



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

---

FACULTAD DE ESTUDIOS SUPERIORES  
CUAUTITLAN

DISEÑO, OPERACION Y PROTECCION DE  
INSTALACIONES ELECTRICAS INDUSTRIALES.  
EL CONTROLADOR LOGICO PROGRAMABLE  
SIMATIC S7-200.

**TRABAJO DE SEMINARIO**  
QUE PARA OBTENER EL TITULO DE:  
**INGENIERO MECANICO ELECTRICISTA**  
P R E S E N T A  
**ALBERTO AGUILAR CANDELARIO**

ASESOR:

ING. JOSE GUSTAVO OROZCO HERNANDEZ

CUAUTITLAN IZCALLI, EDO. DE MEXICO

2003

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ESTA TESIS NO SALE  
DE LA BIBLIOTECA

ESTADÍSTICA  
DE ORIGEN



FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN  
UNIDAD DE LA ADMINISTRACIÓN ESCOLAR  
DEPARTAMENTO DE EXÁMENES PROFESIONALES

DR. JUAN ANTONIO MONTARAZ CRESPO  
DIRECTOR DE LA FES CUAUTITLÁN  
P R E S E N T E

ATN: Q. Ma. del Carmen García Mijares  
Jefe del Departamento de Exámenes  
Profesionales de la FES Cuautitlán

Con base en el art. 51 del Reglamento de Exámenes Profesionales de la FES-Cuautitlán, nos permitimos comunicar a usted que revisamos el Trabajo de Seminario:

Diseño, operación y protección de instalaciones eléctricas  
industriales. El controlador lógico programable simatic S7-200.

que presenta el pasante: Alberto Aguilar Candelario

con número de cuenta: 9659287-0 para obtener el título de:

Ingeniero Mecánico Electricista

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el EXÁMEN PROFESIONAL correspondiente, otorgamos nuestro VISTO BUENO.

**A T E N T A M E N T E**  
**"POR MI RAZA HABLARA EL ESPIRITU"**

Cuautitlán Izcalli, Méx. a 11 de noviembre de 2003

MODULO	PROFESOR	FIRMA
<u>I</u>	<u>MI. Benjamín Contreras Santacruz</u>	<u>[Firma]</u>
<u>II</u>	<u>MC. Ricardo Joaquín Ramírez Verdeja</u>	<u>[Firma]</u>
<u>IV</u>	<u>Ing. José Gustavo Orozco Hernández</u>	<u>[Firma]</u>



## **Dedicatorias**

A dios, por darme la oportunidad de realizar una de mis grandes metas

A la mujer que más quiero y respeto, mi madre, porque me brindo toda su confianza, cariño y apoyo incondicionalmente y sobre todo porque esa linda mujer se merece esto y muchas cosas más por tantas cosas buenas que me ha dado, gracias madre por que tu has sido el eje principal que ha guiado mi vida. Gracias madre por todo esos consejos y regaños que no fueron en vano, sino para ser de mí un hombre de bien

Al hombre que más quiero y respeto, mi padre, que me brindo su apoyo y que siempre se esforzó y se preocupo porque no me faltara nada en todo el transcurso de mi vida como estudiante, gracias por darme tu apoyo, amor y cariño.

## INDICE

Objetivo	7
Introducción	8
<b>1. Definición general de autómatas programables</b>	
<b>1.1</b> Definición	11
1.2 Lógica de cableado y de programación	11
1.3 Principales componentes de un autómata programable	13
<b>2. El controlador lógico programable SIMATIC S7-200</b>	
2.1 Introducción al capítulo	15
2.2 La familia de los controladores SIMATIC	15
2.3 Introducción al controlador programable SIMATIC S7-200	16
2.4 Principales componentes de un Micro-PLC S7-200	17
2.5 Características principales de los diferentes CPUs SIMATIC S7-200	20
2.6 Módulos de ampliación digitales	23
2.7 Módulos de ampliación analógicos	24
2.8 Módulos para termopares y RTD	26
2.9 Módulos para comunicaciones	27
<b>3. Conocimientos básicos para programar una CPU S7-200</b>	
3.1 Creación de una solución de automatización con un Micro-PLC	30
3.2 Referencias a las entradas y salidas en el programa	33
3.3 Lenguajes de programación para las CPUs S7-200	35
3.4 Elementos básicos para estructurar un programa	39
3.5 El ciclo de la CPU	41
3.6 Ajustar el modo de operación de la CPU	44
3.7 Determinar una contraseña para la CPU	45

#### **4. Memoria de la CPU, tipo de datos y direccionamiento**

4.1	Direccionamiento directo de las áreas de memoria de la CPU	49
4.2	Direccionamiento indirecto de las áreas de memoria de la CPU	58
4.3	Respaldo de datos en la CPU S7-200	61
4.4	Guardar el programa en un cartucho de memoria	68

#### **5. Entradas y salidas**

5.1	Entradas y salidas integradas y adicionales	72
5.2	Entradas y salidas rápidas	74
5.3	Potenciómetros analógicos	75

#### **6. Instrucciones de operación**

6.1	Operaciones lógicas con bits	78
6.2	Operaciones lógicas SR y RS	84
6.3	Operaciones de reloj	86
6.4	Operaciones de comunicación	87
6.5	Operaciones de comparación	88
6.6	Operaciones de conversión	90
6.7	Operaciones de codificar y decodificar	95
6.8	Operaciones de conteo	97
6.9	Operaciones aritméticas	99
6.10	Operaciones para incrementar y decrementar	102
6.11	Operaciones de transferencia	104
6.12	Operaciones de temporización	107
6.13	Operaciones de desplazamiento y rotación	109

#### **7. Comunicación en redes con las CPUs S7-200**

7.1	Protocolos de comunicación de las CPUs S7-200	113
7.2	Componentes para la comunicación en redes	115
7.3	Utilizar un cable PC/PPI para la comunicación	119
7.4	Utilizar una tarjeta MPI o CP para la comunicación	122

7.5	Comunicación en redes de periferia descentralizada (DP)	123
<b>8.</b>	<b>Instalar y utilizar el software STEP7-Micro/WIN</b>	
8.1	Instalar el software de STEP7-Micro/WIN	125
8.2	Utilizar STEP7-Micro/WIN para configurar la comunicación	127
8.3	Establecer la comunicación con una CPU S7-200	128
8.4	Configurar las preferencias para STEP7-Micro/WIN	137
8.5	Crear y guardar un proyecto	138
8.6	Crear un programa	139
8.7	Crear un bloque de datos	143
8.8	Utilizar la tabla de estado	144
<b>9.</b>	<b>Aplicación para un mezclador de pintura</b>	
9.1	Requisitos para el programa de ejemplo	147
9.2	Aplicación: Mezclador de pintura	147
9.3	Programa de ejemplo en lenguaje de escalera (KOP) y lista de instrucciones (AWL)	149
	Bibliografía	152

**Objetivo general:**

Comprender la funcionalidad, los diferentes tipos de datos, modos de direccionamiento, comunicación en redes y ciclo del controlador lógico programable SIMATIC S7-200 para estructurar y crear programas que se puedan utilizar en la solución de diversos sistemas de automatización.

## Introducción

La presión existente por bajar los costos en los procesos de producción a hecho que los autómatas programables (PLC) estén cada vez más difundidos en las aplicaciones de automatización. También la rápida evolución de la técnica es un factor que requiere autómatas programables para resolver las diversas tareas de automatización.

El autómata programable S7-200 entra cada vez más en campos donde por motivos económicos, se empleaba hasta la fecha electrónica especializada. Además de que permite la solución económica y compacta para tareas de automatización en la entrada de la gama en reemplazo de contactores y relés temporizadores, optimizada para soluciones con bajo grado de interconexión en red y escasa integración en el sistema.

El siguiente trabajo está estructurado por los siguientes capítulos:

- El capítulo 1 (Definición general de autómata programable) ofrece una visión general del uso de los controladores lógicos programables.
- El capítulo 2 (El controlador lógico programable SIMATIC S7-200) proporciona información de los componentes y características principales de los diferentes CPUs SIMATIC S7-200.
- El capítulo 3 (Conocimientos básicos para programar una CPU S7-200) proporciona información de los diferentes lenguajes de programación de un PLC y la forma de ejecución cíclica del programa.
- El capítulo 4 (Memoria de la CPU, tipo de datos y direccionamiento) proporciona información de cómo la CPU almacena y procesa la información en sus diferentes áreas de memoria.
- El capítulo 5 (Entradas y salidas) proporciona información del direccionamiento de las entradas y salidas de los CPUs S7-200 y de los módulos de ampliación.
- El capítulo 6 (Juego de operaciones) contiene explicaciones y ejemplos de las operaciones de programación utilizadas por las CPUs S7-200.

- El capítulo 7 (Comunicación en redes con CPUs S7-200) describe la forma de como se deben de conectar las CPUs a los diferentes tipos de redes.
- El capítulo 8 (Instalar y utilizar el software STEP7-Micro/WIN) describe cómo instalar el software de programación de los CPUs S7-200 y las funciones básicas del mismo.
- El capítulo 9 (Aplicación para un mezclador de pintura) muestra en forma práctica un ejemplo para la operación de un mezclador de pintura en diagrama de escalera (KOP) y en lista de instrucciones (AWL).

El SIMATIC S7-200 permite además una ampliabilidad modular que permite su adaptación especial a las tareas previstas como por ejemplo:

- Máquinas de serie
- Industria alimentaria
- Cintas transportadoras
- Instalaciones de extracción
- Maquinaria de labrado de madera

## **CAPÍTULO 1**

### **Definición general de autómata programable**



## 1.1 ¿Qué es un autómatas programable?

Un autómatas programable no es más que un aparato electrónico que sustituye los circuitos auxiliares o de mando de los sistemas automáticos utilizados para resolver tareas de mando en maquinaria e instalaciones. Cubre aplicaciones que van de la sustitución de relés y contactores hasta tareas complejas de automatización operando aislado, interconectado en red o en configuraciones descentralizadas.

A él se conectan los captadores (finales de carrera, pulsadores...) por una parte, y los actuadores (bobinas de contactores, lámparas,...) por otra.

## 1.2 Lógica cableada y de programación

Hasta no hace mucho tiempo el control de procesos industriales se venía haciendo de forma cableada por medio de contactores y relés. Al operario que se encontraba a cargo de este tipo de instalaciones, se le exigía tener altos conocimientos técnicos para poder realizarlas y posteriormente mantenerlas. Además cualquier variación en el proceso suponía modificar físicamente gran parte de las conexiones de los montajes, siendo necesario para ello un gran esfuerzo técnico y un mayor desembolso económico.

En la figura 1-2 se muestra un ejemplo de lógica cableada convencional.

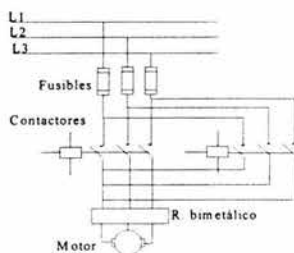


Figura 1-2 Mostrando lógica de cableado de un circuito de fuerza de arrancador reversible

En la figura anterior se puede observar el diagrama convencional del circuito de fuerza de un arrancador reversible en donde el control por cableado sin duda es uno de los más utilizados.

En la figura 1-3 se muestra el circuito de control del arrancador reversible utilizando la lógica cableada.

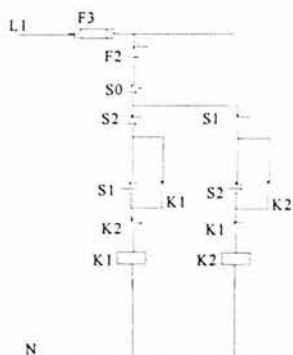


Figura 1-3 Mostrando lógica de cableado del circuito de control del arrancador reversible

En la figura anterior se puede observar el cableado del circuito de control para el arrancador reversible, en donde S1 es el pulsador de arranque de motor de giro a la derecha, S2 es el pulsador de arranque de motor de giro a la izquierda, S0 detiene por completo al motor, K1, K2 son los contactores, F2 es el relevador de sobrecarga y F3 es un fusible de protección.

El diagrama anterior nos da una visión de cómo sería la lógica de cableado del circuito de control, es importante considerar que cualquier tarea de automatización puede realizarse retomando la lógica de control por cableado y es posible implementar las secuencias de control requeridas para resolver alguna tarea de automatización.

En la actualidad no se puede entender un proceso complejo de alto nivel desarrollado por técnicas cableadas. El ordenador y los autómatas programables han intervenido de forma considerable para que este tipo de instalaciones se hayan visto sustituidas por otras controladas de forma programada. Por lo que el autómata programable industrial nació como solución al control de circuitos complejos de automatización.

Para poder desarrollar una lógica de programación es necesario utilizar un dispositivo lógico programable mejor conocido como PLC.

### 1.3 Principales componentes de un autómata programable

Básicamente un autómata programable esta formado por los siguientes elementos.

- Fuente de alimentación.
- Unidad central de procesamiento (CPU).
- Entradas y salidas.
- Interface.
- Diodos luminosos.
- Batería.
- Módulo de memoria.
- Dispositivos de programación.

Cada uno de los elementos que conforman un PLC, tienen características fundamentales que hacen posible que un autómata programable sea capaz de realizar un sin numero de tareas de control total.

Estos elementos se explicaran mas detalladamente en los siguientes capítulos con el autómata programable SIMATIC S7-200 de Siemens, el cual cumple el máximo efecto de automatización al mínimo costo y que principalmente estamos enfocados a estudiar en los siguientes capítulos.

## **CAPÍTULO 2**

El Controlador Lógico Programable SIMATIC S7-200

## 2.1 Introducción al capítulo

Hasta ahora, SIMATIC se ha venido entendiendo en gran parte como sinónimo de autómatas programables o PLC.

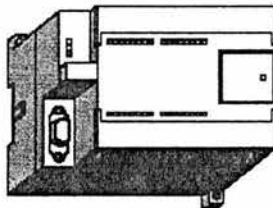
La gama SIMATIC S7 comprende diversos sistemas de automatización, pequeños Micro-PLCs que se pueden utilizar para numerosas tareas. Gracias a su diseño compacto, su capacidad de ampliación, su bajo costo y su amplio juego de operaciones, los sistemas de automatización S7-200 son idóneos para numerosas aplicaciones pequeñas de control. La gran variedad de tamaños y fuentes de alimentación de las CPUs, así como las múltiples opciones de programación proporcionan la flexibilidad necesaria para solucionar las tareas de automatización.

Los siguientes capítulos contienen información acerca de la funcionalidad del S7-200, los tipos de datos, los modos de direccionamiento y el ciclo de la CPU.

## 2.2 Familia de los Controladores SIMATIC S7

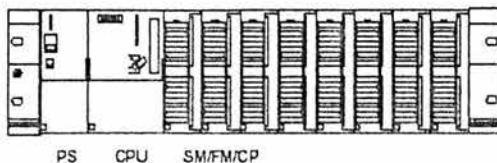
La gama de los autómatas programables está dividida en tres familias principales las cuales son:

- La familia de los Micro-PLCs SIMATIC S7-200 que forman parte de la serie S7-22X para la gama baja.



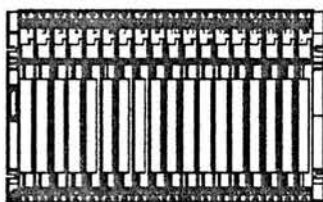
Autómata programable SIMATIC S7-200

- Los autómatas SIMATIC S7-300, los cuales son de aplicación universal con su modularidad aun más amplia, dando pruebas de su eficiencia sobre todo en la automatización de la fabricación.



Autómata programable SIMATIC S7-300

- Los autómatas SIMATIC S7 400, los cuales ofrecen aún más prestaciones gracias a las nuevas CPUs y módulos así como a funciones ampliadas. El S7-400F/FH es el primer autómata SIMATIC S7 de seguridad positiva para plantas e instalaciones que requieren medidas especiales de seguridad.

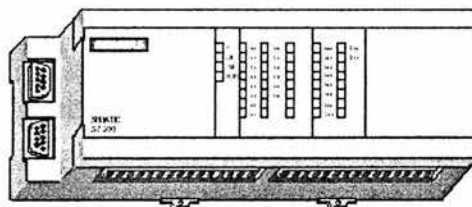


Autómata programable SIMATIC S7-400

### 2.3 Introducción al Controlador Programable SIMATIC S7-200

El SIMATIC S7-200 es el Micro-PLC para resolver tareas de mando y regulación en maquinaria e instalaciones a bajo costo.

En la figura siguiente se muestra un Micro-PLC S7 200.



Micro-PLC de la familia SIMATIC

### **Construcción**

La familia SIMATIC S7 200 está compuesta de los siguientes módulos.

- 4 equipos básicos diferentes en diversas variantes.
- 14 módulos de ampliación digitales y analógicos.
- 2 módulos de comunicaciones para la conexión a PROFIBUS y AS-Interface.

### **Características mecánicas**

- Carcasa robusta y compacta.
- Elementos de conexión y mando fácilmente accesibles, protegidos por tapas frontales.

### **Propiedades**

Las Normas internacionales; el SIMATIC S7-200 satisface las normas VDE, UL, CSA y FM (Clase 1, Categoría 2, grupos de clase de peligro A, B, C y D, T4A). El sistema de gestión de calidad implantado en la fabricación está certificado conforme a ISO 9001.

## **2.4 Principales componentes de un Micro-PLC SIMATIC S7-200**

La CPU S7-200 es un aparato autónomo compacto que incorpora una unidad central de procesamiento (CPU), la fuente de alimentación, así como entradas y salidas.

### **Unidad central de procesamiento (CPU) del SIMATIC S7-200**

La CPU del SIMATIC S7-200 ejecuta el programa de usuario y almacena los datos para la tarea de automatización o el proceso a realizar.

### La fuente de alimentación de los SIMATIC S7-200

La fuente de alimentación proporciona corriente a la unidad central y a los módulos de ampliación conectados entre sí para una operación eficiente.

### Batería

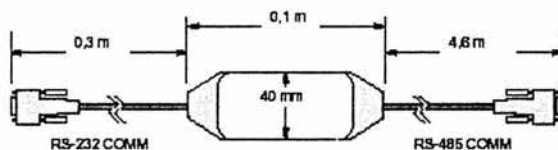
La batería respalda los contenidos de la memoria RAM en caso de pérdida de alimentación, es por eso que es indispensable en todo dispositivo programable SIMATIC S7-200.

### Entradas y salidas digitales incorporadas

Las entradas y salidas controlan el sistema de automatización. Las entradas vigilan las señales de los aparatos de campo y las salidas vigilan las bombas, motores u otros dispositivos del proceso.

### Interface de comunicación del autómeta S7-200

El interface de comunicación permite conectar la CPU a una unidad de programación o a otros dispositivos. Algunas CPUs S7-200 disponen de dos interfaces de comunicación.



Interface de comunicación PC/PPI de un SIMATIC S7-200

### Diodos indicadores de operación del SIMATIC S7-200

Los diodos luminosos indican el modo de operación de la CPU (RUN o STOP), el estado de las entradas y salidas integradas, así como los posibles fallos del sistema que se hayan detectado.



### **Entradas y salidas adicionales**

Entradas y Salidas adicionales pueden ser adicionadas a los CPUs por medio de Módulos de Expansión (El CPU 221 no es expandible).

### **Selector de modo**

Nos permite cambiar de modo STOP, TERM, RUN.

### **Dispositivo de programación**

La Unidad de programación (PG) con tarjeta MPI o cable PC/MPI y software STEP 7 ya instalados, nos ayudan para configurar y programar al autónoma programable.



Unidad de programación PG de Siemens

La figura 2-4 muestran las partes principales de un Micro-PLC SIMATIC S7200.

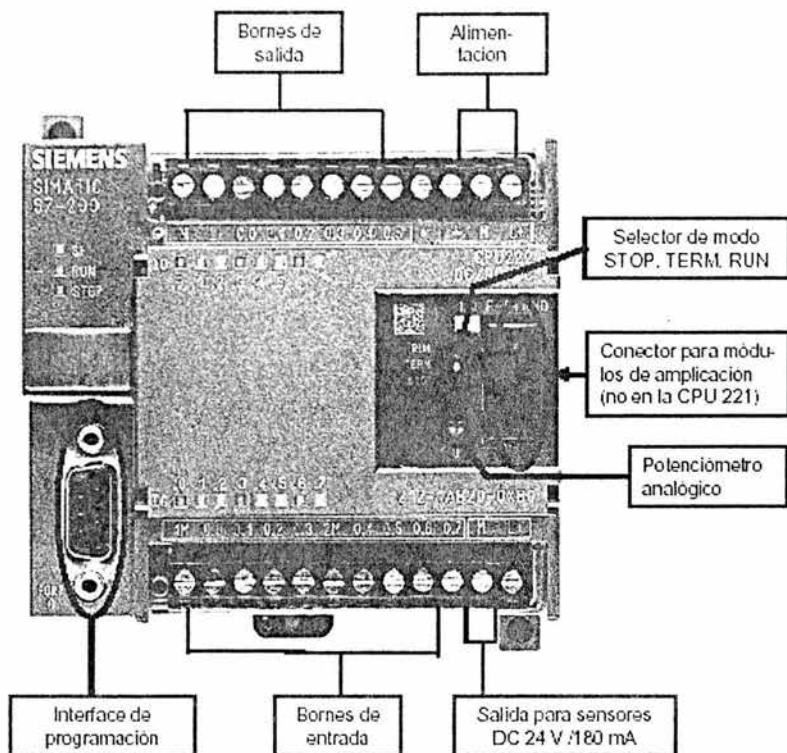


Figura 2-4 Elementos principales de un autómata programable S7-200 marca siemens

## 2.5 Características principales de los CPUs SIMATIC S7-200

La serie S7 200 comprende diversas CPUs. Por lo tanto, se dispone de una amplia gama de funciones que permiten diseñar soluciones de automatización.

A continuación se muestran tablas con datos técnicos de los diferentes CPUs de la familia S7 200.

**Datos técnicos CPU 221**

Alimentación	24 VDC	100 a 230 VAC
Entradas	24 VDC	24 VDC
Salidas	24 VDC	relé
E/S Digitales integradas	6E/4S (transistor)	6E/4S (relés)
E/S Analógicas integradas	no	no
Expansibilidad máxima	no	
Memoria de programa	4 Kbytes	
Memoria de datos	2 Kbytes	
Marcas	256	
Puertos	Interface de comunicación RS 485	
Contadores	256	
Temporizadores	256	
Respaldo de los datos	EEPROM integrada	
Unidades de programación	PG 720 P11, PG 740 P111	
Dimensiones en mm	90 x 80 x 62	

**Datos técnicos CPU 222**

Alimentación	24 VDC	100 a 230 VAC
Entradas	24 VDC	24 VDC
Salidas	24 VDC	relé
E/S Digitales integradas	8E/6S (transistor)	8E/6S (relés)
E/S Analógicas integradas	no	no
Expansibilidad máxima	2 módulos de ampliación	
Memoria de programa	4 Kbytes	
Memoria de datos	2 Kbytes	
Marcas	256	
Puertos	Interface de comunicación RS 485	
Contadores	256	
Temporizadores	256	

Respaldo de los datos	EEPROM integrada
Unidades de programación	PG 720 P11, PG 740 P111
Dimensiones en mm	90 x 80 x 62

#### Datos técnicos CPU 224

Alimentación	24 VDC	100 a 230 VAC
Entradas	24 VDC	24 VDC
Salidas	24 VDC	relé
E/S Digitales integradas	14E/10S (transistor)	14E/10S (relés)
E/S Analógicas integradas	no	no
Expansibilidad máxima	7 módulos de ampliación	
Memoria de programa	8 Kbytes	
Memoria de datos	5 Kbytes	
Marcas	256	
Puertos	Interface de comunicación RS 485	
Contadores	256	
Temporizadores	256	
Respaldo de los datos	EEPROM integrada	
Unidades de programación	PG 720 P11, PG 740 P111	
Dimensiones en mm	120.5 x 80 x 62	

#### Datos técnicos CPU 226

Alimentación	24 VDC	100 a 230 VAC
Entradas	24 VDC	24 VDC
Salidas	24 VDC	relé
E/S Digitales integradas	24E/16S (transistor)	24E/16S (relés)
E/S Analógicas integradas	no	no
Expansibilidad máxima	7 módulos de ampliación	
Memoria de programa	8 Kbytes	
Memoria de datos	5 Kbytes	

Marcas	256
Puertos	Interface de comunicación RS 485
Contadores	256
Temporizadores	256
Respaldo de los datos	EEPROM integrada
Unidades de programación	PG 720 P11, PG 740 P111
Dimensiones en mm	1960 x 80 x 62

## 2.6 Módulos de ampliación digitales

Los módulos de ampliación digitales permiten añadir más entradas/salidas digitales a las integradas, por lo que el usuario puede adaptar exactamente su autómatas a los requerimientos respectivos mediante módulos de ampliación mezclables a voluntad, ahorrando de este modo, inversiones superfluas. Se ofrecen módulos de 8, 16 y 32 entradas/salidas.

Los módulos de ampliación presentan las mismas características que los equipos base.

Las CPUs sólo pueden expandirse con módulos digitales de su misma serie.

1. Montaje en perfil soporte normalizado; en donde los módulos se enganchan en el perfil a la derecha de la CPU y se enlazan entre si y con la CPU mediante cables planos flexibles integrados.
2. El montaje directo; en donde media te los taladros integrados pueden atornillarse directamente a la pared. Este tipo de montaje se recomienda en caso de aplicaciones sometidas a muchas vibraciones.

A continuación se muestran tablas con datos técnicos de los diferentes módulos de ampliación digitales de la familia S7-200.

### Datos técnicos del módulo de entrada digital EM 221

Conectable a	CPU 22x
Cantidad de entradas	8
Bornes de E/S enchufables	sí

Dimensiones en mm	46 x 80 x 62
-------------------	--------------

### Datos técnicos del módulo de salida digital EM 222

	DC	relé
Cantidad de salidas	8	8
Conectable a	Serie S7-22x	Serie S7-22x
Bornes de E/S enchufables	si	si
Dimensiones en mm	45 x 80 x 62	45 x 80 x 62

### Datos técnicos de los módulos de E/S digital EM 223

	4E/4S	8E/8S	16E/16S	4E/4S	8E/8S	16E/16S
Entradas	24 VDC	24 VDC	24 VDC	24 VDC	24 VDC	24 VDC
Salidas	24 VDC	24 VDC	24 VDC	relé	relé	relé
Conectables a	CPU 22x	CPU 22x	CPU 22x	CPU 22x	CPU 22x	CPU 22x
Bornes de E/S enchufables	si	si	si	Si	si	si
Entradas	4	8	16	4	8	16
Salidas	4	8	16	4	8	16
Dimensiones en mm	46 x 80 x 62	71.2 x 80 x 62	137.5 x 80 x 62	46 x 80 x 62	71.2 x 80 x 62	137.5 x 80 x 62

## 2.7 Módulos de ampliación analógicos

Los módulos de ampliación analógicos permiten aplicar entradas/salidas analógicas a las integradas, por lo que el usuario puede adaptar exactamente su autómatas a los requerimientos respectivos mediante módulos de ampliación mezclables a voluntad.

Las CPUs sólo pueden expandirse con módulos analógicos de su misma serie.

Existen módulos de ampliación para la serie S7-21x y para la serie S7-22x.

Los módulos de ampliación presentan las mismas características que los equipos base.

1. Montaje en perfil soporte normalizado; en donde los módulos se enganchan en el perfil a la derecha de la CPU y se enlazan entre sí y con la CPU mediante cables planos flexibles integrados.
2. El montaje directo; en donde media te los taladros integrados pueden atornillarse directamente a la pared. Este tipo de montaje se recomienda en caso de aplicaciones sometidas a muchas vibraciones.

A continuación se muestran tablas con datos técnicos de los diferentes módulos de ampliación analógicos de la familia S7-200.

#### Datos técnicos del módulo de entrada analógica EM 231

Conectable a	Serie S7-22x
Cantidad de entradas	4
Bornes de E/S enchufables	no
Dimensiones en mm	71.2 x 80 x 62

#### Datos técnicos del módulo de entrada analógica EM 232

Conectable a	Serie S7-22x
Cantidad de salidas	4
Bornes de E/S enchufables	si
Dimensiones en mm	46 x 80 x 62

#### Datos técnicos del módulo de E/S analógicas EM 235

Conectable a	Serie S7-22x
Cantidad de salidas	4
Cantidad de salidas	1
Bornes de E/S enchufables	no
Dimensiones en mm	71.2 x 80 x 62

## 2.8 Módulos para termopares y RTD

### Módulos para termopares

El módulo para termopares EM 231 permite captar la temperatura con alta precisión utilizando termopares convencionales. También es posible medir señales analógicas de bajo nivel,  $\pm 80$  mv.

#### Datos técnicos del módulo para termopares EM 231

Conectable a	S7-222/224/226
Cantidad de entradas	4 analógicas
Bornes de E/S enchufables	no
Margen de tensión	$\pm 80$ mV/ $> 1$ M $\Omega$
Dimensiones en mm	71.2 x 80 x 62

### Módulo para RTD

El módulo para RTD EM 231 permite captar la temperatura con alta precisión utilizando termorresistencias convencionales.

#### Datos técnicos del módulo para RTD EM 231

Conectable a	S7-222/224/226
Cantidad de entradas	2 analógicas
Bornes de E/S enchufables	no
Margen de entrada	Tipo RTD <ul style="list-style-type: none"> <li>● Pt 100W, 200W, 500W, 1000W (a 3850 ppm, 3920 ppm, 3850,55 ppm, 3916 ppm, 3902 ppm)</li> <li>● Pt 10000 W (a=3850 ppm)</li> </ul>
Dimensiones en mm	71.2 x 80 x 62



## 2.9 Módulos de comunicaciones

La familia SIMATIC S7-200 está compuesta de 2 módulos de comunicaciones para las conexiones a PROFIBUS y AS-Interface.

### Módulo de comunicaciones EM 277

El módulo EM 277 para PROFIBUS-DP permite conectar a PROFIBUS-DP las CPUs de la serie 22x en calidad de esclavo.

