

## SIMATIC

### STEP 7 De S5 a S7

#### Guía para facilitar la transición

Indicaciones importantes,  
Contenido

---

**1ª parte:  
Planificar la transición**

---

Introducción

---

**1**

Hardware

---

**2**

Software

---

**3**

**2ª parte:  
Conversión de programas**

---

Procedimiento

---

**4**

Preparando la conversión

---

**5**

Conversión

---

**6**

Retocar el programa convertido

---

**7**

Compilar

---

**8**

Ejemplo de aplicación

---

**9**

**Anexos**

---

Listas de operandos y  
operaciones

---

**A**

Índice bibliográfico

---

**B**

Glosario, Índice alfabético

## Consignas de seguridad para el usuario

Este manual contiene las informaciones necesarias para la seguridad personal, así como para la prevención de daños materiales. Las informaciones están puestas de relieve mediante señales de precaución. Las señales que figuran a continuación representan distintos grados de peligro:



### Peligro

Significa que, si no se adoptan las medidas preventivas adecuadas, **se producirá** la muerte, lesiones corporales graves o daños materiales considerables.



### Precaución

Significa que, si no se adoptan las medidas preventivas adecuadas, **puede producirse** la muerte, lesiones corporales graves o daños materiales considerables.



### Cuidado

Significa que, si no se adoptan las medidas preventivas adecuadas, pueden producirse lesiones corporales o daños materiales.

### Nota

Se trata de una información importante sobre el producto y su manejo, o sobre una parte determinada de la documentación sobre la que se desea llamar particularmente la atención.

## Personal cualificado

La puesta en funcionamiento y el servicio del equipo sólo deben ser llevados a cabo por **personal cualificado**. En el sentido del manual se trata de personas que disponen de los conocimientos técnicos necesarios para poner en funcionamiento, conectar a tierra y marcar los aparatos, sistemas y circuitos de acuerdo con las normas estándar de seguridad.

## Uso conforme

Considere lo siguiente:



### Precaución

El equipo o los componentes del sistema sólo se podrán utilizar para los casos de aplicación previstos en el catálogo y en la descripción técnica, y sólo en unión de los equipos y componentes de provendencia tercera recomendados y homologados por Siemens.

El funcionamiento correcto y seguro del producto presupone un transporte, un almacenamiento, una instalación y un montaje conforme a las prácticas de la buena ingeniería, así como una operación y un mantenimiento rigurosos.

## Marcas

SIMATIC® , SIMATIC NET® y SIMATIC HMI® son marcas registradas de la SIEMENS AG.

Los restantes nombres y designaciones contenidos en el presente impreso pueden ser marcas registradas cuya utilización por terceros para sus propios fines puede violar los derechos de los propietarios.

### Copyright © Siemens AG 2006 All rights reserved

La divulgación y reproducción de este documento, así como el uso y la comunicación de su contenido, no están autorizados, a no ser que se obtenga el consentimiento expreso para ello. Se reservan todos los derechos, en particular para el caso de concesión de patentes o de modelos de utilidad.

Siemens AG  
Bereich Automation and Drives  
Geschäftsgebiet Industrial Automation Systems  
Postfach 4848, D- 90327 Nuernberg

### Exención de responsabilidad

Hemos comprobado si el contenido de esta publicación concuerda con el hardware y el software descrito. Sin embargo, es posible que se den algunas desviaciones que nos impidan tomar garantía completa de esta concordancia. El contenido de esta publicación está sometido a revisiones regulamente y en caso necesario se incluyen las correcciones en la siguiente edición. Agradecemos sugerencias.

© Siemens AG 2006  
Sujeto a cambios sin previo aviso.

# Indicaciones importantes

<b>Objetivo del manual</b>	<p>Este manual pretende ser una ayuda para todas aquellas personas que deseen cambiar del sistema S5 al S7.</p> <p>Las informaciones que proporciona este manual le servirán para:</p> <ul style="list-style-type: none"><li>• transcribir sus programas S5 a programas S7 mediante un convertidor y retocarlos a mano si es necesario.</li><li>• integrar funciones de S7 ya convertidas (equivalentes a los bloques de función S5 estándar) en los programas S7.</li></ul>
<b>Destinatarios</b>	<p>El presente manual está dirigido a programadores que deseen utilizar programas S5 en sistemas S7.</p>
<b>Ambito de validez</b>	<p>Este manual rige para el software de programación STEP 7 a partir de la versión 4.0.</p>
<b>Asistencia adicional</b>	<p>Si tiene preguntas relacionadas con el uso de los productos descritos en el manual a las que no encuentre respuesta, diríjase a la sucursal o al representante más próximo de Siemens, en donde le pondrán en contacto con el especialista.</p> <p>Encontrará a su persona de contacto en la página de Internet: <a href="http://www.siemens.com/automation/partner">http://www.siemens.com/automation/partner</a></p> <p>Encontrará una guía sobre el conjunto de la información técnica correspondiente a los distintos productos y sistemas SIMATIC en la página de Internet: <a href="http://www.siemens.com/simatic-tech-doku-portal">http://www.siemens.com/simatic-tech-doku-portal</a></p> <p>Encontrará el catálogo y el sistema de pedidos on-line en: <a href="http://mall.automation.siemens.com">http://mall.automation.siemens.com</a></p>
<b>Centro de formación SIMATIC</b>	<p>Para ofrecer a nuestros clientes un fácil aprendizaje de los sistemas de automatización SIMATIC S7, les ofrecemos distintos cursillos de formación. Diríjase a su centro de formación regional o a la central en D 90327 Nürnberg.</p> <p>Teléfono: +49 (911) 895-3200.</p> <p>Internet: <a href="http://www.sitrain.com">http://www.sitrain.com</a></p>

**Technical Support**

Podrá acceder al Technical Support de todos los productos de A&D

- a través del formulario de Internet para el Support Request  
<http://www.siemens.com/automation/support-request>
- Teléfono: + 49 180 5050 222
- Fax: + 49 180 5050 223

Encontrará más información sobre nuestro Technical Support en la página de Internet:

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

**Service & Support en Internet**

Además de nuestra documentación, en Internet le ponemos a su disposición todo nuestro know-how.

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

En esta página encontrará:

- “Newsletter” que le mantendrán siempre al día ofreciéndole informaciones de última hora
- La rúbrica “Servicios online” con un buscador que le permitirá acceder a la información que necesita
- El “Foro” en el que podrá intercambiar sus experiencias con cientos de expertos en todo el mundo
- El especialista o experto de Automation & Drives de su región.
- Bajo la rúbrica ”Servicios” encontrará información sobre el servicio técnico más próximo, sobre reparaciones, repuestos etc.

# Contenido

<b>1ª parte</b>		
<b>1</b>	<b>Introducción</b> .....	<b>1-1</b>
<b>2</b>	<b>Hardware</b> .....	<b>2-1</b>
2.1	Sistemas de automatización .....	2-2
2.2	Módulos S7 .....	2-4
2.2.1	Módulos centrales (CPUs) .....	2-6
2.2.2	Fuentes de alimentación (PS) .....	2-8
2.2.3	Módulos interfase (IM) .....	2-9
2.2.4	Módulos de comunicación (CP) .....	2-10
2.2.5	Módulos de función (FM) .....	2-13
2.2.6	Módulos de señales (SM) .....	2-15
2.2.7	Módulos de simulación (S7-300) .....	2-16
2.3	Periferia descentralizada .....	2-17
2.4	Comunicación .....	2-18
2.4.1	Interface con el programa de usuario .....	2-20
2.5	Manejo y visualización (M+V) .....	2-21
<b>3</b>	<b>Software</b> .....	<b>3-1</b>
3.1	Filosofía de manejo .....	3-1
3.1.1	Requisitos para la instalación .....	3-1
3.1.2	Instalación del software STEP 7 .....	3-2
3.1.3	Arrancar el software STEP 7 .....	3-3
3.2	Estructura de un proyecto S7 .....	3-4
3.3	Elaborar un proyecto con el Administrador SIMATIC .....	3-7
3.3.1	Crear el proyecto .....	3-7
3.3.2	Crear copias de seguridad de los proyectos .....	3-8
3.4	Configurar el hardware en STEP 7 .....	3-9
3.5	Configurar enlaces en la tabla de enlaces .....	3-11
3.6	Insertar y elaborar un programa .....	3-13
3.6.1	Procedimiento general para crear software .....	3-13
3.6.2	Insertar componentes para crear software en programas S7/M7 .....	3-15
3.7	Bloques .....	3-17
3.7.1	Comparativa .....	3-17
3.7.2	Funciones y bloques de función .....	3-18
3.7.3	Bloques de datos .....	3-18
3.7.4	Bloques de sistema .....	3-19
3.7.5	Bloques de organización .....	3-20

