento y rotación

Desplazar a la derecha y Desplazar a la izquierda

Las operaciones de desplazamiento desplazan el valor de entrada IN a la derecha o a la izquierda tantas posiciones como indique el valor de desplazamiento N y cargan el resultado en la salida OUT.

Las operaciones de desplazamiento se rellenan con ceros cada vez que se desplaza un bit. Si el valor de desplazamiento (N) es mayor o igual al valor máximo permitido (8 en las operaciones con bytes, 16 en las operaciones con palabras y 32 en las operaciones con palabras dobles), se desplazará el valor máximo permitido para la operación en cuestión. Si el valor de desplazamiento es mayor que 0, la marca de desbordamiento (SM1.1) adoptará el valor del último bit desplazado hacia afuera. La marca cero (SM1.0) se activará si el resultado de la operación de desplazamiento es cero. Las operaciones de desplazamiento de bytes no llevan signo. En el caso de las operaciones con palabras y con palabras dobles, el bit de signo se desplaza cuando se utilizan tipos de datos con signo.

Condiciones de error que ponen Marcas especiales

ENO a 0:

- 0006 (direccionamiento indirecto)
- SM1.0(cero)

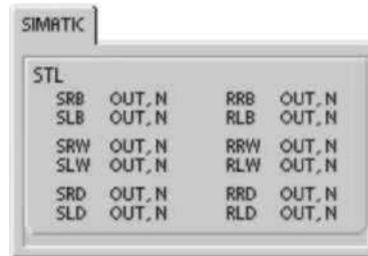
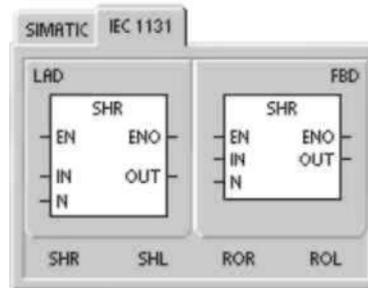
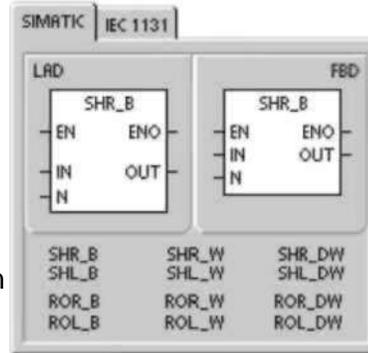
afectadas:

- SM1.1 (desbordamiento)

Rotar a la derecha y Rotar a la izquierda

Las operaciones de rotación rotan el valor de entrada (IN) a la derecha o a la izquierda tantas posiciones como indique el valor de desplazamiento (N) y cargan el resultado en la dirección de la memoria (OUT). La rotación es circular.

Si el valor de desplazamiento es mayor o igual al valor máximo permitido (8 en las operaciones con bytes, 16 en las operaciones con palabras y 32 en las operaciones con palabras dobles), el S7-200 ejecutará una operación módulo en el valor de desplazamiento para obtener un valor válido antes de ejecutarse la rotación. De ello resulta un valor de desplazamiento de 0 a 7 en las operaciones con bytes, de 0 a 15 en las operaciones con palabras y de 0 a 31 en las operaciones con palabras dobles.



Si el valor de desplazamiento es igual a 0, no se rotará el valor. Si se ejecuta la rotación, el valor del último bit rotado se copiará en la marca de desbordamiento (SM1.1).

Si el valor de desplazamiento no es un entero múltiplo de 8 (en las operaciones con bytes), de 16 (en las operaciones con palabras) o de 32 (en las operaciones con palabras dobles), el último bit rotado se copiará en la marca de desbordamiento (SM1.1). La marca cero (SM1.0) se activará si el valor a rotar es igual a cero.

Las operaciones de desplazamiento de bytes no llevan signo. En el caso de las operaciones con palabras y con palabras dobles, el bit de signo se desplaza cuando se utilizan tipos de datos con signo.

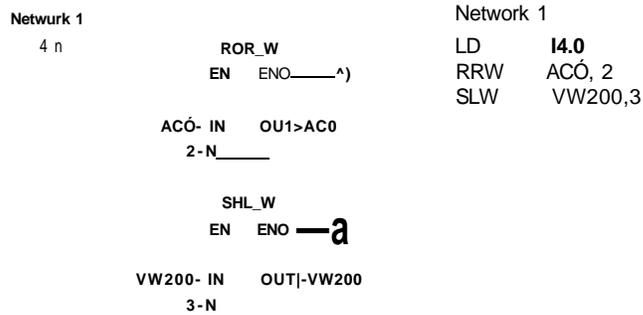
Condiciones de error que ponen Marcas especiales afectadas:

- ENOa0:**
- SM1.0(cero)
 - 0006 (direccionamiento indirecto)
 - SM1.1 (desbordamiento)

Tabla 6-61 Operandos válidos para las operaciones de desplazamiento y rotación

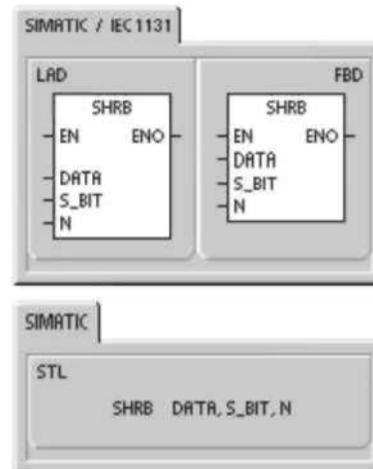
Entradas/sal	Tipos de	Operandos
IN	BYTE WORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante
	DWORD	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD,
OUT	BYTE WORD	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,
	DWORD	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD,
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante

Ejemplo Operaciones de desplazamiento y rotación



Rotación	Antes de la rotación	Rebase por exceso	Desplazamiento	Antes del desplazamiento	Rebase por exceso
ACÓ	0100 0000 0000 0001	S	VW200	111000101010 1101	S
	Después de la primera rotación	Rebase por exceso		Después del primer desplazamiento	Rebase por exceso
ACÓ	1010 0000 0000 0000 1	*Q]	VW200	<u>1100 0101 0101 1010</u>	U f T]
	Después de la segunda rotación	Rebase por exceso		Después del segundo desplazamiento	Rebase por exceso
ACÓ	0101 0000 0000 0000	*R)	VW200	<u>1000 10101011 0100</u>	L » [T]
Marca cero (SM1.0)	. 0	. 0		Después del tercer desplazamiento	Rebase por exceso
Marca de desbordamiento (SM1.1)	0	0	VW200	<u>0001 pipi olio bop</u>	L » f T]
				Marca cero (SM1.0)	= 0
				Marca de desbordamiento (SM1.1)	= 1

Registro de desplazamiento



La operación Registro de desplazamiento desplaza un valor al registro de desplazamiento. La operación Registro de desplazamiento permite secuenciar y controlar fácilmente el flujo de productos o de datos. Esta operación se debe utilizar para desplazar todo el registro un bit en cada ciclo.

La operación Registro de desplazamiento desplaza el valor de DATA al registro de desplazamiento. S_BIT señala el bit menos significativo de este registro. N indica la longitud del registro y el sentido de desplazamiento (valor positivo = N, valor negativo = -N).

Los bits desplazados por la operación Registro de desplazamiento se depositan en la marca de desbordamiento (SM1.1).

El registro de desplazamiento está definido por el bit menos significativo (S_BIT) y por el número de bits indicados por la longitud (N).

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)
- 0092 (error en campo de contaje)

Marcas especiales afectadas:

- SM1.1 (desbordamiento)

Tabla 6-62 Operandos válidos para la operación Registro de desplazamiento

Entradas/sali	Tipos de	Operandos
DATA, S_BIT	BOOL	I, Q, V, M, SM, S, T, C, L
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante

La dirección del bit más significativo del registro de desplazamiento (MSB.b) se calcula con la ecuación siguiente:

$$\text{MSB.b} = [(\text{byte de S_BIT}) + ([N] - 1 + (\text{bit de S_BIT})) / 8].[\text{resto de la división por } 8]$$