#### Datos técnicos del módulo para el módulo EM 277 para PROFIBUS-DP

Conectable a	CPU 22x
Número de salidas	1 tipo R S485
Bornes de E/S enchufables	no
Protocolos	<ul style="list-style-type: none"> <li>● PROFIBUS-DP (esclavo)</li> <li>● MPI (esclavo)</li> </ul>
Velocidades de transmisión	9,6/19,2/45,45/93,75/187,5/500/1000/1500/3000/6000/ 12000 Kbits/s
Dimensiones en mm	71.2 x 80 x 62

### Módulo de comunicaciones CP 243-2

El CP 243-2 constituye un elemento de conexión de maestro a AS-Interface pensado exclusivamente para las CPUs 22x del SIMATIC S7-200. La conexión a AS-Interface incrementa sensiblemente el número de entradas y salidas digitales disponibles para el S7-200.

#### Datos técnicos del módulo para el módulo de comunicaciones CP-243-2

Tensión de alimentación	5 VDC del bus posterior
Perfil de maestro AS-Interface	M0 e M1e
Tiempo del ciclo del bus	5 ms con 31 esclavos

Espacio de direcciones ocupado en PLC	Correspondiente a 2 módulos E/S (8ED/8SD y 8EA y 8SA)
Conexión AS-Interface	Regletero

## **CAPÍTULO 3**

Conocimientos básicos para programar una CPU S7-200

### 3.1 Crear una solución de automatización con un Micro-PLC

Hay diversos métodos para crear una solución de automatización con un Micro-PLC. En el presente apartado se indican algunas reglas generales aplicables a numerosos proyectos. No obstante, también deberá tener en cuenta las reglas de su empresa y sus propias experiencias. La tabla 3-1 muestra los pasos básicos al respecto.

Estructurar el proceso o la instalación
Especificar las unidades funcionales
Diseñar los circuitos de seguridad cableados
Definir las operaciones de operador
Crear los planos de configuración del PLC
Elaborar una lista de direcciones simbólicas (opcional)

Tabla 3-1 Pasos básicos para crear una solución de automatización

#### **Estructurar el proceso o la instalación**

Estructure el proceso o la instalación en secciones independientes entre sí. Dichas secciones determinarán los límites entre los diversos sistemas de automatización e influirán en las descripciones de las áreas de funciones y en la asignación de recursos.

### **Especificar las unidades funcionales**

Describa las funciones de cada sección del proceso o de la instalación. Incorpore los siguientes aspectos.

- Entradas y salidas.
- Descripción del funcionamiento.
- Condiciones de habilitación (es decir, los estados que se deben alcanzar antes de ejecutar una función) de cada actuador (electroválvulas, motores, accionamientos, etc.).
- Descripción del interface del operador.
- Interfaces con otras secciones del proceso o de la instalación.

### **Diseñar los circuitos de seguridad cableados**

Determine qué aparatos requieren un cableado permanente por motivos de seguridad. Si fallan los sistemas de automatización, puede producirse un arranque inesperado o un cambio de funcionamiento de las máquinas que controlan. En tal caso, se pueden causar heridas graves o deteriorar objetos. Por lo tanto, es preciso utilizar dispositivos de protección contra sobrecargas electromecánicas que funcionen independientemente de la CPU, evitando así las condiciones inseguras.

Para diseñar los circuitos de seguridad cableados.

- Defina el funcionamiento erróneo o inesperado de los actuadores que pudieran causar peligros.
- Defina las condiciones que garanticen el funcionamiento seguro y determine cómo reconocer dichas condiciones, independientemente de la CPU.
- Defina cómo la CPU y los módulos de ampliación deberán influir el proceso cuando se conecte y desconecte la alimentación, así como al detectarse errores. Estas informaciones se deberán utilizar únicamente para diseñar el funcionamiento normal y el funcionamiento anormal esperado, sin poderse aplicar para fines de seguridad.
- Prevea dispositivos de parada de emergencia manual o de protección contra sobrecargas

electromagnéticas que impidan el funcionamiento peligroso, independientemente de la CPU.

- Desde los circuitos independientes, transmita informaciones de estado apropiadas a la CPU para que el programa y los interfaces de operador dispongan de los datos necesarios.
- Defina otros requisitos adicionales de seguridad para que el proceso se lleve a cabo de forma segura y fiable.

### **Definir las estaciones de operador**

Conforme a las funciones exigidas, cree planos de las estaciones de operador incorporando los siguientes puntos.

- Panorámica de la ubicación de todas las estaciones de operador con respecto al proceso o a la instalación.
- Disposición mecánica de los aparatos (pantalla, interruptores, lámparas, etc) de la estación de operador.
- Esquemas eléctricos con las correspondientes entradas y salidas de la CPU o de los módulos de ampliación.

### **Crear los planos de configuración del PLC**

Conforme a las funciones exigidas, cree planos de configuración del sistema de automatización incorporando los siguientes puntos.

- Ubicación de todas las CPUs y de todos los módulos de ampliación con respecto al proceso o a la instalación.
- Disposición mecánica de las CPUs y de los módulos de ampliación.

- Esquemas eléctricos de todas las CPUs y de los módulos de ampliación (incluyendo los números de referencia, las direcciones de comunicación y las direcciones de las entradas y salidas).

### **Elaborar una lista de nombres simbólicos**

Si desea utilizar nombres simbólicos para el direccionamiento, elabore una lista de nombres simbólicos para las direcciones absolutas. Incluya no sólo las entradas y salidas físicas, sino también todos los demás elementos que utilizará en su programa.

## **3.2 Referencias a las entradas y salidas en el programa**

El funcionamiento básico de la CPU S7-200 es muy sencillo.

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

La figura 3-2 muestra la conexión de un esquema de circuitos simple con la CPU S7-200. En este ejemplo, el estado del interruptor de la estación de operador para abrir la válvula de vaciado se suma a los estados de otras entradas. Los cálculos de los mismos determinan entonces el estado de la salida para la electroválvula que cierra la válvula de vaciado.

La CPU procesa el programa cíclicamente, leyendo y escribiendo los datos.

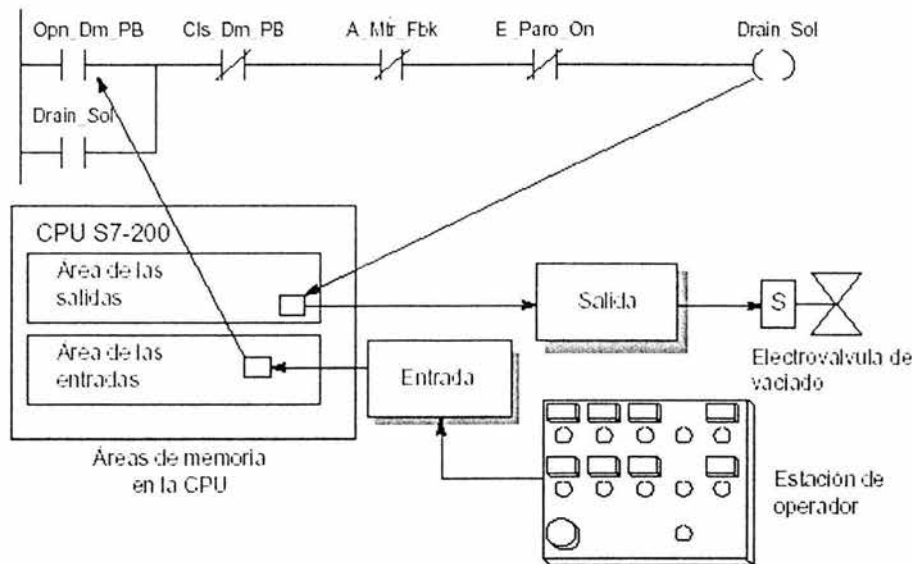


Figura 3-2 Referencias a las entradas y salidas en el programa

### Acceder a los datos en las áreas de memoria

La CPU almacena el estado de las entradas y salidas en determinadas áreas de la memoria. La figura 3-2 muestra el flujo simplificado de la información: entrada ► área de memoria ► programa ► área de memoria ► salida. A cada área de la memoria se ha asignado un identificador nemotécnico (por ejemplo “I” para las entradas y “Q” para las salidas) que se utiliza para acceder a los datos almacenados allí.

STEP 7-Micro/WIN provee direcciones absolutas para todas las áreas de memoria. Para acceder a una dirección determinada es preciso indicar el operando (por ejemplo. **I0.0** que es la primera entrada).

STEP 7-Micro/WIN también permite asignar nombres simbólicos a las direcciones absolutas. Una dirección absoluta de un área de memoria incluye no sólo el identificador de área (por ejemplo “V”), sino también el tamaño (hasta 4 bytes o 32 bits) de los datos a los que se desea acceder: B (byte), W (palabra = 2 bytes) o D (palabra doble = 4 bytes). La dirección absoluta incluye también un valor numérico: bien sea el número de bytes desde el comienzo del área de memoria (desplazamiento), o bien el número del elemento. Este valor depende del identificador de área.



### 3.3 Lenguaje de programación para las CPUs S7-200

Las CPUs S7-200 y STEP 7-Micro/WIN, asisten los siguientes tres editores de programas de lenguajes de programación:

- **El esquema de contactos (KOP):** Es un lenguaje de programación gráfico con componentes similares a los elementos de un esquema de circuitos. La lógica se divide en segmentos (“networks”). El programa se ejecuta un segmento tras otro, de izquierda a derecha y luego de arriba a abajo.
- **La lista de instrucciones (AWL):** Comprende un juego de operaciones nemotécnicas que representan las funciones de la CPU.
- **Esquema de funciones (FUP):** Es un lenguaje de programación que visualiza el programa gráficamente, de forma similar a los circuitos de puertas lógicas.

#### El esquema de contactos (KOP)

Al programar con KOP, se crean y se disponen componentes gráficos que conforman un segmento de operaciones lógicas. Como muestra la figura 3-3, se ofrecen los siguientes elementos básicos para crear programas:

**Contactos:** Un contacto representa un interruptor por el que circula la corriente cuando está cerrado.

**Bobinas:** Una bobina representa un relé que se excita cuando se le aplica tensión.

**Cuadros:** Un cuadro representa una función que se ejecuta cuando la corriente circula por él.

**Segmentos:** Cada uno de estos elementos constituye un circuito completo. La corriente circula desde la barra de alimentación izquierda pasando por los contactos cerrados para excitar las bobinas o cuadros.

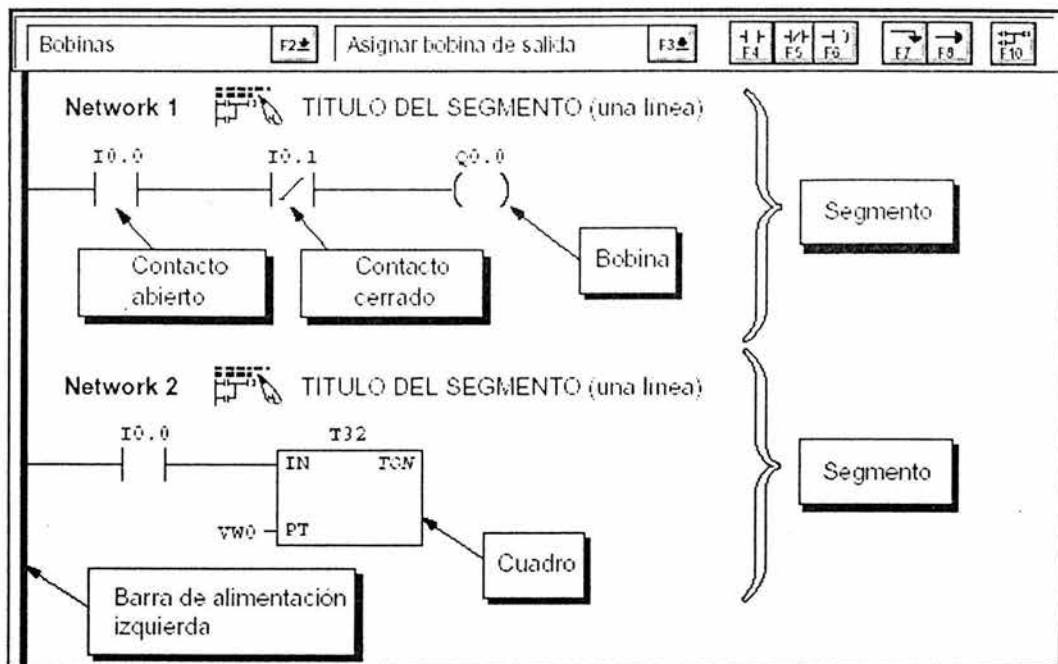


Figura 3-3 Elementos básicos del esquema de contactos (KOP)

### La lista de instrucciones (AWL)

La lista de instrucciones (AWL) es un lenguaje de programación en el que cada línea del programa contiene una operación que utiliza una abreviatura nemotécnica para representar una función de la CPU. Las operaciones se combinan en un programa, creando así la lógica de control de la aplicación.

La figura 3-4 muestra los elementos básicos de un programa AWL.

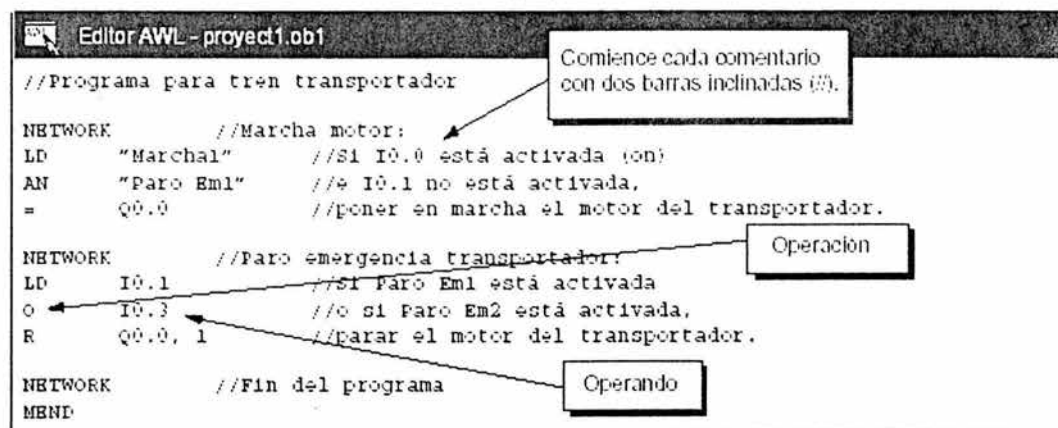


Figura 3-4 Ventana del editor AWL con un programa del ejemplo

Las operaciones AWL utilizan una pila lógica en la CPU para resolver la lógica. Como muestra la figura 3-5, dicha pila tiene nueve bits de profundidad y uno de ancho. La mayoría de las operaciones AWL utilizan el primer bit, o bien el primero y el segundo bit de la pila. Combinando los primeros dos bits de la pila, se pierde el valor del nivel superior y en su lugar se coloca el valor del nivel siguiente, es decir que a la pila se le resta un bit.

En tanto que la mayoría de las operaciones AWL leen sólo los valores de la pila lógica, muchas otras modifican también los valores ahí almacenados. La figura 3-5 muestra ejemplos de cómo tres operaciones utilizan la pila lógica.

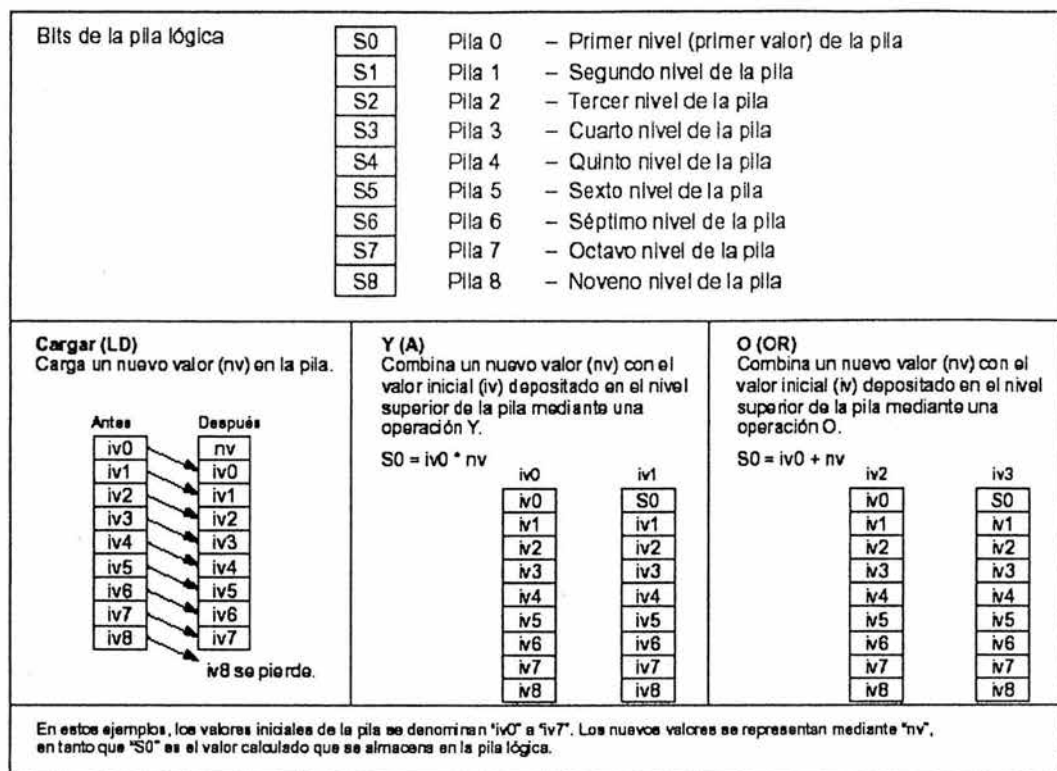


Figura 3-5 Pila lógica de la CPU S7-200

### El esquema de funciones (FUP)

El lenguaje de programación FUP no utiliza las barras de alimentación izquierda ni derecha. Sin embargo, el término "circulación de corriente" se utiliza para expresar el concepto análogo del flujo de señales por los bloques lógicos FUP.

La figura 3-6 muestra un ejemplo de programación de esquema de funciones (FUP).

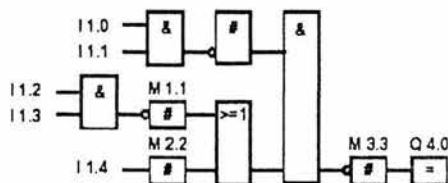


Figura 3-6 Programación en esquema de funciones (FUP)

### 3.4 Elementos básicos para estructurar un programa

La CPU S7-200 ejecuta continuamente el programa para controlar una tarea o un proceso. El programa se crea con STEP 7-Micro/WIN y se carga en la CPU. Desde el programa principal se pueden llamar diversas subrutinas o rutinas de interrupción.

#### Estructurar el programa

Los programas para la CPU S7-200 comprenden tres partes básicas: el programa principal, las subrutinas (opcional) y las rutinas de interrupción (opcional). Como muestra la figura 3-7, un programa S7-200 se estructura mediante los siguientes elementos:

**Programa Principal:** En esta parte del programa se disponen las operaciones que controlan la aplicación. Las operaciones del programa principal se ejecutan de forma secuencial en cada ciclo de la CPU. Para terminar el programa principal, utilice en KOP una bobina absoluta Finalizar programa principal, o en AWL una operación Finalizar programa principal (MEND). Véase (1) en la figura 3-7.

**Subrutinas:** Estos elementos opcionales del programa se ejecutan sólo cuando se llaman desde el programa principal. Se deben añadir siempre al final del programa principal (detrás de la bobina absoluta finalizar programa principal en KOP o detrás de la operación MEND en AWL). Utilice siempre una operación Retorno absoluto (RET) para terminar cada subrutina. Véase (2) en la figura 3-7.

**Rutinas de Interrupción:** Estos elementos opcionales del programa se ejecutan cada vez que se presente el correspondiente evento de interrupción. Se deben añadir siempre al final del programa principal (detrás

de la bobina absoluta Finalizar programa principal en KOP o detrás de la operación MEND en AWL). Utilice siempre una operación Retorno absoluto desde rutina de interrupción (RETI) para terminar cada rutina de interrupción. Véase (3) en la figura 3-7.

Las subrutinas y las rutinas de interrupción se deben añadir detrás de la bobina absoluta Finalizar programa principal en KOP o detrás de la operación MEND en AWL. No hay reglas adicionales en lo relativo a su disposición en el programa de usuario.

Las subrutinas y las rutinas de interrupción se pueden mezclar a voluntad después del programa principal, no obstante, para que la estructura del programa sea fácil de leer y comprender, es recomendable agrupar al final del programa principal primero todas las subrutinas y, después, todas las rutinas de interrupción.

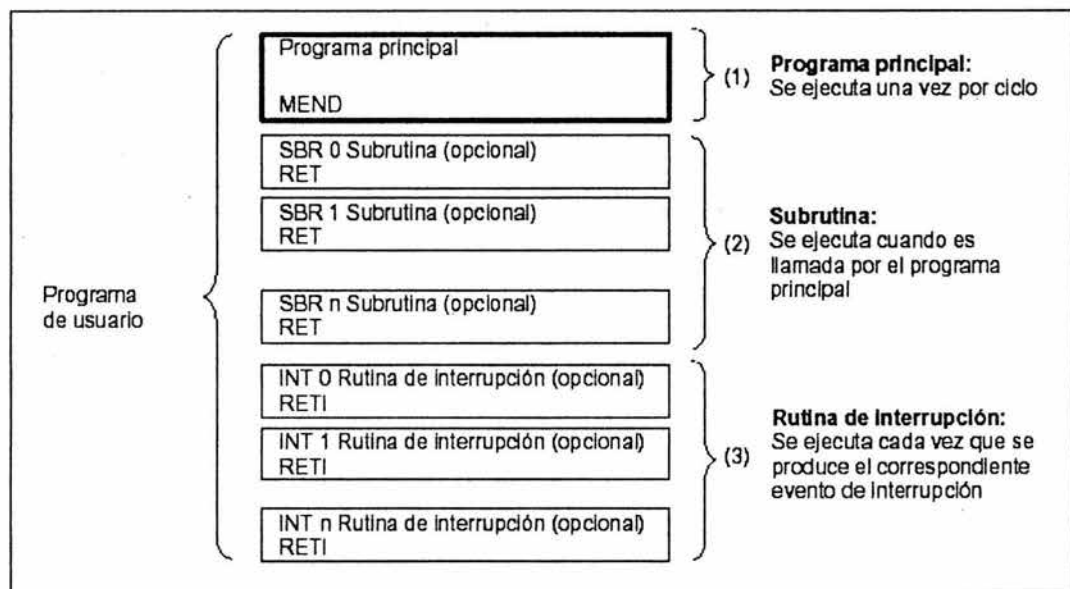


Figura 3-7 Estructura del programa de una CPU S7-200

A continuación se muestra un programa de ejemplo utilizando una subrutina y una rutina de interrupción, recordemos que ambas instrucciones se ejecutarán siempre y cuando el programa principal los mande llamar.

### 3.5 El ciclo de la CPU

La CPU S7-200 se ha previsto para que ejecute cíclicamente una serie de tareas, incluyendo el programa de usuario. Dicha ejecución se denomina ciclo. Durante el ciclo que se muestra en la figura 3-8, la CPU ejecuta la mayoría de las tareas siguientes (o todas ellas):

- Lee las entradas.
- Ejecuta el programa de usuario.
- Procesa las peticiones de comunicación.
- Efectúa un autodiagnóstico.
- Escribe las salidas.



Figura 3-8 ciclo de la CPU S7-200

#### Leer las 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.

La CPU reserva un espacio de la imagen del proceso de las entradas en incrementos de ocho bits (un byte). Si la CPU o el módulo de ampliación no proporcionan una entrada física para cada bit del byte reservado, no será posible asignar dichos bits a los módulos siguientes en la cadena de E/S o utilizarlos en el programa de usuario. Al comienzo de cada ciclo, la CPU pone a 0 estos bits no utilizados en la imagen del proceso, no obstante, si la CPU asiste varios módulos de ampliación y no se está utilizando su

capacidad de E/S (porque no se han instalado los módulos de ampliación), los bits de entradas de ampliación no utilizados se pueden usar como marcas internas adicionales.

La CPU no actualiza automáticamente las entradas analógicas como parte del ciclo y no ofrece una imagen del proceso para las mismas. A las entradas analógicas se debe acceder directamente desde el programa de usuario.

### **Ejecuta el Programa**

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

Si se utilizan interrupciones, las rutinas asociadas a los eventos de interrupción se almacenan 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, la CPU procesa los mensajes que haya recibido por el interface de comunicación.

### **Efectuar el autodiagnóstico de la CPU**

Durante el autodiagnóstico se comprueba el firmware de la CPU y la memoria del programa (sólo en modo RUN), así como el estado de los módulos de ampliación.

### **Escribir las salidas digitales**

Al final de cada ciclo, la CPU escribe los valores de la imagen del proceso de las salidas en las salidas digitales.



La CPU reserva un espacio de la imagen del proceso de las salidas en incrementos de ocho bits (un byte). Si la CPU o el módulo de ampliación no proveen una salida física para cada bit del byte reservado, no será posible asignar dichos bits a los módulos siguientes en la cadena de E/S. No obstante, los bits no utilizados de la imagen del proceso de las salidas se pueden usar como marcas internas adicionales.

La CPU no actualiza automáticamente las salidas analógicas como parte del ciclo y no ofrece una imagen del proceso para las mismas. A las salidas analógicas se debe acceder directamente desde el programa de usuario.

Cuando el modo de operación de la CPU se cambia de RUN a STOP, las salidas digitales adoptan los valores definidos en la tabla de salidas o conservan su estado actual

Las salidas analógicas conservan su último valor.

### **Interrumpir el ciclo**

Si se utilizan interrupciones, las rutinas asociadas a los eventos de interrupción se almacenan 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). La CPU procesa las interrupciones según su prioridad y después en el orden que aparecen.

### **Imagen del proceso de las entradas y salidas**

Por lo general, es recomendable utilizar la imagen del proceso, en vez de acceder directamente a las entradas o salidas durante la ejecución del programa. Las imágenes del proceso existen por tres razones.

- El sistema verifica todas las entradas al comenzar el ciclo. De este modo se sincronizan y congelan los valores de estas entradas durante la ejecución del programa. La imagen del proceso actualiza las salidas cuando termina de ejecutarse el programa. Ello tiene un efecto estabilizador en el sistema.
- El programa de usuario puede acceder a la imagen del proceso mucho más rápido de lo que podría acceder directamente a las entradas y salidas físicas, con lo cual se acelera su tiempo de ejecución.
- Las entradas y salidas son unidades de bit a las que se debe acceder en formato de bit. No obstante, la

imagen del proceso permite acceder a ellas en formato de bits, bytes, palabras y palabras dobles, lo que ofrece flexibilidad adicional.

### Control directo de las entradas y salidas

Las operaciones de control directo de las entradas y salidas (E/S) permiten acceder a la entrada o salida física, aunque el acceso a las E/S se efectúa por lo general a través de las imágenes del proceso. El acceso directo a una entrada no modifica la dirección correspondiente en la imagen del proceso de las entradas, en cambio, el acceso directo a una salida actualiza simultáneamente la dirección correspondiente en la imagen del proceso de las salidas.

## 3.6 Ajustar el modo de operación de la CPU

La CPU S7-200 tiene dos modos de operación:

**STOP:** La CPU no ejecuta el programa. Cuando está en modo STOP, es posible cargar programas o configurar la CPU.

**RUN:** La CPU ejecuta el programa, cuando está en modo RUN, no es posible cargar programas ni configurar la CPU.

El diodo luminoso (LED) en la parte frontal de la CPU indica el modo de operación actual. Para poder cargar un programa en la memoria de la CPU es preciso cambiar a modo STOP.

### Cambiar el modo de operación con el selector

El modo de operación de la CPU se puede cambiar manualmente accionando el selector ubicado debajo de la tapa de acceso a la CPU:

- Si el selector se pone en STOP, se detendrá la ejecución del programa.
- Si el selector se pone en RUN, se iniciará la ejecución del programa.

- Si el selector se pone en TERM (terminal), no cambiará el modo de operación de la CPU, no obstante, será posible cambiarlo utilizando el software de programación (STEP 7-Micro/WIN).

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

### Cambiar el modo de operación con STEP 7-Micro/WIN

Como muestra la figura 3-9, el modo de operación de la CPU se puede cambiar también con STEP 7-Micro/WIN. Para que ello sea posible, el selector de la CPU deberá estar en posición TERM o RUN.

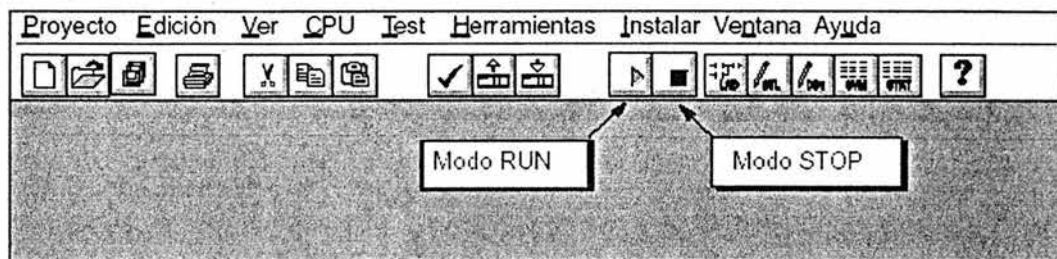


Figura 3-9 Utilizar STEP7-Micro/WIN para cambiar el modo de operación de la CPU

### 3.7 Determinar una contraseña para la CPU

Todas las CPUs S7-200 ofrecen una protección con contraseña para restringir el acceso a determinadas funciones. Con una contraseña se puede acceder a las funciones y a la memoria de la CPU. Si no se utiliza la opción de contraseña, la CPU permite un acceso ilimitado. Si está protegida con una contraseña, la CPU prohíbe todas las operaciones restringidas conforme a la configuración definida al determinar la contraseña.