3.7.6	Conversión de los bloques .....	3-24
3.8	Ajustes del sistema .....	3-26
3.9	Funciones estándar .....	3-28
3.9.1	Aritmética en coma flotante .....	3-28
3.9.2	Funciones de señales .....	3-28
3.9.3	Funciones integradas .....	3-28
3.9.4	Funciones básicas .....	3-29
3.9.5	Funciones analógicas .....	3-29
3.9.6	Funciones matemáticas .....	3-29
3.10	Tipos de datos .....	3-30
3.11	Areas de operandos .....	3-32
3.11.1	Panorámica .....	3-32
3.11.2	Nuevos operandos de S7: Datos locales .....	3-33
3.12	Operaciones .....	3-35
3.13	Direccionamiento .....	3-39
3.13.1	Direccionamiento absoluto .....	3-39
3.13.2	Direccionamiento simbólico .....	3-39
3.13.3	Novedad: Direccionamiento completo de operandos de datos .....	3-41
3.13.4	Direccionamiento indirecto .....	3-43
<b>2ª parte</b>		
<b>4</b>	<b>Procedimiento .....</b>	<b>4-1</b>
4.1	Analizar el sistema S5 .....	4-2
4.2	Crear un proyecto S7 .....	4-4
4.3	Configurar el hardware .....	4-4
<b>5</b>	<b>Preparando la conversión .....</b>	<b>5-1</b>
5.1	Preparar los archivos necesarios .....	5-2
5.2	Comprobar los operandos .....	5-3
5.3	Preparar el programa S5 .....	5-4
5.4	Crear macros .....	5-5
5.4.1	Macros de instrucciones .....	5-6
5.4.2	Macros de OBs .....	5-7
5.4.3	Editar macros .....	5-8
<b>6</b>	<b>Conversión .....</b>	<b>6-1</b>
6.1	Iniciar la conversión .....	6-1
6.2	Archivos generados .....	6-5
6.3	Evaluar mensajes de error .....	6-8

<b>7</b>	<b>Retocar el programa convertido</b> .....	<b>7-1</b>
7.1	Cambiar direcciones .....	7-2
7.1.1	Métodos para cambiar las direcciones .....	7-2
7.2	Funciones no convertibles .....	7-3
7.3	Direccionamiento indirecto: conversión .....	7-4
7.4	Accesos directos a la memoria .....	7-5
7.5	Asignación de parámetros .....	7-5
7.6	Funciones estándar .....	7-6
<b>8</b>	<b>Compilar</b> .....	<b>8-1</b>
<b>9</b>	<b>Ejemplo de aplicación</b> .....	<b>9-1</b>
9.1	Tratamiento de valores analógicos .....	9-2
9.2	Datos locales temporales .....	9-5
9.3	Evaluar la información de arranque del OB de alarma de diagnóstico (OB 82) .....	9-9
9.4	Transferencia en bloque .....	9-12
9.5	Llamada de los ejemplos .....	9-15
	<b>Anexos</b>	
<b>A</b>	<b>Listas de operandos y operaciones</b> .....	<b>A-1</b>
A.1	Operandos .....	A-1
A.2	Operaciones .....	A-3
<b>B</b>	<b>Índice bibliográfico</b> .....	<b>B-1</b>
	<b>Glosario</b> .....	<b>Glosario-1</b>
	<b>Índice alfabético</b> .....	<b>Índice-1</b>





**1ª parte:**  
**Planificar la transición**

---

Introducción

---

**1**

---

Hardware

---

**2**

---

Software

---

**3**



Hasta ahora el nombre SIMATIC se utilizaba como sinónimo de nuestro autómatas programables S5, pero entre tanto el SIMATIC ha pasado a definir la Integración Total en Automatización.

El concepto de **Integración Total en Automatización** describe un método revolucionario que unifica el mundo de la industria manufacturera y la de procesos, integrando todos los componentes de hardware y software en un único sistema: SIMATIC.

La integración total se hace posible gracias a la triple homogeneidad que ofrece el sistema S7:

- En la gestión de datos  
Los datos se introducen una sola vez y son accesibles desde cualquier punto de la fábrica. Con ello se acabaron los errores de transferencia y los problemas de coherencia de los datos.
- En la configuración y programación  
Todos los componentes de una solución de automatización se configuran, se programan, se ponen en servicio, se comprueban y supervisan con un único software totalmente integrado y modular - sobre una misma plataforma y con la herramienta apropiada para cada tarea.
- En la comunicación  
La cuestión del "Quién se conecta con Quién" se define cómodamente en una tabla de enlaces, pudiéndose modificar en cualquier momento desde cualquiera de los interlocutores. Las diferentes redes se configuran de forma sencilla y unificada.

Para poder describir esta nueva filosofía del SIMATIC como sistema totalmente integrado, hemos tenido que introducir nuevos conceptos en el SIMATIC S7. Así, por ejemplo, algunas funciones están implementadas de otra manera que en el S5.

También en el tema del software de programación STEP 7 hemos apostado por una tecnología punta y por las ideas más avanzadas. Por ejemplo, hemos diseñado todo el interface de usuario bajo Windows 95/NT y siguiendo los criterios ergonómicos más modernos. En cuanto a los lenguajes de programación, hemos dado mucha importancia al cumplimiento casi estricto de la norma IEC 1131, sin por ello renunciar a la compatibilidad con STEP 5.

Estamos convencidos de haber cumplido en gran medida con lo que se debe exigir a STEP 7:

- Software estándar para la Integración Total en Automatización
- Programación según la norma IEC 1131
- Compatibilidad con STEP 5

También somos conscientes de que la transición de un sistema ya consolidado a uno nuevo crea nuevas preguntas y - especialmente en el software - requiere determinadas adaptaciones.

La presente documentación pretende responderle a estas preguntas y, al mismo tiempo, enseñarle un método sencillo para seguir utilizando los programas creados con STEP 5 en el mundo de los S7.

También es importante mencionar que por motivos de claridad se han realizado cambios terminológicos en S7 con respecto al S5. Por ejemplo, para evitar confusiones entre módulos software y hardware, en STEP 7 los módulos software se denominan "Bloques" que es además el nombre utilizado en las normas. Las "tarjetas" de S5 se denominan "módulos" en S7. Tenga en cuenta que el presente manual ya utiliza la nueva terminología del S7.

# Hardware

# 2

Este capítulo describe el hardware que se puede utilizar en el S7 y, en caso necesario, se compara con el hardware de S5, para facilitarle la transición del S5 al S7.

## **Del hardware S5 al S7 con el catálogo electrónico de Siemens en CD-ROM**

Para cambiar del S5 al S7 en lo que respecta al hardware está disponible en el CD-ROM “Komponenten für die Automation” / Catálogo CA01 (desde 04/97) una aplicación que encontrará en el catálogo de productos (Produktkatalog) bajo el comando de menú **Auswahlhilfen > Simatic**. Aquí puede introducir un equipo S5 cualquiera. La aplicación creará una configuración del hardware y una lista de señales a partir de los datos indicados. Seguidamente puede convertir la configuración recién creada en una configuración S7.

## 2.1 Sistemas de automatización

SIMATIC S7 se compone de tres gamas de sistemas de automatización que se distinguen por sus prestaciones.

### SIMATIC S7-200

El SIMATIC S7-200 es un Micro PLC compacto para la gama inferior de prestaciones. Este autómatas dispone de un paquete de software propio que no ha sido incluido en la presente guía de transición, ya que las características del sistema S7-200 no permiten realizar una conversión mediante software.