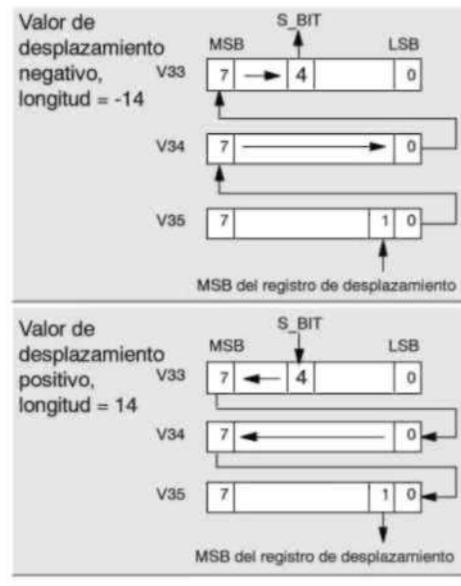


Figura 6-36 Entrada y salida en el registro de desplazamiento

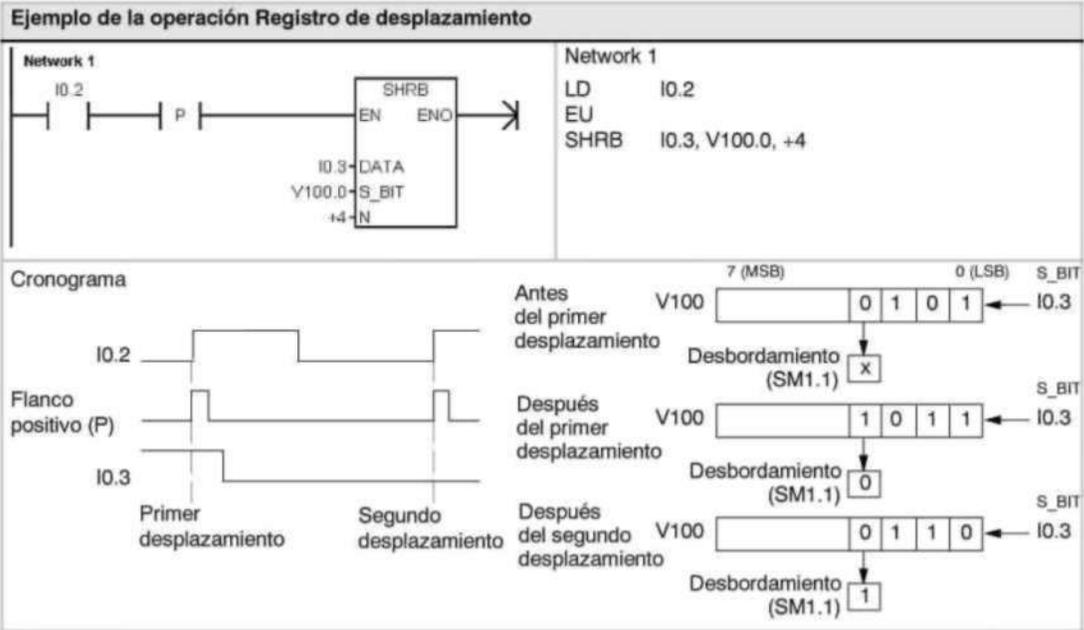
Por ejemplo, si S_BIT es V33.4 y N es 14, el bit MSB.b será V35.1.

$$\begin{aligned} \text{MSB.b} &= V33 + ([14]-1 + 4)/8 \\ &= V33 + 17/8 \\ &= V33 + 2 \text{ con un resto "1"} = V35.1 \end{aligned}$$

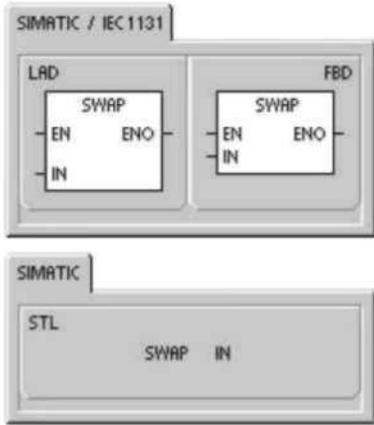
Si el valor de desplazamiento es negativo, es decir, si la longitud (N) indicada es negativa, los datos de entrada se desplazarán desde el bit menos significativo (S_BIT) al bit más significativo del registro de desplazamiento. Los datos desplazados se depositan en la marca de desbordamiento (SM1.1).

Si el valor de desplazamiento es positivo, es decir, si la longitud (N) indicada es positiva, los datos de entrada (DATA) se desplazarán desde el bit más significativo al bit menos significativo (indicado por S_BIT) del registro de desplazamiento. Los datos desplazados se depositan en la marca de desbordamiento (SM1.1).

El registro de desplazamiento puede tener una longitud máxima de 64 bits (positiva o negativa). La figura 6-36 muestra el desplazamiento de bits de un valor N positivo y de un valor N negativo.



Invertir bytes de una palabra



La operación Invertir bytes de una palabra intercambia el byte más significativo y el byte menos significativo de una palabra IN.

Condiciones de error que ponen ENO a 0:

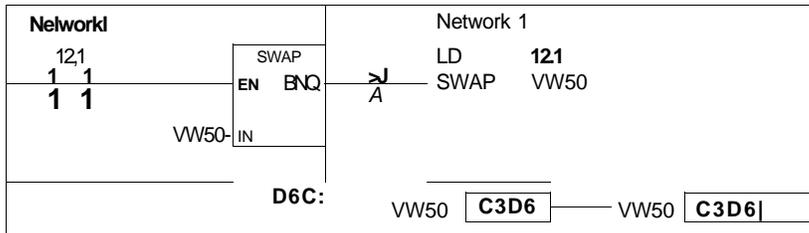
- 0006 (direccionamiento indirecto)

Tabla 6-63 Operandos válidos para la operación Invertir bytes de una palabra

Entradas/salidas Tipos Operandos

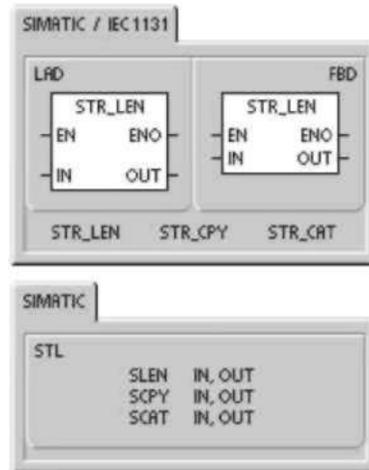
IN _____ IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD,

Ejemplo de la operación Invertir bytes de una palabra



Operaciones con cadenas Longitud de cadena

La operación Longitud de cadena (SLEN) indica la longitud de la cadena



especificada por IN.

Copiar cadena

La operación Copiar cadena (SCPY) copia la cadena indicada por IN en la cadena indicada por OUT.

Concatenar cadena

La operación Concatenar cadena (SCAT) cuelga la cadena indicada por IN al final de la cadena indicada por OUT.

Marcas especiales y ENO

Las condiciones siguientes afectan a ENO en las operaciones Longitud de cadena, Copiar cadena y Concatenar cadena.

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0091 (error de rango)

Tabla 6-64 Operandos válidos para la operación Longitud de cadena

Entradas/salidas	Tipos	Operandos
IN		VB, LB, *VD, *LD, *AC, cadena de constante
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD,

Tabla 6-65 Operandos válidos para las operaciones Copiar cadena y Concatenar cadena

Entradas/salidas	Tipos	Operandos
IN	STRING	VB, LB, *VD, *LD, *AC, cadena de constante
OUT	STRING	VB, LB, *VD, *AC, *LD

Ejemplo de las operaciones Concatenar cadena, Copiar cadena y Longitud de cadena

```

Network 1 //1º Colgar la cadena en "WORLD"
           // a la cadena en VB0.
           //2º Copiar la cadena en VB0
           // a la nueva cadena en VB100.
           //3º Leer la longitud de la cadena
           // que comienza en VB100.

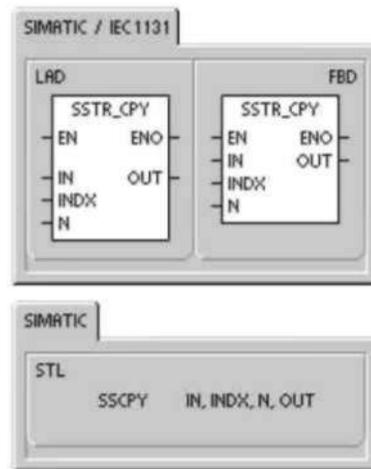
LD      10.0
SCAT   "WORLD", VB0
STRCPY VB0, VB100
STRLEN VB100, AC0
    
```

Antes de ejecutar el programa

VB0		VB6
	6 'H' 'E' 'L' 'L' 'O' ' '	

Tras ejecutar el programa

VB0		VB11
	11 'H' 'E' 'L' 'L' 'O' ' ' 'W' 'O' 'R' 'L' 'D'	
VB100		VB111
	11 'H' 'E' 'L' 'L' 'O' ' ' 'W' 'O' 'R' 'L' 'D'	
AC0		
	11	



Copiar subcadena de cadena

La operación Copiar subcadena de cadena (SSCPY) copia el número indicado de caracteres N de la cadena indicada por IN, comenzando en el índice (INDX), a la nueva cadena indicada por OUT.

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0091 (error de rango)
- 009B (índice=0)

Tabla 6-66 Operandos válidos para la operación Copiar subcadena de cadena

Entradas/salidas	Tipos	Operandos
IN	STRING	VB, LB, *VD, *LD, *AC, cadena de constante
OUT	STRING	VB, LB, *VD, *LD, *AC
INDX, N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD,

Ejemplo de la operación Copiar subcadena

```

Network 1 //Comenzando en el 7º carácter de
           //la cadena en VBO. copiar 5 caracteres
           //a la nueva cadena en VB20
10 :      SSTR_CPY
           EN  ENO
           VBO- IN  OUT ^ 2 0
           7- INDX
           5- N_____
           LD      I0.0
           SSCPYP VBO, 7. 5. VB20

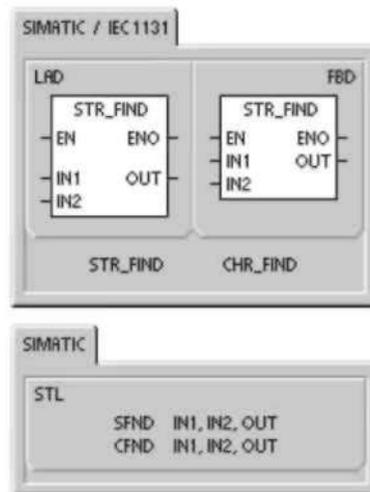
```

Antes de ejecutar el programa

VBO	VB11
11	H' E L 1' 0' •• W 'O' R L 'D'

Tras ejecutar el programa

VB20	VB25
5	W O' 'R' V D'



Buscar cadena en cadena

La operación Buscar cadena en cadena (SFND) busca la primera aparición de la cadena IN2 dentro de la cadena IN1. La búsqueda comienza en la posición inicial indicada por OUT. Si se encuentra una secuencia de caracteres que coincida exactamente con la cadena IN2, la posición del primer carácter de esa secuencia se escribirá en OUT. Si la cadena IN2 no se encuentra en la cadena IN1, OUT se pondrá a "0".