### Restringir el acceso a la CPU

Como muestra la tabla 3-10, las CPUs S7-200 ofrecen tres niveles de protección para acceder a sus funciones. Cada uno de dichos niveles permite ejecutar determinadas funciones sin la contraseña. Si se introduce la contraseña correcta, es posible acceder a todas las funciones de la CPU. El ajuste estándar para las CPUs S7-200 es el nivel 1 (privilegios totales).

Si se introduce la contraseña a través de una red, no se afecta la protección con contraseña de la CPU. Si un usuario tiene acceso a las funciones restringidas de la CPU, ello no autoriza a los demás usuarios a acceder a dichas funciones. El acceso ilimitado a las funciones de la CPU sólo se permite a un usuario a la vez.

Una vez introducida la contraseña, el nivel de protección se conservará aproximadamente durante un minuto después de haber desconectado la unidad de programación de la CPU.

Tarea	Nivel 1	Nivel 2	Nivel 3
Leer y escribir datos de usuario	No restringido	No restringido	No restringido
Arrancar, detener y reprogramar la CPU			
Leer y escribir el reloj de tiempo real			
Leer los datos forzados en la CPU			
Cargar en la PG el programa de usuario, los datos y la configuración		Restringido	Restringido
Cargar en la CPU			
Borrar el programa de usuario, los datos y la configuración <sup>1</sup>			
Forzar datos o ejecutar uno o varios ciclo(s)			
Copiar en el cartucho de memoria			

Tabla 3-10 Niveles de acceso a la CPU S7-200

### Configurar la contraseña para la CPU S7-200

STEP 7-Micro/WIN permite determinar una contraseña para acceder a las funciones de la CPU. Elija el comando de menú **CPU ► Configurar** y seleccione la ficha “Contraseña” ver figura 3-11.

Indique el nivel de protección deseado. Introduzca y verifique luego la contraseña.

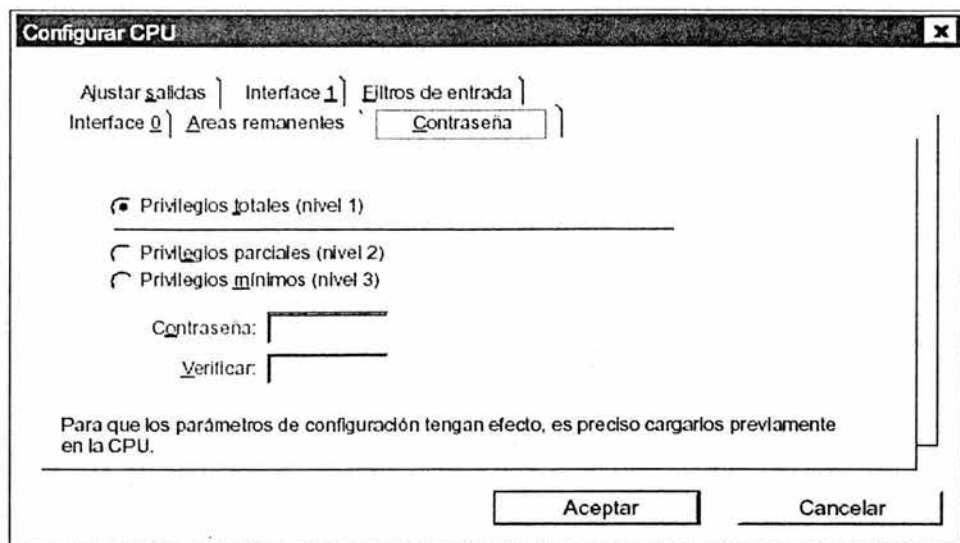


Figura 3-11 Configuración de contraseña para la CPU S7-200

### En caso de olvido de contraseña

Si se olvida la contraseña es preciso efectuar un borrado total de la memoria de la CPU y volver a cargar el programa. Al borrar la memoria de la CPU, ésta pasa a modo STOP y recupera los ajustes predeterminados, con excepción de la dirección de estación y del reloj de tiempo real.

Si desea borrar el programa de la CPU, elija el comando de menú **CPU ► Borrar** para visualizar el cuadro de diálogo “Borrar CPU”. Elija la opción “Todo” y confirme su acción haciendo clic en el botón “Aceptar”. Entonces se visualizará un cuadro de diálogo donde deberá introducir la contraseña “clearplc” que permitirá iniciar el borrado total.

La función de borrado total no borra el programa contenido en el cartucho de memoria. Puesto que en éste último se encuentra almacenado no sólo el programa, sino también la contraseña, es preciso volver a programar también dicho cartucho para borrar la contraseña olvidada.

## **CAPÍTULO 4**

Memoria de la CPU, tipo de datos y direccionamiento

## 4.1 Direccionamiento directo de las áreas de memoria de la CPU

La CPU S7-200 almacena información en diferentes áreas de la memoria que tienen direcciones unívocas.

Es posible indicar explícitamente la dirección a la que se desea acceder.

El programa puede acceder entonces directamente a la información.

### Acceder a los datos a través de direcciones

Para acceder a un bit en un área de memoria es preciso indicar la dirección del mismo, la cual está formada por un identificador de área, la dirección del byte y el número del bit. La figura 4-1 muestra un ejemplo de direccionamiento de un bit (denominado también direccionamiento “byte.bit”). En el ejemplo, el área de memoria y la dirección del byte (I=entrada y 3=byte 3) van seguidas de un punto decimal (“.”) que separa la dirección del bit (bit 4).

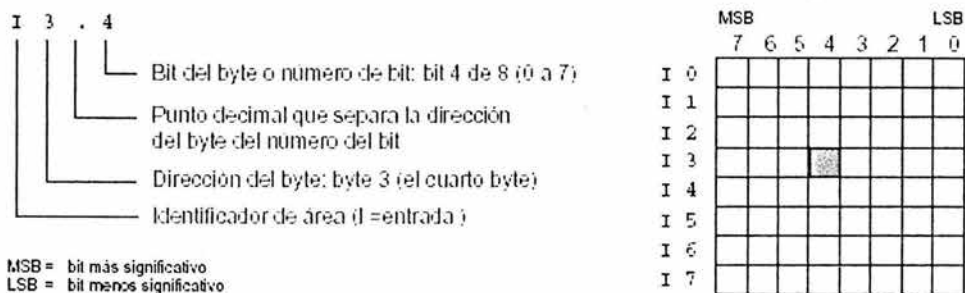


Figura 4-1 Acceder a un bit de datos en la memoria de la CPU (direccionamiento “byte.bit”)

Utilizando la dirección del byte se puede acceder a los datos de diversas áreas de memoria de la CPU (V, I, Q, M y SM) en formato de byte, palabra o palabra doble, ver figura 4-2. La dirección de un byte, de una palabra o de una palabra doble de datos en la memoria de la CPU se indica de forma similar a la dirección de un bit.

Esta última está formada por un identificador de área, el tamaño de los datos y la dirección inicial del valor del byte, de la palabra o de la palabra doble, como muestra la figura. Para acceder a los datos comprendidos en otras áreas de la memoria de la CPU (por ejemplo, T, C, HC y acumuladores) es preciso utilizar una dirección compuesta por un identificador de área y un número de elemento.

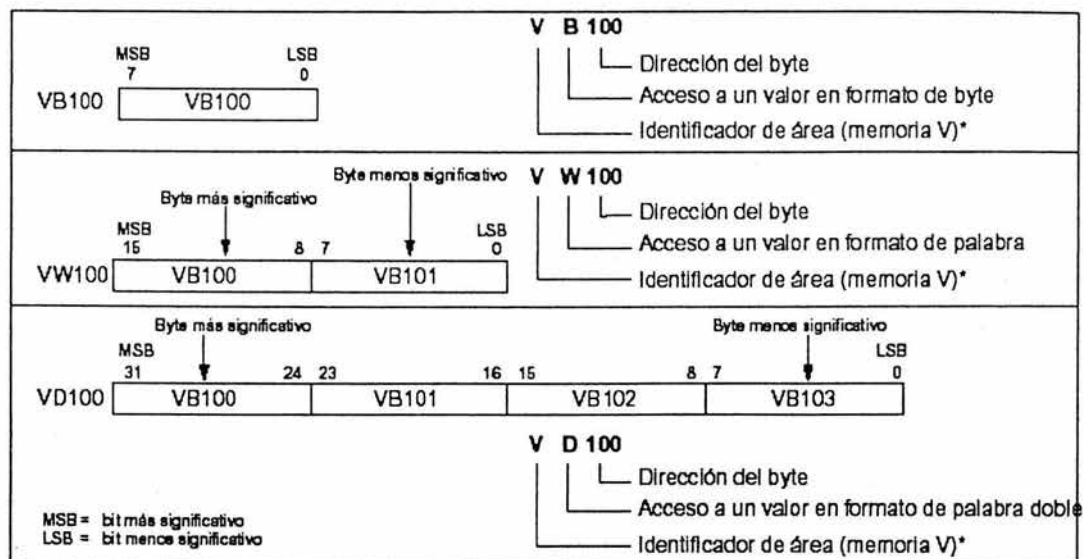


Figura 4-2 Acceso a la misma dirección en formato byte, palabra y palabra doble

### Representación numérica

La Figura 4-3 muestra el margen de números enteros representables en diversos tamaños de datos. Los números reales (en coma flotante) se representan como números de precisión simple de 32 bits. A los valores de números reales se accede en formato de palabra doble.

Tamaño de los datos	Margen de enteros sin signo		Margen de enteros con signo	
	Decimal	Hexadecimal	Decimal	Hexadecimal
B (byte): valor de 8 bits	0 a 255	0 a FF	-128 a 127	80 a 7F
W (palabra): valor de 16 bits	0 a 65.535	0 a FFFF	-32.768 a 32.767	8000 a 7FFF
D (palabra doble): valor de 32 bits	0 a 4.294.967.295	0 a FFFF FFFF	-2.147.483.647 a 2.147.483.647	8000 0000 a 7FFF FFFF

Figura 4-3 Indicadores de tamaño



### Direccionamiento de la imagen del proceso de las entradas (I)

Como se describe, la CPU lee las entradas físicas al comienzo de cada ciclo y escribe los correspondientes valores en la imagen del proceso de las entradas. A ésta última se puede acceder en formato de bit, byte, palabra o palabra doble.

Formato: Bit	I [direcc. del byte] . [direcc. del bit]	<b>I0.1</b>
Byte, palabra, palabra doble	I [tamaño] [direcc. del byte inicial]	<b>IB4</b>

### Direccionamiento de la imagen del proceso de las salidas (Q)

Al final de cada ciclo, la CPU copia en las salidas físicas el valor almacenado en la imagen del proceso de las salidas. A ésta última se puede acceder en formato de bit, byte, palabra o palabra doble.

Formato: Bit	Q [direcc. del byte] . [direcc. del bit]	<b>Q1.1</b>
Byte, palabra, palabra doble	Q [tamaño] [direcc. del byte inicial]	<b>QB5</b>

### Direccionamiento de la memoria de variables (V)

La memoria de variables (memoria V) se puede utilizar para depositar los resultados intermedios calculados por las operaciones en el programa. La memoria V también permite almacenar otros datos que pertenezcan al proceso o tarea actuales. A la memoria de variables se puede acceder en formato de bit, byte, palabra o palabra doble.

Formato: Bit	V [direcc. del byte] . [direcc. del bit]	<b>V10.2</b>
Byte, palabra, palabra doble	V [tamaño] [direcc. del byte inicial]	<b>VW100</b>

### Direccionamiento del área de marcas (M)

Las marcas internas (área de marcas M) se pueden utilizar como relés de control para almacenar el estado intermedio de una operación u otras informaciones de control. Al área de marcas se puede acceder en formato de bit, byte, palabra o palabra doble.

Formato: Bit	M [direcc. del byte] . [direcc. del bit]	<b>M26.7</b>
Byte, palabra, palabra doble	M [tamaño] [direcc. del byte inicial]	<b>MD20</b>

### Direccionamiento de los relés de control (S)

Los relés de control secuencial (S) permiten organizar los pasos del funcionamiento de una máquina en segmentos equivalentes en el programa. Dichos relés permiten segmentar lógicamente el programa de usuario. A los relés de control secuencial (SCR) se puede acceder en formato de bit, byte, palabra o palabra doble.

Formato: Bit	S [direcc. del byte] . [direcc. del bit]	<b>S3.1</b>
Byte, palabra, palabra doble	S [tamaño] [direcc. del byte inicial]	<b>SB4</b>

### Direccionamiento de las marcas especiales (SM)

Las marcas especiales permiten intercambiar datos entre la CPU y el programa. Dichas marcas se puede utilizar para seleccionar y controlar algunas funciones especiales de la CPU S7-200, tales como:

- Un bit que se activa sólo en el primer ciclo.
- Bits que se activan y se desactivan en determinados intervalos.
- Bits que muestran el estado de operaciones matemáticas y de otras operaciones.

El área de marcas especiales se basa en bits, es posible acceder a los datos en formato de bit, byte, palabra o palabra doble.

Formato: Bit SM [direcc. del byte] . [direcc. del bit] **M0.1**  
 Byte, palabra, palabra doble SM [tamaño] [direcc. del byte inicial] **SMB86**

### Direccionamiento del área de temporizadores (T)

En las CPUs S7-200, los temporizadores son elementos que cuentan intervalos de tiempo. Los temporizadores de las CPUs S7-200 tienen resoluciones (intervalos) de 1 ms, 10 ms y 100 ms.

Hay dos variables asociadas a los temporizadores:

**Valor actual:** En este número entero de 16 bits con signo se deposita el valor de tiempo contado por el temporizador.

**Bit del temporizador (bit T):** Este bit se activa (se pone a 1) cuando el valor actual del temporizador es mayor o igual al valor predeterminado. (Éste último se introduce como parte de la operación).

A estas dos variables se accede utilizando la dirección del temporizador (T + número del temporizador). Dependiendo de la operación utilizada, se accede al bit del temporizador o al valor actual. Las operaciones con operandos en formato de bit acceden al bit del temporizador, en tanto que las operaciones con operandos en formato de palabra acceden al valor actual, como muestra la figura 4-4.

Formato: T [número del temporizador] **T24**

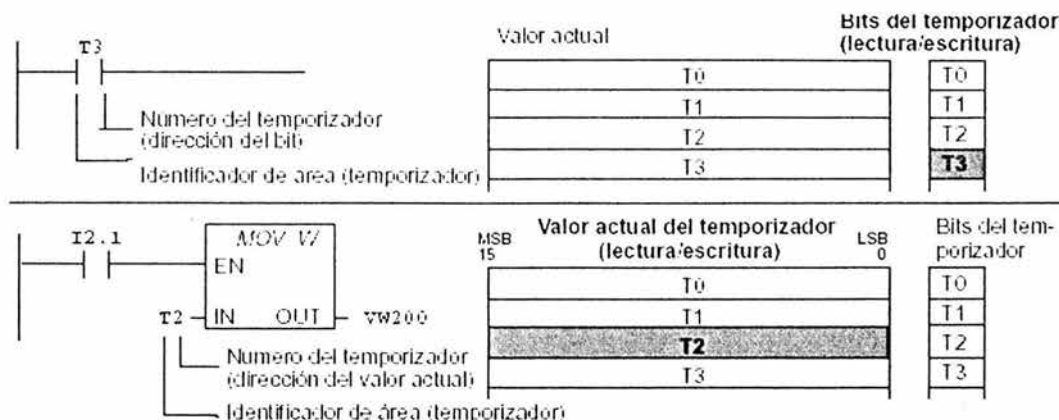


Figura 4-4 Acceder a los datos del temporizador

### Direccionamiento de los contadores (C)

Los contadores de las CPUs S7-200 son elementos que cuentan los cambios de negativo a positivo en las entradas de contaje.

Hay dos tipos de contadores: uno que cuenta sólo adelante y uno que cuenta tanto adelante como atrás, hay dos variables asociadas a los contadores:

**Valor actual:** En este número entero de 16 bits con signo se deposita el valor de contaje acumulado.

**Bit del contador (bit C):** Este bit se activa (se pone a 1) cuando el valor actual del contador es mayor o igual al valor predeterminado. (Éste último se introduce como parte de la operación).

A estas dos variables se accede utilizando la dirección del contador (C + número del contador). Dependiendo de la operación utilizada, se accede al bit del contador o al valor actual. Las operaciones con operandos en formato de bit acceden al bit del contador, en tanto que las operaciones con operandos en formato de palabra acceden al valor actual, como muestra la figura 4-5, la operación contacto abierto accede al bit del contador, en tanto que la operación transferir palabra (MOV\_W) accede al valor actual del contador.

Formato:

C [número del contador]      C20

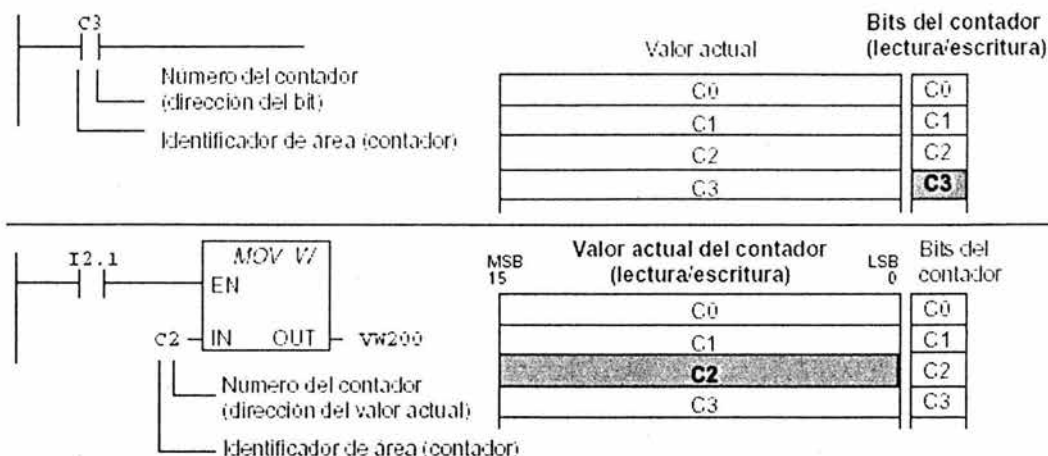


Figura 4-5 Acceder a los datos del contador

### Direccionamiento de las entradas analógicas (AI)

La CPU S7-200 convierte valores reales analógicos (por ejemplo, temperatura, tensión, etc.) en valores digitales en formato de palabra (de 16 bits). A estos valores se accede con un identificador de área (AI), seguido del tamaño de los datos (W) y de la dirección del byte inicial.

Puesto que las entradas analógicas son palabras que comienzan siempre en bytes pares (por ejemplo, 0, 2, 4, etc.) es preciso utilizar direcciones con bytes pares (por ejemplo, AIW0, AIW2, AIW4, etc.) para acceder a las mismas, como muestra la figura 4-6. Las entradas analógicas son valores de sólo lectura.

Formato: AIW [dirección del byte inicial] AIW4

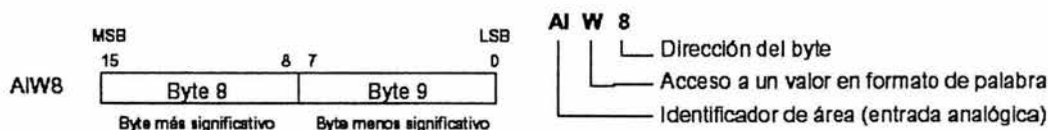


Figura 4-6 Acceso a una entrada analógica

### Direccionamiento de las salidas analógicas (AQ)

La CPU S7-200 convierte valores digitales en formato de palabra (de 16 bits) en valores reales analógicos (por ejemplo, corriente o voltaje), proporcionales al valor digital. A estos valores se accede con un identificador de área (AQ), seguido del tamaño de los datos (W) y de la dirección del byte inicial.

Puesto que las salidas analógicas son palabras que comienzan siempre en bytes pares (por ejemplo, 0, 2, 4, etc.) es preciso utilizar direcciones con bytes pares (por ejemplo, AQW0, AQW2, AQW4, etc.). Para acceder a las mismas, como muestra la figura 4-7. El programa no puede leer los valores de las salidas analógicas.

Formato: AQW [dirección del byte inicial]

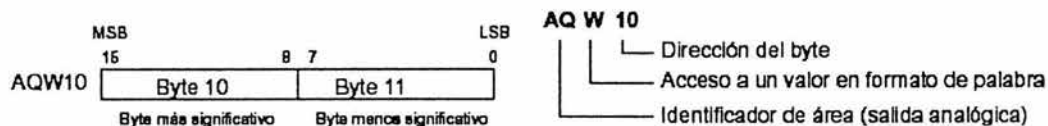


Figura 4-7 Acceso a una salida analógica

### Direccionamiento de los acumuladores (AC)

Los acumuladores son elementos de lectura/escritura que se utilizan igual que una memoria. Los acumuladores se pueden usar por ejemplo, para transferir parámetros de y a subrutinas, así como para almacenar valores intermedios utilizados en cálculos. La CPU dispone de cuatro acumuladores de 32 bits (AC0, AC1, AC2 y AC3). A los acumuladores se puede acceder en formato de byte, palabra o palabra doble. Como muestra la figura 4-8, cuando se accede a un acumulador en formato de byte o de palabra se utilizan los 8 ó 16 bits menos significativos del valor almacenado en el acumulador. Cuando se accede a un acumulador en formato de palabra doble, se usan todos los 32 bits. La operación utilizada para el acceso al acumulador determina el tamaño de los datos a los que se accede.

Formato: AC [número del acumulador] AC0

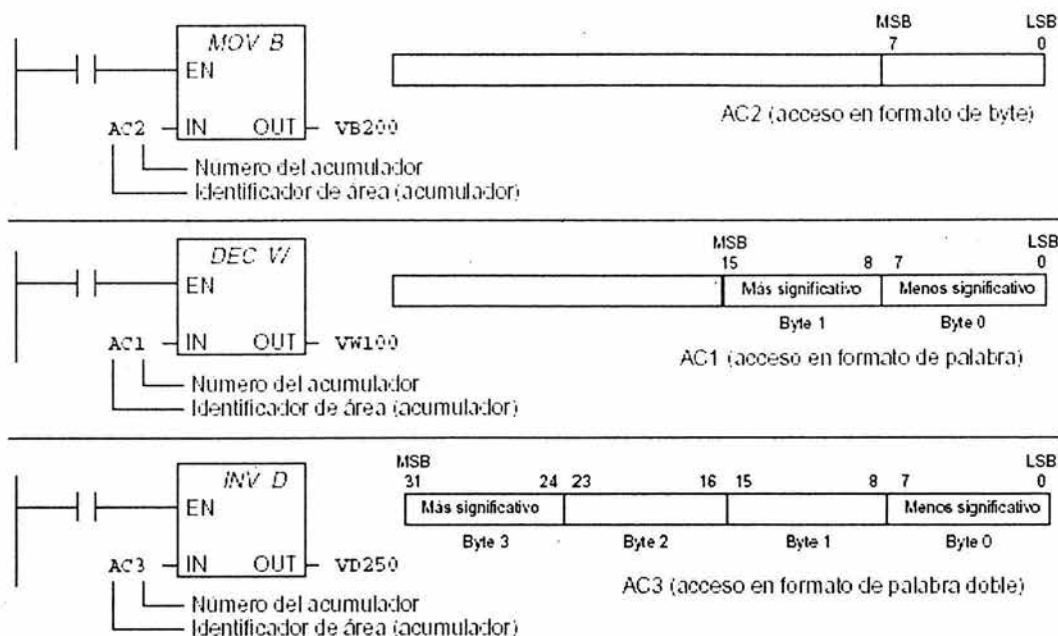


Figura 4-8 Acceso a los acumuladores

### Direccionamiento de los contadores rápidos (HC)

Los contadores rápidos son elementos que cuentan los eventos más deprisa de lo que puede explorarlos la CPU. Tienen un valor de contaje de entero de 32 bits con signo (denominado también valor actual). Para acceder al valor de contaje del contador rápido, se indica la dirección del mismo (utilizando el identificador HC) y el número del contador (por ejemplo, HC0). El valor actual del contador rápido es de sólo lectura, pudiéndose acceder al mismo sólo en formato de palabra doble de 32 bits, como muestra la figura 4-9.

Formato: HC [número del contador rápido] HC1

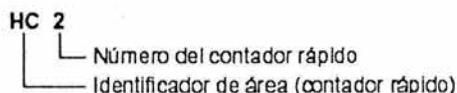
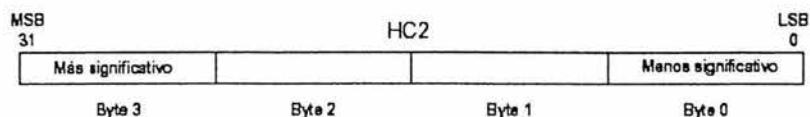


Figura 4-9 Acceso a los valores actuales del contador rápido

### Utilizar valores de constantes

Las constantes se pueden utilizar en numerosas operaciones de la CPU S7-200. Pueden ser valores de bytes, palabras o palabras dobles. La CPU almacena todas las constantes como números binarios que se pueden representar en formato decimal, hexadecimal o ASCII.

Formato decimal:	[valor decimal]
Formato hexadecimal:	16#[valor hexadecimal]
Formato ASCII:	' [texto ASCII] '

La CPU S7-200 no permite indicar tipos de datos específicos ni comprobar datos (por ejemplo, indicar si la constante es un entero de 16 bits, un entero con signo o un entero de 32 bits). Por ejemplo, la operación sumar puede utilizar el valor depositado en VW100 como entero con signo, en tanto que una operación de combinación con O-exclusiva puede emplear ese mismo valor de VW100 como valor binario sin signo.

## 4.2 Direccionamiento indirecto de las áreas de memoria de la CPU

El direccionamiento indirecto utiliza un puntero para acceder a los datos de la memoria. La CPU S7-200 permite utilizar punteros para direccionar indirectamente las siguientes áreas de memoria: I, Q, V, M, S, T (sólo el valor actual) y C (sólo el valor actual). Los valores analógicos o de bits individuales no se pueden direccionar de forma indirecta.



## Crear un puntero

Para acceder indirectamente a una dirección en la memoria es preciso crear primero un puntero que señale a esa dirección. Los punteros son valores de palabra doble que señalan a otra dirección en la memoria. Como punteros sólo se pueden utilizar direcciones de la memoria V o los acumuladores (AC1, AC2 y AC3). Para crear un puntero se debe utilizar la operación transferir palabra doble (MOVD) con objeto de transferir la dirección indirecta a la del puntero. El operando de entrada de la operación debe ir precedido de un carácter "&" para determinar que a la dirección indicada por el operando de salida (es decir, el puntero) se debe transferir la dirección y no su contenido.

Ejemplo:

MOVD	&VB100, VD204
MOVD	&MB4, AC2
MOVD	&C4, VD6

Si desea acceder indirectamente a un valor de palabra o de palabra doble en las áreas de memoria I, Q, V, M o S, debe indicar la dirección del byte inicial del valor como operando de entrada de la operación MOVD utilizada para crear el puntero. Por ejemplo, VB100 es la dirección del byte inicial de VW100, en tanto que MB4 es la dirección del byte inicial de MD4. Si al valor de palabra o de palabra doble se le ha asignado un nombre simbólico, éste último no se puede usar en la operación MOVD utilizada para crear el puntero, puesto que la dirección del byte inicial del valor se debe indicar en el operando de entrada de la operación. En estos casos es preciso asignarle un nombre simbólico diferente a la dirección del byte inicial de la palabra o de la palabra doble utilizada para crear el puntero.

Ejemplo:

- “Velocidad\_Bomba” es el nombre simbólico de VW100
- “Velocidad\_Bomba\_IB” es el nombre simbólico de VB100  
(que es el byte inicial del valor de palabra almacenado en VW100)

**MOVD &“Velocidad\_Bomba”, AC1** no válido (&VW100 no está permitido)  
**MOVD &“Velocidad\_Bomba\_IB”, AC1** correcto (&VB100 está permitido)

### Utilizar un puntero para acceder a los datos

Introduciendo un asterisco (\*) delante de un operando de una operación, se indica que el operando es un puntero. En el ejemplo que muestra la figura 4-10, \*AC1 significa que AC1 es el puntero del valor de palabra indicado por la operación Transferir palabra (MOVW). En este ejemplo, los valores almacenados en V200 y V201 se transfieren al acumulador AC0.

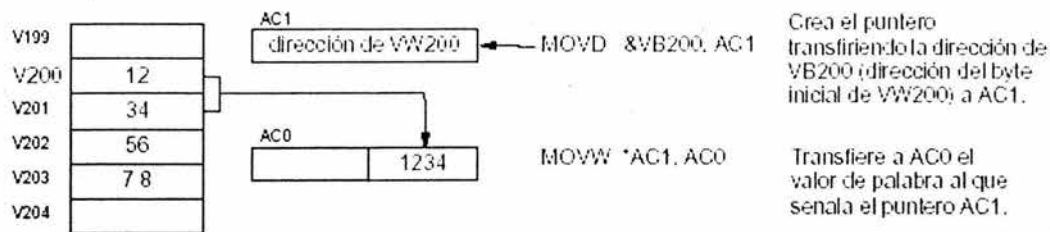


Figura 4-10 Utilizar un puntero para el direccionamiento indirecto

### Modificar punteros

Los valores de los punteros se pueden modificar, puesto que los punteros son valores de 32 bits, para cambiarlos es preciso utilizar operaciones de palabra doble. Las operaciones aritméticas simples, tales como sumar o incrementar, se pueden utilizar para modificar los valores de los punteros. Recuerde que debe indicar el tamaño de los datos a los que desea acceder:

- Para acceder a una byte, sume o incremente el valor del puntero en 1.
- Para acceder a una palabra, o bien al valor actual de un temporizador o de un contador, sume o incremente el valor del puntero en 2.
- Para acceder a una palabra doble, sume o incremente el valor del puntero en 4.

La figura 4-11 muestra un ejemplo de cómo crear un puntero de direccionamiento indirecto y de cómo acceder indirectamente a los datos e incrementar el puntero.