### SIMATIC S7-300

El SIMATIC S7-300 es el miniautómata modular para la gama baja.

### SIMATIC S7-400

El SIMATIC S7-400 cubre la gama media, alta y superior.

Para orientarse mejor, recuerde que los nombres de los módulos S7-300 comienzan con un 3, y los de los módulos S7-400 con un 4.

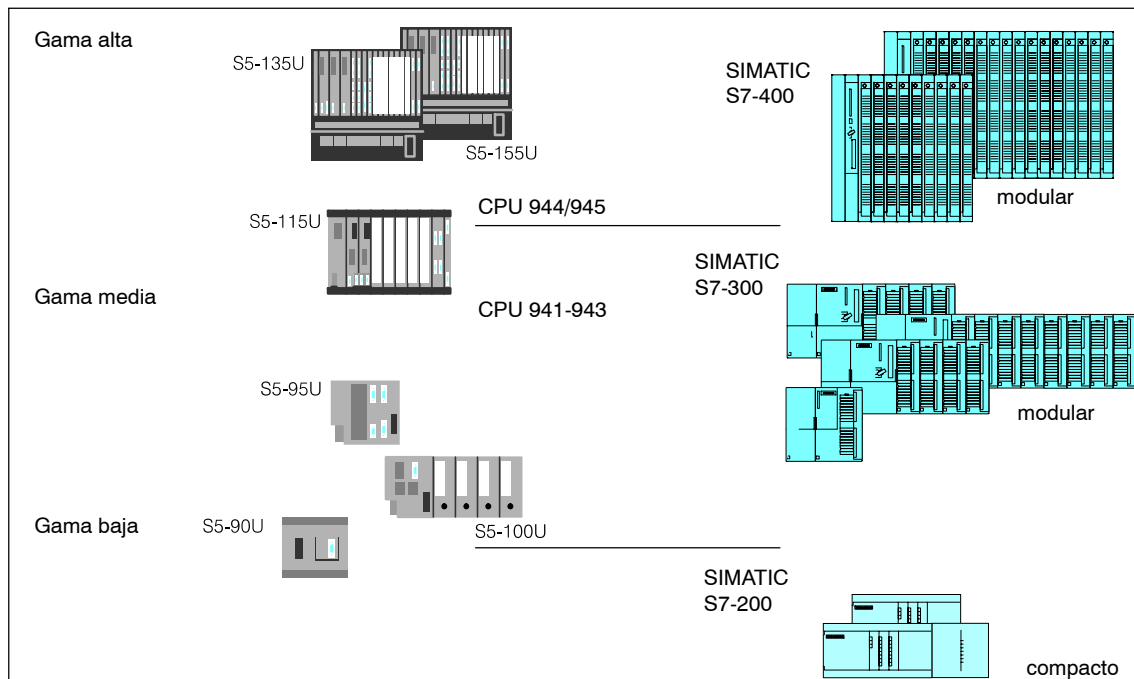


Figura 2-1 Sistemas de automatización SIMATIC

## Conexión de PG y OP al SIMATIC S7

### Interface de programación MPI (Multi Point Interface) para PG y OP

El puerto de programación AS511 del SIMATIC S5 ha sido sustituido por el interface multipunto (para S7-300 y S7-400). El interface MPI sirve para realizar una conexión eléctrica directa entre los aparatos HMI (HMI: Human Machine Interface, anteriormente denominado COROS) y las unidades de programación a través del interface de programación del SIMATIC S7. Los interfaces están integrados físicamente.

La tabla siguiente muestra una comparativa de los interfaces.

AS511	MPI
Interface TTY 25 polos TTY (20 mA)	Conexión subminiatura 9 polos con RS485
Velocidad de transferencia: 9,6 kbaudios	Velocidad de transferencia: 187,5 kbaudios
Protocolo: 3964R	Protocolo: funciones S7
	Extensión de la red: 50 m con repetidores de bus o cables especiales de hasta 1000 m Todos los módulos programables de la configuración son accesibles a través de MPI.
Permite conectar un aparato	Permite conectar hasta 31 aparatos

### Interface de bus para OP

A través del sistema de red en bus **PROFIBUS** (antes denominado SINEC L2) se pueden conectar autómatas programables/sistemas de automatización de las familias SIMATIC S5 y SIMATIC S7. La conexión depende del bus.

## 2.2 Módulos S7

### **No hay cambios fundamentales con respecto al S5**

La gama de módulos del sistema S7 sigue en la línea del ya conocido y acreditado concepto del SIMATIC S5. Esta gama se irá ampliando y completando progresivamente.

Para el S7 se dispone de los tipos de módulos siguientes:

- módulos centrales (CPU),
- puentes de alimentación (PS),
- módulos interfase (IM),
- módulos de comunicación CP; (p. ej.: para conexión al PROFIBUS),
- módulos de función FM; (p. ej.: para contaje, posicionamiento y regulación),
- los módulos digitales y analógicos se denominan ahora módulos de señales (SM).

En este capítulo se explican las características comunes y las diferencias entre las gamas de módulos SIMATIC S5 y SIMATIC S7.

### **Nuevas prestaciones**

Los módulos STEP 7 se distinguen por estas nuevas prestaciones:

- Los módulos ya no disponen de puentes e interruptores.
- Todos los módulos funcionan sin ventilador y tienen el mismo grado de protección IP 20 que el S5.
- Existen módulos parametrizables y módulos con funciones de diagnóstico.
- La ocupación de los slots del S7 es más flexible que la del S5.
- Los aparatos de ampliación y los sistemas de periferia descentralizada ET 200 pueden disparar alarmas.



**Parametrización de los módulos/tarjetas S5/S7** La tabla siguiente muestra una comparativa de la parametrización en SIMATIC S5 y SIMATIC S7:

<b>SIMATIC S5</b>	<b>SIMATIC S7</b>
	Los módulos se disponen en un bastidor (tabla de configuración) con la herramienta de configuración de hardware (HW-Config) de STEP 7.
Las direcciones se ajustan con interruptores DIL	Las direcciones se ajustan para cada slot con la herramienta de configuración de hardware de STEP 7.
El comportamiento del sistema se ajusta con interruptores DIL	Los módulos se parametrizan con la herramienta de configuración del hardware de STEP 7.
El funcionamiento de los módulos centrales se parametriza a través de las áreas de datos de sistema o del DB 1 / DX 0	La CPU se parametriza con la herramienta de configuración de hardware de STEP 7.
	Los datos de configuración se compilan y transfieren a la CPU; los parámetros de los módulos se transfieren automáticamente durante el arranque.

## 2.2.1 Módulos centrales (CPUs)

### CPUs del S7-300

La tabla 2-1 muestra las características más importantes de las distintas CPUs del S7-300. Si desea sustituir una CPU S5, utilice la tabla siguiente para comparar sus características y así elegir la CPU apropiada.

Tabla 2-1 Características de las CPUs S7-300

Característica	312 IFM	313	314	314 IFM	315	315-2 DP
Memoria de trabajo (integrada)	6 kbytes	12 kbytes	24 kbytes	24 kbytes	48 kbytes	
Memoria de carga	20 kbytes RAM; 20 kbytes EEPROM	20 kbytes RAM	40 kbytes RAM	40 kbytes RAM; 40 kbytes EEPROM	80 kbytes RAM	
• integrada						
• ampliable con Memory Card	-	hasta 512 kbytes	hasta 512 kbytes	-	hasta 512 kbytes (en la CPU hasta 256 kbytes programables)	
Tamaño de la imagen del proceso; entradas y salidas, respectivamente	32 bytes + 4 on-board	128 bytes	128 bytes	124 bytes + 4 on-board	128 bytes	
Area de direccionamiento de la periferia	Entradas: 128 + 10 on-board Salidas: 128 + 6 on-board	128	512	Entradas: 496 + 20 on-board Salidas: 496 + 16 on-board	1024	
• Entradas/salidas digitales						
• Entradas/salidas analógicas	32		64	Entradas: 64 + 4 on-board Salidas: 64 + 1 on-board	128	
Marcas	1024	2048				
Contadores	32	64				
Temporizadores	64	128				
Suma máxima de los datos remanentes	72 bytes		4736 bytes	144 bytes	4736 bytes	
Datos locales	512 bytes en total; 256 bytes por prioridad	1536 bytes en total; 256 bytes por prioridad				
Bloques:						
OBs	3	13	13	13	13	14
FBs	32	128	128	128	128	128
FCs	32	128	128	128	128	128
DBs	63	127	127	127	127	127
SFCs	25	44	48	48	48	53
SFBs	2	7	7	14	7	7

**Las CPUs del S7-400**

Las CPUs del S7-400 se distinguen por sus características. La tabla 2-2 muestra una comparativa de las características de estas CPUs.