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0091 (error de rango)
- 009B (índice = 0)

Buscar carácter en cadena

La operación Buscar carácter en cadena (CFND) busca en la cadena IN1 hasta encontrar la primera aparición de un carácter cualquiera que pertenezca al juego de caracteres descrito en la cadena IN2. La búsqueda comienza en la posición inicial indicada por OUT. Si el carácter se encuentra en IN1, la posición del carácter se escribirá en OUT. Si no se encuentra el carácter, OUT se pondrá a "0".

Condiciones de error que ponen ENO a 0:

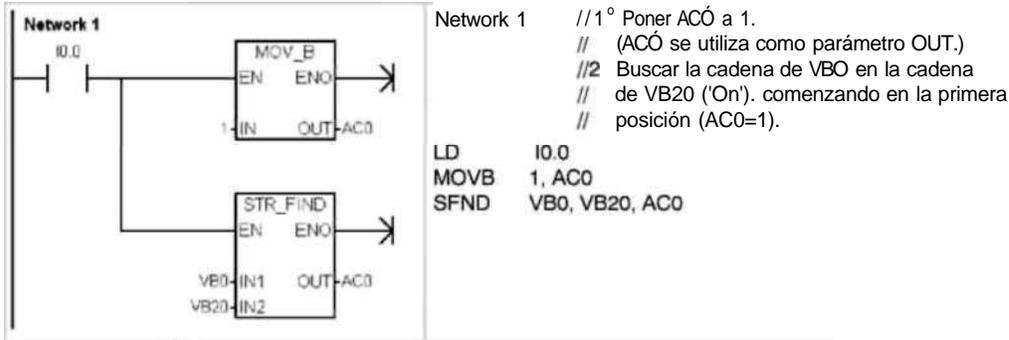
- 0006 (direccionamiento indirecto)
- 0091 (error de rango)
- 009B (índice = 0)

Tabla 6-67 Operandos válidos para las operaciones Buscar cadena en cadena y Buscar carácter en cadena

Entradas/salidas	Tipos	Operandos
IN1, IN2		VB, LB, *VD, *LD, *AC, cadena de
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Ejemplo de la operación Buscar cadena en cadena

El ejemplo siguiente utiliza una cadena almacenada en VBO como comando para conectar ('On') o desconectar ('Off') una bomba. Una cadena 'On' se almacena en VB20 y una cadena 'Off' se almacena en VB30. El resultado de la operación Buscar cadena en cadena se almacena en ACÓ (que representa el parámetro OUT). Si el resultado no es 0, significa que la cadena 'On' se ha encontrado en la cadena de comando (VB12).



VBO												VB12															
12	T	'u'	t	'n'		F	'u'	m'	'p'		0'	W															
VB20				VB22				VB30				VB33															
				3				'O' T 'f																			

Si la cadena se encuentra en VB20:

ACÓ

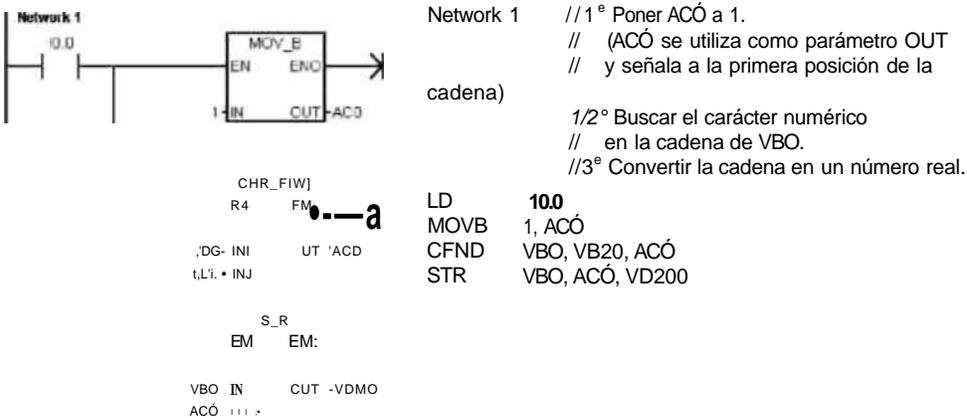
11

Si la cadena no se encuentra en VB20:

ACÓ

Ejemplo de la operación Buscar carácter en cadena

En el ejemplo siguiente, una cadena almacenada en VBO contiene la temperatura. La cadena en VB20 almacena todos los valores numéricos (y los signos "+" y "-") que pueden indicar la temperatura en una cadena. El programa de ejemplo busca la posición inicial de un número en esa cadena y convierte luego los caracteres numéricos en un número real. VD200 almacena el valor de número real correspondiente a la temperatura.



VBO												VB11												
11	T	'e'	'm'	'p'	'•'	'•'	'g'	'8'	'/	'6'	F													
VB20												VB32												
12	T	'2'	'3'	'4'	'5'	'6'	7	'8'	'9'	'0'	V	'.'												

Posición Inicial de la temperatura almacenada en VBO:

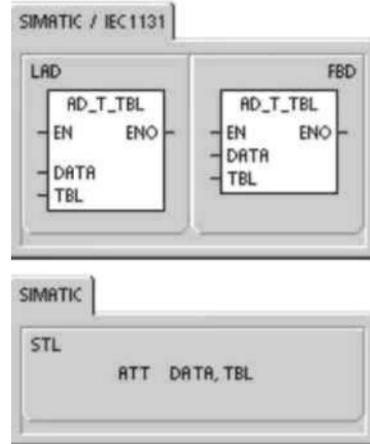
ACÓ

Valor de número real de la temperatura:

VD200

Operaciones de tabla Registrar valor en tabla

La operación Registrar valor en tabla registra valores de palabra (DATA) en una tabla (TBL). El primer valor de la tabla indica la longitud máxima de la misma (TL). El segundo valor (EC) indica el número de registros que contiene la tabla. Los



nuevos datos se agregan al final de la tabla, debajo del último registro. Cada vez que se agrega un registro a la tabla, se incrementa el número efectivo de registros. Una tabla puede tener como máximo 100 registros.

Condiciones de error que ponen ENO a 0:

- SM1.4 (desbordamiento de tabla)
- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)

Marcas especiales afectadas:

- SM1.4 se activa si se intenta introducir demasiados registros en la tabla.

Tabla 6-68 Operandos válidos para las operaciones de tabla

Entradas/sali	Tipos de	Operandos
DATOS	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, *A AC, AIW, *VD, *LD, constante C,
TBL	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, *VD, *LD, *AC

Ejemplo de la operación Registrar valor en tabla

```

SMO 1
H      !/OV_W
      EN  ENÜ

      ijijl -WWUU

Helwok 2
10.0   AD_T_TBL
      EN  ENO_____^|

      VW1UU UAIA
      VW20Ü TBL
    
```

```

Network 1 //Cargar la longitud máxima de la tabla
LD      SMO.1
MOVW   +6. VW200
    
```

```

Network 2
LD      IO.0
ATT     VW100,VW200
    
```

Antes de la operación ATT

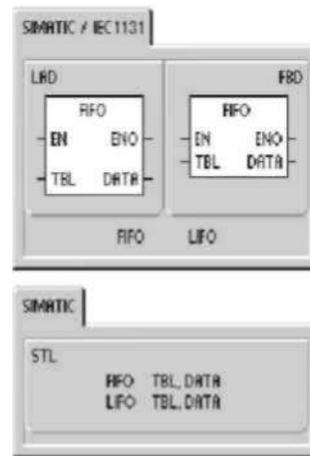
VW200	Kl'h	TL (n* max de registros)
VW202	000,"	EC (n' de registros)
VW204	5-131	dO (datos 0)
VW206	8942	d1 (datos 1)
VW208	xxxx	
VW210	xxxx	
VW212	xxxx	
VW214	xxxx	

Después de la operación ATT

VW200	"0006-	TL (n' máx. de registros)
VW202		EC (n' de registros)
VW204		dO (datos 0)
VW206		d1 (datos 1)
-VW208		d2 (datos 2)
VW210		
VW212		
VW214		

Borrar primer registro de la tabla y Borrar último registro de la tabla

Una tabla puede tener como máximo 100 registros.