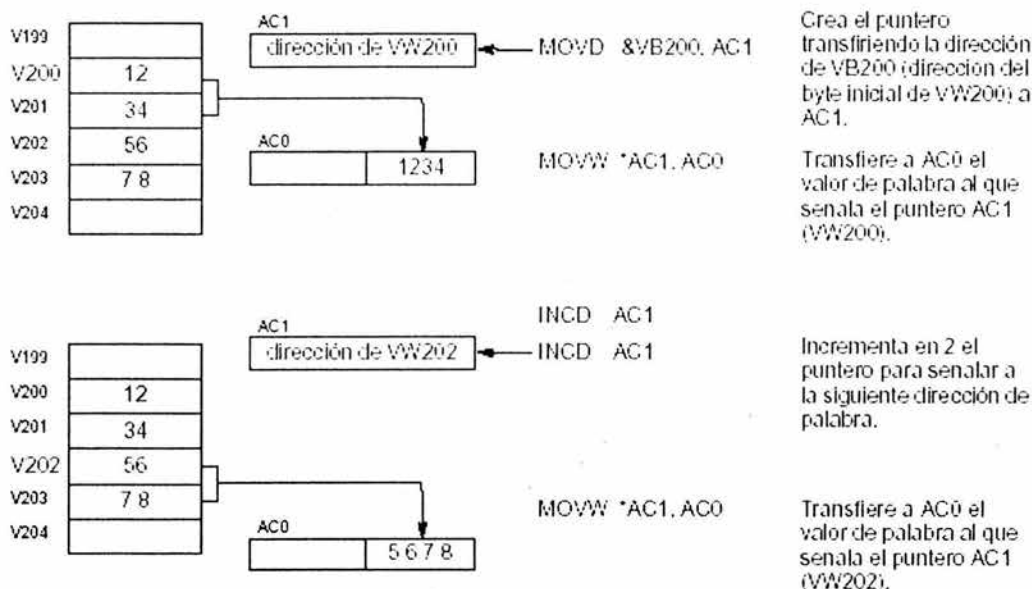


Figura 4-11 Modificar un puntero al acceder a un valor de palabra

### 4.3 Respaldo de datos en la CPU S7-200

La CPU S7-200 ofrece diversos métodos para garantizar que el programa, los datos del mismo y los datos de configuración de la CPU se almacenen de forma segura:

- La CPU dispone de una EEPROM no volátil para almacenar todo el programa, así como algunas áreas de datos y la configuración de la CPU, ver figura 4-12.
- La CPU dispone de un condensador de alto rendimiento que conserva todo el contenido de la memoria RAM después de un corte de alimentación. Según el tipo de CPU, el condensador puede respaldar la memoria durante varios días.
- Algunas CPUs asientan un cartucho de pila opcional que prolonga el tiempo durante el que se puede respaldar la memoria RAM después de un corte de alimentación. El cartucho de pila se activa sólo cuando se descarga el condensador de alto rendimiento.

RAM: respaldada por el condensador de alto rendimiento y por el cartucho de pila opcional



EEPROM: almacenamiento no volátil



Figura 4-12 Áreas de memoria de la CPU S7-200

### Cargar programas en la CPU y en la PG

El proyecto comprende tres elementos: el programa de usuario, el bloque de datos (opcional) y la configuración de la CPU (opcional). Como muestra la figura 4-13, cargando el programa en la CPU se almacenan dichos elementos en la memoria RAM (de la CPU). La CPU también copia automáticamente el programa de usuario, el bloque de datos (DB1) y la configuración de la CPU en la EEPROM no volátil para que se almacenen allí.

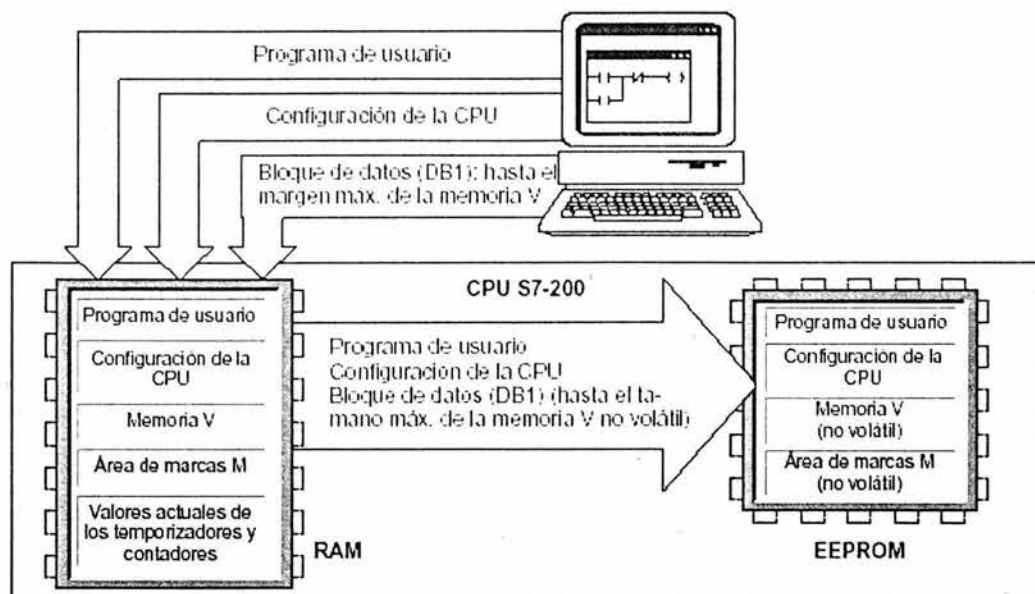


Figura 4-13 Cargar los elementos del programa en la CPU

Como muestra la figura 4-14, cuando un programa se carga de la CPU en el PC (o la PG), el programa de usuario y la configuración de la CPU se cargan de la RAM en el PC (o la PG). Cuando se carga el bloque de datos en el PC (o la PG), el área no volátil del bloque de datos (almacenada en la EEPROM) se fusiona con el resto del bloque de datos (si existe) que esté almacenado en la RAM. El bloque de datos completo se transfiere entonces al PC (ó a la PG). El tamaño de la memoria V no volátil depende de la CPU.

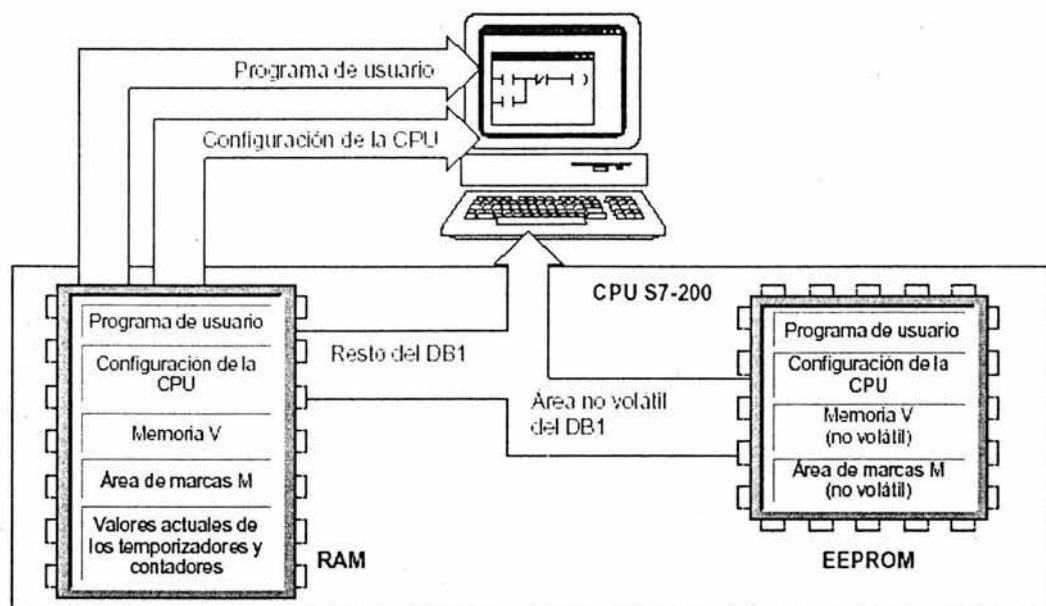


Figura 4-14 Cargar los elementos del programa en el PC (o la PG)

#### Almacenar automáticamente los datos del área de marcas (M) en caso de un corte de alimentación

Si se define que los primeros 14 bytes del área de marcas (MB0 a MB13) sean remanentes, se copiarán automáticamente en la EEPROM no volátil cuando se produzca un corte de alimentación de la CPU. Como muestra la figura 4-15, la CPU transfiere dichas áreas remanentes del área de marcas a la EEPROM.

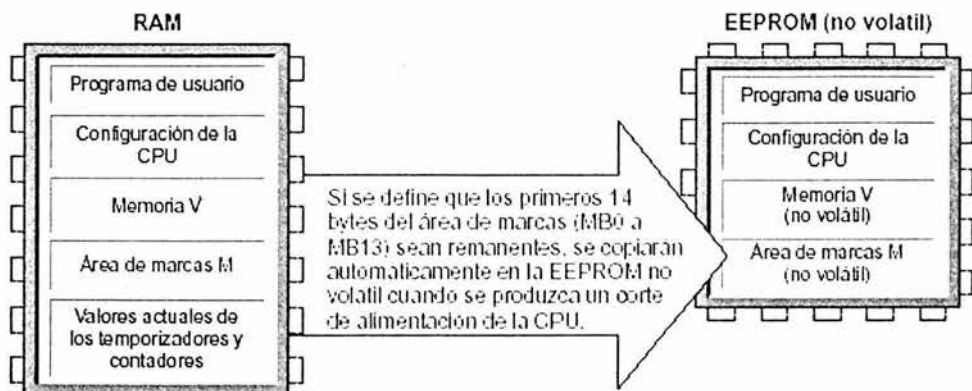


Figura 4-15 Almacenar partes del área de marcas (M) en la EEPROM en caso de un corte de alimentación

### Respalda la memoria al conectar la alimentación

Cuando se conecta la alimentación, la CPU restablece el programa de usuario y la configuración (de la CPU) que se han depositado en la memoria EEPROM ver figura 4-16.

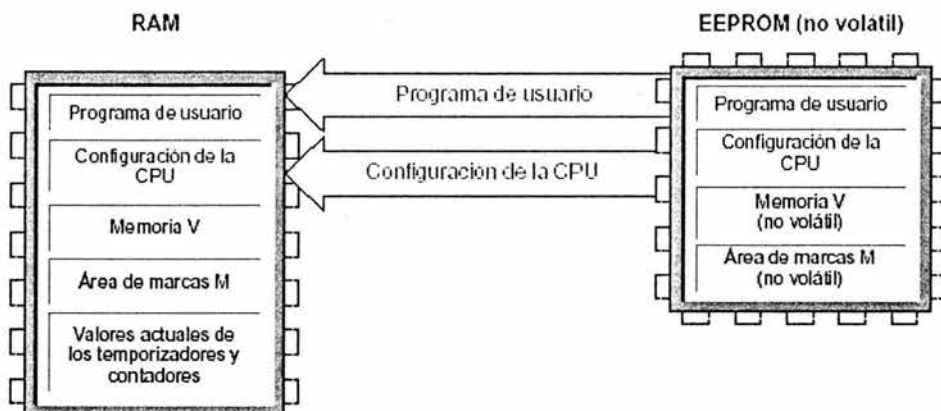


Figura 4-16 Restablecer el programa de usuario y la configuración de la CPU al conectar la alimentación

Cuando se conecta la alimentación, la CPU comprueba la memoria RAM para verificar si el condensador de alto rendimiento ha respaldado los datos almacenados en la RAM. En caso afirmativo, no se modificarán las áreas remanentes de la misma. Como muestra la figura 4-17, las áreas no remanentes de la

memoria V se restablecen conforme a la correspondiente área no volátil de la memoria V contenida en la EEPROM.

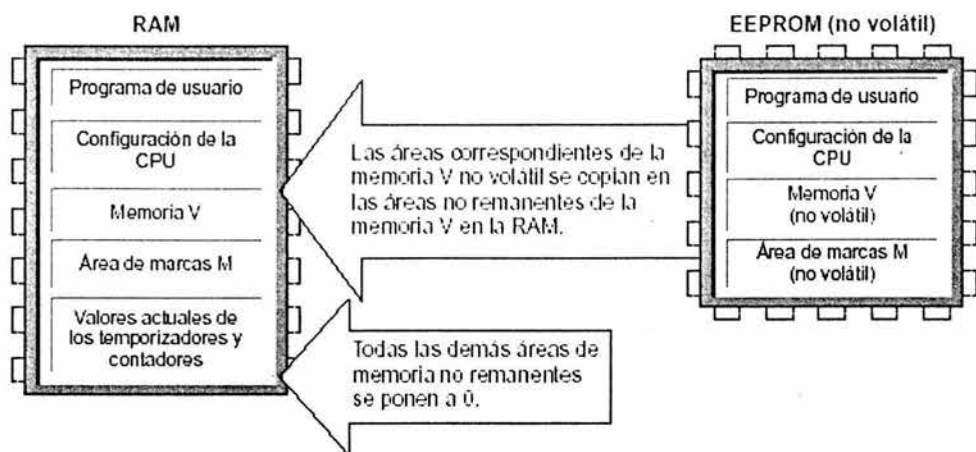


Figura 4-17 Restablecer los datos del programa al conectarse la alimentación habiéndose respaldado los datos en la RAM

Si el contenido de la RAM no se ha respaldado (por ejemplo, en el caso de un corte de alimentación prolongado), la CPU borra la memoria RAM (tanto las áreas remanentes como las no remanentes) y activa la marca datos remanentes perdidos (SM0.2) en el primer ciclo que le sigue a la puesta en marcha. Como muestra la figura 4-18, los datos almacenados en la EEPROM no volátil se copian entonces en la memoria RAM.



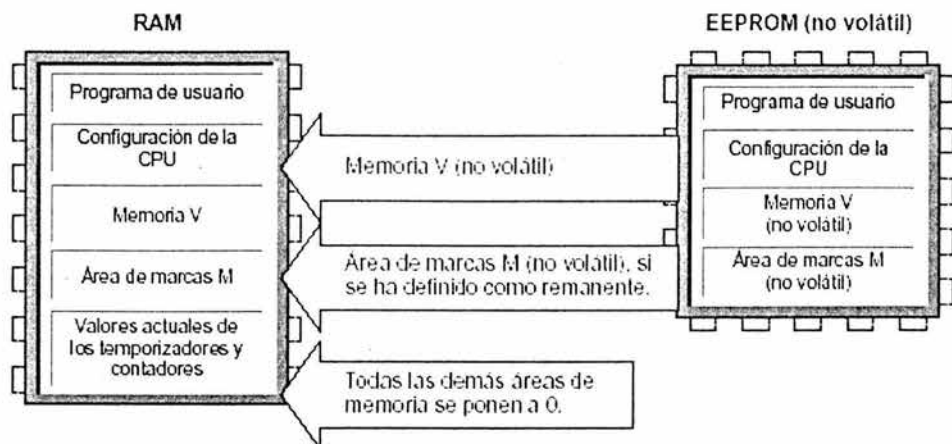


Figura 4-18 Restablecer los datos del programa al conectarse la alimentación sin haberse respaldado los datos en la RAM

### Definir las áreas remanentes

Como muestra la figura 4-19, el usuario puede definir hasta seis áreas remanentes para elegir las áreas de memoria que se deberán respaldar cuando se interrumpa la alimentación. Se puede determinar que sean remanentes los márgenes de direcciones en las áreas de memoria V, M, C y T. En el caso de los temporizadores, sólo es posible respaldar los de retardo a la conexión memorizado (TONR).

Para definir las áreas remanentes de la memoria, elija el comando de menú **CPU ► Configurar** y haga clic en la ficha "Áreas remanentes".

La figura 4-10 muestra el cuadro de diálogo donde se definen dichas áreas. Para visualizar las áreas remanentes predeterminadas de su CPU, haga clic en el botón **Estándar**.

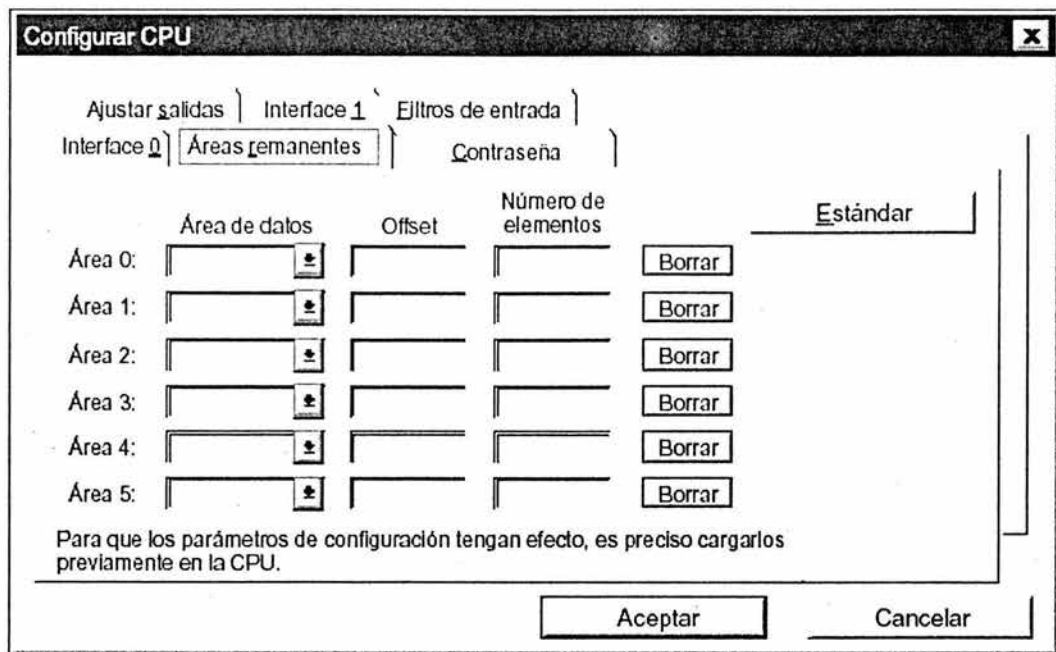


Figura 4-19 Configurar las áreas remanentes de la memoria de la CPU

#### 4.4 Guardar el programa en un cartucho de memoria

Algunas CPUs asisten un cartucho de memoria opcional que permite guardar el programa en una EEPROM portátil. Dicho cartucho se puede utilizar como un disquete. La CPU guarda los siguientes datos en el cartucho de memoria:

- Programa de usuario.
- Datos almacenados en la memoria de variables no volátil de la EEPROM.
- Configuración de la CPU.

##### Copiar en el cartucho de memoria

El programa se puede copiar en el cartucho de memoria desde la RAM sólo si se ha arrancado la CPU y si está insertado dicho cartucho.

El cartucho de memoria se puede instalar o extraer estando conectada la alimentación de la CPU. Para instalarlo, retire la cinta protectora del receptáculo e inserte allí el cartucho. El receptáculo está ubicado debajo de la tapa de acceso de la CPU. (El cartucho de memoria se ha diseñado de forma que sólo se pueda insertar en un sólo sentido en el receptáculo). Una vez insertado el cartucho, copie el programa como se indica a continuación:

1. Cargue el programa en la CPU si no lo ha hecho todavía.
2. Elija el comando de menú **CPU ► Cartucho de memoria** para copiar el programa en el cartucho de memoria. La figura 4-20 muestra los componentes de la memoria de la CPU que se almacenan en dicho cartucho.
3. Extraiga el cartucho de memoria del receptáculo (opcional).

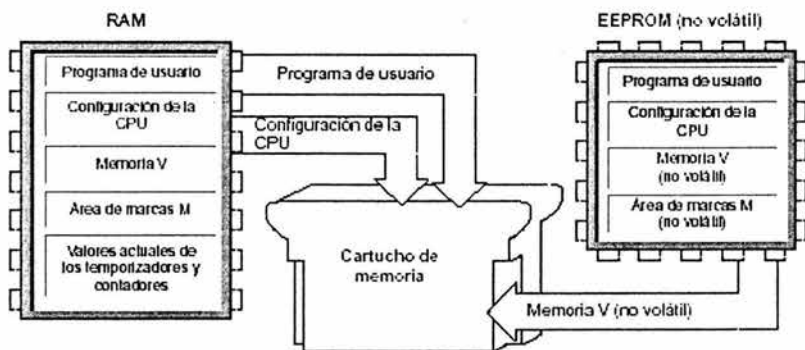


Figura 4-20 Copiar la memoria de la CPU en el cartucho de memoria

### Restablecer el programa y la memoria mediante un cartucho de memoria

Para transferir el programa de un cartucho de memoria en la CPU, es preciso desconectar la alimentación de la CPU y conectarla de nuevo con el cartucho insertado. Como muestra la figura 4-21, la CPU ejecuta las siguientes tareas después del arranque (si está insertado el cartucho de memoria).

- Se borra la memoria RAM.
- El contenido del cartucho de memoria se copia en la memoria RAM.
- El programa de usuario, la configuración de la CPU y la memoria V (hasta el tamaño máximo del área no volátil de la misma) se copian en la EEPROM no volátil.

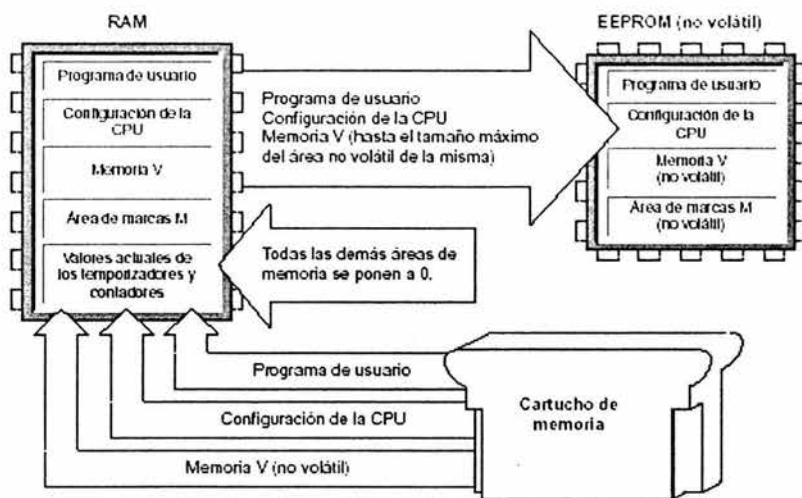


Figura 4-21 Restablecer la memoria durante el arranque estando insertado un cartucho de memoria

## **CAPÍTULO 5**

### **Entradas y salidas**

## 5.1 Entradas y salidas integradas y adicionales

El sistema se controla mediante entradas y salidas (E/S). Las entradas vigilan las señales de los dispositivos de campo (por ejemplo, sensores e interruptores), mientras que las salidas supervisan las bombas, motores u otros aparatos del proceso. Se dispone de entradas y salidas integradas (en la CPU), así como de E/S adicionales (en los módulos de ampliación).

- Las CPUs S7-200 disponen de un número determinado de entradas y salidas digitales.
- Las CPUs S7-200 asisten módulos de ampliación con entradas y salidas tanto digitales como analógicas.

### Direccionar las E/S integradas y adicionales

Las entradas y salidas integradas de la unidad central (CPU) tienen direcciones fijas. Para añadir a la CPU entradas y salidas adicionales, se pueden conectar módulos de ampliación a la derecha de la CPU, formando una cadena de E/S. Las direcciones de las E/S de cada módulo vienen determinadas por el tipo de E/S y la posición del módulo en la cadena, con respecto al anterior módulo de entradas o de salidas del mismo tipo. Por ejemplo, un módulo de salidas no afecta las direcciones de un módulo de entradas y viceversa, igualmente, los módulos analógicos no afectan el direccionamiento de los módulos digitales y viceversa.

Los módulos de ampliación digitales reservan siempre un espacio de la imagen del proceso en incrementos de ocho bits (un byte). Si un módulo no dispone de un punto físico para cada bit de cada byte reservado, se pierden estos bits no utilizados y no se pueden asignar a los módulos siguientes en la cadena de E/S. Por lo que respecta a los módulos de salidas, los bits no utilizados en los bytes reservados pueden servir de marcas internas (M).

En cuanto a los módulos de entradas, los bits no utilizados en los bytes reservados se ponen a cero cada vez que se actualizan las entradas, por lo que no pueden servir de marcas internas.

Los módulos de ampliación analógicos se prevén siempre en incrementos de dos puntos. Si un módulo no ofrece E/S físicas para cada uno de dichos puntos, se pierden los mismos y no se pueden asignar a los módulos siguientes en la cadena de E/S. Puesto que para las E/S analógicas no se prevé imagen del proceso, no hay ninguna manera de aprovechar estas E/S no utilizadas.

Todos los accesos de E/S analógicas se establecen en el mismo instante de ejecutarse la operación.

La figura siguiente muestra un ejemplo de cómo las diferentes configuraciones del hardware afectan la numeración de las entradas y salidas.



I0.0	Q0.0
I0.1	Q0.1
I0.2	Q0.2
I0.3	Q0.3
I0.4	
I0.5	

Ejemplo de numeración de E/S para una CPU 221

	Module 0	Module 1	Module 2	Module 3	Module 4
<b>CPU 224</b>	<b>4 In / 4 Out</b>	<b>8 In</b>	<b>4 AI/ 1 AQ</b>	<b>8 Out</b>	<b>4 AI/ 1 AQ</b>

I0.0	Q0.0	I2.0	Q2.0	I3.0	AIW0	AQW0	Q3.0	AIW8	AQW4
I0.1	Q0.1	I2.1	Q2.1	I3.1	AIW2		Q3.1	AIW10	
I0.2	Q0.2	I2.2	Q2.2	I3.2	AIW4		Q3.2	AIW12	
I0.3	Q0.3	I2.3	Q2.3	I3.3	AIW6		Q3.3	AIW14	
I0.4	Q0.4			I3.4			Q3.4		
I0.5	Q0.5			I3.5			Q3.5		
I0.6	Q0.6			I3.6			Q3.6		
I0.7	Q0.7			I3.7			Q3.7		
I1.0	Q1.0								
I1.1	Q1.1								
I1.2									
I1.3									
I1.4									
I1.5									

Ejemplo de numeración de E/S para una CPU 224

## 5.2 Entradas y salidas rápidas

Las CPUs S7-200 disponen de entradas y salidas para controlar los eventos rápidos.

### Contadores rápidos

Los contadores rápidos cuentan eventos que se ejecutan más deprisa de lo que pueden controlarlos las CPUs

S7-200. Según el modelo de CPU, se asiste un contador rápido en el software y hasta dos contadores rápidos en el hardware:

- HSC0 es un contador (software) que permite contar adelante y atrás, asistiendo una sola entrada de reloj. El programa controla el sentido de contaje (adelante o atrás) mediante el bit del control del sentido. La frecuencia máxima de este contador es de 2 KHz.
- HSC1 y HSC2 son contadores universales (hardware) que se pueden configurar según uno de doce modos de operación distintos. La frecuencia máxima de contaje de HSC1 y HSC2 depende de la CPU.

Cada contador dispone de entradas que asisten funciones tales como relojes, control del sentido, puesta a 0 y arranque. Los contadores A/B permiten elegir una velocidad simple o cuádruple para el contaje. HSC1 y HSC2 trabajan de forma completamente independiente y sin afectar a otras funciones rápidas. Ambos contadores funcionan a velocidades máximas sin interferirse mutuamente.

### Salidas de impulsos rápidos

Las CPUs S7-200 asisten salidas de impulsos rápidos. En dichas CPUs, Q0.0 y Q0.1 pueden generar trenes de impulsos rápidos (PTO) o controlar la modulación del ancho de impulsos (PWM).

La función PTO ofrece una salida en cuadratura (con un ancho de impulsos de 50%) para un número determinado de impulsos y un tiempo de ciclo determinado. El número de impulsos puede estar comprendido entre 1 y 4.294.967.295. El tiempo de ciclo se puede indicar en microsegundos o milisegundos, bien sea de 250 ms a 65.535 ms, o bien de 2 ms a 65.535 ms.



Un número impar de microsegundos o milisegundos (por ejemplo, 75 ms) causa una distorsión del factor de trabajo relativo.

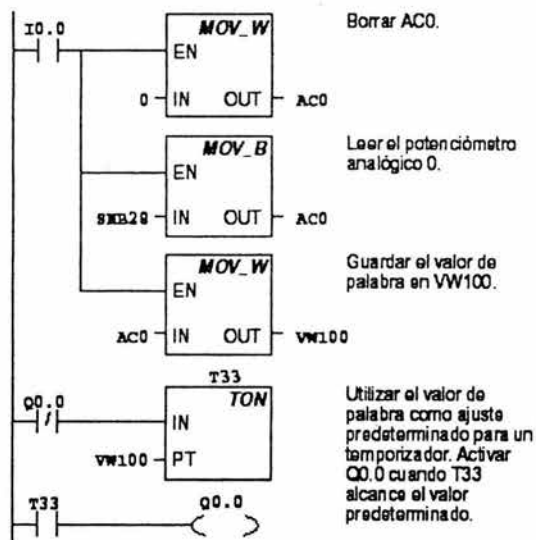
La función PWM ofrece un tiempo de ciclo fijo con una salida de ancho de impulsos variable. El tiempo de ciclo y el ancho de impulsos pueden indicarse en incrementos de microsegundos o milisegundos. El tiempo de ciclo puede estar comprendido entre 250 ms y 65.535 ms, o bien entre 2 ms y 65.535 ms. El tiempo del ancho de impulsos puede estar comprendido entre 0 ms y 65.535 ms, o bien entre 0 ms y 65.535 ms. Si el ancho de impulsos y el tiempo de ciclo son iguales, entonces el factor de trabajo relativo (relación impulso-pausa) será de 100% y la salida se activará continuamente. Si el ancho de impulsos es cero, el factor de trabajo relativo (relación impulso-pausa) será de 0% y se desactivará la salida.

### 5.3 Potenciómetros analógicos

Las CPUs S7-200 incorporan uno o dos potenciómetros analógicos (dependiendo del tipo de CPU), ubicados debajo de la tapa de protección del módulo, dichos potenciómetros permiten incrementar o decrementar valores almacenados en los bytes de marcas especiales SMB28 y SMB29. El programa puede utilizar estos valores de sólo lectura para diversas funciones, por ejemplo, para actualizar el valor actual de un temporizador o de un contador, para introducir o modificar los valores predeterminados, o bien, para ajustar límites.