Tabla 2-2 Características de las CPUs del S7-400

Características	CPU 412-1	CPU 413-1	CPU 413-2 DP	CPU 414-1	CPU 414-2 DP	CPU 416-1	CPU 416-2 DP
Memoria de trabajo integrada	48 kbytes	72 kbytes		128 kbytes	128/384 kbytes	512 kbytes	0,8/1,6 Mbytes
Memoria de carga <ul style="list-style-type: none"> <li>integrada</li> <li>ampliable con Memory Card</li> </ul>	8 kbytes hasta 15 Mbytes			8 kbytes hasta 15 Mbytes		16 kbytes hasta 15 Mbytes	
Tamaño de la imagen del proceso; entradas y salidas, respectivamente	128 bytes			256 bytes		512 bytes	
Area de direccionamiento de la periferia <ul style="list-style-type: none"> <li>Entradas/salidas digitales máx.</li> <li>Entradas/salidas analógicas máx.</li> </ul>	2 kbytes 16384 1024			8 kbytes 65536 4096		16 kbytes 131072 8192	
Marcas	4096 M 0.0 hasta M 511.7			8192 M 0.0 hasta M 1023.7		16384 M 0.0 hasta M 2047.7	
Contadores	256 Z 0 hasta Z 255			256 Z 0 hasta Z 255		512 Z 0 hasta Z 511	
Temporizadores	256 T 0 hasta T 255			256 T 0 hasta T 255		512 T 0 hasta T 511	
Datos locales	4 Kbytes en total			8 Kbytes en total		16 Kbytes en total	
Bloques:							
OBs	23			31		44	
FBs	256			512		2048	
FCs	256			1024		2048	
DBs	511			1023		4095	
SFBs	24			24		24	
SDBs	512			512		512	
SFCs	55	55	58	55	58	55	58

**Remanencia en el S7-400**

Los módulos centrales del SIMATIC S7-400 requieren una pila de respaldo para memorizar los temporizadores, contadores y marcas.

**Remanencia sin pila de respaldo en el S7-300**

El S7-300 no necesita ninguna pila adicional en el módulo central para respaldar temporizadores, contadores y marcas. De igual modo, tampoco se pierde el contenido de los bloques de datos en caso de que se corte la alimentación eléctrica. Los módulos centrales del SIMATIC S7-300 disponen de un sistema de backup libre de mantenimiento que, en caso de un corte de alimentación, memoriza los operandos y datos parametrizados de forma remanente.

El número y el tamaño de las áreas remanentes dependen de la CPU utilizada.

**Parametrizar la remanencia**

Las áreas remanentes se definen en los diálogos de parametrización de la herramienta de configuración de hardware (HW-Config) de STEP 7.

## 2.2.2 Fuentes de alimentación (PS)

Para cada sistema de automatización están disponibles distintas fuentes de alimentación.

### Fuentes de alimentación del S7-300

Para la alimentación eléctrica de la CPU se puede utilizar cualquier red eléctrica de 24V (industrial).

La gama de módulos del sistema S7 incluye las siguientes fuentes de alimentación, las cuales han sido construidas especialmente para el S7-300:

Denominación	Intensidad de salida	Tensión de salida	Tensión de entrada
PS 307	2A	DC 24V	AC 120V / 230V
PS 307	5A	DC 24V	AC 120V / 230V
PS 307	10A	DC 24V	AC 120V / 230V

### Fuentes de alimentación del S7-400

Denominación	Intensidad de salida	Tensión de salida	Tensión de entrada
PS 407 4A	4A 0,5A	DC 5V DC 24V	AC 120V / 230V
PS 407 10A	10A 1A	DC 5V DC 24V	AC 120V / 230V
PS 407 20A	20A 1A	DC 5V DC 24V	AC 120V / 230V
PS 405 4A	4A 0,5A	DC 5V DC 24V	DC 24V
PS 405 10A	10A 1A	DC 5V DC 24V	DC 24V
PS 405 20A	20A 1A	DC 5V DC 24V	DC 24V

Para más información consulte los manuales de referencia [/71/](#) y [/101/](#).

### 2.2.3 Módulos interfase (IM)

Para algunos módulos interfase que están disponibles en el sistema S5 existen módulos equivalentes en el S7, principalmente para funciones de conexión local. Para realizar conexiones remotas en S7 se recomienda enviar las señales a través de la red PROFIBUS.

#### Comparativa de los módulos IM

Módulo S5	Módulo S7-300	Módulo S7-400	Descripción
IM 305 IM 306 IM 300 / IM 312	IM 365 IM 360 / IM 361	IM 460-0 / IM 461-0 IM 460-1 / IM 461-1	Configuración centralizada
-	-	IM 460-3 / IM 461-3	Conexiones remotas (hasta 100 m)
IM 301 / IM 310	Conexión vía PROFIBUS	Conexión vía PROFIBUS	Conexión de módulos periféricos y módulos preprocesadores de señales (hasta 200 m)
IM 304 / IM 314	Conexión vía PROFIBUS	Conexión vía PROFIBUS	Uso de periferia descentralizada remota (hasta 600 m)
	-	IM 463-2	Conexión descentralizada de aparatos de ampliación S5 remota (hasta 600 m)
IM 307 / IM 317	Conexión vía PROFIBUS	Conexión vía PROFIBUS	Conexión vía cable de fibra óptica (hasta 1500 m)
IM 308 / IM 318	Conexión vía PROFIBUS	Conexión vía PROFIBUS	Distancias de hasta 3000 m

Alternativamente al módulo interfase IM 308C, en S7 se puede utilizar el IM 467.

Los módulos digitales y analógicos S5 se pueden conectar a un bastidor S7 con un IM 314 a través de un bastidor de ampliación S5 con un IM 463-2.

#### Bastidores de ampliación S5 conectables

Se pueden conectar los siguientes bastidores de ampliación S5:

- EG 183
- EG 185
- ER 701-2
- ER 701-3

## 2.2.4 Módulos de comunicación (CP)

A continuación figura una lista de los módulos de comunicación S5 y S7 que se pueden conectar a las distintas subredes. Además se indican los servicios que soporta cada uno de dichos módulos.

### Subredes del mundo SIMATIC

Para satisfacer las distintas exigencias de los niveles de automatización (control central, célula, campo, actuador/sensor) SIMATIC ofrece las subredes siguientes:

- **AS-Interface**

El AS-Interface o Actuador-Sensor-Interface es un sistema de conexión para el nivel inferior de la pirámide de automatización, el nivel de campo. Sirve especialmente para interconectar sensores y actuadores binarios. La cantidad de datos está limitada a un máximo de 4 bits por esclavo.

- **MPI**

La subred MPI se utiliza para los niveles de campo y de célula de poca extensión. La red MPI está constituida por un interface multipunto integrado en los SIMATIC S7/M7 y C7 y está concebida como puerto de programación de PG, así como para interconectar un número reducido de CPUs que intercambien pequeñas cantidades de datos (hasta 70 bytes).

- **PROFIBUS**

PROFIBUS constituye la red para el nivel de campo y de célula en el sistema de comunicación abierto y no propietario de SIMATIC. PROFIBUS es especialmente apropiado para una transferencia rápida de cantidades medianas de datos (aproximadamente 200 bytes).

- **Industrial Ethernet**

Industrial Ethernet es la red para el nivel de control central y el de célula en el sistema de comunicación abierto y no propietario de SIMATIC. Industrial Ethernet es especialmente apropiado para una transferencia rápida de grandes cantidades de datos.