Borrar primer registro de la tabla

La operación Borrar primer registro de la tabla (FIFO) transfiere el registro más antiguo (es decir, el primer registro) de una tabla a la dirección de salida, quitando el primer registro de la tabla (TBL) y transfiriendo el valor a la dirección indicada por DATA. Todos los demás registros se desplazan una posición hacia arriba. El número de registros (EC) de la tabla se decrementa cada vez que se ejecuta esta operación.

Borrar último registro de la tabla

La operación Borrar último registro de la tabla (LIFO) transfiere el registro más reciente (es decir, el último registro) de una tabla a la dirección de salida, quitando el último registro de la tabla (TBL) y transfiriendo el valor a la dirección indicada por DATA. El número de registros (EC) de la tabla se decrementa cada vez que se ejecuta esta operación.

Condiciones de error que ponen ENO a 0: Marcas especiales afectadas:

- SM1.5 (tabla vacía)
- SM1.5 se activa si se intenta borrar un registro de una

tabla vacía.

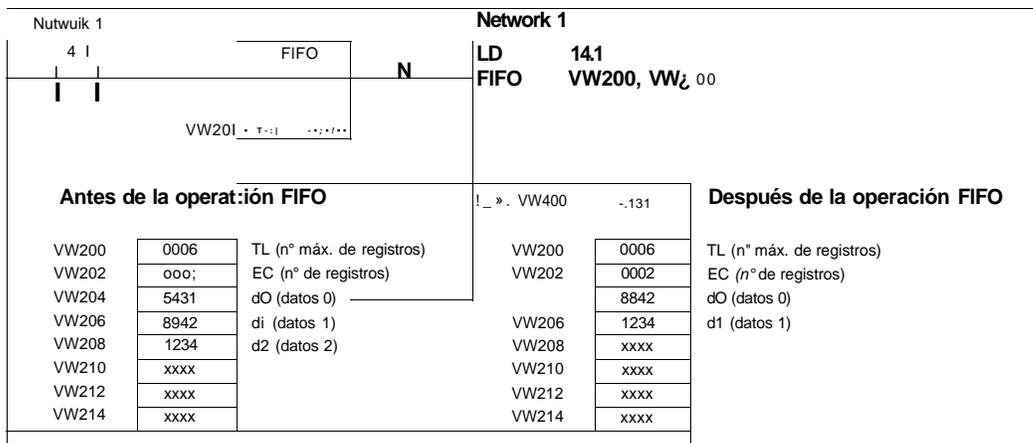
- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)

Tabla 6-69 Operandos válidos para las operaciones Borrar primer registro de la tabla y Borrar último registro de la tabla

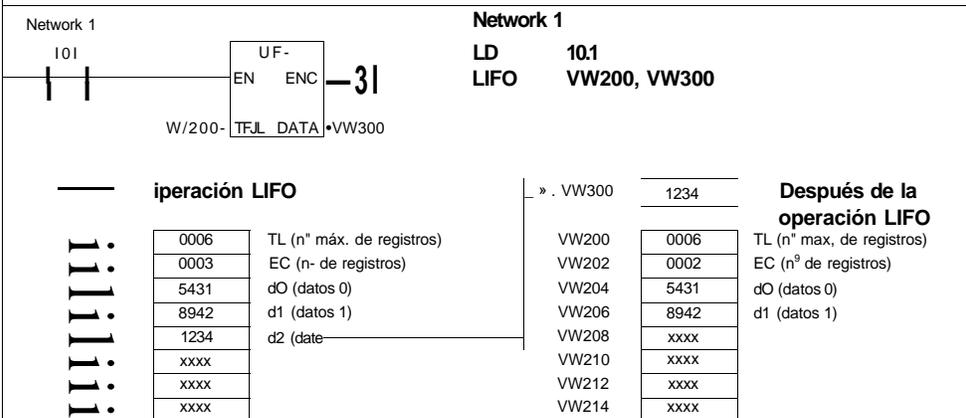
Entradas/salidas Tipos Operandos

TBL WORD IW, QW, VW, MW, SMW, SW, T, C, LW, *VD, *LD,
 DATOS INT IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW,

Ejemplo de la operación Borrar primer registro de la tabla



Ejemplo de la operación Borrar último registro de la tabla



Buscar valor en tabla

La operación Buscar valor en tabla (FND) busca en una tabla los datos que correspondan a determinados criterios. Esta operación rastrea la tabla TBL, comenzando en el registro indicado por INDX, y busca el valor o patrón PTN que corresponda a los criterios de búsqueda definidos por CMD. El parámetro de comando (CMD) indica un valor numérico comprendido entre 1 y 4 que corresponde a la relación =, <>, <, y >, respectivamente.

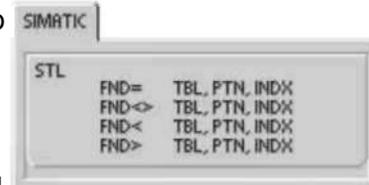
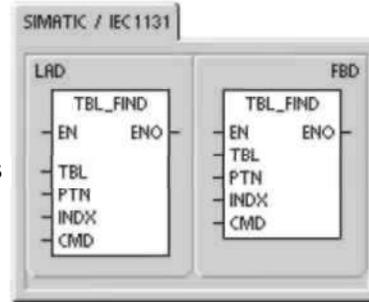
Si se cumple un criterio, INDX señalará el registro en cuestión. Para buscar el siguiente registro se habrá de incrementar INDX antes de volver a llamar nuevamente a la operación Buscar valor en tabla. Si no se encuentra ningún registro que corresponda al criterio, el valor INDX será igual al número de registros que contiene la tabla.

Una tabla puede tener como máximo 100 registros. Los registros de la tabla (el área donde se desea buscar) están numerados de 0 hasta el valor máximo (99).

Condiciones de error que ponen ENO a 0:

- 0006 (direccionamiento indirecto)
- 0091 (operando fuera de rango)

Tabla 6-71 Operandos válidos para la operación Buscar valor en tabla



Entradas/sali	Tipos de	Operandos
TBL	WORD	IW, QW, VW, MW, SMW, T, C, LW, *VD, *LD,
PTN	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante
INDX	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC,
CMD	BYTE	(constante) 1: igual (=), 2: diferente (<>), 3: menor que (<), 4: mayor que (>)

Consejo



Si la operación Buscar valor en tabla se utiliza en tablas creadas con las operaciones Registrar valor en tabla, Borrar último registro de la tabla y Borrar primer registro de la tabla, el valor de conteo corresponderá directamente al número de registros. La palabra que indica el número máximo de registros en las operaciones Registrar valor en tabla, Borrar último registro de la tabla y Borrar primer registro de la tabla no se necesita para la operación Buscar valor en tabla (v. fig. 6-37).

Por consiguiente, la dirección del operando TBL de una operación de búsqueda debería exceder en una palabra (dos bytes) al operando TBL correspondiente a la operación Registrar valor en tabla, Borrar último registro de la tabla o Borrar primer registro de la tabla.

Formato de tabla para ATT, LIFO y FIFO

VW200	0006	TL (nº máx. de registros)
VW202	0006	EC (nº de registros)
VW204	XXXX	d0 (datos 0)
VW206	XXXX	d1 (datos 1)
VW208	XXXX	d2 (datos 2)
VW210	XXXX	d3 (datos 3)
VW212	XXXX	d4 (datos 4)
VW214	XXXX	d5 (datos 5)

Formato de tabla para TBL_FIND

VW202	0006	EC (nº de registros)
VW204	XXXX	d0 (datos 0)
VW206	XXXX	d1 (datos 1)
VW208	XXXX	d2 (datos 2)
VW210	XXXX	d3 (datos 3)
VW212	XXXX	d4 (datos 4)
VW214	XXXX	d5 (datos 5)

Figura 6-37 Diferentes formatos de tabla entre la operación Buscar valor en tabla y las operaciones ATT, LIFO y FIFO

Ejemplo de la operación Buscar valor en tabla

Network 1

Network 1

LD I2.1
FND= VW202, 16#3130, AC1

Si I2.1 está activada, buscar un valor en la tabla que sea igual a 3130 HEX.

VW202	0006	EC (nº de registros)
VW204	3133	d0 (datos 0)
VW206	4142	d1 (datos 1)
VW208	3130	d2 (datos 2)
VW210	3030	d3 (datos 3)
VW212	3130	d4 (datos 4)
VW214	4541	d5 (datos 5)

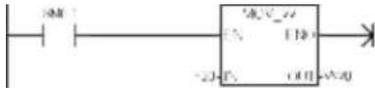
Si la tabla se creó utilizando las operaciones ATT, LIFO y FIFO, VW200 contendrá el número máximo de registros posibles y no será requerido por las operaciones de búsqueda.

AC1	<input type="text" value="0"/>	AC1 se debe poner a "0" para poder iniciar la búsqueda desde el primer registro de la tabla.
Buscar		
AC1	<input type="text" value="2"/>	AC1 contiene el número del primer registro que corresponde al criterio de búsqueda (d2).
Buscar		
AC1	<input type="text" value="3"/>	Incrementar INDX en pasos de 1 antes de buscar los demás registros de la tabla.
Buscar		
AC1	<input type="text" value="4"/>	AC1 contiene el número del registro que corresponde al segundo criterio de búsqueda (d4).
Buscar		
AC1	<input type="text" value="5"/>	Incrementar INDX en pasos de 1 antes de buscar los demás registros de la tabla.
Buscar		
AC1	<input type="text" value="6"/>	AC1 contiene un valor igual al número de registros. Se ha rastreado toda la tabla sin encontrar ningún otro registro que corresponda al criterio de búsqueda.
Buscar		
AC1	<input type="text" value="0"/>	Antes de que la tabla se pueda rastrear de nuevo, el valor de INDX debe ponerse a "0".