SMB28 almacena el valor digital que representa la posición del potenciómetro analógico 0. SMB29 almacena el valor digital que representa la posición del potenciómetro analógico 1. Los potenciómetros analógicos tienen un margen nominal comprendido entre 0 y 255, y un margen garantizado de 10 a 200.

Utilice un destornillador pequeño para ajustar los potenciómetros. Gire el potenciómetro hacia la derecha para incrementar el valor, o bien hacia la izquierda para decrementarlo. La figura 5-1 muestra un programa de ejemplo donde se utiliza el potenciómetro analógico.



Programa en KOP

```

LD I0.0
MOVW 0, AC0
MOVE SMB20, AC0
MOVW AC0, VW100

LDN Q0.0
TON T33, VW100

LD T33
- Q0.0
  
```

Programa en AWL

Figura 5-1 Ejemplo de un programa utilizando el potenciómetro analógico

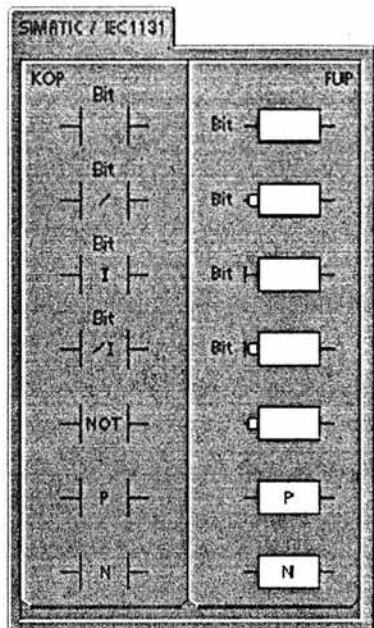
## **CAPÍTULO 6**

### **Instrucciones de operación**

## 6.1 Operaciones lógicas con bits

### Contactos

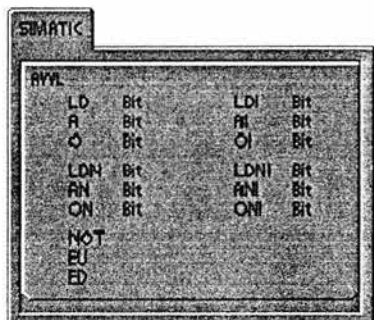
#### Contactos estándar



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.

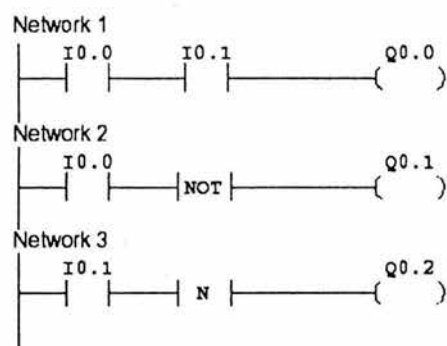
Tabla con operandos válidos para las operaciones lógicas con bits de entrada

Entradas/ salidas	Tipo de datos	Operandos
Bit	BOOL	I, Q, V, SM, S, T, C, L, circulación de corriente
Bit (directo)	BOOL	I

A continuación se muestran algunos ejemplos de operaciones con contactos y podemos observar que para que las bobinas de salida se activen, es necesario que los contactos de entrada estén activados.

ESTE LIBRO NO SALE  
DE LA BIBLIOTECA

## Ejemplos de operaciones lógicas con contactos



Operaciones lógicas de contactos en KOP

```

NETWORK
LD      I0.0
A       I0.1
=       Q0.0

```

```

NETWORK
LD      I0.0
NOT
=       Q0.1

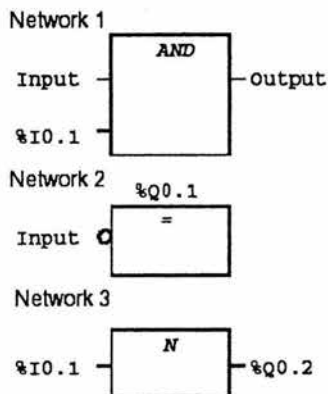
```

```

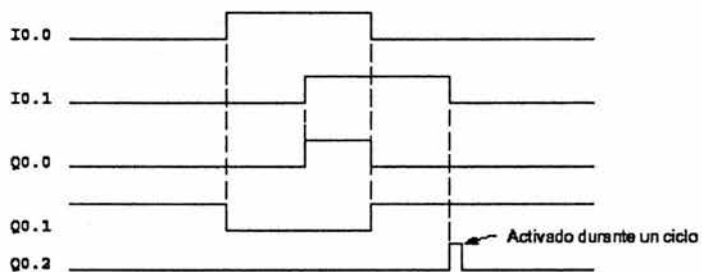
NETWORK
LD      I0.1
ED
=       Q0.2

```

Operaciones lógicas de contactos en AWL

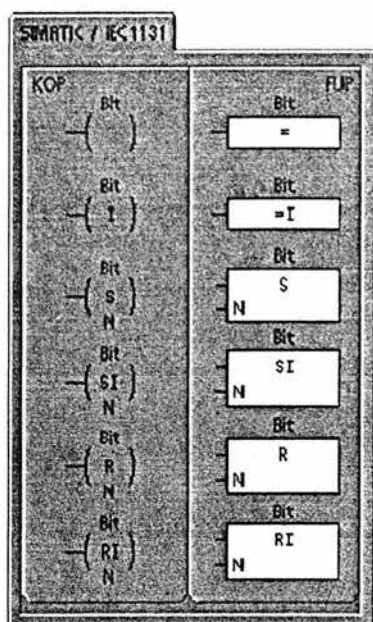


Operación lógicas de contactos en FUP



Cronograma

## Bobinas



## Salida

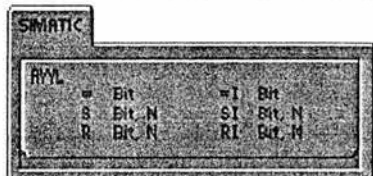
La operación Asignar (=) escribe el nuevo valor del bit de salida en la imagen del proceso. Cuando se ejecuta la operación Asignar, el S7-200 activa o desactiva el bit de salida en la imagen del proceso. En KOP y FUP, el bit indicado se ajusta de forma equivalente a la circulación de la corriente. En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



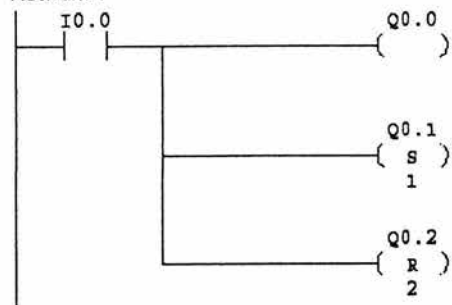
**Falta página**

**N° 82**



## Ejemplos operaciones lógicas con salidas en KOP y AWL y FUP.

Network 1

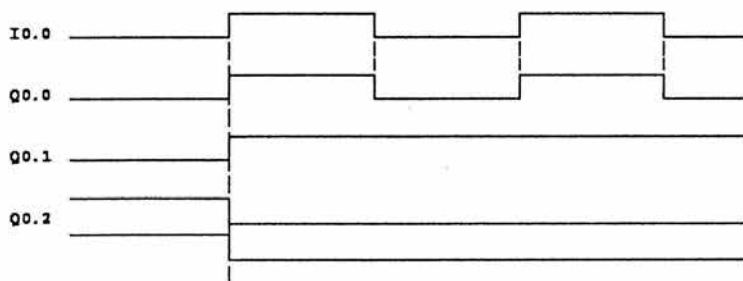


NETWORK

```
LD I0.0
= Q0.0
S Q0.1, 1
R Q0.2, 2
```

Operaciones lógicas con salidas en KOP

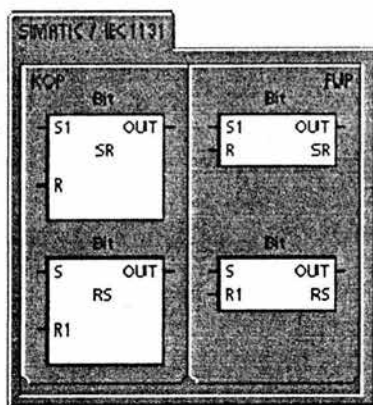
Operaciones lógicas con salidas en AWL



Cronograma

## 6.2 Operaciones lógicas SR y RS

### 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”.

Tabla de operandos válidos para las operaciones posicionar dominante biestable y rearmar dominante biestable

Entradas/ salidas	Tipo de datos	Operandos
S1, R	BOOL	I, Q, V, SM, S, T, C, L, circulación de corriente
S, R1, OUT	BOOL	I, Q, V, SM, S, T, C, L, circulación de corriente
Bit	BOOL	I, Q, V, M, S

A continuación se muestra un programa de ejemplo con las operaciones posicionar y rearmar dominante biestable.

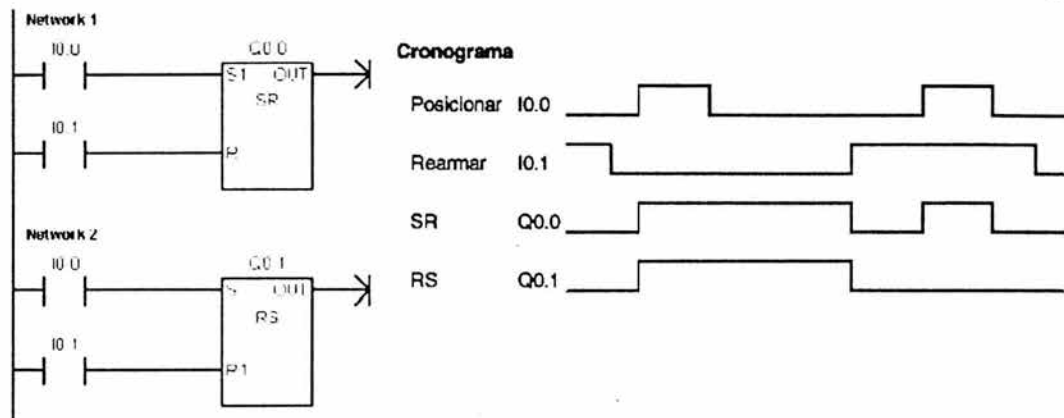


Diagrama en KOP

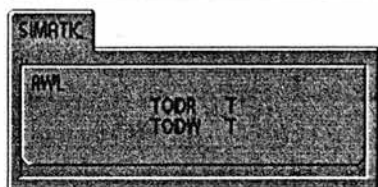
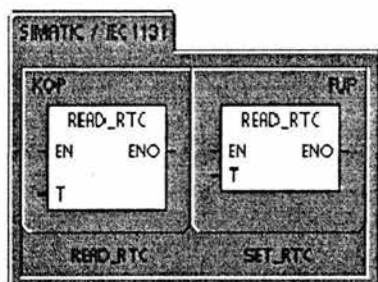
Cronograma

Tabla de verdad del ejemplo 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

## 6.3 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 6-3 muestra el formato del búfer de tiempo (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

Tabla de operandos válidos para las operaciones de reloj

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

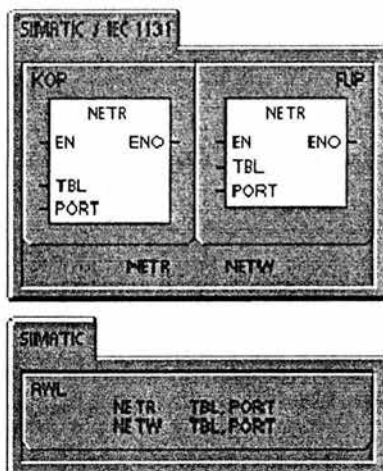
T	T+1	T+2	T+3	T+4	T+5	T+6	T+7
Año: 00 a 99	Mes: 01 a 12	Día: 01 a 31	Horas: 00 a 23	Minutos: 00 a 59	Segundos: 00 a 59	0	Día de la semana: 1 a 7*

1= Domingo, 7= Sábado, 0 desactiva el día de la semana

Tabla del búfer de tiempo de 8 bytes (T)

## 6.4 Operaciones de comunicaciones

### Leer de red y escribir en 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).

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.

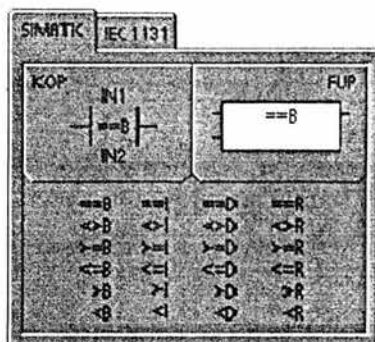
El programa puede contener un número cualquiera de operaciones NETR y NETW, pero sólo ocho de ellas (en total) pueden estar activadas simultáneamente. Por ejemplo, en un sistema de automatización S7-200 pueden estar activadas cuatro operaciones NETR y cuatro NETW, o bien 2 operaciones NETR y 6 NETW.

Tabla de operandos válidos para las operaciones leer de la red y escribir en la red.

Entradas/ salidas	Tipo de datos	Operandos
TBL	BYTE	VB, MB, *VD, *LD, *AC
PORT	BYTE	Constante para las CPUs 221, 222 y 224: 0  para las CPUs 226 y 226XM: 0 ó 1

## 6.5 Operaciones de comparación

### Comparar valores numéricos



Las operaciones de comparación se utilizan para comparar dos valores:

$IN1 = IN2$   $IN1 \geq IN2$   $IN1 \leq IN2$

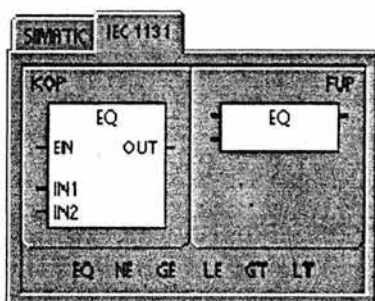
$IN1 > IN2$   $IN1 < IN2$   $IN1 \diamond IN2$

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.

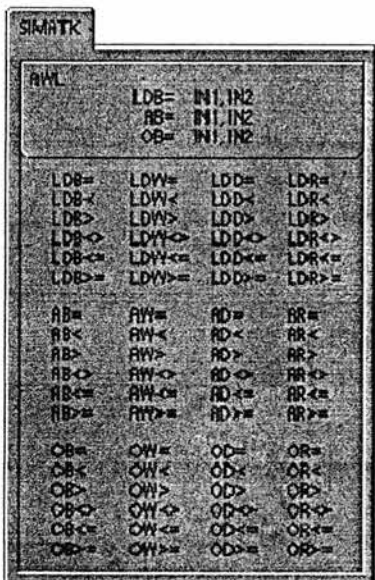


Tabla de operandos válidos para las operaciones de comparación

Entradas/ salidas	Tipo de datos	Operandos
IN1, IN2	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante
	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, 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
Salida (o OUT)	BOOL	I, Q, V, M, SM, S, T, C, L, circulación de corriente

A continuación se muestran ejemplos de operaciones de comparación con en KOP y AWL y FUP. Si la comparación es verdadera, se activa la bobina de salida.



NETWORK  
LDW>= VW4, VW8  
= Q0.3

Operación lógica de comparación en KOP

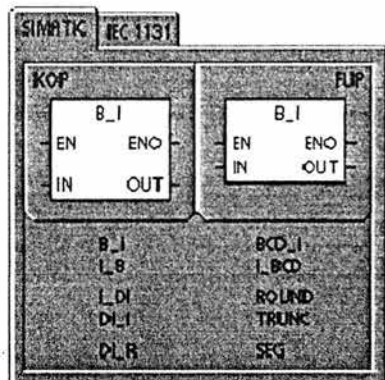
Operación lógica de comparación en AWL



Cronograma

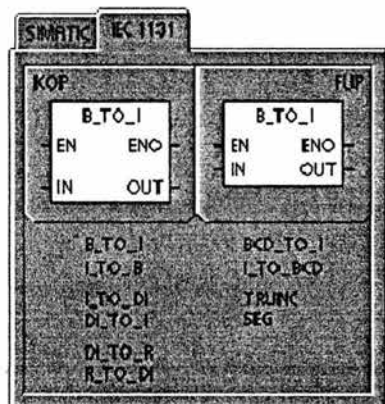
## 6.6 Operaciones de conversión

### Operaciones de conversión normalizadas



#### Conversiones numéricas

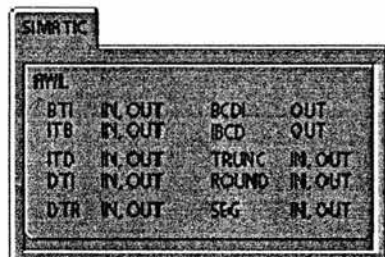
Las operaciones Convertir byte en entero (BTI), Convertir entero en byte (ITB), Convertir entero en entero doble (ITD), Convertir entero doble en entero (DTI), Convertir entero doble en real (DTR), Convertir BCD en entero (BCDI) y Convertir entero en BCD (IBCD) convierten un valor de entrada IN en el 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 de operandos válidos para las operaciones de conversión normalizadas

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

### Convertir BCD a entero y de entero a BCD

La operación Convertir BCD en entero (BCDI) convierte el 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.

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 de IN está comprendido entre 0 y 9999 enteros.

### Convertir entero doble en real

La operación Convertir entero doble en real (DTR) convierte 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.

### **Convertir entero doble en entero**

La operación Convertir entero doble en entero (DTI) convierte el valor de entero doble IN en un valor de entero y deposita el resultado en la variable indicada por OUT.

Si el valor a convertir es demasiado grande para ser representado en la salida, la marca de desbordamiento se activará y la salida no se verá afectada.

### **Convertir entero en entero doble**

La operación Convertir entero en entero doble (ITD) 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.

### **Convertir byte en entero**

La operación Convertir byte en entero (BTI) convierte el valor de byte IN en un valor de entero y deposita el resultado en la variable indicada por OUT.

El byte no tiene signo. Por tanto, no hay ampliación de signo.

### **Convertir entero en byte**

La operación Convertir entero en byte (ITB) convierte el valor de entero IN en un valor de byte y deposita el resultado en la variable indicada por OUT. Se convierten los valores comprendidos entre 0 y 255. Todos los demás valores producen un desbordamiento y la salida no se ve afectada.

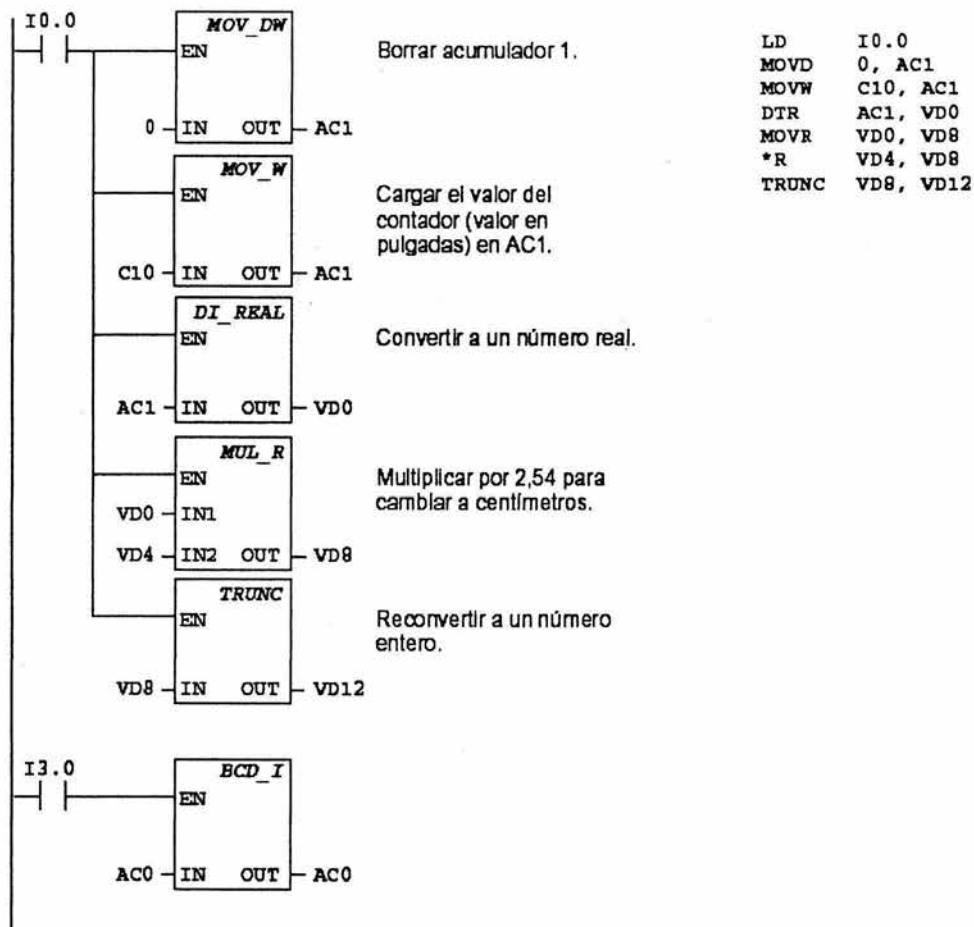
### **Redondear a entero doble y truncar**

La operación Redondear (ROUND) convierte un número real (IN) en un valor de entero doble y deposita el resultado en la variable indicada por OUT. Si la fracción es 0,5 o superior, el número se redondeará al próximo entero superior.

La operación Truncar (TRUNC) convierte un número real (IN) en un entero doble y carga el resultado en la variable 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.

En la figura siguiente tenemos un ejemplo aplicando las operaciones de conversión normalizadas, en donde se realiza una conversión de pulgadas a centímetros, el programa se encuentra en KOP y AWL.



Operación de conversión en KOP

Operación de conversión en AWL

Convertir entero de palabra doble a real y truncar

BCD a entero

C10 

Contar = 101 pulgadas

VD0 AC0 VD4 

Factor 2.54 (cambiar de pulgadas a centímetros)

BCDI

VD8 

256,54 centímetros es un número real.

AC0 V12 

256 centímetros es un número entero.

Aplicación en los acumuladores

### 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.

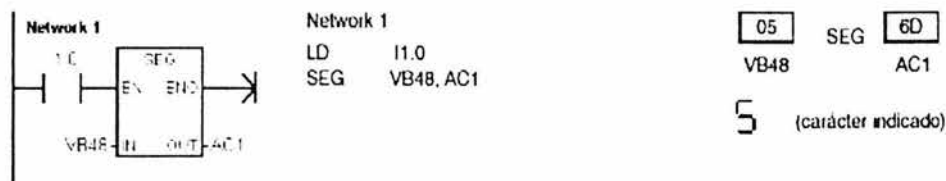
Los segmentos iluminados representan el carácter depositado en el dígito menos significativo del byte de entrada. La figura 6-6 muestra la codificación del indicador de siete segmentos utilizado por la operación Segmento.

(IN) LSD	Indicador	(OUT) -gfe dcba
0		0 0 1 1 1 1 1 1
1		0 0 0 0 0 1 1 0
2		0 1 0 1 1 0 1 1
3		0 1 0 0 1 1 1 1
4		0 1 1 0 0 1 1 0
5		0 1 1 0 1 1 0 1
6		0 1 1 1 1 1 0 1
7		0 0 0 0 0 1 1 1

(IN) LSD	Indicador	(OUT) -gfe dcba
8		0 1 1 1 1 1 1 1
9		0 1 1 0 0 1 1 1
A		0 1 1 1 0 1 1 1
B		0 1 1 1 1 1 0 0
C		0 0 1 1 1 0 0 1
D		0 1 0 1 1 1 1 0
E		0 1 1 1 1 0 0 1
F		0 1 1 1 0 0 0 1

Figura 6-6 Codificación de un indicador de siete segmentos

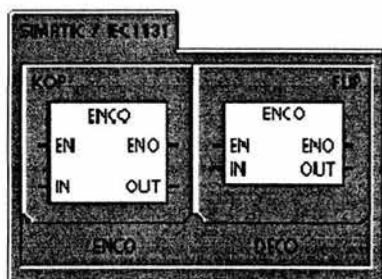
Ejemplo:



Ejemplo de operación segmento en KOP y AWL

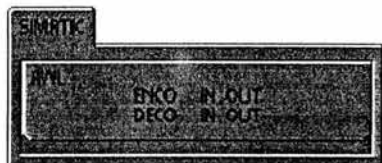
## 6.7 Operaciones de codificar y decodificar

### 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



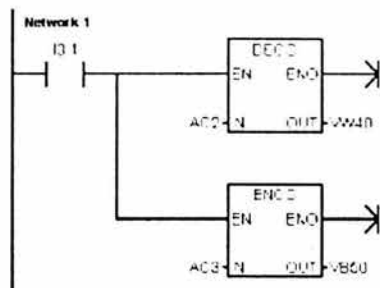
La operación Decodificar (DECO) activa el bit de la palabra de salida OUT. Este bit corresponde al número de bit representado por el medio byte menos significativo (4 bits) del byte de entrada IN. Todos los demás bits de la palabra de salida se ponen a 0.

Tabla de operandos válidos para las operaciones codificar y decodificar

Entradas/ salidas	Tipo de datos	Operandos
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante
	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante

OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW *VD, *LD, *AC

## Ejemplo de las operación codificar y decodificar



Network 1

```

//AC2 contiene bits de error.
//1: La operación DECO activa el bit en VW40
//   que corresponde a este código de error.
//2: La operación ENCO convierte
//   el bit menos significativo en un código de error
//   que se almacena en VB50.

```

```

LD    I3.1
DECO  AC2, VW40
ENCO  AC3, VB50

```

AC2      3

DECO      15      3      0

VW40      0000 0000 0000 1000

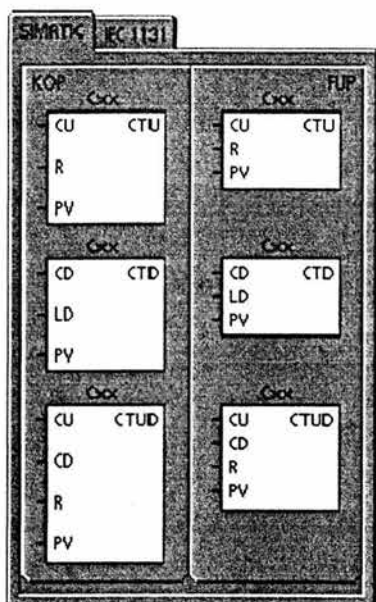
AC3      15      9      0

1000 0010 0000 0000

ENCO

VB50      9

## 6.8 Operaciones de contaje



### 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).

### 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.

### 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).

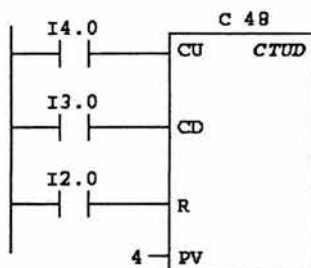
Si el valor actual (Cxx) es mayor o igual al valor de preselección PV, se activa el bit de contaje Cxx. En caso contrario, se desactiva el bit. El contador se inicializa cuando se activa la entrada de desactivación (R) o al ejecutarse la operación Poner a 0. El contador adelante/atrás se detiene al alcanzar el valor de preselección (PV).

Tabla de operandos válidos para las operaciones de contaje

Entradas/ salidas	Tipo de datos	Operandos
Cxx	WORD	Constante (C0 a C255)
CU, CD, LD, R	BOOL	I, Q, V, M, MS, S, T, C, L, circulación de corriente
PV	INT	IW, QW, VW, MW, SMW, SW, LW, T, C, AC, AIW, *VD, *LD, *AC, constante

Ejemplo de una operación de contaje en KOP y AWL

El contador utilizado es el C48, el cual se activa atrás o adelante, dependiendo del contacto de entrada activado.

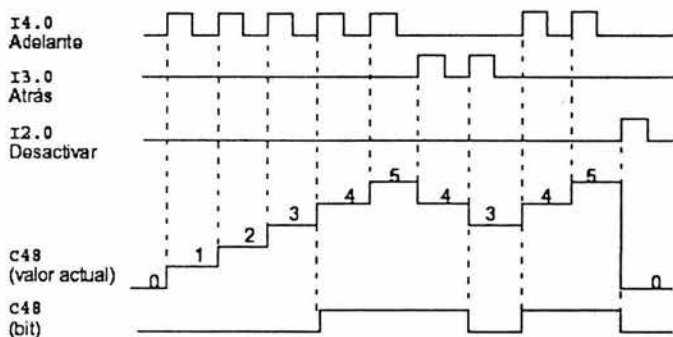


```
LD I4.0 //Contar adelante
LD I3.0 //Contar atrás
LD I2.0 //Poner a 0
CTUD C48, 4
```

Ejemplo de operación de contaje en KOP

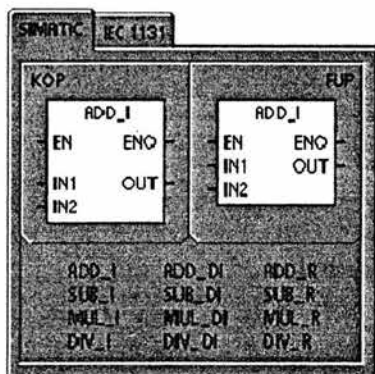
Ejemplo de operación de contaje en AWL





Cronograma

## 6.9 Operaciones aritméticas



### Operaciones de sumar, restar, multiplicar y dividir

#### Sumar

$$IN1 + IN2 = OUT$$

$$IN1 + OUT = OUT$$

#### Restar

$$IN1 - IN2 = OUT$$

$$OUT - IN1 = OUT$$

KOP y FUP

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.