- **Conexión punto a punto**

Una conexión punto a punto no es una subred en el sentido tradicional de la palabra. Dicha conexión se realiza en SIMATIC a través de procesadores de comunicación punto a punto (CP), conectándose dos interlocutores (PLC, Scanner, PC, etc.).

**AS-Interface  
(SINEC S1)**

La tabla siguiente muestra los módulos disponibles para la comunicación a través del AS-Interface.

Módulo S5	Módulo S7-300	Módulo S7-400
CP 2433 (funciones AS-i) CP 2430 (funciones AS-i)	CP 342-2 (funciones AS-i)	-

**MPI (SINEC L1)**

La comunicación que se realizaba en S5 a través de SINEC L1 ha sido sustituida en S7 por la comunicación de datos globales a través de la red MPI.

Todas las CPUs del S7-300 y S7-400, así como las PGs y los OPs, poseen un interface MPI.

**PROFIBUS  
(SINEC L2)**

La tabla siguiente muestra los módulos disponibles para la comunicación a través de la red PROFIBUS e indica los servicios soportados por dichos módulos.

Módulo S5	Módulo S7-300	Módulo S7-400
CP5431 (FMS, FDL, DP) CPU 95U (FDL, DP *)	CP 342-5 (funciones S7, FDL, DP) CP 343-5 (funciones S7, FDL, FMS)	CP 443-5 Ext. (funciones S7, FDL, DP) CP 443-5 Basic (funciones S7, FDL, FMS)
IM 308-B/C (DP)	CPU 315-2 DP (DP)	CPU 413-2 DP (DP) CPU 414-2 DP (DP) CPU 416-2 DP (DP) IM 467 (DP)

\*) depende de la versión pedida

**Industrial Ethernet  
(SINEC H1)**

La tabla siguiente muestra los módulos disponibles para la comunicación a través de Industrial Ethernet y los servicios soportados por dichos módulos.

Módulo S5	Módulo S7-300	Módulo S7-400
CP1430 TF (transporte ISO)	CP 343-1 (funciones S7, transporte ISO)	CP 443-1 (funciones S7, transporte ISO)
CP 1430 TCP (ISO on TCP)	CP 343-1 TCP (funciones S7, ISO on TCP)	CP 443-1 TCP (funciones S7, ISO on TCP)

**Conexión punto a punto**

La tabla siguiente muestra los módulos disponibles para conexiones punto a punto e indica los servicios que soportan dichos módulos.

<b>Módulo S5</b>	<b>Módulo S7-300</b>	<b>Módulo S7-400</b>
CP 521 (3964 (R), ASCII) CP 523 (3964 (R), ASCII)	CP 340-RS 232C (3964 (R), ASCII) CP 340-20 mA (3964 (R), ASCII) CP 340-RS 422/485 (3964 (R), ASCII)	CP 441-1 (3964 (R), RK512, ASCII)
CP 544 (3964 (R), RK 512, ASCII)	-	
CP 524/525 (3964 (R), RK 512, ASCII, drivers especiales para carga a posteriori) CP 544 B (3964 (R), RK 512, ASCII, drivers especiales para carga a posteriori)	-	CP 441-2 (3964 (R), RK512, ASCII, drivers especiales para carga a posteriori)



## 2.2.5 Módulos de función (FM)

Para algunos módulos IP y WF existe la posibilidad de integrarlos en las cápsulas de adaptación del S7-400. En otro caso hay nuevos módulos de función disponibles para el S7 que ofrecen las mismas funciones.

La tabla siguiente muestra una comparativa de los módulos preprocesadores de señales del S5 y del S7.

Tabla 2-3 Comparativa de los módulos preprocesadores de señales del S5 y del S7

Módulo S5	Cápsula de adaptación	Módulo S7	Descripción
IP 240	sí	FM 451 (limitadamente)	Módulo de contaje, de lectura de recorrido y de posicionamiento
IP 241	no	FM 451 / FM 452 (limitadamente)	Módulo digital de lectura de recorrido
IP 242A	no	no	Módulo contador
IP 242B	sí	no	Módulo contador
IP 244	sí	FM 455	Módulo de regulación
IP 246I/A	sí	FM 354 / FM 357 / FM 453	Módulo de posicionamiento para accionamientos de velocidad variable
IP 247	sí	FM 353 / FM 357 / FM 453	Módulo de posicionamiento para motores paso a paso
IP 252	no	FM 455 (limitadamente)	Módulo regulador
IP 260	no	FM 355 (limitadamente)	Módulo regulador
IP 261	no	no	Módulo dosificador
IP 281	no	FM 350-1 / FM 450-1	Módulo contador
IP 288	no	FM 451 / FM 452	Módulo de posicionamiento para control de ejes mecánicos en accionamientos de marcha lenta/ rápida y para control de levas
WF 705	sí	FM 451 (limitadamente)	Módulo de lectura de recorrido
WF 706	no	FM 451 (limitadamente)	Módulo de posicionamiento y contaje
WF 707	no	FM 452 (limitadamente)	Leva electrónica

Tabla 2-3 Comparativa de los módulos preprocesadores de señales del S5 y del S7, continuación

<b>Módulo S5</b>	<b>Cápsula de adaptación</b>	<b>Módulo S7</b>	<b>Descripción</b>
WF 721	sí	FM 354 (limitadamente por motivos de ensamblaje)	Módulo de posicionamiento
WF 723A	sí	FM 453	Módulo de posicionamiento
WF 723 B	sí	FM 357 (limitadamente por motivos de ensamblaje)	Módulo de posicionamiento
WF 723 C	sí	no	Módulo de posicionamiento
-	-	FM 456-4	Módulo tecnológico personalizable (M7-FM)
-	-	SINUMERIK FM-NC	Control NC
-	-	FM STEPDRIVE	Control de motores paso a paso
-	-	SIMOSTEP	Motor paso a paso

## 2.2.6 Módulos de señales (SM)

Los módulos de señales del SIMATIC S7 son comparables a las tarjetas de entrada y salida del S5. Sin embargo, además de los módulos de señales sencillos, también existen módulos parametrizables y módulos con funciones de diagnóstico.

### SMs parametrizables

Para los módulos de entradas digitales, la herramienta de configuración del hardware de STEP 7 ofrece, por ejemplo, la posibilidad de ajustar qué canales deberán disparar una alarma de proceso cuando se produzca un cambio de flanco.

Las áreas de entrada de los módulos de entradas analógicas se parametrizan cómodamente con el software STEP 7.

### SMs con funciones de diagnóstico

Los módulos con funciones de diagnóstico detectan tanto fallos/averías externos - la rotura de un hilo o un cortocircuito externo - como errores internos - errores de la RAM o un cortocircuito interno de un módulo.

El autómata “trata” los eventos de diagnóstico de dos maneras distintas:

- o bien se dispara una alarma de diagnóstico (a continuación se llama al OB correspondiente en el programa de usuario para que interrumpa el programa cíclico)
- o bien se registra una entrada en el búfer de diagnóstico de la CPU (que se puede leer con una PG o un equipo de manejo y visualización (M+V)).

Las tablas siguientes muestran los módulos de señales del S7:

Tabla 2-4 Módulos de señales del SIMATIC S7-300

DI (SM 321)	DO (SM 322)	AI (SM 331)	AO (SM 332)
32 x DC 24V	32 x DC 24V/0,5A	8 x 12 bits	2 x 12 bits
16 x DC 24V	16 x DC 24V/0,5A	2 x 12 bits	
16 x DC 24V con alarma de proceso y de diagnóstico	8 x DC 24V/0,5A con alarma de diagnóstico	Ex: 4 x 15 bits	Ex: 4 x 15 bits
16 x DC 24V tipo M	8 x DC 24V/2A	Ex: 12 x 15 bits	
8 x AC 120V/230V	8 x AC 120V/230V/2A	AI 4/AO 2 X 8/8 bits (SM 334)	
Ex: 4 x DC 24V	Ex: 4 x DC 15V/20mA		
	Ex: 4 x DC 24V/20mA		

Tabla 2-5 Módulos de señales del SIMATIC S7-400

DI (SM 421)	DO (SM 422)	AI (SM 431)	AO (SM 432 )
32 x DC 24V	32 x DC 24V/0,5A	8 x 13 bits	8 x 13 bits
16 x UC 24V/60V con alarma de proceso y de diagnóstico	16 x DC 24V/2A	8 x 14 bits (para medir temperaturas)	
16 x UC 120V/230V	16 x AC 120V/230V/ 5A	8 x 14 bits	
32 x UC 120V	16 x AC 120V/230V/ 2A	16 x 16 bits	
	16 x UC 30V/230V/ Rel 5A		

### 2.2.7 Módulos de simulación (S7-300)

Para probar el programa de usuario, en el S7-300 está disponible el módulo de simulación SM 374.