Ejemplo de cómo crear una tabla

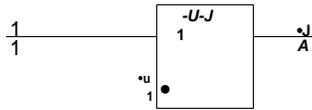
El programa siguiente sirve para crear una tabla que comprende 20 registros. La primera dirección de la tabla indica la longitud de la misma (en este ejemplo, 20 registros). La segunda dirección muestra el número actual de registros de la tabla. Las demás direcciones contienen los registros. Una tabla puede tener como máximo 100 registros, a excepción de los parámetros que definen la longitud máxima de la misma y el número actual de registros (en este ejemplo, VW0 y VW2). Con cada comando, la CPU incrementa o decrementa automáticamente el número real de registros de la tabla (en este ejemplo, VW2). Antes de trabajar con una tabla es preciso asignar el número máximo de registros. En caso contrario, no será posible hacer introducciones en la misma. Vigile también que todos los comandos de lectura y escritura se Para poder rastrear la tabla, el índice (VW106) se deberá poner a "0" antes de la operación de búsqueda. Si se encuentra una coincidencia, el índice indicará el número de registros de la tabla. En caso contrario, el índice concordará con el contaje actual de registros de la tabla (VW2).

Ejemplo de cómo crear una tabla



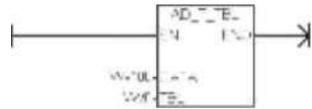
Network 1 //Crear una tabla con 20 registros
//a partir de la dirección 4.
//1^º En el primer ciclo, definir la
//longitud máxima de la tabla.

LD SM0.1
MOVW +20, VW0



Network 2 //Poner a "0" la tabla con la entrada I0.0
//En un flanco positivo de I0.0,
//rellenar con "+0" las direcciones
//de VW2 .

LD I0.0
EU
FILL +0, VW2, 21



Network 3 //Escribir el valor en la tabla con la
//entrada I0.1
//En un flanco positivo de I0.1,
//copiar el valor de la dirección
//VW100 a la tabla

LD I0.1
EU
ATT VW100, VW0



Network 4 //Leer el primer valor de la tabla con la
//entrada I0.2.
//Transferir el último valor de la tabla a
//VW102.

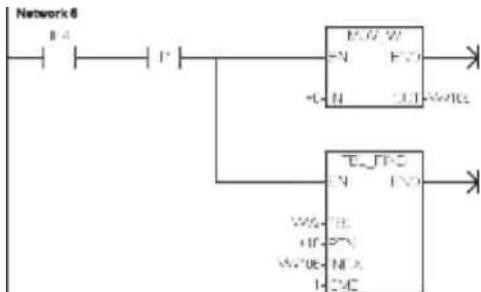
//Así se reduce el número de registros.
//En un flanco positivo de I0.2,
//Transferir el último valor de la tabla
//a VW102.

LD I0.2
EU
UFO VW0, VW102



Network 5 //Leer el último valor de la tabla
//con la entrada I0.3, Transferir el
//último valor de la tabla a VW102.
//Así se reduce el número de registros.
//En un flanco positivo de I0.0,
//Transferir el primer valor de la tabla
//a VW104.

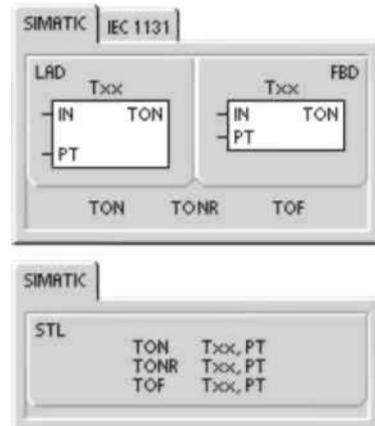
LD I0.3
EU
FIFO VW0, VW104



Network 6 //Buscar en la tabla la primera
//dirección que tenga el valor 10.
//1^º En un flanco positivo de I0.4,
// poner a "0" el puntero del índice.
//12º Buscar un registro de la tabla que
//sea igual a 10.

LD I0.4
EU
MOVW +0, VW106
FND= VW2, +10, VW106

Operaciones de temporización
Operaciones de temporización (SIMATIC)



Temporizador como retardo a la conexión Temporizador como retardo a la conexión con memoria

Las operaciones Temporizador como retardo a la conexión (TON) y Temporizador como retardo a la conexión con memoria (TONR) cuentan el tiempo al estar activada (ON) la entrada de habilitación. El número de temporizador (Txx) determina la resolución del mismo. Ésta se visualiza entonces en el cuadro de la operación.

Temporizador como retardo a la desconexión

El Temporizador como retardo a la desconexión (TOF) se utiliza para retardar la puesta a "0" (OFF) de una salida durante un período determinado tras haberse desactivado (OFF) una entrada. El número del temporizador (Txx) determina la resolución del mismo.

Tabla 6-72 Operandos válidos para las operaciones de temporización SIMATIC

Entradas/salidas	Tipos	Operandos
Txx	WORD	Constante (T0 a T255)
IN	BOOL	I, Q, V, M, SM, S, T, C, L, circulación de
PT	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante



Consejo

No se pueden utilizar números iguales (Txx) para un temporizador como retardo a la conexión (TON) y un temporizador como retardo a la desconexión (TOF). Por ejemplo, no puede haber tanto un TON T32 como un TOF T32.

Como muestra la tabla 6-73, los tres tipos de temporizadores ejecutan diferentes tareas de temporización:

- Los temporizadores como retardo a la conexión se utilizan para temporizar un solo intervalo.
- Los temporizadores como retardo a la conexión con memoria se utilizan para acumular varios intervalos temporizados.
- Los temporizadores con retardo a la desconexión se utilizan para ampliar el tiempo después de un cambio a OFF, por ejemplo, para enfriar un motor tras haber sido desconectado.

Tabla 6-73 Funcionamiento de las operaciones de temporización

Tipos de	Actual >= Preselección	Estado de la entrada de habilitación (IN)	Alimentación/p rimer ciclo
TON	Bit de temporización ON	ON: El valor actual cuenta el tiempo.	Bit de temporización OFF
TONR	El valor actual Bit de temporización ON	OFF: Bit de temporización ON: El valor actual cuenta el tiempo.	Valor actual = 0 Bit de temporización OFF
TOF	El valor actual Bit de temporización OFF. Valor actual = valor de	OFF: El bit de temporización ON: Bit de temporización ON. Valor actual = 0 OFF: El temporizador	El valor actual se Bit de temporización OFF Valor actual = 0

¹ El valor actual del temporizador como retardo a o de ^{Se} detiene el contador. ^{ON a OFF} para que quede memorizado cuando se interrumpe la alimentación. Para más información sobre el respaldo de la memoria de la CPU CP-200, consulte el capítulo 4.

Fⁿ ¿y) Los ejemplos de programación del CD de documentación contienen un programa que utiliza un ^{AA} temporizador como retardo a la conexión (TON) (consulte el ejemplo 31).

Tipos de

Operación.

Las operaciones TON y TONR cuentan el tiempo al estar activada (ON) la entrada de habilitación. Si el valor actual es mayor o igual al valor de preselección, se activará el bit de temporización (bit T).

- Cuando la entrada de habilitación está activada (OFF), el valor actual se borra en el temporizador TON. En cambio, se conserva en el temporizador TONR.
- El temporizador TONR sirve para acumular tiempo cuando la entrada se activa (ON) y se desactiva (OFF). Utilice la operación Poner a 0 (R) para borrar el valor actual del temporizador TONR.
- Tanto el temporizador TON como el temporizador TONR continúan contando tras haberse alcanzado el valor de preselección y paran de contar al alcanzar el valor máximo de 32.767.

La operación TOF se utiliza para retardar la puesta a "0" (OFF) de una salida

durante un período determinado tras haberse desactivado (OFF) una entrada. Cuando la entrada de habilitación se activa (ON), el bit de temporización se activa (ON) inmediatamente y el valor actual se pone a 0. Cuando la entrada se desactiva (OFF), el temporizador cuenta hasta que el tiempo transcurrido alcance el valor de preselección.

- Cuando se alcanza el valor de preselección, el bit de temporización (bit T) se desactiva y el valor actual no se incrementa más. Sin embargo, si la entrada se activa de nuevo antes de que el temporizador TOF alcance el valor de preselección, el bit de temporización permanecerá activado (ON).
- La entrada de habilitación debe cambiar de ON a OFF para que el temporizador TOF comience a contar intervalos de tiempo.
- Si un temporizador TOF se encuentra dentro de una sección SCR y ésta se encuentra desactivada, el valor actual se pone a 0, el bit de temporización se desactiva (OFF) y el valor actual no se incrementa.

Consejo

El temporizador TONR sólo se puede inicializar mediante la operación Poner a 0 (R).

Esta operación también sirve para reinicializar los temporizadores TON o TOF, arrojando los resultados siguientes:

- Bit de temporización = OFF
- Valor actual del temporizador = 0

Tras inicializarse un temporizador TOF, la entrada de habilitación debe cambiar de ON a OFF para poder rearrancar el temporizador.