#### Multiplicar

$$IN1 * IN2 =$$

$$IN1 + OUT =$$

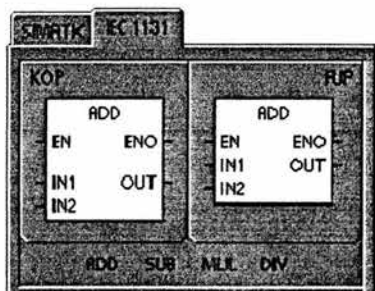
#### Dividir

$$OUT IN1 / IN2 = OUT$$

$$OUT OUT / IN1 = OUT$$

KOP y FUP

AWL

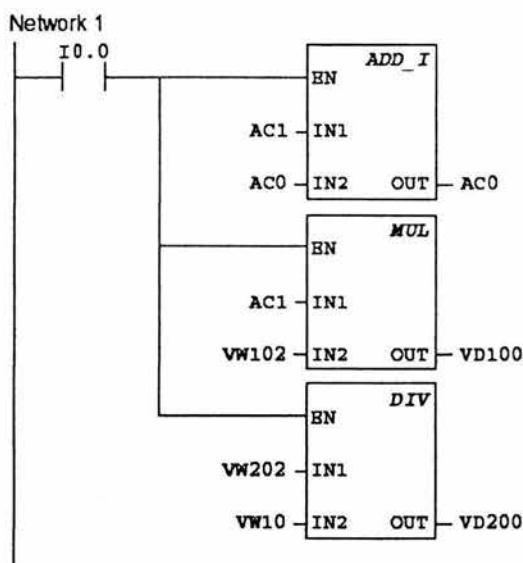


Las operaciones Multiplicar enteros (\*I) y Dividir enteros (/I) multiplican o dividen dos enteros de 16 bits, respectivamente, arrojando un resultado de 16 bits. (En la división no se conserva un resto.) Las operaciones multiplicar enteros dobles (\*D) y Dividir enteros dobles (/D) multiplican o dividen 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.

Tabla de operandos válidos para las operaciones de sumar, restar, multiplicar y dividir

Entradas/ salidas	Tipo de datos	Operandos
INI, IN2	INT  DINT  REAL	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *AC, *LD, constante  ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, constante  ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, constante
OUT	INT  DINT, REAL	IW, QW, VW, MW, SMW, SW, LW, T, C, AC, *VD, *AC, *LD  ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Ejemplo de operaciones aritméticas de sumar, multiplicar y dividir.



```

NETWORK
LD      I0.0
+I      AC1, AC0
MUL     AC1, VD100
DIV     VW10, VD200

```

Operaciones aritméticas en KOP

Operaciones aritméticas en AWL

Sumar

AC1	4000
	más
AC0	6000
	igual a
AC0	10000

Multiplicar

AC1	4000
	multiplicado por
VD100	200
	igual a
VD100	800000

Dividir

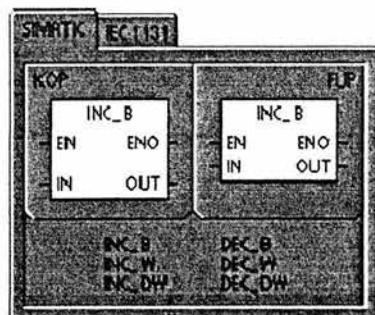
VD200	4000	
	dividido por	
VW10	41	
	igual a	
VD200	23	97
	resto	cociente
VW200	VW202	

Nota: VD100 contiene VW100 y VW102.  
VD200 contiene VW200 y VW202.

Aplicación en los acumuladores

## 6.10 Incrementar y decrementar

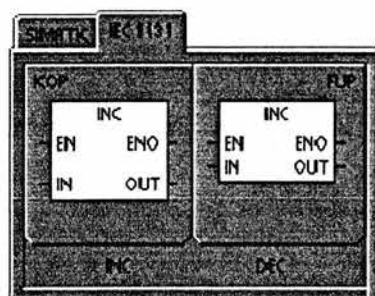
### Incrementar



$IN + 1 = OUT$   
 $OUT + 1 = OUT$

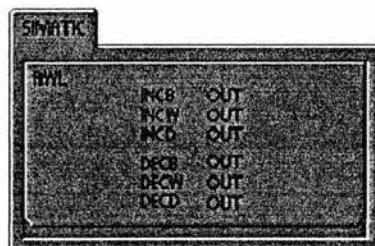
KOP y FUP  
 AWL

### Decrementar



$IN - 1 = OUT$   
 $OUT - 1 = OUT$

KOP y FUP  
 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.

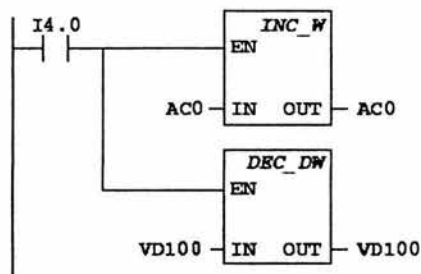
Las operaciones incrementar palabra doble (INCD) y decrementar palabra doble (DECD) llevan signo.

Al programar en KOP se puede reducir el espacio de memoria necesario indicando que IN sea igual a OUT.

Tabla de operandos válidos para las operaciones incrementar y decrementar

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

Ejemplo con las operaciones incrementar y decrementar valor



```
LD    I4.0
INCW  AC0
DECD  VD100
```

Operaciones en KOP

Operaciones en AWL

Incrementar palabra

AC0	125
incremento	

AC0	126
-----	-----

Decrementar palabra

VD100	128000
decremento	

VD100	127999
-------	--------

Aplicación en los acumuladores

## 6.11 Operaciones de transferencia

### Transferir byte, 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.

Los tipos de los datos de entrada y salida pueden ser diferentes, pero su tamaño debe ser igual.

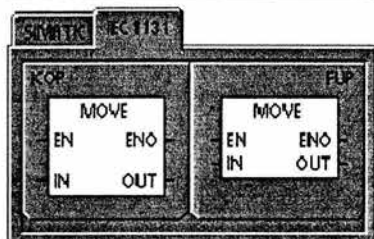
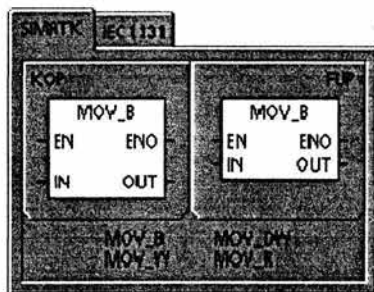
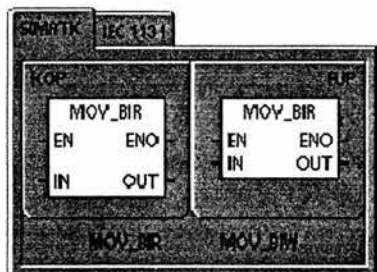


Tabla de operandos válidos para las operaciones de transferencia

Entradas/ salidas	Tipo de datos	Operandos
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante
	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, AC, HC, &IB, &QB, &VB, &MB, &SB, &T, &C, *VD, *LD, *AC, constante
	REAL	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, constante
OUT	BYTE	OB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	WORD, INT	OW, QW, VW, MW, SMW, SW, T, C, LW, AC, AQW, *VD, *LD, *AC
	DWORD, DINT, REAL	OD, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

### Transferir byte 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.

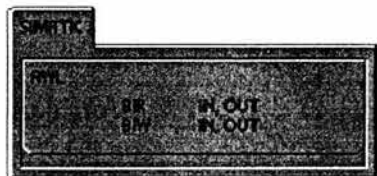


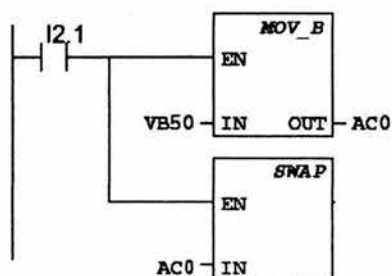
Tabla de operandos válidos para la operación lectura y transferencia directa de bytes

Entradas/ salidas	Tipo de datos	Operandos
IN	BYTE	IB, *VD, *LD, *AC
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Tabla de operandos válidos para la operación escritura y transferencia directa de bytes

Entradas/ salidas	Tipo de datos	Operandos
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante
OUT	BYTE	QB, *VD, *LD, *AC

Ejemplo con las operaciones de transferir byte.

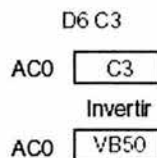


```
LD    I2.1
MOV   VB50, AC0
SWAP  AC0
```

Operación transferir en KOP



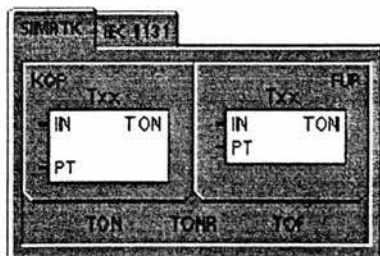
Operación transferir operación en AWL



Aplicación en los acumuladores



## 6.12 Operaciones de temporización



### 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 del temporizador (Txx) determina la resolución del mismo.

### 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 de operandos válidos para las operaciones de temporización

Entradas/ salidas	Tipo de datos	Operandos
Txx	BYTE	Constante (T0 a T255)
IN	BOOL	I, Q, V, M, SM, S, T, C, L, circulación de corriente
PT	INT	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante

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.

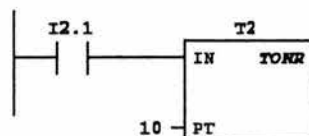
Los temporizadores cuentan intervalos de tiempo. La resolución (o base de tiempo) del temporizador determina el lapso de tiempo de cada intervalo. Por ejemplo, un temporizador TON con una resolución de 10 ms cuenta el número de intervalos de 10 ms que han transcurrido desde que se habilitó el TON. Un 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 6-1, el número del temporizador determina su resolución.

Tabla 6-1 Temporizadores y sus resoluciones

TONR (con memoria)	1 ms	32,767 s	T0, T64
	10 ms	327,67 s	T1 a T4, T65 a T68
	100 ms	3276,7 s	T5 a T31, T69 a T95
TON, TOF (sin memoria)	1 ms	32,767 s	T32, T96
	10 ms	327,67 s	T33 a T36, T97 a T100
	100 ms	3276,7 s	T37 a T63, T101 a T255

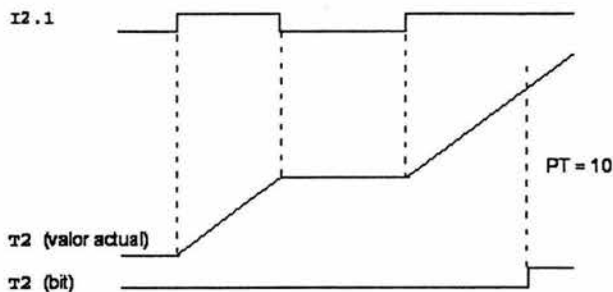
En la figura siguiente se muestra un ejemplo de temporizador con retardo a la conexión memorizado.



```
LD I2.1
TONR T2, 10
```

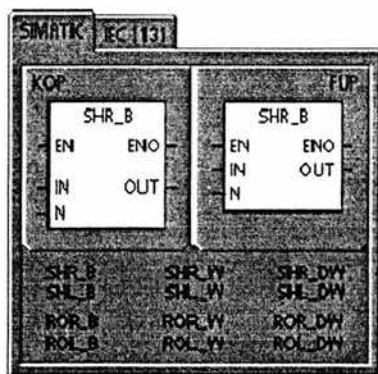
Diagrama de temporizador en KOP

Diagrama de temporizador en AWL



Cronograma

## 6.13 Operaciones de desplazamiento y rotación

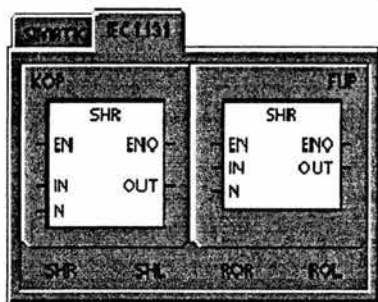


### Desplazar a la derecha y desplazar a la izquierda

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.

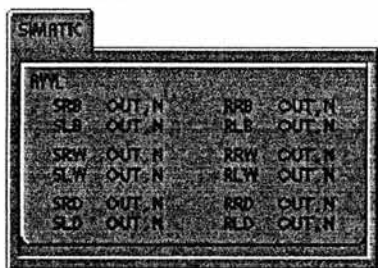
Las operaciones de desplazamiento de bytes no llevan signo. En el caso de las operaciones con palabras y con palabras dobles, el bit de signo se desplaza cuando se utilizan tipos de datos con signo.



### 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).

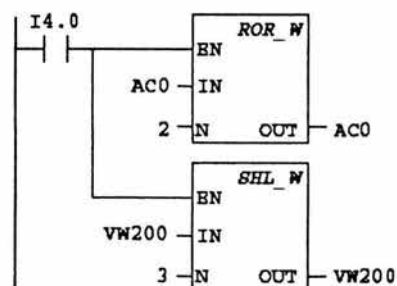
Si el valor de desplazamiento no es un entero múltiplo de 8 (en las operaciones con bytes), de 16 (en las operaciones con palabras) o de 32 (en las operaciones con palabras dobles), el último bit rotado se copiará en la marca de desbordamiento (SM1.1). La marca cero (SM1.0) se activará si el valor a 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.

Tabla de operandos válidos para las operaciones de desplazamiento y rotación

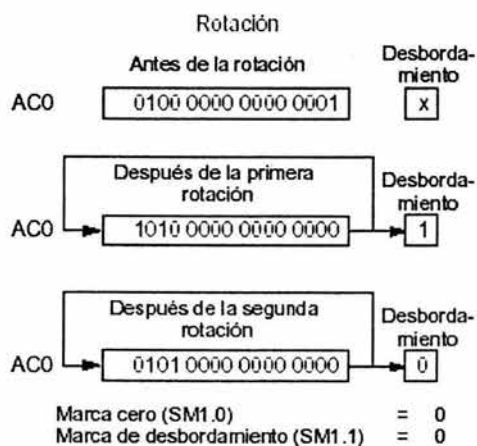
<b>Entradas/ salidas</b>	<b>Tipo de datos</b>	<b>Operandos</b>
IN	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante
	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, AIW, *VD, *LD, *AC, constante
	DWORD	ID, QD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, constante
OUT	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	WORD	IW, QW, VW, MW, SMW, SW, T, C, LW, AC, *VD, *LD, *AC
	DWORD	ID, QD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC
N	BYTE	IB, QB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, constante

En la figura siguiente se muestra un ejemplo de operación de rotación y desplazamiento.

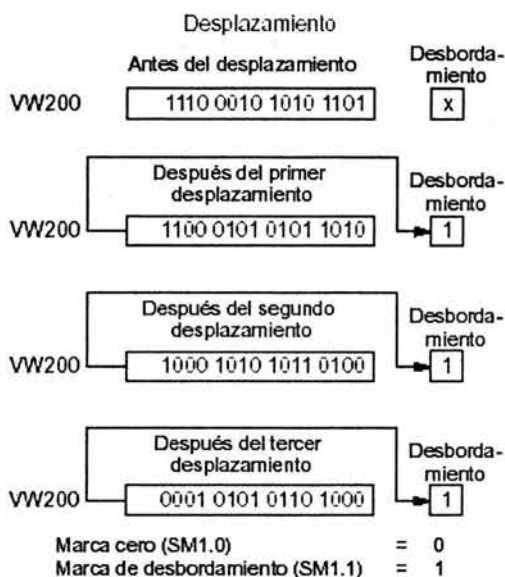


```
LD    I4.0
RRW   AC0, 2
SLW   VW200, 3
```

Operación en KOP



Operación en AWL



Aplicación en acumuladores

## **CAPÍTULO 7**

### **Comunicación en redes con CPUs S7-200**

## 7.1 Protocolos de comunicación de las CPUs S7-200

Las CPUs S7-200 asisten diversos métodos de comunicación. Dependiendo de la CPU S7-200 utilizada, la red puede asistir uno o varios de los siguientes protocolos de comunicación:

- Interface punto a punto (PPI)
- Interface multipunto (MPI)
- PROFIBUS-DP

Estos protocolos se basan en la intercomunicación de sistemas abiertos (OSI) de la arquitectura de siete capas. Los protocolos PPI, MPI y PROFIBUS-DP se implementan en una red “token ring” (red de anillo con testigo) conforme al estándar Process Field Bus (PROFIBUS) que se describe en la norma europea EN 50170.

Los protocolos prevén dos tipos de dispositivos de red: los maestros y los esclavos. Los maestros pueden enviar una petición a otros dispositivos. En cambio, los esclavos sólo pueden responder a las peticiones de los maestros, sin poder lanzar nunca una petición por su propia cuenta.

Los protocolos asisten 127 direcciones (0 a 126) en una red. Una red puede comprender 32 maestros como máximo. Todos los dispositivos que formen parte de una red deberán tener direcciones unívocas para poder comunicarse entre sí. El ajuste estándar para las unidades de programación SIMATIC y para los PCs con STEP 7-Micro/WIN es la dirección “0”. El visualizador de textos TD 200 y los paneles de operador OP3 y OP7 tienen la dirección predeterminada “1”. La dirección estándar de los sistemas de automatización es “2”. La dirección predeterminada del interface DP de la CPU 215 es “126”.

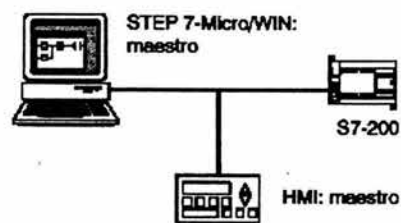


Figura 7-5 Red PPI

### Protocolo PPI

PPI es un protocolo maestro/esclavo. Los maestros (otras CPUs, unidades de programación SIMATIC o visualizadores de textos TD 200) envían peticiones a los esclavos y éstos últimos responden. Los esclavos no inician mensajes, sino que esperan a que un maestro les envíe una petición o solicite una respuesta.

Todas las CPUs S7-200 actúan de estaciones esclavas en la red, ver figura 7-5.

Estando en modo RUN, algunas CPUs S7-200 pueden actuar de estaciones maestras en la red si se encuentra habilitado el modo maestro PPI en el programa de usuario. Una vez habilitado el modo maestro PPI, se podrán enviar mensajes a otras CPUs, usando las operaciones Leer de la red (NETR) y Escribir en la red (NETW). Mientras actúa de estación maestra PPI, la CPU S7-200 sigue respondiendo en calidad de esclava a las peticiones de otros maestros.

El protocolo PPI no limita la cantidad de maestros que pueden comunicarse con una CPU cualquiera que actúe de esclava, pero la red no puede comprender más de 32 maestros.

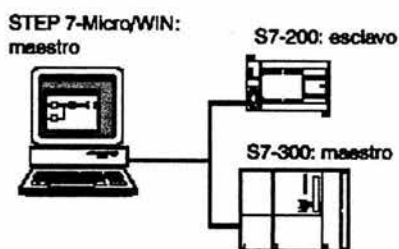


Figura 7-6 Red MPI

### Protocolo MPI

MPI puede ser un protocolo maestro/maestro, o bien maestro/esclavo. El funcionamiento de dicho protocolo depende de los equipos utilizados. Si el dispositivo de destino es una CPU S7-300, se establece un enlace maestro/maestro, puesto que todas las CPUs S7-300 son estaciones maestras en la red. Si es una CPU S7-200, se establece un enlace

maestro/esclavo, ya que las CPUs S7-200 son unidades esclavas, ver figura 7-6.

El protocolo MPI crea siempre un enlace entre los dos dispositivos comunicados entre sí. Un enlace es una conexión privada entre los dos dispositivos. Ningún otro maestro puede interferir en un enlace establecido entre dos dispositivos. Un maestro puede establecer un enlace para utilizarlo durante un tiempo breve o indefinido.

Puesto que los enlaces son conexiones privadas entre los dispositivos y utilizan recursos en la CPU, cada CPU puede asistir sólo una cantidad limitada de enlaces. Cada CPU reserva algunos de sus enlaces para las unidades de programación SIMATIC y para los paneles de operador. El enlace reservado para una unidad de programación (PG) SIMATIC o para un PC con STEP 7-Micro/WIN garantiza que el usuario pueda conectar siempre por lo menos una PG SIMATIC a la CPU para poder acceder a ésta última. Algunas CPUs reservan también un enlace para un panel de operador. Dichos enlaces reservados no pueden ser utilizados por otros maestros (por ejemplo. CPUs).



## Protocolo PROFIBUS - DP

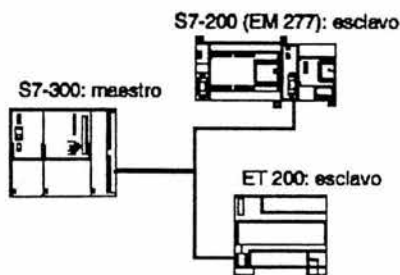


Figura 7-7 Red PROFIBUS

El protocolo PROFIBUS-DP se ha diseñado para la comunicación rápida con unidades periféricas descentralizadas. Hay numerosos dispositivos PROFIBUS ofrecidos por diversos fabricantes, dichos dispositivos abarcan desde módulos sencillos de entradas o de salidas hasta controladores de motores y sistemas de automatización.

Por lo general, las redes PROFIBUS-DP incorporan un maestro y varios esclavos. La configuración del maestro le permite reconocer cuáles tipos de esclavos están conectados, así como sus respectivas direcciones. El maestro inicializa la red y verifica si los esclavos coinciden con la configuración, continuamente, el maestro escribe los datos de salida en los esclavos y lee de allí los datos de entrada. Una vez que un maestro DP haya configurado correctamente a un esclavo, éste último le pertenecerá, si hay otro maestro en la red, tendrá apenas un acceso muy limitado a los esclavos del primer maestro, ver figura 7-7.

## 7.2 Componentes para la comunicación en redes

Un sistema de automatización S7-200 se puede conectar a través del interface de comunicación a un bus de red. A continuación se describen dicho interface, los conectores para el bus, el cable de conexión y los repetidores utilizados para ampliar la red.

### Interface de comunicación

Los interfaces de comunicación de las CPUs S7-200 son compatibles con el estándar RS-485 mediante un conector D subminiatura de 9 pines conforme al estándar PROFIBUS definido en la norma europea EN 50170. La figura 7-8 muestra el conector que ofrece el enlace físico para el interface de comunicación y la tabla 7-9 describe las señales.

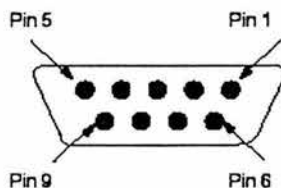


Figura 7-8 Pines del interface de comunicación de la CPU S7-200

Tabla 7-9 Asignación de pines del interface de comunicación de la CPU S7-200

Pin	Denominación PROFIBUS	Interfaces 0 y 1	Interface DP
1	Blindaje	Hilo lògico	Hilo lògico
2	Hilo de retorno 24 Volts	Hilo lògico	Hilo lògico
3	Señal B RS-485	Señal B RS-485	Señal B RS-485
4	Request-to-send	Sin conexión	Request-to-send1
5	Hilo de retorno 5 Volts	Hilo lògico	Isolated + 5 Volts Return2
6	+ 5 Volts	+ 5 V, 100 $\Omega$ series limit	+ 5 V, con separación galvànica, 90 mA
7	+ 24 Volts	+ 24 Volts	+ 24 Volts
8	Señal A RS-485	Señal A RS-485	Señal A RS-485
9	No aplicable	Sin conexión	Sin conexión
Carcasa del enchufe	Blindaje	Hilo lògico ò Tierra	Tierra

### Conectores de bus

Siemens ofrece dos tipos de conectores de bus que permiten conectar fácilmente varios dispositivos a una red. Ambos conectores poseen dos juegos de tornillos para fijar los cables de entrada y salida. Asimismo, disponen de interruptores para polarizar y cerrar la red de forma selectiva. Uno de ellos ofrece sólo un enlace a la CPU, en tanto que el otro agrega un interface de programación ver figura 7-10.

El conector que provee un interface de programación permite añadir a la red una unidad de programación SIMATIC o un panel de operador, sin perturbar ningún enlace existente. Dicho conector transmite todas

las señales de la CPU a través del interface de programación, adecuándose para conectar dispositivos alimentados por la CPU (por ejemplo un TD 200 o un OP3). Los pines de alimentación del conector del puerto de comunicación se pasan por el interface de programación.

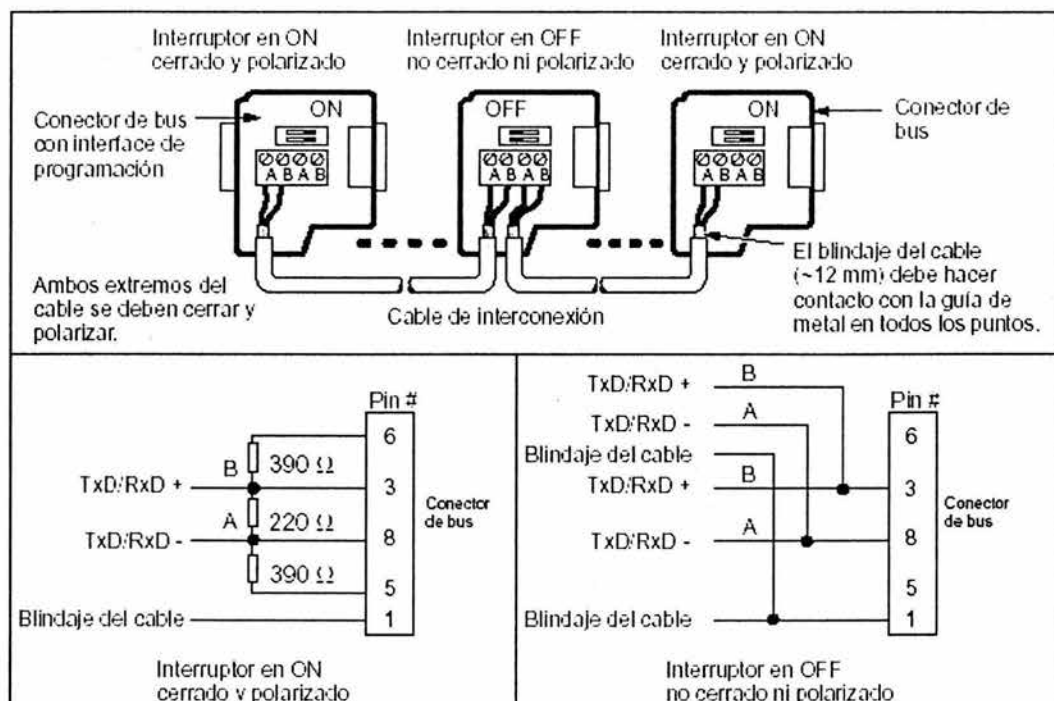


Figura 7-10 Polarizar y cerrar el cable de interconexión

### Cable para una red PROFIBUS

La tabla 7-11 muestra los datos técnicos generales de un cable para una red PROFIBUS.

Propiedades generales	Datos técnicos
Tipo	Apantallado, con par trenzado
Sección transversal del cable	24 AWG ò superior
Capacidad del cable	< 60 pF/m
Impedancia nominal	100 Ω a 120 Ω

La longitud máxima de un segmento de red PROFIBUS depende de la velocidad de transferencia y del tipo de cable utilizados. En la tabla 7-12 figuran las longitudes máximas de los segmentos para el cable indicado en la tabla 7-11.

Tabla 7-12 Longitud máxima del cable en un segmento de una red PROFIBUS

Velocidad de transferencia	Longitud máxima del cable en un segmento
9.6 kbit/s a 93.75 kbit/s	1,200 m
187.5 kbit/s	1,000 m
500 kbit/s	400 m
1.5 Mbit/s	200 m
3 Mbit/s a 12 Mbit/s	100 m

### Repetidores

Siemens ofrece repetidores para interconectar segmentos de redes PROFIBUS (ver figura 7-13). Utilizando repetidores es posible ampliar la longitud total de la red y/o agregar dispositivos a la misma. El protocolo PROFIBUS admite máximo 32 dispositivos en un segmento de red de hasta 1.200 m a una velocidad de transferencia de 9.600 bit/s. Cada repetidor permite agregar 32 dispositivos adicionales a la red y ampliar ésta última en 1.200 m a una velocidad de transferencia de 9.600 bit/s. En una red se pueden utilizar 9 repetidores como máximo. Cada repetidor permite polarizar y cerrar el segmento de red en cuestión.

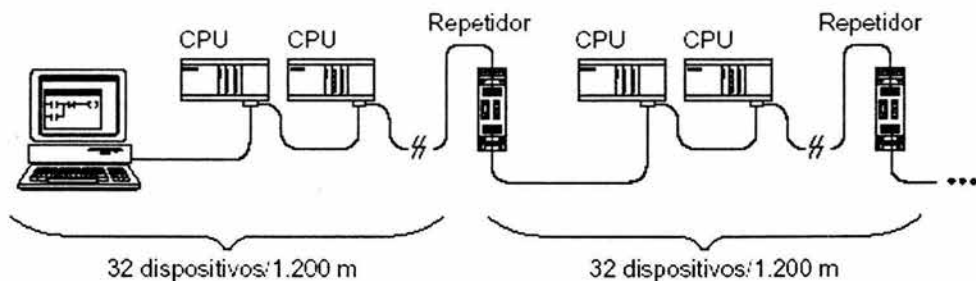


Figura 7-13 Red con repetidores

### 7.3 Utilizar un cable PC/PPI para la comunicación

#### Cable PC/PPI

Por lo general, los puertos de comunicación de un PC son compatibles con el estándar RS-232. Los interfaces de comunicación de la CPU S7-200 utilizan el estándar RS-485 para poder agregar varios dispositivos a una misma red. El cable PC/PPI permite conectar el puerto RS-232 de un PC al interface RS-485 de una CPU S7-200 (ver figura 7-14). Dicho cable se puede utilizar también para conectar el interface de comunicación de una CPU S7-200 a otros dispositivos compatibles con el estándar RS-232.