El módulo de simulación tiene las siguientes características:

- Simulación de
  - 16 entradas o
  - 16 salidas o
  - 8 entradas y 8 salidas  
(¡con las mismas direcciones iniciales, respectivamente!)
- Función ajustable mediante destornillador
- Indicadores de estado (LEDs) para simulación de entradas o salidas

## 2.3 Periferia descentralizada

Los módulos de periferia descentralizada del sistema ET 200 que se utilizaban en el SIMATIC S5 se pueden seguir utilizando en el SIMATIC S7.

La gama de módulos ha sido ampliada con nuevos módulos ET 200.

### Maestro DP

Los siguientes módulos pueden ser **maestro DP** del sistema de periferia descentralizada:

- S7-300 con CPU 315-2 DP o CP 342-5 como maestro DP
- S7-400 con CPU 413-2 DP / 414-2 DP / 416-2DP o CP 443-5 Extended como maestro DP

### Esclavos DP

Los **esclavos DP** del sistema de periferia descentralizada pueden ser, por ejemplo:

- Sistemas de periferia descentralizada ET 200B, ET 200C, ET 200M, ET 200X (hasta 12 Mbaudios) y ET 200U, ET 200L (hasta 1,5 Mbaudios)
- Automatas programables, como p. ej.:
  - S5-115U, S5-135U o S5-155U con IM 308-C como esclavo DP
  - S5-95U con interface de esclavo DP (hasta 1,5 Mbaudios)
  - S7-300 con CPU 315-2 DP o CP 342-5 como esclavo DP
  - S7-400 con CP 443-5 como esclavo DP
- Interface para el Actuator-Sensor-Interface con el DP/AS-i Link
- Visualizadores de textos y paneles de operador para manejo y visualización a pie de máquina
- Sistemas de identificación MOBY
- Aparatos de baja tensión
- Aparatos de campo de Siemens o de terceros como p. ej. accionamientos, grupos de válvulas, etc.

### Maestros FMS

**Maestros FMS** pueden ser:

- S7-300 con CP 343-5 como maestro FMS
- S7-400 con CP 443-5 Basic como maestro FMS

### Esclavos FMS

Los **esclavos FMS** pueden ser, por ejemplo, estaciones ET 200U o aparatos de mando y protección de motores SIMOCODE.

Para más información consulte los manuales correspondientes o el catálogo de Siemens CA01.

## 2.4 Comunicación

### Servicios y subredes

La comunicación del SIMATIC S7 se basa en distintas subredes que ofrecen los más variados servicios.

Servicios	Funciones de comunicación S7 (funciones S7)		
	transporte ISO ISO-on-TCP	FDL (SDA) FMS DP	GD
Subredes	Industrial Ethernet	PROFIBUS	MPI

A continuación figura un resumen de los servicios de comunicación utilizados en el SIMATIC.

### Funciones S7

Las funciones S7 ofrecen servicios de comunicación para CPUs S7/M7, OPs S7/Oss y PCs. Las funciones S7 ya están integradas en todos los sistemas SIMATIC S7/M7. Ya que las funciones S7 corresponden a un servicio del nivel de transporte ISO, no dependen de la subred y se pueden utilizar en todas las subredes (MPI, PROFIBUS, Industrial Ethernet).

### Transporte ISO

Estas funciones cumplen la tarea de asegurar la transferencia de datos desde un SIMATIC S7 a un SIMATIC S5.

Las funciones ISO sirven para transferir cantidades medianas de datos (hasta 240 bytes) en el nivel número 4 del modelo de referencia ISO/OSI para comunicación abierta en Industrial Ethernet.

### ISO-on-TCP

Estas funciones cumplen la tarea de asegurar la transferencia de datos desde un SIMATIC S7 a un SIMATIC S5.

Las funciones ISO-on-TCP permiten transferir cantidades medianas de datos (hasta 240 bytes) a través del protocolo TCP/IP en el nivel 4 del modelo de referencia ISO/OSI para comunicación abierta.

El servicio ISO-on-TCP requiere el estándar ampliado RFC1006.

### FDL (SDA)

Estas funciones cumplen la tarea de asegurar la transferencia de datos desde un SIMATIC S7 a un SIMATIC S5.

Las funciones FDL permiten transferir cantidades medianas de datos (hasta 240 bytes) en el nivel 2 Fieldbus Data Link (FDL) del modelo de referencia ISO para comunicación abierta en Industrial Ethernet.

- FMS** PROFIBUS FMS (Fieldbus Message Specification) ofrece servicios para transferir datos estructurados (variables FMS) a través de enlaces FMS.
- El servicio FMS corresponde al nivel 7 del modelo de referencia ISO. Este servicio cumple la norma europea EN 50170 vol. 2 PROFIBUS y ofrece servicios para transferir datos estructurados (variables).
- DP** Los servicios PROFIBUS-DP permiten una comunicación transparente con la periferia descentralizada. El programa de control accede a la periferia descentralizada de igual modo que a la periferia centralizada.
- GD** La comunicación de datos globales (GD) es una función de comunicación sencilla que está integrada en el sistema operativo de las CPUs de los sistemas S7-300/400.
- La comunicación GD permite un intercambio cíclico de datos. En el caso del S7-400, el intercambio entre las CPUs puede controlarse incluso por eventos y realizarse a través del interface MPI.

### 2.4.1 Interface con el programa de usuario

El interface de comunicación con el programa de usuario está constituido por los siguientes bloques:

- SFCs (sin configuración de enlaces)
- SFBs (con configuración de enlaces) (sólo S7-400)
- FCs / FBs cargables

Estos bloques sustituyen a los bloques de manejo S5, los denominados módulos de manejo. La funcionalidad es similar, si bien utilizan las convenciones sintácticas de STEP 7. Si dispone de un programa S5 con funciones de manejo tiene que adaptarlo a los nuevos bloques para realizar la comunicación.

Red	Servicio	Interface del programa de usuario S5	Interface del programa de usuario S7
Conexión punto a punto	-	Bloques de manejo *	S7-300: FBs cargables S7-400: SFBs cargables
PROFIBUS	FDL (AG - AG) Nivel libre 2 FMS	Bloques de manejo * Bloques de manejo * Bloques de manejo *	FCs cargables - FBs cargables
Industrial Ethernet	ISO 4 ISO 4 + AP STF  MAP	Bloques de manejo * Bloques de manejo * Bloques de manejo * + FBs cargables Bloques de manejo * + FBs cargables	FCs cargables - -  FBs cargables

\* Bloque integrado o cargable que depende de la CPU utilizada.



## 2.5 Manejo y visualización (M+V)

**Introducción** Los apartados siguientes explican en qué medida es posible utilizar los paneles de operador SIMATIC HMI (HMI: Human Machine Interface, antes: COROS) en el SIMATIC S7.

**Paneles de operador (OP)** Los paneles de operador (Operator Panels) SIMATIC HMI ofrecen funciones de manejo y visualización M+V para SIMATIC S5, SIMATIC S7 y SIMATIC TI (además de otros controladores).

**STEP 5** Para integrar paneles de operador **OP SIMATIC en SIMATIC S5** rige lo siguiente: en el autómatas programable tiene que haber un bloque de función estándar que se habrá de llamar en función del OP que se haya conectado.

Con el S5 se pueden utilizar los siguientes OPs:

- TD17, OP5/A1, OP7/PP, OP7/DP-12, OP15/x1, OP17/PP, OP17/DP-12
- OP25, OP35, OP37, TP37

**STEP 7** En lo que respecta a la conexión **de OPs SIMATIC a SIMATIC S7/M7** cabe distinguir entre las conexiones PPI, MPI y PROFIBUS (como estaciones MPI).