Determinar la resolución de los temporizadores

Los temporizadores cuentan intervalos de tiempo. La resolución (o base de tiempo) del temporizador determina el lapso de tiempo de cada intervalo. Por ejemplo, un temporizador TON con una resolución de 10 ms cuenta el número de intervalos de 10 ms que han transcurrido desde que se habilitó el TON. Un valor de conteo de 50 en un temporizador de 10 ms equivale a 500 ms. Se dispone de temporizadores SIMATIC con tres resoluciones, a saber: 1 ms, 10 ms y 100 ms. Como muestra la tabla 6-74, el número del temporizador determina su resolución.



Consejo

Para garantizar un intervalo de tiempo mínimo, incremente en 1 el valor de preselección (PV). Por ejemplo, para garantizar un intervalo mínimo de 2.100 ms utilizando un temporizador de 100 ms, ajuste el PV a 22.

Tabla 6-74 Temporizadores y sus resoluciones

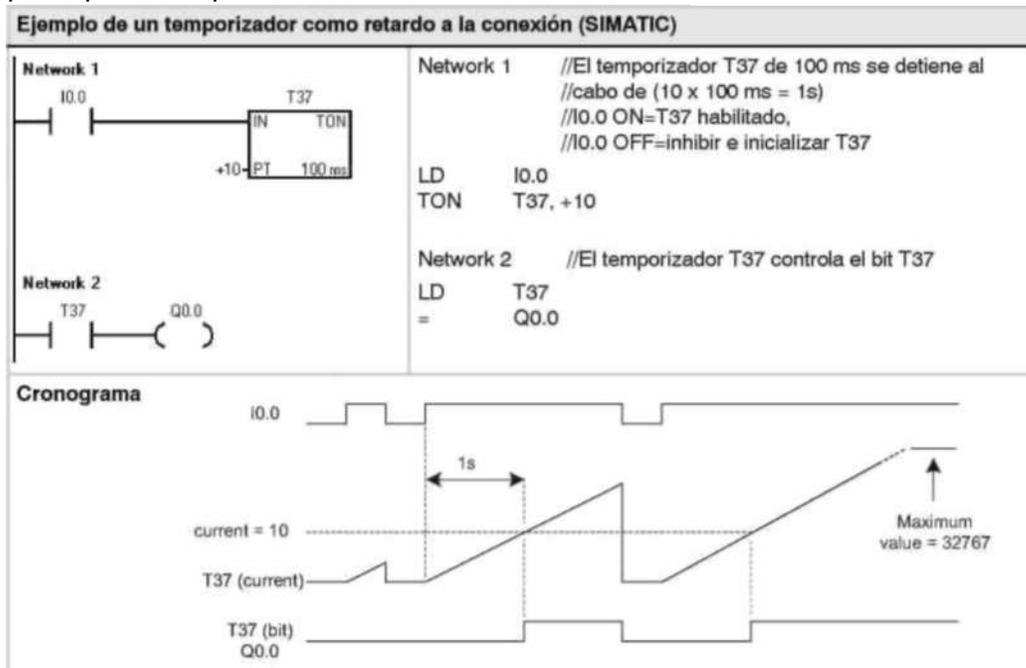
Tipo de temporiza	Resolución	Valor máximo	Nº de temporizador
TONR (con memoria)	1 ms	32.767 s (0,546	T0, T64
	10 ms	327,67 s (5,46	T1 a T4, T65 a T68
	100 ms	3276,7 s (54,6	T5 a T31, T69 a T95
TON, TOF (sin memoria)	1 ms	32.767 s (0,546	T32, T96
	10 ms	327,67 s (5,46	T33 a T36, T97 a
	100 ms	3276,7 s (54,6	T37 a T63, T101 a

Efecto de la resolución en la actividad el temporizador

En los temporizadores con una resolución de 1 ms, el bit de temporización y el valor actual se actualizan de forma asíncrona al ciclo. En ciclos que duren más de 1 ms, el bit de temporización y el valor actual se actualizan varias veces durante el ciclo.

En los temporizadores con una resolución de 10 ms, el bit de temporización y el valor actual se actualizan al comienzo de cada ciclo. El bit de temporización y el valor actual permanecen constantes durante el ciclo. Los intervalos de tiempo que se acumulan durante el ciclo se suman al valor actual al comienzo de cada ciclo.

En los temporizadores con una resolución de 100 ms, el bit de temporización y el valor actual se actualizan cuando se ejecuta la operación. Por consiguiente, vigile que el programa ejecute la operación de un temporizador de 100 ms sólo una vez por ciclo para que el temporizador conserve el valor correcto.

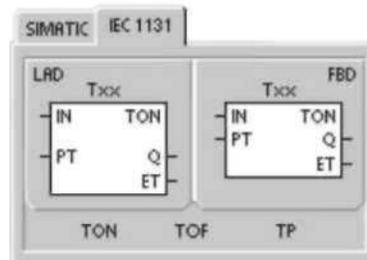




Consejo

Para garantizar que la salida de un temporizador que se inicialice a sí mismo se active durante un ciclo cada vez que el temporizador alcance el valor de preselección, utilice un contacto normalmente cerrado (en vez del bit de temporización) como entrada de habilitación del temporizador.

Ejemplo de un temporizador como retardo a la conexión (SIMATIC)	
<p>Network 1</p>	<p>Network 1 //El temporizador T33 de 10 ms se detiene //(100 x 10 ms = 1s). //El impulso de M0.0 es demasiado rápido //para poder observarlo en la vista "Estado".</p> <p>LDN M0.0 TON T33, +100</p> <p>Network 2</p>
<p>Network 3</p>	<p>Network 2 //La comparación es verdadera a una velocidad //visible en la vista "Estado". //Activar Q0.0 al cabo de (40 x 10 ms), //para crear una forma de onda 40% OFF/60% ON</p> <p>LDW>= T33, +40 = Q0.0</p> <p>Network 3 //Impulso de T33 (bit) demasiado rápido //para observarlo en la vista "Estado". //Inicializar el temporizador con M0.0 //al finalizar el período (100 x 10 ms).</p> <p>LD T33 = M0.0</p>
<p>Cronograma</p>	
Ejemplo de un temporizador como retardo a la desconexión (SIMATIC)	
<p>Network 1</p>	<p>Network 1 //El temporizador T33 de 10 ms se detiene al cabo //de (100 x 10 ms = 1s) //I0.0 ON a OFF=T33 habilitado //I0.0 OFF a ON=inhibir e inicializar T33</p> <p>LD I0.0 TOF T33, +100</p> <p>Network 2</p>
<p>Network 2</p>	<p>Network 2 //El temporizador T33 controla Q0.0 a través del //contacto T33.</p> <p>LD T33 = Q0.0</p>
<p>Cronograma</p>	



Operaciones de temporización (IEC)

Temporizador con retardo al conectar

La operación Temporizador con retardo al conectar (TON) cuenta el tiempo al estar activada (ON) la entrada de habilitación.

Temporizador como retardo a la desconexión

El Temporizador con retardo al desconectar (TOF) retarda la puesta a "0" (OFF) de una salida durante un período determinado tras haberse desactivado (OFF) una entrada.

Temporizador por impulsos

El Temporizador por impulsos (TP) genera impulsos de una duración determinada.

Tabla 6-75 Operandos válidos para las operaciones de temporización IEC

Entradas/sali	Tipos de	Operandos
Txx	TON, TOF,	Constante (T32 a T63, T96 a T255)
IN	BOOL	I, Q, V, M, SM, S, T, C, L, circulación de corriente
PT	INT	IW, QW, VW, MW, SMW, SW, LW, AC, AIW, *VD,
Q	BOOL	I, Q, V, M, SM, S, L
ET	INT	IW, QW, VW, MW, SMW, SW, LW, AC, AQW, *VD, *LD, *AC



Consejo

No se pueden compartir números iguales para los temporizadores TOF, TON y TP. Por ejemplo, no puede haber tanto un TON T32 como un TOF T32.

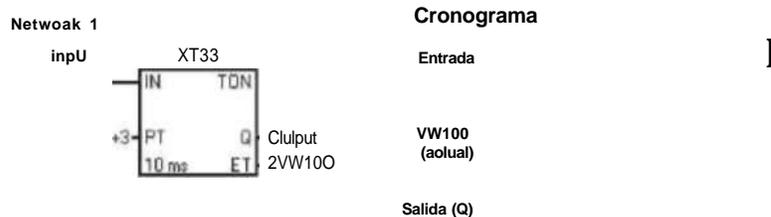
- El Temporizador con retardo al conectar (TON) cuenta intervalos de tiempo hasta alcanzar el valor prefijado, cuando la entrada de habilitación (IN) cambia a “verdadero”. Si el tiempo transcurrido (ET) es igual al tiempo prefijado (PT), se activará el bit de salida del temporizador (Q). El bit de salida se pone a “0” cuando se desactiva la entrada de habilitación. Cuando se alcanza el tiempo prefijado (PT), la temporización se detiene y el temporizador se inhibe.
- El Temporizador con retardo al desconectar (TOF) retarda la puesta a “0” de una salida durante un período determinado tras haberse desactivado una entrada. Temporiza hasta el valor prefijado cuando se desactiva la entrada de habilitación (IN). Si el tiempo transcurrido (ET) es igual al tiempo prefijado (PT), se desactivará el bit de salida del temporizador (Q). Una vez alcanzado el valor prefijado, se desactiva el bit de salida del temporizador y el tiempo transcurrido se mantiene hasta que se activa la entrada de habilitación (IN). Si la entrada de habilitación (IN) se desactiva durante un período inferior al tiempo prefijado, el bit de salida seguirá activado.
- El Temporizador por impulsos (TP) genera impulsos de una duración determinada. Cuando el estado de señal de la entrada de habilitación (IN) es “1”, se activa el bit de salida (Q). Éste último permanece activado durante el impulso indicado en el tiempo prefijado (PT). Cuando el tiempo transcurrido (ET) alcanza el valor prefijado (PT), se desactiva el bit de salida (Q). El tiempo transcurrido se conserva hasta que se desactiva la entrada de habilitación. Cuando se active el bit de salida, permanecerá en ese estado hasta que haya transcurrido el tiempo prefijado.