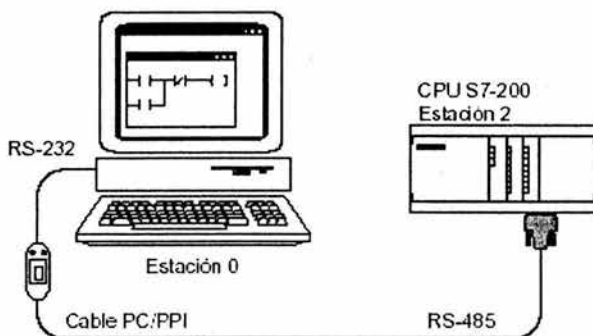
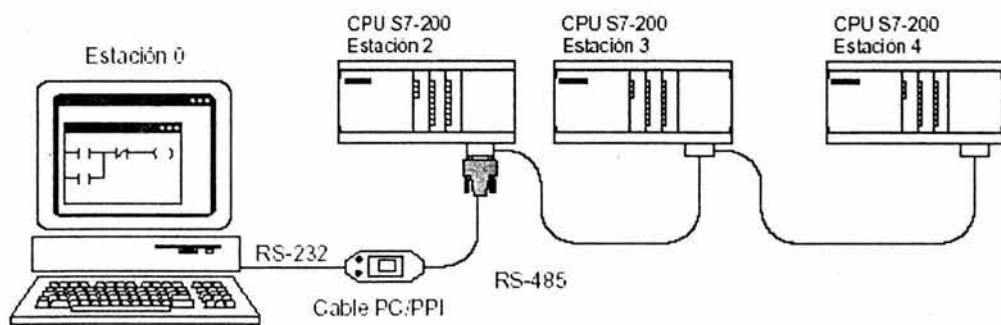


Figura 7-14 Utilizar un cable PC/PPI para la comunicación con una CPU S7-200

#### Utilizar STEP 7-Micro/WIN con un cable PC/PPI

STEP 7-Micro/WIN puede utilizar el cable PC/PPI para comunicarse con una o varias CPUs S7-200 (ver figura 7-15). Al utilizar STEP 7-Micro/WIN, verifique que la velocidad de transferencia del cable PC/PPI se ajuste a la exigida en la red. STEP 7-Micro/WIN asiste sólo velocidades de transferencia de 9.600 bit/s y 19.200 bit/s.



Polarizar y cerrar las estaciones 2 y 4. Estas estaciones se encuentran en los extremos de la red.

El conector utilizado en la estación 2 tiene un interface de programación. Los conectores de las demás estaciones no disponen de dicho interface.

Figura 7-15 Utilizar un cable PC/PPI para la comunicación con una CPU a la vez

El cable PC/PPI asiste velocidades de transferencia comprendidas entre 600 bit/s y 38.400 bit/s. Utilice los interruptores DIP dispuestos en la carcasa del cable PC/PPI para configurar la velocidad de transferencia correcta. La tabla 7-16 muestra las velocidades de transferencia y las correspondientes posiciones de los interruptores.

Tabla 7-16 Seleccionar la velocidad de transferencia del cable PC/PPI

Velocidad de transferencia	Interruptor DIP (1= arriba)
38400	0000
19200	0010
9600	0100
4800	0110
2400	1000
1200	1010
600	1100

El interface RS-232 del cable PC/PPI se considera un equipo de comunicación de datos (DCE o Data Communications Equipment). Las únicas señales presentes en dicho interface son: transmitir datos, recibir datos y tierra. La tabla 7-17 muestra los números de los pines y las funciones del interface RS-232 del cable PC/PPI. El cable PC/PPI no utiliza ni envía ninguna de las señales de control del RS-232, tales como Request to Send (RTS) y Clear to Send (CTS).

Tabla 7-17 Cable PC/PPI definición de pines del interface RS-232

<b>Nº de pin</b>	<b>Función</b>
2	Recibir datos (de DCE)
3	Transmitir datos (de DTE a DCE)
5	Tierra

El cable PC/PPI se encuentra en el modo de transmisión cuando los datos se envían del interface RS-232 al RS-485. En cambio, se encuentra en modo de recepción al estar en vacío, o bien cuando los datos se transmiten del interface RS-485 al RS-232. El cable cambia inmediatamente de modo de recepción a transmisión cuando detecta caracteres en el canal de transmisión del RS-232. El cable cambia nuevamente a modo de recepción cuando el canal de transmisión del RS-232 está en vacío durante el tiempo de inversión del cable. Dicho tiempo depende de la velocidad de transferencia seleccionada con los interruptores DIP del cable (ver tabla 7-18).

Tabla 7-18 Tiempo de inversión del cable PC/PPI cambio de transmisión a recepción

<b>Velocidad de transferencia</b>	<b>Tiempo de inversión (en milisegundos)</b>
38400	1
19200	1
9600	2
4800	4
2400	7
1200	14
600	28

## 7.4 Utilizar una tarjeta MPI o CP para la comunicación

Se ofrecen diversas tarjetas de interface que se pueden incorporar en un PC o en una PG (unidad de programación) SIMATIC. Las tarjetas permiten que el PC o la PG actúen de estaciones maestras en la red. Dichas tarjetas contienen componentes de hardware especiales para asistir al PC o a la PG en la gestión de la red multimaestro, soportando diferentes protocolos y varias velocidades de transferencia (ver tabla 7-19).

Tabla 7-19 Tarjetas de conexión a una red multimaestro

Nombre	Tipo	Sistemas operativos Asistidos	Comentarios
MPI	Short AT ISA o Integrado en CP	MS-DOS Windows 3.1x	Asiste el protocolo PPI, 9.600 bit/s y 19.200 bit/s
		Windows 95 Windows NT	Asiste los protocolos PPI, MPI y PROFIBUS-DP, 9.600 bit/s a 1,5 Mbit/s para PCs y PGs
CP 5411	Short AT ISA	Windows 95 Windows NT	Asiste los protocolos PPI, 1 MPI y PROFIBUS-DP, 9.600 bit/s a 12 Mbit/s para PCs y PGs
CP 5511	PCMCIA TIPO 11 Hardware "plug & play"	Windows 95 Windows NT	Asiste los protocolos PPI, 1 MPI y PROFIBUS-DP, 9.600 bit/s a 12 Mbit/s para PCs portátiles
CP 5611	Short PCI Hardware "plug & play"	Windows 95 Windows NT	Asiste los protocolos PPI, 1 MPI y PROFIBUS-DP, 9.600 bit/s a 12 Mbit/s para PCs

La tarjeta y el protocolo en cuestión se ajustan en el cuadro de diálogo "Ajustar interface PG/PC" accesible desde STEP 7-Micro/WIN, o bien en el panel de control de Windows.

En Windows 95 ó Windows NT se puede seleccionar cualquier protocolo (PPI, MPI o PROFIBUS) a utilizar con cualquiera de las tarjetas disponibles. Como regla general, se deberá seleccionar el protocolo PPI a 9.600 bit/s ó a 19.200 bit/s para la comunicación con las CPUs S7-200.



Cada tarjeta incorpora un interface RS-485 sencillo para la conexión a la red PROFIBUS. La tarjeta CP 5511 PCMCIA dispone de un adaptador que incorpora el conector D subminiatura de 9 pines.

Uno de los extremos del cable MPI se conecta al interface RS-485 de la tarjeta y el otro, al conector del interface de programación en la red

## **7.5 Comunicación en redes de periferia descentralizadas (DP)**

PROFIBUS-DP (o estándar DP) es un protocolo de telecomunicación definido en la norma europea EN 50170. Los dispositivos que cumplen con dicha norma son compatibles entre sí, aunque sean de diferentes fabricantes. “DP” es la abreviatura inglesa de “Distributed Peripherals” que significa periferia descentralizada (o periferia distribuida). “PROFIBUS” es la abreviatura de “Process Field Bus”.

La configuración de un maestro DP le permite reconocer las direcciones, los tipos de esclavos y las informaciones relativas a la parametrización que éstos necesitan. Al maestro se le indica también dónde depositar los datos que haya leído de los esclavos (entradas) y de dónde obtener los datos a escribir en los esclavos (salidas). El maestro DP establece la red e inicializa sus esclavos DP, escribiendo posteriormente los parámetros y la configuración de E/S en el esclavo. Luego lee las informaciones de diagnóstico del esclavo DP para verificar que éste haya aceptado los parámetros y la configuración de E/S. El maestro comienza entonces a intercambiar datos con el esclavo. En cada intercambio con el esclavo, escribe en las salidas y lee de las entradas. Dicho intercambio de datos continúa indefinidamente. Los esclavos pueden informar al maestro si se presenta una condición excepcional. Entonces, el maestro lee la información de diagnóstico del esclavo. Una vez que un maestro DP haya escrito los parámetros y la configuración de E/S en un esclavo DP y éste los haya aceptado, el esclavo será propiedad del maestro. El esclavo sólo acepta peticiones de escritura de su respectivo maestro. Los demás maestros de la red pueden leer las entradas y salidas del esclavo, pero no escribir datos en él.

## **CAPÍTULO 8**

**Instalar y utilizar el software STEP7-Micro/WIN**

## 8.1 Instalar el software de STEP7- Micro/WIN

STEP 7-Micro/WIN es el software de programación para el SIMATIC S7-200. Es una aplicación que se puede ejecutar tanto en el entorno Windows 3.1 de 16 bits (STEP 7-Micro/WIN 16) como en los entornos Windows 95 y Windows NT de 32 bits (STEP 7-Micro/WIN 32). Para poder utilizar STEP 7-Micro/WIN se requieren los siguientes equipos:

- Recomendable: PC con procesador 80586 o superior y 16 MB de RAM o una unidad de programación Siemens (por ejemplo. PG 740); requisito mínimo: procesador 80486 con 8 MB de RAM
- Uno de los equipos siguientes:
  - Un cable PC/PPI conectado al puerto de comunicación del PC o la PG (COM1 ó COM2).
  - Un procesador de comunicaciones (CP) y un cable de interface multipunto (MPI).
  - Una tarjeta de interface multipunto (MPI) (junto con la tarjeta MPI se suministra un cable de comunicación).
- 50 MB libres en el disco duro (como mínimo)
- Microsoft Windows 3.1, Windows para Trabajo en Grupo 3.11, Windows 95 o Windows NT 4.0 o superior

STEP 7-Micro/WIN incorpora una Ayuda online detallada. Para obtener más información sobre cualquier pregunta, elija uno de los comandos del menú **ayuda** o pulse el botón F1.

### Instalación en Windows 3.1

Para instalar el software STEP 7-Micro/WIN 16 en un PC con Windows 3.1 (Windows para Trabajo en Grupo 3.11):



## 8.1 Instalar el software de STEP7- Micro/WIN

STEP 7-Micro/WIN es el software de programación para el SIMATIC S7-200. Es una aplicación que se puede ejecutar tanto en el entorno Windows 3.1 de 16 bits (STEP 7-Micro/WIN 16) como en los entornos Windows 95 y Windows NT de 32 bits (STEP 7-Micro/WIN 32). Para poder utilizar STEP 7-Micro/WIN se requieren los siguientes equipos:

- Recomendable: PC con procesador 80586 o superior y 16 MB de RAM o una unidad de programación Siemens (por ejemplo. PG 740); requisito mínimo: procesador 80486 con 8 MB de RAM
- Uno de los equipos siguientes:
  - Un cable PC/PPI conectado al puerto de comunicación del PC o la PG (COM1 ó COM2).
  - Un procesador de comunicaciones (CP) y un cable de interface multipunto (MPI).
  - Una tarjeta de interface multipunto (MPI) (junto con la tarjeta MPI se suministra un cable de comunicación).
- 50 MB libres en el disco duro (como mínimo)
- Microsoft Windows 3.1, Windows para Trabajo en Grupo 3.11, Windows 95 o Windows NT 4.0 o superior

STEP 7-Micro/WIN incorpora una Ayuda online detallada. Para obtener más información sobre cualquier pregunta, elija uno de los comandos del menú **ayuda** o pulse el botón F1.

### Instalación en Windows 3.1

Para instalar el software STEP 7-Micro/WIN 16 en un PC con Windows 3.1 (Windows para Trabajo en Grupo 3.11):

1. Inserte el disquete 1 en la correspondiente unidad del PC (que es, por lo general, la unidad de disquete A: o B:).
2. En el Administrador de programas, elija el comando de menú **Archivo ► Ejecutar**.
3. En el cuadro de diálogo “Ejecutar”, introduzca **a:\setup** y haga clic en “Aceptar” o pulse la tecla ENTER. Así se inicia la instalación.
4. Siga las instrucciones que vayan apareciendo en pantalla hasta finalizar la instalación.

### **Instalación en Windows 95 o Windows NT 4.0**

Para instalar el software STEP 7-Micro/WIN 32 en un PC con Windows 95 o Windows NT 4.0:

1. Inserte el disquete 1 en la correspondiente unidad del PC (que es, por lo general, la unidad de disquete A: o B:).
2. Haga clic en el botón “Inicio” para abrir el menú de arranque de Windows 95.
3. Haga clic en el comando **Ejecutar**.
4. En el cuadro de diálogo “Ejecutar”, introduzca **a:\setup** y haga clic en “Aceptar” o pulse la tecla ENTER. Así se inicia la instalación.
5. Siga las instrucciones que vayan apareciendo en pantalla hasta finalizar la instalación.
6. Al finalizar la instalación aparece automáticamente el cuadro de diálogo donde se pueden instalar o desinstalar las tarjetas (ver figura 8-2).

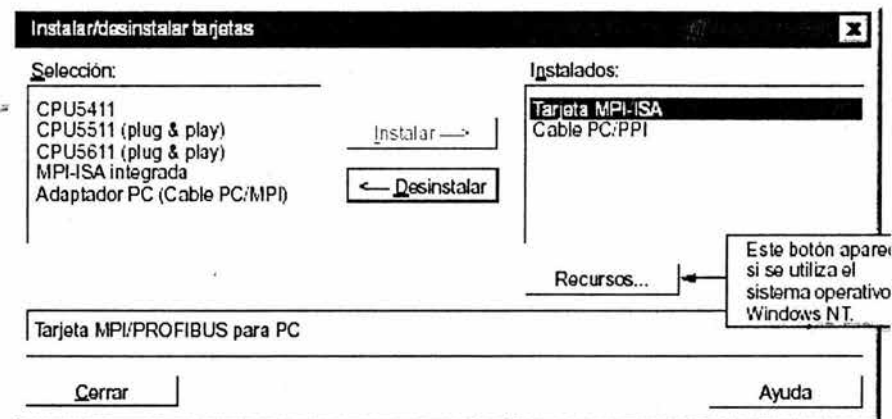


Figura 8-2 Cuadro de diálogo “instalar/desinstalar tarjetas”.

## 8.2 Utilizar STEP 7-Micro/WIN para configurar la comunicación

Si utiliza Windows 95 o Windows NT 4.0, el cuadro de diálogo “Instalar/desinstalar tarjetas” automáticamente cuando termine de instalar el software ver. figura 9-2. En Windows 3.1, por la siguiente forma:

1. Elija el comando de menú **Instalar ► Comunicación** para acceder al cuadro de diálogo “Comunicación”.
2. Haga clic en el botón “Interface PG/PC” para acceder al cuadro de diálogo “Ajustar interface”.
3. Haga clic en el botón “Instalar” para acceder al cuadro de diálogo “Instalar/desinstalar tarjetas”.  
Figura 9-2.

La instalación del hardware de comunicación depende de los criterios siguientes:

1. El sistema operativo utilizado (Windows 3.1, Windows 95 o Windows NT 4.0).
2. El tipo de hardware utilizado:

- PC con cable PC/PPI
- PC o unidad de programación SIMATIC con tarjeta de interface multipunto (MPI) o con procesador de comunicaciones (CP)
- CPU 221, CPU 222, CPU 224, CPU 226

3. La velocidad de transferencia utilizada.

### 8.3 Establecer la comunicación con una CPU S7-200

Las CPUs S7-200 se pueden disponer en diversas configuraciones para asistir la comunicación en redes. El software STEP 7-Micro/WIN se puede instalar en un PC con Windows 3.1x, Windows 95 o Windows NT, o bien en una unidad de programación SIMATIC (por ejemplo. la PG 740). El PC o la PG se pueden utilizar como unidades maestras en cualquiera de las siguientes configuraciones:

- Un sólo maestro se conecta a uno o varios esclavos.
- Un sólo maestro se conecta a uno o varios esclavos y maestros.
- Un sólo maestro se conecta a uno o varios esclavos., este maestro se conecta a través de módems de 11 bits bien sea a una CPU S7-200 que actúe de esclava, o bien a una red de CPUs S7-200 también esclavas.

#### Conectar el PC a la CPU S7-200 utilizando el cable PC/PPI

La figura 8-4 muestra una configuración típica para conectar el PC a la CPU mediante el cable PC/PPI. Para establecer un enlace correcto entre los componentes:

1. Ajuste los interruptores DIP del cable PC/PPI para determinar la velocidad de transferencia deseada.

2. Conecte el extremo RS-232 ("PC") del cable PC/PPI al puerto de comunicación de su PC (COM1 o COM2) y apriete los tornillos de conexión.

3. Conecte el otro extremo (RS-485) del cable PC/PPI al interface de comunicación de la CPU y apriete los tornillos de conexión.

Ajustes de los interruptores DIP (abajo = 0, arriba = 1):  
 0 1 0 0 = 9600 bit/s (v. fig).  
 0 0 1 0 = 19200 bit/s

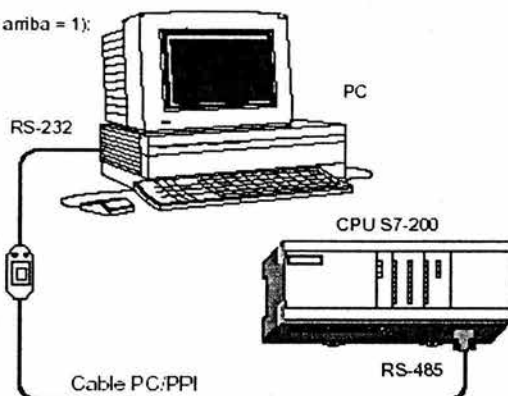
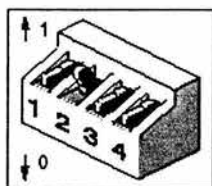


Figura 8-4 Comunicación con una CPU en el modo PPI

La figura 8-5 muestra una configuración compuesta por un PC conectado a varias CPUs S7-200. STEP 7-Micro/WIN se ha diseñado para que se comunique con una CPU S7-200 a la vez. No obstante, se puede acceder a cualquier CPU incorporada en la red.

Las CPUs que muestra la figura 8-5 podrían ser tanto esclavas como maestras. El TD 200 es una unidad maestra.

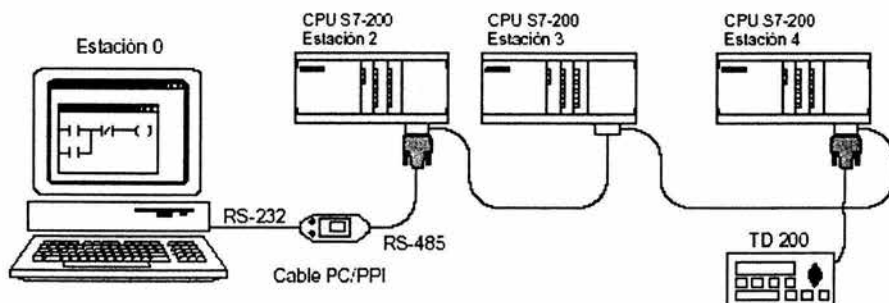


Figura 8-5 Utilizando un cable PC/PPI para la comunicación con varias CPUs S7-200



## Configurar los parámetros de comunicación en STEP7- Micro/WIN

STEP 7-Micro/WIN incorpora el cuadro de diálogo “Comunicación” donde se pueden configurar los ajustes de comunicación como se puede observar en la figura 8-6.

Para acceder a dicho cuadro puede optar por uno de los siguientes métodos:

- Elija el comando de menú **Instalar ► Comunicación**
- Cree un nuevo proyecto y haga clic en el botón “Comunicación” en el cuadro de diálogo “Tipo de CPU”.
- Si tiene un proyecto abierto, elija el comando de menú **CPU ► Tipo** y haga clic en el botón “Comunicación” del cuadro de diálogo “Tipo de CPU”.

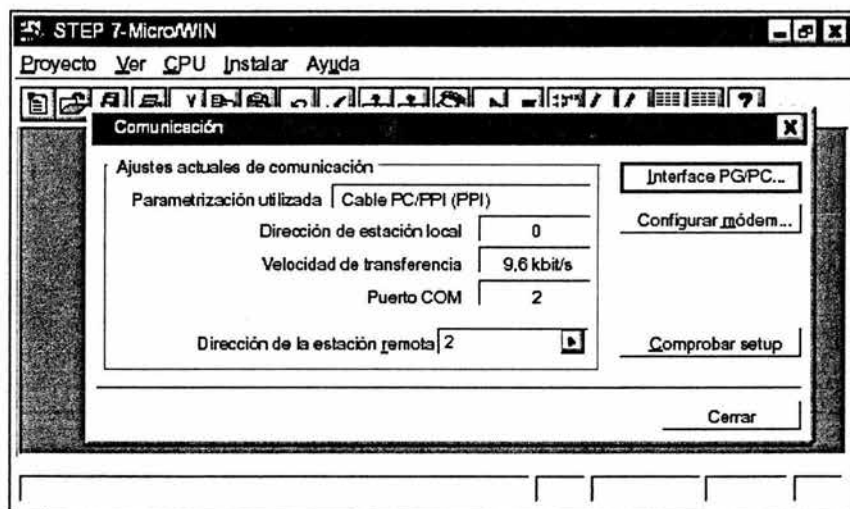


Figura 8-6 Configurar los parámetros de comunicación entre la PG o el PC y la CPU

En el cuadro de diálogo “Comunicación”, haga clic en el botón “Interface PG/PC” para acceder al cuadro de diálogo “Ajustar interface PG/PC” ver figura 8-7.

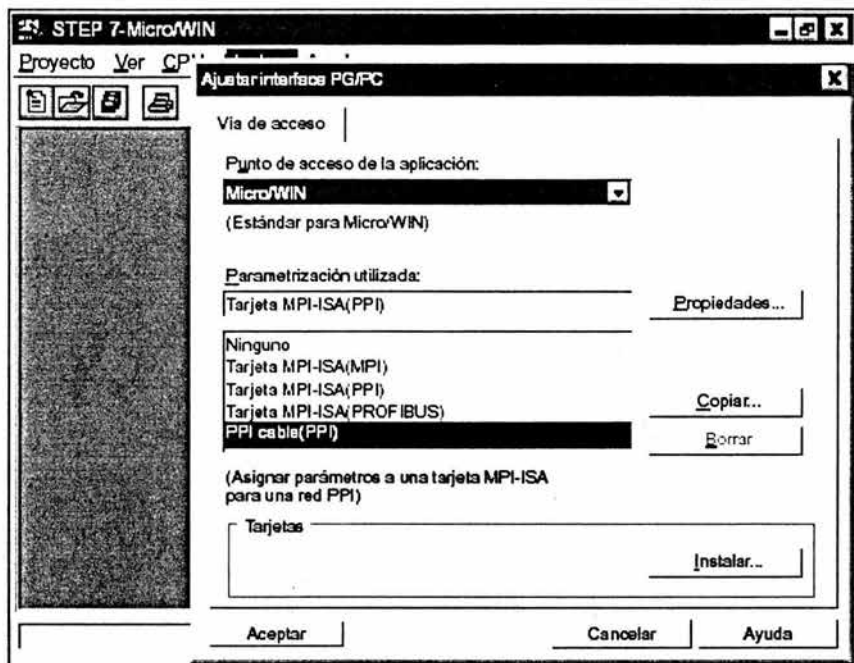


Figura 8-7 Ajustes en el cuadro de diálogo “Interface PG/PC”

### Seleccionar y configurar la parametrización correcta

Para seleccionar una parametrización:

1. Estando en el cuadro de diálogo “Ajustar interface PG/PC” (ver figura 8-7), elija “Micro/WIN” en el cuadro de lista “Punto de acceso de la aplicación” de la ficha “Vía de acceso”.
2. Asegúrese de que su componente de hardware esté instalado.
3. Determine el protocolo a utilizar.
4. Elija la configuración correcta en el cuadro de lista “Parametrización utilizada”.
5. Haga clic en el botón “Propiedades”.

Allí puede efectuar los ajustes conforme a la parametrización elegida.

### **Ajuste de los parámetros del cable PC/PPI (PPI)**

Si en el cuadro de diálogo “Ajustar interface PG/PC” está seleccionado el cable PC/PPI (PPI) y se hace clic en el botón “Propiedades.”, aparecerá la ficha de propiedades del mismo ver figura 8-8.

Para ajustar las propiedades:

1. En la ficha “Red PPI”, seleccione un número en el cuadro “Dirección de la estación local”, dicho número corresponde a la dirección de STEP 7-Micro/WIN en la red.
2. Elija un valor en el cuadro “Timeout”. Éste representa el tiempo durante el que los drivers de comunicación pueden intentar establecer enlaces. El valor predeterminado debería ser suficiente.
3. Determine si desea que STEP 7-Micro/WIN participe en una red donde existan varios maestros. Puede dejar marcada la casilla “Red multimaestro”, a menos que utilice un módem. En éste último caso, la casilla no se podrá marcar, puesto que STEP 7-Micro/WIN no asiste esa función.
4. Ajuste la velocidad de transferencia que desee utilizar para la comunicación de STEP 7-Micro/WIN en la red.
5. Elija la dirección de estación más alta. A partir de esta dirección, STEP 7-Micro/WIN no busca más maestros en la red.

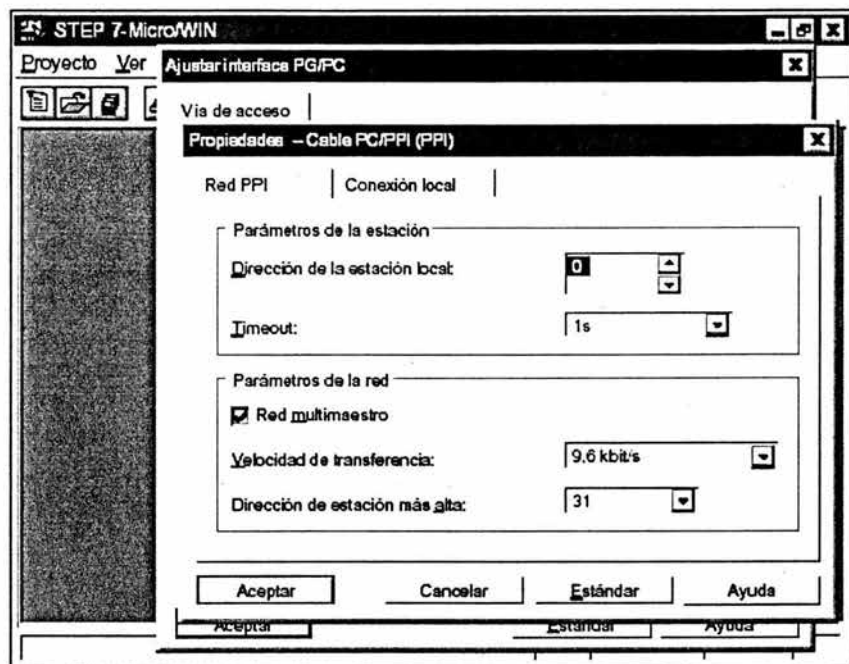


Figura 8-8 Propiedades del cable PC/PPI, red PPI

6. Haga clic en la ficha “Conexión local” ver figura 8-9.
7. En la ficha “Conexión local”, seleccione el puerto COM al que esté conectado el cable PC/PPI. Si utiliza un módem, seleccione el puerto COM al que esté conectado el módem y marque la casilla de verificación “Utilizar módem”.
8. Haga clic en el botón “Aceptar” para cerrar el cuadro de diálogo “Ajustar interface PG/PC”.

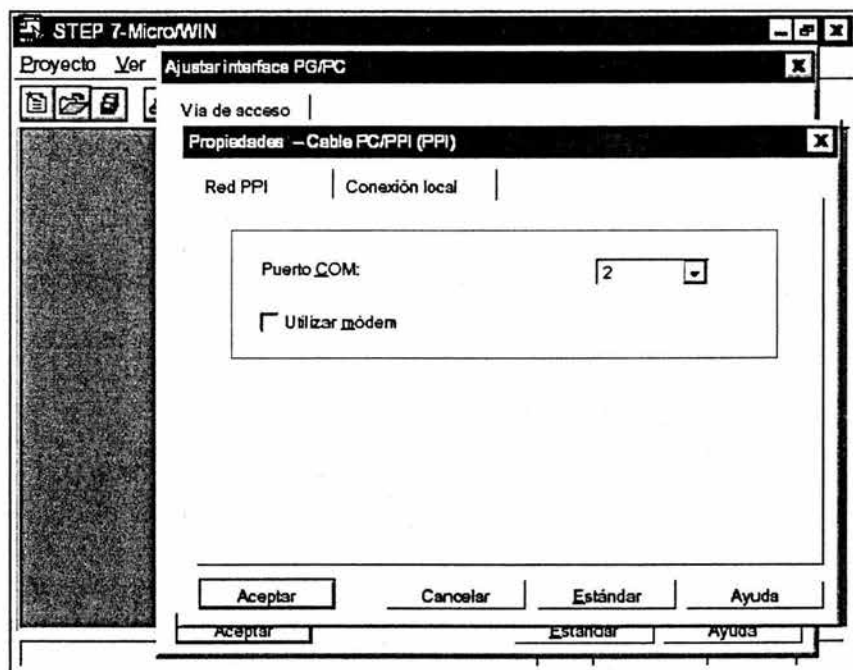


Figura 8-9 Propiedades del cable PC/PPI

### Ajustar los parámetros de la tarjeta MPI (PPI)

Para ajustar las propiedades:

1. En la ficha “Red PPI”, seleccione un número en el cuadro “Dirección de la estación local”. Dicho número corresponde a la dirección de STEP 7-Micro/WIN en la red ver figura 8-10.
2. Elija un valor en el cuadro “Timeout”. Éste representa el tiempo durante el que los drivers de comunicación pueden intentar establecer enlaces. El valor predeterminado debería ser suficiente.
3. Determine si desea que STEP 7-Micro/WIN participe en una red donde existan varios maestros. Puede dejar marcada la casilla “Red multimaestro”.

4. Ajuste la velocidad de transferencia que desee utilizar para la comunicación de STEP 7-Micro/WIN en la red.
5. Elija la dirección de estación más alta. A partir de esta dirección, STEP 7-Micro/WIN no busca más maestros en la red.
6. Haga clic en el botón “Aceptar” para cerrar el cuadro de diálogo “Ajustar interface PG/PC”.