Las conexiones PPI y MPI se conducen a través del puerto de programación PG de la CPU. Los OPs SIMATIC utilizan los servicios de comunicación del SIMATIC S7/M7 (funciones S7), por lo que no se requiere ningún bloque de función estándar.

Una conexión PROFIBUS entre un OP SIMATIC y un SIMATIC S7/M7 establecerá una comunicación basada en funciones S7; ¡aquí tampoco se requiere ningún FB estándar! (los OPs SIMATIC son “estaciones activas” y no esclavos PROFIBUS-DP como en la conexión del SIMATIC S5 al PROFIBUS). En cuanto al número de estaciones admite la misma cantidad que en el caso de una conexión MPI.

Con el S7 se pueden utilizar los siguientes OPs:

- TD17, OP3, OP5/A2, OP7/DP, OP7/DP-12, OP15/x2, OP17/DP, OP17/DP-12
- OP25, OP35, OP37, TP37

Para la conexión de los OPs SIMATIC existen las siguientes limitaciones:

- OP3: máx. 2 conexiones
- OP5/15/25: máx. 4 conexiones
- TD17, OP7/17: máx 4 conexiones
- OP35: máx. 6 conexiones
- OP37, TP37: máx. 8 conexiones

**Configuración**

SIMATIC ProTool y SIMATIC ProTool/Lite son dos modernas herramientas para configurar paneles de operador. Mientras que SIMATIC ProTool permite configurar todo tipo de equipos, SIMATIC ProTool/Lite se limita a configurar los paneles de operador orientados a líneas. Desde el punto de vista funcional, ProTool/Lite se puede considerar un subconjunto de ProTool.

**Integración en el software STEP 7 de SIMATIC**

ProTool se puede integrar en el software de configuración STEP 7 de SIMATIC, lo cual permite acceder directamente a los datos de configuración, como por ejemplo a la tabla de símbolos y a los parámetros de comunicación de los equipos de un proyecto S7. Ello no sólo ahorra tiempo y dinero, sino que además reduce el riesgo de cometer errores, ya que ahora solamente se tienen que introducir una sola vez.

Tabla 2-6 Herramientas de configuración para aparatos de manejo y visualización

Aparato	Herramienta de configuración
OP orientado a líneas (TD17, OP 3, OP 5, OP7, OP 15, OP17)	ProTool/Lite o ProTool
OP orientado a gráficos (OP 25, OP 35, OP37, TP37)	ProTool

**WinCC**

WinCC se puede utilizar como sistema monopuesto y multipuesto (estructura cliente-servidor).

WinCC es un sistema universal que no obedece a un ramo o tecnología especial y que sirve para solucionar tareas de visualización y de control de procesos tanto continuos como discontinuos. Este software ofrece módulos de función industriales con representación de gráficos, sistemas de notificación de mensajes, sistemas de archivado y de listado de informes. Su potente acoplamiento al proceso, el rápido refresco de imagen, así como un archivado seguro de los datos, hacen de WinCC un software altamente disponible.

Además de estas funciones de sistema, WinCC ofrece interfaces abiertos para soluciones personalizadas que permiten integrar WinCC en soluciones de automatización complejas y extensibles a toda la empresa. Por ejemplo, ofrece la posibilidad de acceder a los datos archivados vía ODBC y SQL, dos interfaces estándar. Asimismo, permite integrar objetos y documentos a través de OLE2.0 y OLE-Custom-Controls (OCX). Estos mecanismos hacen de WinCC un partner competente y comunicativo de la plataforma Windows.

La plataforma de WinCC la constituyen los sistemas operativos de 32 bits MS-Windows 95 o MS-Windows NT. Ambos sistemas utilizan el modo multitarea "preemptivo" o con derecho preferente que permite responder rápidamente a los eventos del proceso y que garantiza una seguridad muy elevada ante la pérdida de datos. Windows NT ofrece además funciones de seguridad y constituye la base para el funcionamiento cliente-servidor en un sistema WinCC multipuesto. En cuanto al software WinCC, se trata de una aplicación de 32 bits que ha sido desarrollada con el más avanzado sistema de programación orientado a objetos.

## Software

### 3.1 Filosofía de manejo

**Resumen** El software de configuración y programación de SIMATIC S7/M7/C7 está diseñado según los criterios ergonómicos más avanzados, por lo que es ampliamente autoexplicativo.

#### 3.1.1 Requisitos para la instalación

**Sistema operativo** Microsoft Windows 95.

**Hardware básico** Unidad de programación (PG) o PC con

- un procesador 80486 (o superior) y
- una capacidad de memoria RAM de 16 MB como mínimo, aunque se recomiendan 32 MB,
- un monitor VGA u otro tipo de monitor soportado por Microsoft Windows 95,
- un teclado y - opcional, pero recomendable - un ratón soportado por Microsoft Windows 95.

**Capacidad de memoria** Memoria disponible en el disco duro:

- El paquete básico ocupa 105 MB en caso de instalarlo en un solo idioma. Por esta razón, el requerimiento de memoria dependerá del tipo de instalación que se elija.
- STEP 7 debería disponer de unos 64 MB menos la memoria principal para crear archivos de intercambio STEP 7 (es decir, aprox. 32 MB con una memoria principal de 32 MB).
- Se habrán de reservar aproximadamente 50 MB para los datos de usuario.
- Como mínimo se requiere 1 MB libre en la unidad C: para el programa de instalación Setup (los archivos del setup se borran al terminar la instalación).

### 3.1.2 Instalación del software STEP 7

#### Resumen

STEP 7 contiene un programa de instalación llamado Setup que ejecuta la instalación automáticamente. Las instrucciones que aparecen en pantalla le guiarán paso a paso a través de todo el proceso de instalación.

#### Autorización

Para poder utilizar el software de programación STEP 7 se requiere una autorización especial para el producto (licencia de uso). Para poder emplear el software así protegido tiene que estar instalada en el disco duro la autorización necesaria para el programa o el paquete de software en cuestión.

Para instalar la autorización se requiere el disquete de autorización suministrado con el producto, el cual está protegido contra escritura. Este disquete contiene la autorización y el programa AUTHORS necesario para visualizarla, instalarla y desinstalarla.

El procedimiento para instalar y desinstalar la autorización está descrito en el manual del usuario **/231/** de STEP 7.

---

#### Nota

Las unidades de programación de Siemens (como por ejemplo la PG 740) se suministran con el software STEP 7 instalado de fábrica en el disco duro.

---

Si desea obtener más información sobre cómo instalar STEP 7, consulte el manual del usuario **/231/**.

### 3.1.3 Arrancar el software STEP 7

#### Arranque

Después de arrancar Windows 95/NT encontrará el icono del Administrador SIMATIC en el escritorio de Windows, el cual le permitirá acceder al software STEP 7.

La forma más rápida de arrancar STEP 7 es haciendo un doble clic en el icono "Administrador SIMATIC". Seguidamente se abre la ventana del Administrador SIMATIC, desde donde se accede a todas las funciones que haya instalado, es decir, tanto a las del paquete básico como a las del software opcional.

Alternativamente es posible arrancar el Administrador SIMATIC haciendo clic en el botón "Inicio" de la barra de tareas de Windows 95/NT: elija, a continuación, el comando de menú "Simatic/STEP 7".

#### Administrador SIMATIC

El Administrador SIMATIC es el interface de acceso a las funciones de configuración y programación. Estas permiten:

- crear proyectos,
- configurar y parametrizar hardware,
- configurar enlaces de comunicación,
- crear programas,
- comprobar los programas creados y ponerlos en servicio.

El acceso a las funciones, que se basa en la programación orientada a objetos, es intuitivo y fácil de aprender.

El Administrador SIMATIC permite operar

- offline, es decir, sin tener conectado el autómata programable y
- online, es decir, con el autómata programable conectado.

(Aténgase a las consignas de seguridad aplicables.)