El valor actual resulta del valor de contaje multiplicado por la base de tiempo. Por ejemplo, el valor de contaje 50 en un temporizador de 10 ms equivale a 500 ms. Se dispone de temporizadores IEC (TON, TOF y TP) con tres resoluciones. La resolución viene determinada por el número del temporizador (v. tabla 6-76).

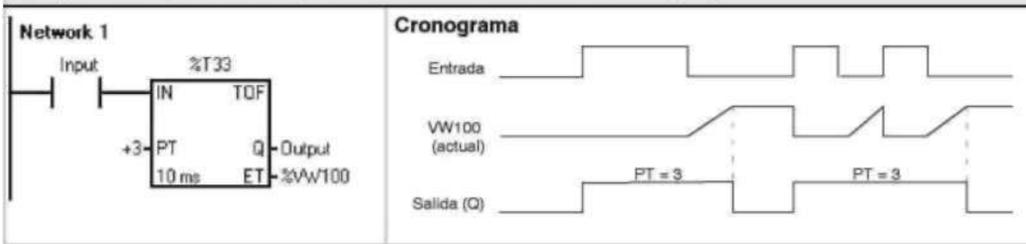
Tabla 6-76 Resolución de los temporizadores

Resolución	Valor máximo	Nº de temporizador
1 ms	32,767 s (0,546 minutos)	T32, T96
10 ms	327,67 s (5,46 minutos)	T33 a T36, T97 a T100
100 ms	3276,7 s (54,6 minutos)	T37 a T63, T101 a T255

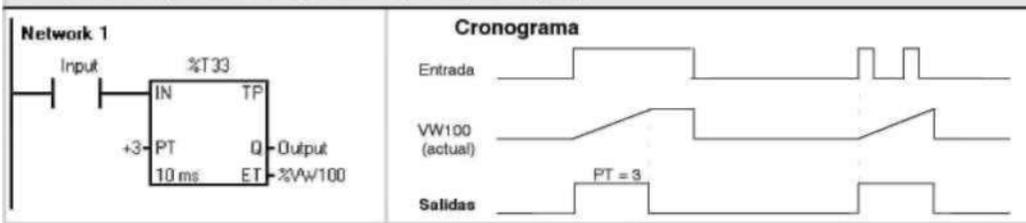
Ejemplo de la operación Temporizador con retardo al conectar (IEC)



Ejemplo de la operación Temporizador con retardo al desconectar (IEC)



Ejemplo de la operación Temporizador por impulsos (IEC)



Temporizadores de intervalos

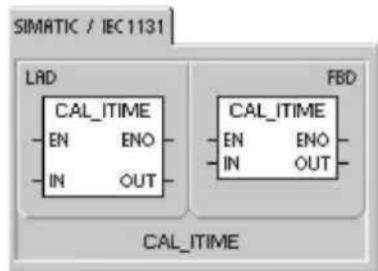
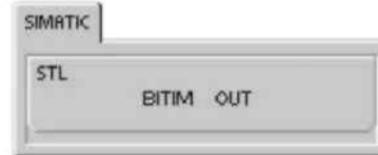
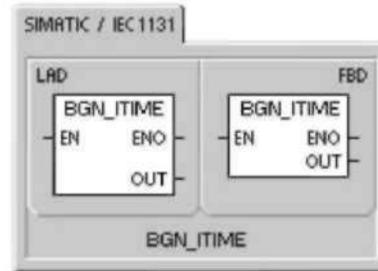
Capturar intervalo inicial

La operación Capturar intervalo inicial (BITIM) lee el valor actual del contador integrado de 1 milisegundo y almacena el valor en OUT. El intervalo de tiempo máximo para un valor en milisegundos en formato DWORD es 2 elevado a la potencia 32 (o 49,7 días).

Calcular intervalo

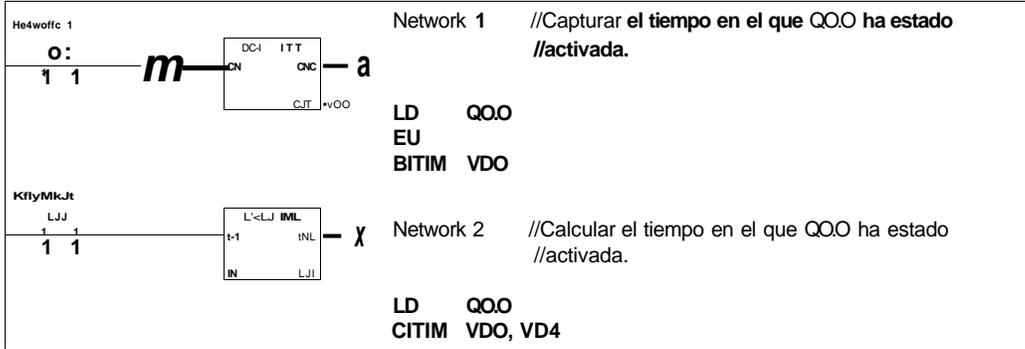
La operación Calcular intervalo (CITIM) calcula la diferencia horaria entre el tiempo actual y el tiempo indicado en IN. Esta diferencia se almacena en OUT. El intervalo de tiempo máximo para un valor en milisegundos en formato DWORD es 2 elevado a la potencia 32 (o 49,7 días). CITIM procesa automáticamente la ejecución del temporizador de 1 milisegundo que ocurre dentro del intervalo máximo, dependiendo de cuándo se ha ejecutado la operación BITIM.

Tabla 6-77 Operandos válidos para las operaciones con temporizadores de intervalos

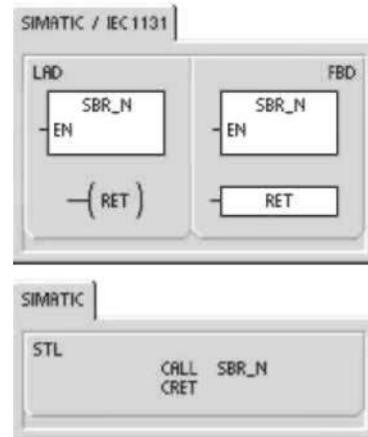


Entradas/sali	Tipos de	Operandos
IN	DWORD	VD, ID, QD, MD, SMD, SD, LD, HC, AC, *VD, *LD,
OUT	DWORD	VD, ID, QD, MD, SMD, SD, LD, AC, *VD, *LD,

Ejemplo de las operaciones Intervalo inicial y Calcular intervalo (SIMATIC)



Operaciones con subrutinas



La operación Llamar a subrutina (CALL) transfiere el control a la subrutina SBR_N. Esta operación se puede utilizar con o sin parámetros. Una vez ejecutada la subrutina, el control vuelve a la operación que sigue a la llamada de la subrutina (CALL). La operación Retorno condicionado desde subrutina (CRET) finaliza la subrutina en función de la combinación lógica precedente.

Para insertar una subrutina, elija los comandos de menú **Edición > Insertar > Subrutina**.

Condiciones de error que ponen ENO a 0:

- 0008 (excedida la profundidad máxima de anidado)
- 0006 (direccionamiento indirecto)

En el programa principal, se pueden anidar (situar una llamada a subrutina en otra) hasta ocho subrutinas. Las subrutinas no se pueden anidar en una rutina de interrupción.

Una llamada a subrutina no se puede disponer en ninguna otra subrutina a la que se llame desde una rutina de interrupción. Si bien la recursión (la subrutina se llama a sí misma) está permitida, hay que utilizarla con gran precaución.