Figura 8-10 Propiedades de la tarjeta MPI-ISA (PPI)

### Ajustar los parámetros de la tarjeta MPI (MPI)

Partiendo del cuadro de diálogo “Ajustar interface PG/PC”, si alguna de las tarjetas MPI o CP mencionadas se utiliza junto con el protocolo MPI y se hace clic en el botón “Propiedades”, aparece la ficha de propiedades de la tarjeta XXX (MPI), donde “XXX” es el tipo de tarjeta que se ha instalado (Por ejemplo, MPI-ISA) (ver figura 8-11).

Para ajustar las propiedades:

1. En la ficha “Red MPI”, seleccione un número en el cuadro “Dirección de la estación local”. Dicho número corresponde a la dirección de STEP 7-Micro/WIN en la red.
2. Verifique que la casilla “No se conectará como único maestro” no esté activada, independientemente de la cantidad de maestros que conformen la red. Si dicha casilla contiene una marca de verificación, haga clic allí para desactivarla. Asegúrese de conectar el cable de comunicación entre la unidad de programación (PG) y la CPU antes de iniciar la comunicación.  
Si la comunicación se establece antes de conectar la PG a una red de CPUs que incorpore uno o más maestros, se interrumpirá mientras se reinicializa la red.
3. Elija un valor en el cuadro “Timeout”. Éste representa el tiempo durante el que los drivers de comunicación pueden intentar establecer enlaces. El valor predeterminado debería ser suficiente.
4. Ajuste la velocidad de transferencia que desee utilizar para la comunicación de STEP 7-Micro/WIN en la red. Puesto que es probable que se esté utilizando el interface DP de una CPU 215, se puede seleccionar cualquier velocidad de transferencia disponible (hasta 12 Mbit/s).
5. Elija la dirección de estación más alta. A partir de esta dirección, STEP 7-Micro/WIN no busca más maestros en la red.
6. Haga clic en el botón “Aceptar” para cerrar el cuadro de diálogo “Ajustar interface PG/PC”.



Figura 8-11 Propiedades de la tarjeta MPI-ISA(MPI)

## 8.4 Configurar las preferencias para STEP7 Micro/WIN

Antes de crear un proyecto, indique sus preferencias para el entorno de programación. Para seleccionar las preferencias:

1. Elija el comando de menú **Instalar ► Preferencias** como muestra la figura 8-12.
2. Elija sus preferencias de programación en el cuadro de diálogo que aparece a continuación.
3. Confirme sus ajustes pulsando la tecla ENTER o haciendo clic en el botón “Aceptar”.



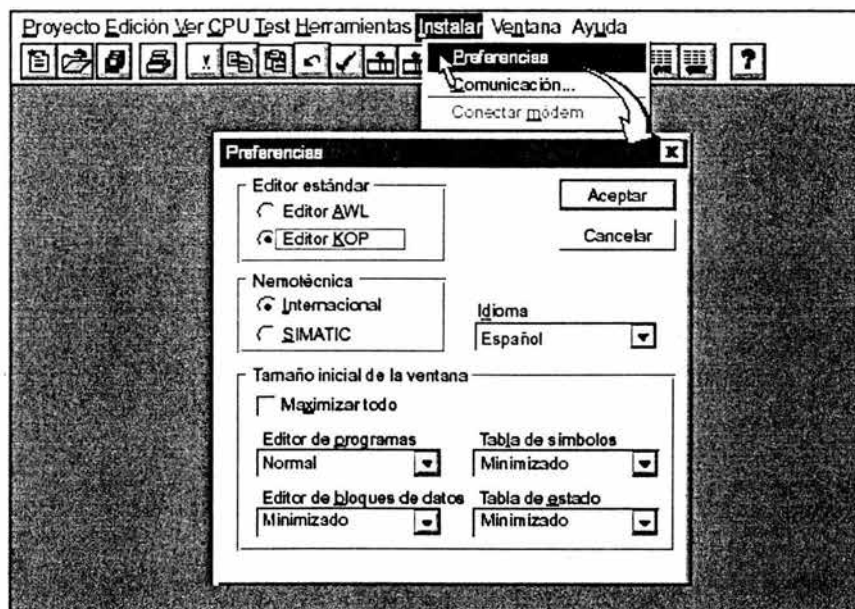


Figura 8-12 Seleccionar las preferencias de programación

## 8.5 Crear y guardar un proyecto

Antes de poder generar un programa es necesario crear o abrir un proyecto. Al crear un proyecto, STEP 7-Micro/WIN abre los siguientes editores:

- \_ Editor KOP, FUP o AWL (dependiendo del editor estándar elegido)
- \_ Editor de bloque de datos
- \_ Tabla de estado
- \_ Tabla de símbolos

### Crear un nuevo proyecto

Mediante el menú Proyecto se puede crear un nuevo proyecto, como muestra la figura 8-13. Elija el comando de menú **Proyecto ► Nuevo**. Aparecerá el cuadro de diálogo “Tipo de CPU”. Al elegir una CPU

determinada en la lista desplegable, el software visualizará sólo las opciones disponibles para la misma. Si elige “Ninguno”, el programa no se restringirá por lo que respecta a la CPU.

Al cargar el programa en la CPU, ésta verifica si se utilizan funciones que no están disponibles. Por ejemplo, si su programa contiene una operación no asistida por la CPU correspondiente, el programa se rechazará.

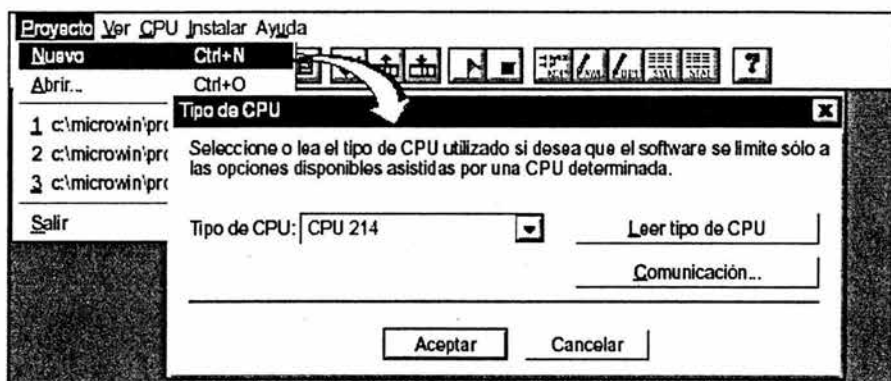


Figura 8-13 Crear un nuevo proyecto

### Guardar un proyecto

Para guardar todos los componentes de su proyecto, elija el comando de menú **Proyecto ► Guardar todo** o haga clic en el botón correspondiente:



Para guardar una copia del proyecto actual bajo otro nombre o en otro directorio, elija el comando de menú **Proyecto ► Guardar como**.

## 8.6 Crear un programa

STEP 7-Micro/WIN permite crear el programa de usuario (OB1) con el editor KOP o el editor AWL.

## Introducir programas en KOP

La ventana del editor KOP que muestra la figura 8-14 permite escribir programas utilizando símbolos gráficos. La barra de herramientas incluye algunos de los elementos KOP más usuales para introducir los programas. El primer cuadro de lista desplegable (a la izquierda) contiene los grupos de operaciones. Para acceder a dichos grupos, haga clic allí o pulse la tecla F2. Una vez seleccionado un grupo, las operaciones correspondientes al mismo aparecerán en la segunda lista desplegable (a la derecha). Para visualizar una lista de todas las operaciones en orden alfabético, pulse la tecla F9 o elija "Todas las categorías". Alternativamente puede elegir el comando de menú **Ver ► Barra de operaciones** para visualizar la barra de operaciones KOP.

En cada segmento se pueden introducir dos tipos de comentarios:

1. Los comentarios de segmento de una sola línea siempre son visibles en el editor KOP. Para acceder haga clic en cualquier parte del título del segmento.
2. Para acceder a los comentarios del segmento de más de una línea haga doble clic en el número del segmento. Dichos comentarios sólo se pueden visualizar en un cuadro de diálogo, pero se imprimen siempre por completo.

Para comenzar a introducir su programa:

3. Para introducir el título del programa, elija el comando de menú **Edición ► Título**. Introduzca el título y haga clic en el botón "Aceptar".
4. Para introducir elementos KOP, elija el tipo de elemento deseado haciendo clic en el botón correspondiente. Alternativamente, puede seleccionarlo de la lista de operaciones.
3. Introduzca la dirección o el parámetro en cada cuadro de texto y pulse la tecla ENTER.

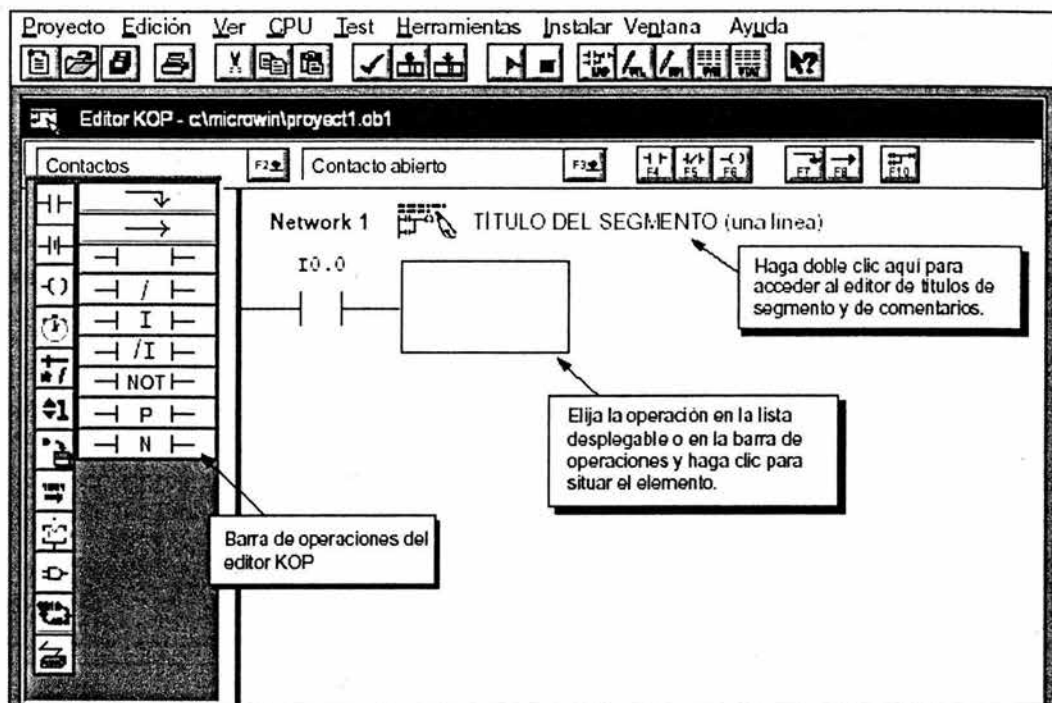


Figura 8-14 Ventana del editor KOP

### Introducir el Programa en AWL o FUP

El editor AWL y el editor FUP son editores que brinda cierta flexibilidad en cuanto a la forma de introducir las operaciones del programa.

### Compilar el programa

Una vez completado un segmento o una serie de segmentos puede comprobar la sintaxis lógica eligiendo el comando de menú **CPU ► Compilar** o haciendo clic en el botón correspondiente:

## Cargar el programa en la CPU

Una vez finalizado el programa, el proyecto se puede cargar en la CPU. Elija el comando de menú **Proyecto ► Cargar en CPU** o haga clic en el botón correspondiente de la ventana principal, ver figura 8-16.

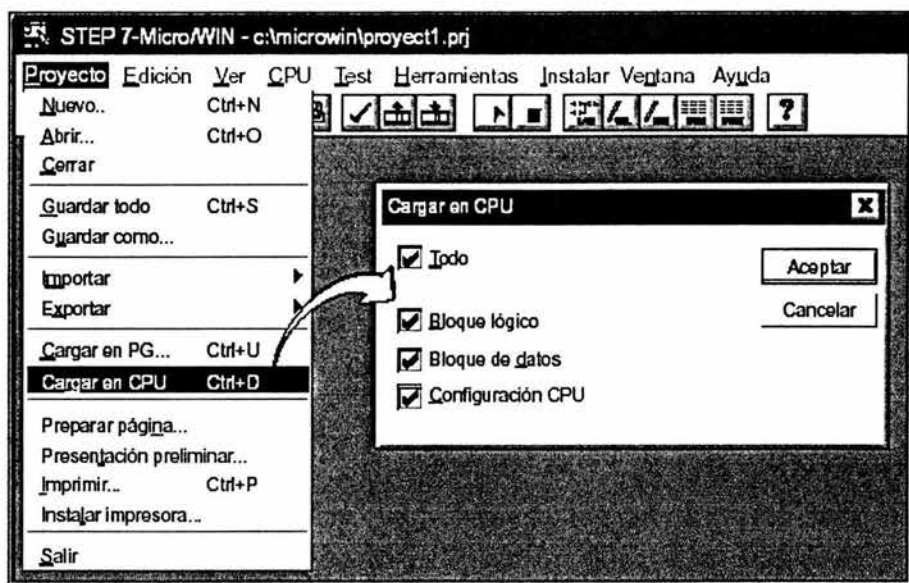


Figura 8-16 Cargar los componentes del proyecto en la CPU

- El bloque lógico (OB1) contiene el programa a ejecutar por la CPU.
- El bloque de datos (DB1) contiene los valores de inicialización utilizados en el programa de usuario.
- La configuración de la CPU (CFG) contiene la información de instalación del sistema, incluyendo los parámetros de comunicación, las áreas remanentes, los ajustes de los filtros de entrada, las contraseñas y los ajustes de las salidas.

Para confirmar los ajustes y cargarlos en la CPU haga clic en el botón "Aceptar" o pulse la tecla ENTER.

## Visualizar un programa en KOP o AWL o FUP

Para visualizar un programa en AWL o KOP, elija el comando de menú **Ver ▶ AWL, FUP o Ver ▶ KOP**, como muestra la figura 8-17.

Para obtener esa misma representación de las operaciones AWL, puede elegir el comando de menú **CPU ▶ Compilar** estando activo el editor AWL.

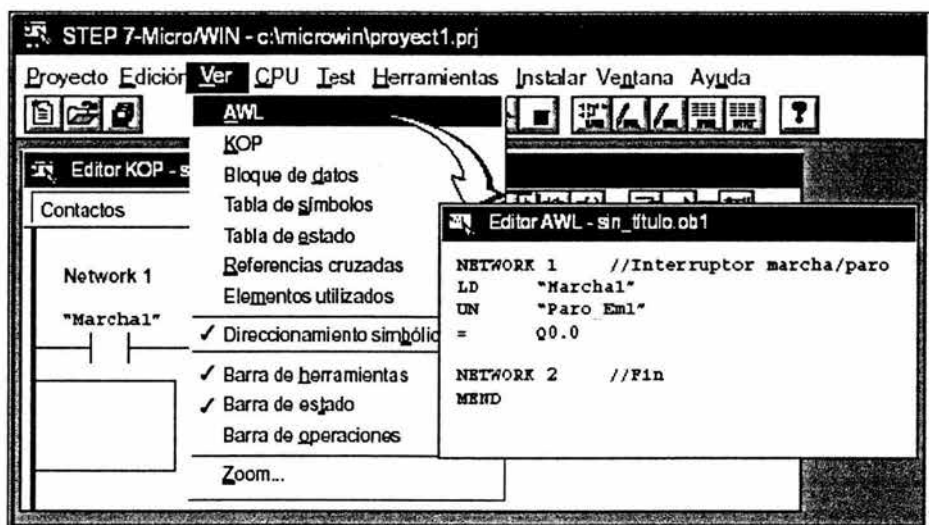


Figura 8-17 Cambiar la visualización del programa de KOP a AWL

## 8.7 Crear un bloque de datos

El bloque de datos permite asignar datos iniciales sólo a la memoria V (memoria de variables). Se pueden efectuar asignaciones a bytes, palabras o palabras dobles de la memoria V. Los comentarios (precedidos de dos barras inclinadas) son opcionales.

Con el editor de bloques de datos puede predefinir e inicializar las variables utilizadas en el programa. El uso del bloque de datos es opcional ver figura 8-18.

El editor de bloques de datos aparece minimizado al tamaño de un icono en la parte inferior de la pantalla principal (si se ha seleccionado así con el comando de menú **Instalar ► Preferencias**).

Para acceder al editor de bloques de datos, haga doble clic en el icono o un solo clic en el botón Restaurar o Maximizar.

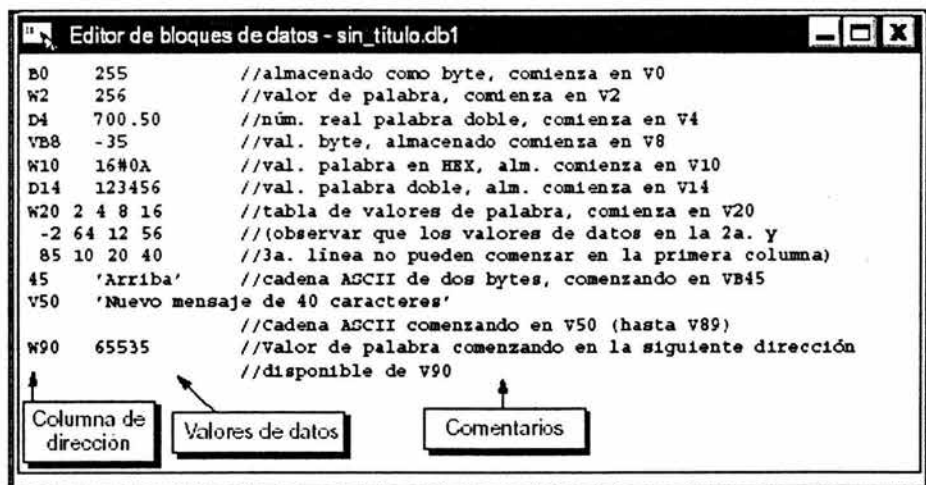


Figura 8-18 Ejemplo de un bloque de datos

## 8.8 Utilizar la tabla de estado

Con la tabla de estado es posible leer, escribir o forzar variables en el programa de usuario.

La tabla de estado aparece minimizada al tamaño de un icono en la parte inferior de la pantalla principal (si se ha seleccionado así con el comando de menú **Instalar ► Preferencias**) ver figura 8-19.

Para acceder a la tabla de estado, haga doble clic en el icono o un clic en el botón Restaurar o Maximizar (en Windows 95).

Tabla de estado

Dirección	Formato	Valor actual	Valor nuevo
*Marcha 1*	Binario	2#0	1
I0.2	Binario	2#0	
*Luz Verde 1*	Binario	2#0	
Q1.2	Binario	2#1	
VB0	Con signo	+84	
VW2	Sin signo	4400	
VW4	Binario	2#0000001000110010	
VW6	Hexadecimal	16#0064	16#65
VD10	Coma flotante	0.0000	10.0
VD14	ASCII	'TEMP'	
VW20	Hexadecimal	16#0027	16#28
VW24	ASCII	'AB'	BA

Pulse la BARRA ESPACIADORA o haga doble clic en la celda para elegir un formato válido.

Para cambiar un valor, introduzca aquí el nuevo valor y haga clic en el botón 'Escribir'.

Figura 8-19 Ejemplo de una tabla de estado



## **CAPÍTULO 9**

**Aplicación para un mezclador de pintura**

## 9.1 Requisitos para el programa de ejemplo

Después de crear y cargar el programa de ejemplo en la CPU S7-200, ésta podrá procesarlo. La figura 9-2 muestra los componentes necesarios para procesar y observar el programa:

- Cable PC/PPI o tarjeta MPI instalada en el PC y cable RS-485 para conectar a la CPU S7-200
- CPU S7-200
- Simulador de entradas
- Cable y fuente de alimentación
- STEP 7-Micro/WIN 32

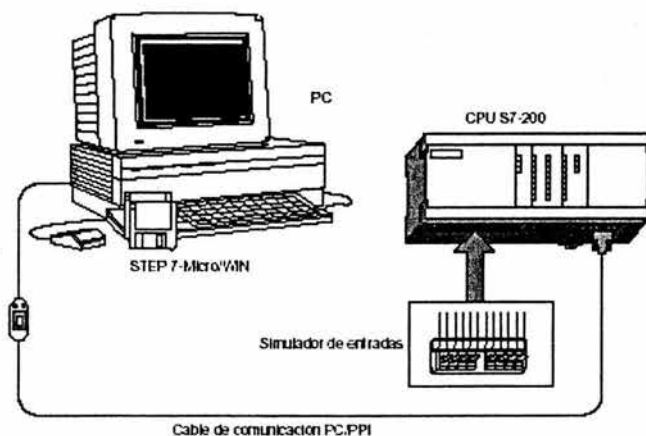


Figura 9-2 Requisitos para ejecutar el programa de ejemplo

## 9.2 Aplicación: Mezclador de pintura

La figura 9-3 muestra un mezclador de pintura. En la parte superior del mezclador hay dos tuberías por donde se suministran dos componentes diferentes. En la parte inferior se aprecia una sola tubería que transporta la mezcla de pintura preparada. Por una tubería en la parte inferior del tanque se conduce la mezcla ya lista. El programa de ejemplo controla la operación de llenado, supervisa el nivel del tanque y controla un ciclo de mezcla y de calefacción como se describe a continuación:

- Paso 1: Llenar el mezclador con el componente 1.
- Paso 2: Llenar el mezclador con el componente 2.
- Paso 3: Supervisar el nivel del mezclador para cerrar el interruptor de nivel superior.
- Paso 4: Mantener el estado de la bomba si se abre el interruptor de arranque.
- Paso 5: Arrancar el ciclo de mezcla y calefacción.
- Paso 6: Poner en marcha el motor del mezclador y la válvula de vapor.
- Paso 7: Vaciar el mezclador.
- Paso 8: Contar cada ciclo.

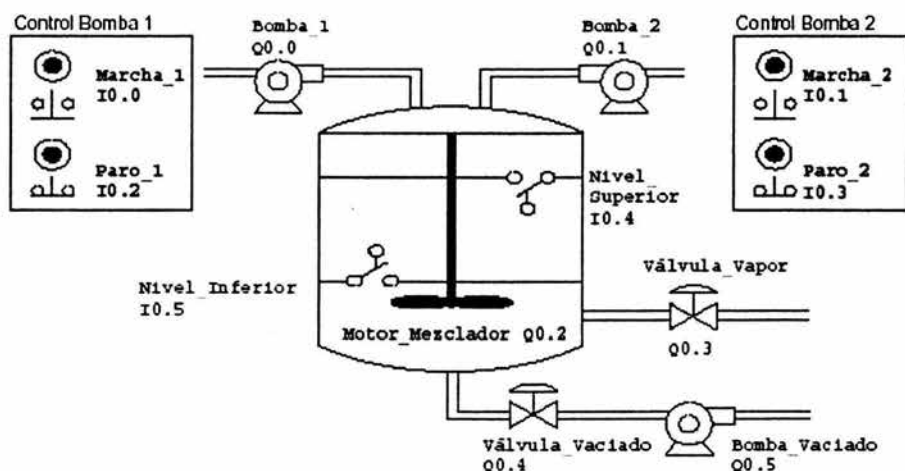
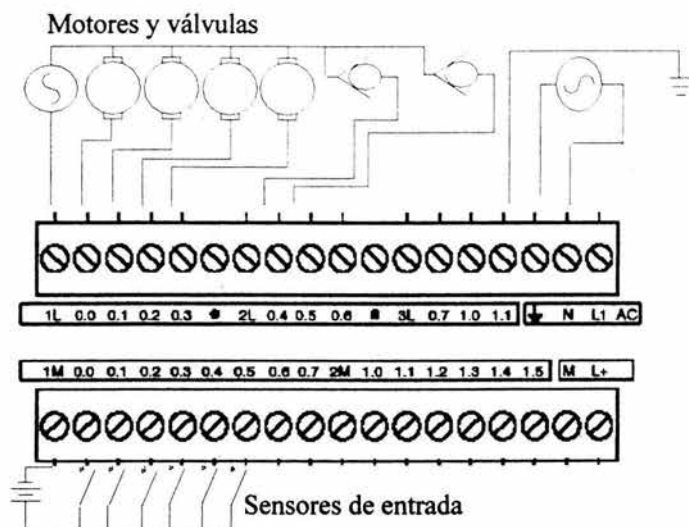


Figura 9-3 Mezclador de pintura

El cableado en el PLC es como se muestra en la figura siguiente, los motores y válvulas se conectan en las salidas del PLC recordando que cada salida le corresponde un solo elemento, y los sensores se conectan en las entradas del PLC.



Conexión de los motores, válvulas y sensores en el PLC

### 9.3 Programa de ejemplo en AWL y en KOP

El programa de ejemplo se puede representar en AWL (Lista de instrucciones) o en KOP (Esquema de contactos). En la tabla 9-3 se representa dicho programa en el lenguaje AWL y la figura 9-4, lo muestra en KOP.

AWL	Descripción
NETWORK 1	//Llenar el mezclador con el componente 1.
LD "Marcha_1"	
O "Bomba_1"	
A "Paro_1"	
AN "Nivel_Superior"	
= "Bomba_1"	
NETWORK 2	//Llenar el mezclador con el componente 2.
LD "Marcha_2"	
O "Bomba_2"	
A "Paro_2"	
AN "Nivel_Superior"	
= "Bomba_2"	
NETWORK 3	//Activar marca si se alcanza el nivel superior.
LD "Nivel_Superior"	
S "Niv_Sup_Alcanz", 1	
NETWORK 4	//Arrancar temporizador si se alcanza el nivel superior.
LD "Niv_Sup_Alcanz"	
TCN "Temporis_Mezcla"	
NETWORK 5	//Poner en marcha el motor mezclador.
LDN "Temporis_Mezcla"	
A "Niv_Sup_Alcanz"	
= "Motor_Mezclador"	
= "Válvula_Vapor"	
NETWORK 6	//Vaciar el mezclador.
LD "Temporis_Mezcla"	
AN "Nivel_Inferior"	
= "Válvula_Vaciado"	
= "Bomba_Vaciado"	
NETWORK 7	//Contar cada ciclo.
LD "Nivel_Inferior"	
A "Temporis_Mezcla"	
LD "Desactivar"	
CTU "Contador_Ciclos",	
+12	
NETWORK 8	//Desactivar marca si se alcanza el nivel inferior //Y si se detiene el temporizador.
LD "Nivel_Inferior"	
A "Temporis_Mezcla"	
R "Niv_Sup_Alcanz", 1	
NETWORK 9	//Fin del programa principal.
MBND	

Tabla 9-3 Programa de ejemplo en AWL

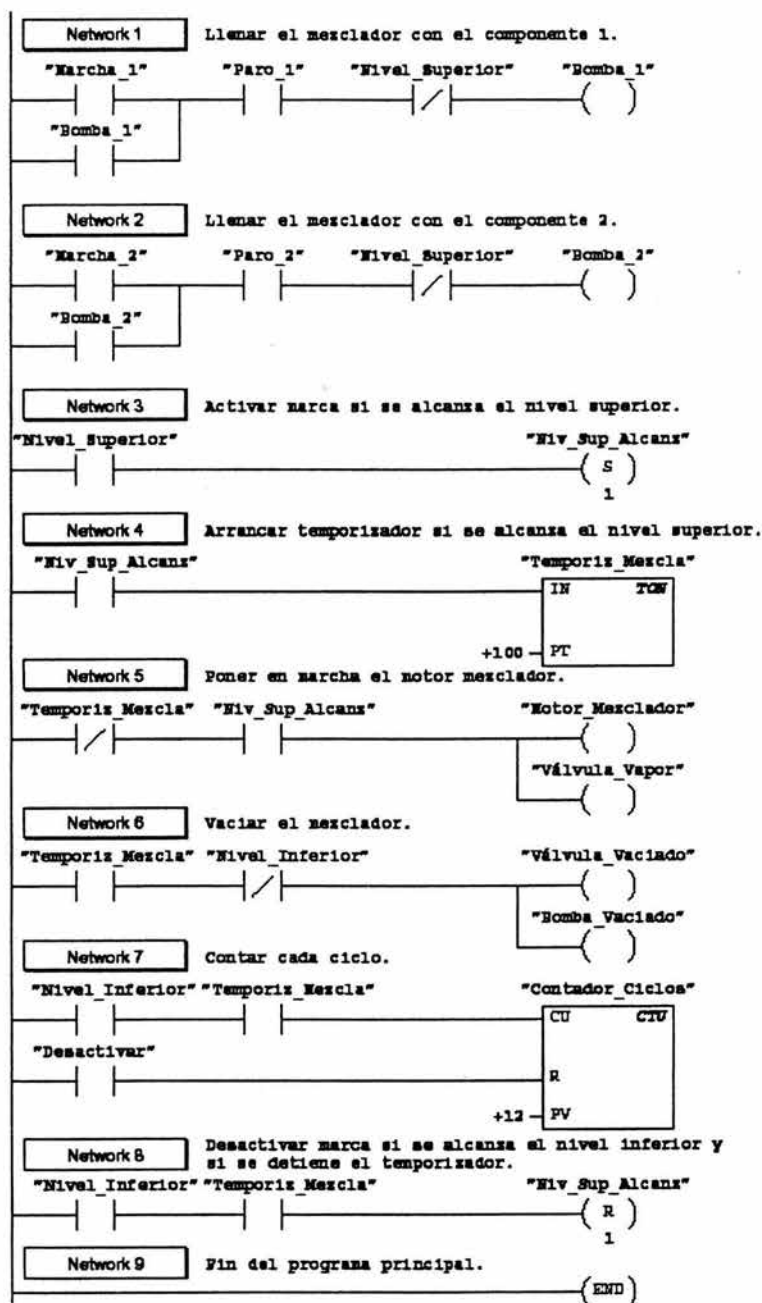


Figura 9-4 Programa de ejemplo en KOP

## **BIBLIOGRAFÍA**

SIMATIC S7-200

Siemens

Referencia de manual 6ES7298-8FA01-8DH0

S7-200 Manual de sistema

Siemens

Referencia de manual 6ES7 298-8FA22-8DH0

Componentes para la totally integrated automation

Catálogo ST 70

Siemens