Tabla 6-78 Operandos válidos para las operaciones con subrutinas

Entradas/sali	Tipos de	Operandos
SBR n	WORD	Constante para las CPUs 221, 222 y 224: <u>0 a 63 para las CPUs 224XP y 226</u>
IN	BOOL	V, I, Q, M, SM, S, T, C, L, circulación de corriente
	BYTE	VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD, *AC ¹ ,
	WORD, INT	VW, T, C, IW, QW, MW, SMW, SW, LW, AC, AIW, *VD, *LD, *AC ¹ , constante
	DWORD, DINT	VD, ID, QD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC ¹ ,
	STRING	*VD, *LD, *AC, constante <u>&C, &SB, &AI, &AQ,</u>
IN/OUT	BOOL	&V S, IM, QB, cMo, n Ss tMa n ² , Se, T, C, 2L
	BYTE	VB, IB, QB, MB, SMB, SB, LB, AC, *VD, *LD,
	WORD, INT	VW, T, C, IW, QW, MW, SMW ² , SW, LW, AC, *VD,
	DWORD,	VD, ID, QD, MD, SMD ² , SD, LD, AC, *VD, *LD,
OUT	BOOL	V, I, ¹ Q, M, SM ² , S, T, C, L
	BYTE	VB, IB, QB, MB, SMB ² , SB, LB, AC, *VD, *LD,
	WORD, INT	VW, T, C, IW, QW, MW, SMW ² , SW, LW, AC, AQW,
	DWORD, DINT	VD, ID, QD, MD, SMD ² , SD, LD, AC, *VD, *LD, *AC ¹

¹ El offset debe ser 1 o superior

² El offset debe ser 30 o superior

Consejo



STEP 7-Micro/WIN inserta automáticamente un retorno absoluto desde cada subrutina. Cuando se llama a una subrutina, se almacena toda la pila lógica, poniéndose a “1” el nivel superior de la pila. Sus demás niveles se ponen a “0” y la ejecución se transfiere a la subrutina que se ha llamado. Cuando ésta se termina de ejecutar, se restablece la pila con los valores almacenados al llamar a la subrutina y se retorna a la rutina que ha efectuado la llamada.

Los acumuladores son comunes a las subrutinas y a la rutina de llamada. Los acumuladores no se almacenan ni se restablecen si se utilizan con subrutinas.

No utilice las operaciones de detección de flancos ni de contaje si una rutina se llama más de una vez en un mismo ciclo.

Llamar a una subrutina con parámetros

Las subrutinas pueden contener parámetros que hayan sido transferidos. Los parámetros se definen en la tabla de variables locales de la subrutina. Estos parámetros deben tener un nombre simbólico (de 23 caracteres como máximo), un tipo de variable y un tipo de datos. Se pueden transferir 16 parámetros a o desde una subrutina.

El campo “Tipo de variable” en la tabla de variables locales define si la variable se transfiere a la subrutina (IN), a y desde la subrutina (IN_OUT) o desde la subrutina (OUT). En la tabla 6-79 figuran los tipos de parámetros de las subrutinas. Para insertar un registro de parámetro, en el campo “Tipo de variable” sitúe el cursor en el tipo (IN, IN_OUT o OUT) que desea insertar. Pulse el botón derecho del ratón para visualizar un menú contextual que ofrece diversas opciones. Seleccione el comando “Insertar” y luego la opción “Fila inferior”. Debajo del registro actual aparecerá un nuevo registro de

parámetro.

Tabla 6-79 Tipos de parámetros de subrutinas

Parámetr Descripción

- IN** Los parámetros se transfieren a la subrutina. Si el parámetro es una dirección directa (por ejemplo, VB10), el valor de la dirección indicada se transfiere a la subrutina. Si el parámetro es una dirección indirecta (por ejemplo, *AC1), el valor de la dirección a la
- IN_OUT** El valor de la dirección del parámetro indicado se transfiere a la subrutina y el valor resultante de la subrutina se devuelve luego a la misma dirección. Para los parámetros de entrada/salida no se
- OUT** El valor resultante de la subrutina se devuelve a la dirección del parámetro indicado. Para los parámetros de salida no se pueden utilizar ni constantes (por ejemplo, 16#1234) ni direcciones (por ejemplo, &VB100).
- TEMP** Cualquier memoria local que no se utilice para la transferencia de parámetros se puede emplear para el almacenamiento

Como muestra la figura 6-38, el campo "Tipo de datos" de la tabla de variables locales indica el tamaño y el formato del parámetro. Los tipos de parámetros figuran a continuación:

- **BOOL:** Este tipo de datos se utiliza para entradas y salidas binarias sencillas. En el ejemplo siguiente, IN3 es una entrada booleana.

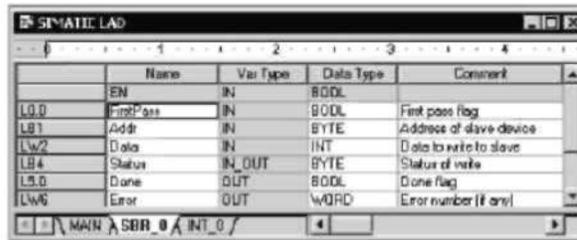


Figura 6-38 Tabla de variables locales

- **BYTE, WORD, DWORD:** Estos tipos de datos identifican parámetros de entrada o de salida sin signo compuestos por 1, 2 ó 4 bytes, respectivamente.
- **INT, DINT:** Estos tipos de datos **identifican parámetros de entrada** o de salida con signo compuestos por 2 ó 4 bytes, respectivamente.

- REAL: Este tipo de datos identifica un valor en coma flotante IEEE de precisión simple (4 bytes).
- STRING: Este tipo de datos sirve de puntero de 4 bytes a una cadena.
- Circulación de corriente: La circulación de corriente booleana sólo se permite en las entradas binarias (booleanas). Esta declaración le indica a STEP 7-Micro/WIN que este tipo de parámetro de entrada es el resultado de la circulación de la corriente conforme a una combinación de operaciones lógicas con bits. Las entradas booleanas de circulación de corriente deben aparecer en la tabla de variables locales antes de cualquier otro tipo de entrada. Aquí se permite utilizar sólo parámetros de entrada. La entrada de habilitación (EN) y las entradas IN1 en el siguiente ejemplo usan la lógica booleana.

Ejemplo de la operación Llamada a subrutina	
<p>A continuación figuran dos ejemplos en AWL. El primer juego de instrucciones AWL se puede visualizar únicamente en el editor AWL, puesto que los parámetros BOOL utilizados como entradas de circulación de corriente no se almacenan en la memoria L.</p> <p>El segundo juego de instrucciones AWL se puede visualizar también en los editores KOP y FUP, puesto que la memoria L se utiliza para guardar el estado de los parámetros de entrada BOOL que se visualizan como entradas de circulación de corriente en KOP y FUP.</p>	
	<p>Sólo en AWL:</p> <p>Network 1</p> <pre>LD I0.0 CALL SBR_0, I0.1, VB10, I1.0, &VB100, *AC1, VD200</pre> <p>Para poder visualizar el programa correctamente en KOP y FUP:</p> <p>Network 1</p> <pre>LD I0.0 = L60.0 LD I0.1 = L63.7 LD L60.0 CALL SBR_0, L63.7, VB10, I1.0, &VB100, *AC1, VD200</pre>

Los parámetros de dirección tales como IN4 (&VB100) se transfieren a una subrutina como valor DWORD (palabra doble sin signo). El tipo de parámetro de una constante se debe indicar en la rutina de llamada mediante un descriptor delante del valor de la constante. Por ejemplo, para transferir como parámetro una constante de palabra doble sin signo cuyo valor sea 12.345, el parámetro de esta constante se deberá indicar de la siguiente forma: DW#12345. Si se omite el descriptor de la constante, se podría deducir que la constante es de un tipo diferente.

En el caso de los parámetros de entrada o de salida no se realiza una conversión automática de datos. Por ejemplo, si en la tabla de variables locales se indica que un parámetro es del tipo de datos REAL y en la rutina de llamada se indica una palabra doble (DWORD) para ese parámetro, el valor en la subrutina será una palabra doble.

Los valores que se transfieren a una subrutina se depositan en la memoria local de la misma. La columna situada en el extremo izquierdo de la tabla de variables locales muestra la dirección local de todos los parámetros que se han transferido. Cuando se llama a la subrutina, los valores de los parámetros de entrada se copian a la memoria local de la subrutina. Cuando se finaliza la ejecución de la subrutina, los valores de los parámetros de salida se copian de la memoria local de la subrutina a las direcciones indicadas de los parámetros de salida.

El tamaño y el tipo de los elementos de datos está codificado en los parámetros. Los valores de los parámetros se asignan de la siguiente

Ejemplo de una llamada a subrutina con cadenas

En este ejemplo, un literal de cadena diferente se copia a una dirección unívoca en función de la entrada dada. La dirección unívoca de esta cadena se almacena. A continuación, la dirección de la cadena se transfiere a la subrutina, utilizando para ello una dirección indirecta. El tipo de datos del parámetro de entrada de la subrutina es STRING. A continuación, la subrutina desplaza la cadena a una dirección diferente.

Un literal de cadena también se puede transferir a la subrutina. La referencia a la cadena dentro de la subrutina es siempre igual.

P
R
I
N
C
I
P
A
L

```

                    5TF^CF>
                    N   ENO
                    iirrai"iIN.  OÜ' moa  '.*.iii.

```

-31

```

                    QIII.CPr
                    N   EHO
                    *i.,  r-|  *i.,

```

-31

I rΔ.u.

*i., r-| *i.,

```

Network 1 //
LD      I0.0
SSCPY   "string1". VB100
AENO
MOVD    &VB100. VDO

```

```

Network2 //
LD      10.1
SSCPY   "string2". VB200
AENO
MOVD    &VB200. VDO

```

```

Network3 //
LD      I0.2
CALL    SBR 0. *VDO

```

```

Network 1 //
LD      SMOO
SSCPY   *LD0, VB300

```