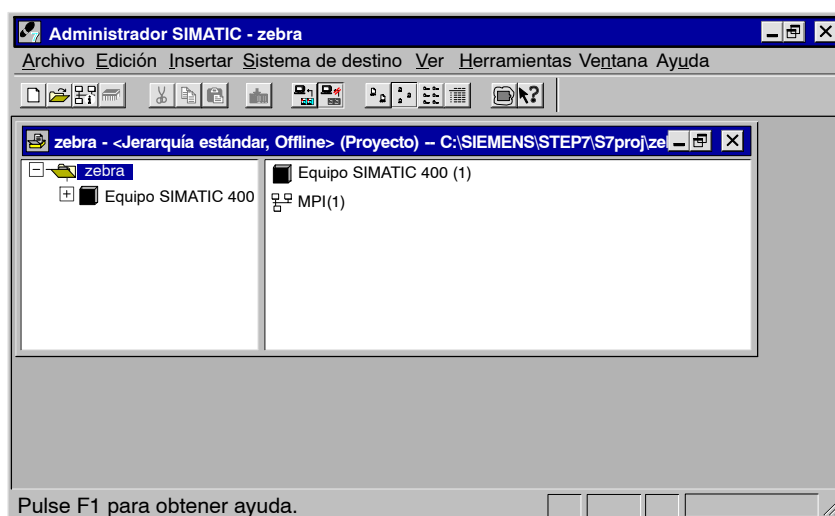


Figura 3-1 Proyecto abierto en el Administrador SIMATIC

## 3.2 Estructura de un proyecto S7

<b>Definición</b>	Los proyectos representan la totalidad de datos y programas de una solución de automatización. Sirven para almacenar de forma ordenada todos los datos y programas requeridos para elaborar una solución de automatización determinada.
<b>Proyectos de STEP 5</b>	<p>El concepto de proyecto ya existe en STEP 5 y abarca todos los ficheros <b>STEP 5</b> creados en un fichero de proyecto para <b>un</b> programa de usuario.</p> <p>En el fichero de proyecto se almacenan datos que permiten elaborar y retocar cómodamente el programa de usuario; p. ej.: los parámetros ajustados y los nombres de los directorios y ficheros.</p>
<b>Proyectos de STEP 7</b>	Un proyecto de STEP 7 abarca toda la gestión de programas y datos de una solución de automatización, independientemente del número de módulos centrales utilizados y de cómo estén interconectados. Por consiguiente, el proyecto no se limita solamente a un programa de usuario destinado a un módulo programable, sino que puede englobar varios programas de usuario para varios módulos programables que se encuentren bajo un mismo nombre de proyecto.
<b>Observación</b>	<p>Por supuesto, también es posible crear un programa de usuario sencillo para un solo módulo programable, como es habitual con STEP 5. En este caso el proyecto se limitará a un módulo central.</p> <p>A continuación explicaremos el árbol de directorios que ofrece STEP 7 para los programas de usuario y los datos que cree en su proyecto.</p>

## Componentes de un proyecto

Un proyecto de STEP 7 abarca fundamentalmente los objetos ilustrados en la figura 3-2. Estos objetos se explican a continuación.

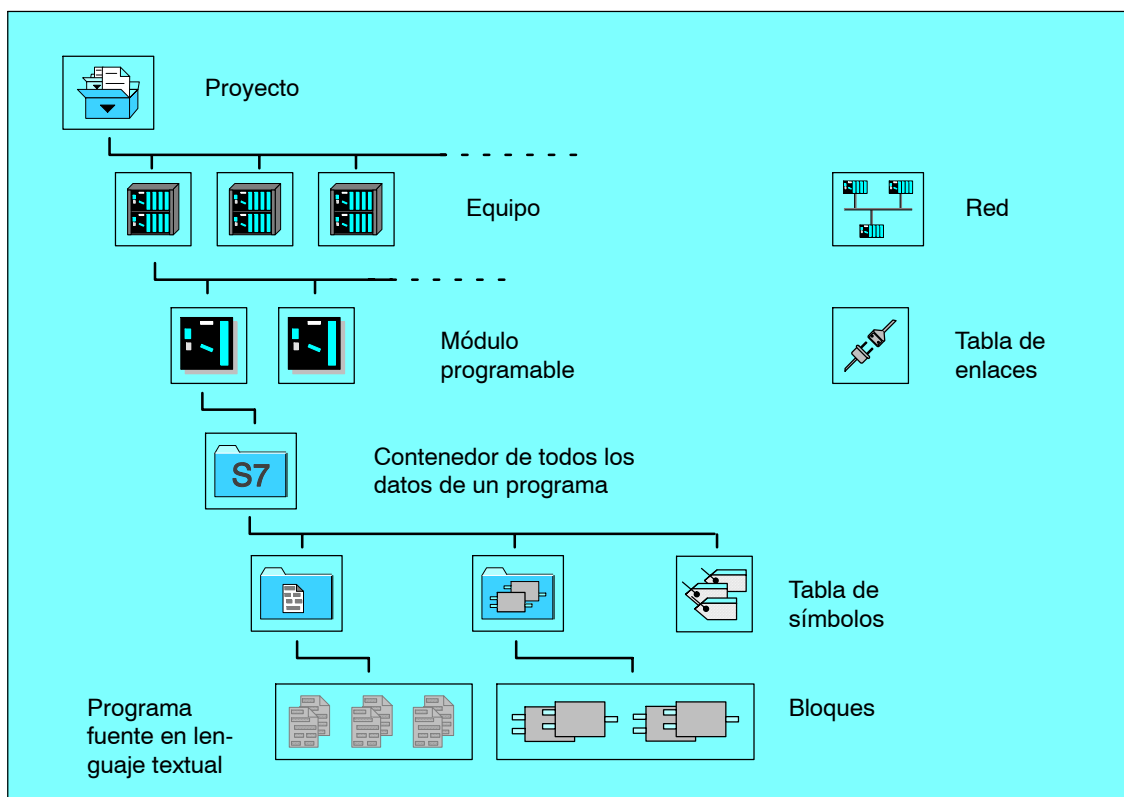


Figura 3-2 Principales objetos de un proyecto de STEP 7 y su jerarquía

### Red

El objeto "Red" representa las características determinantes de una subred, como puede ser MPI o PROFIBUS. Asignando un equipo o un módulo de comunicación a una red, STEP 7 podrá comprobar si los parámetros de comunicación son coherentes.

### Equipo

Un equipo representa la configuración real de un sistema de automatización con sus bastidores. Si un equipo tiene enchufado un módulo con interface DP, todo el sistema maestro (es decir, los esclavos DP correspondientes) formará parte de este equipo.

Un equipo puede contener uno o varios módulos programables (p. ej.: una CPU).

### Hardware

El hardware es un objeto que contiene los datos de configuración y los parámetros de un equipo, los cuales se depositan en bloques de datos de sistema (SDBs).

**Módulo programable**

A diferencia de otros módulos, los módulos programables pueden contener programas de usuario. En las carpetas - también llamadas “contenedores” en STEP 7 - que se encuentran debajo de los módulos programables, se guardan todos los datos pertenecientes al programa cargado en el módulo:

- Programas fuente textuales (se escriben con editores de textos)  
Al compilar los programas fuente se crean bloques ejecutables en el contenedor “Bloques”.
- Bloques (se cargan en el módulo programable)
- Tabla de símbolos

**Tabla de enlaces**

La tabla de enlaces representa todos los enlaces de un módulo programable (p. ej.: de una CPU) perteneciente a un equipo. Un enlace define las propiedades de comunicación entre dos estaciones y se identifica mediante un ID de enlace. El ID de enlace es todo lo que necesitará para programar una comunicación controlada por eventos con bloques de comunicación estándar - comparables a los módulos (bloques) de manejo en STEP 5.

**Fuentes**

En la programación de sistemas S7, las fuentes se utilizan de plataforma para crear los bloques y no se pueden cargar en una CPU S7.

**Bloques**

Los bloques son secciones del programa de usuario que se distinguen por su función, su estructura o su campo de aplicación. A diferencia de las fuentes, los bloques sí se pueden cargar en las CPUs S7.

Además de los bloques ejecutables, el contenedor “Bloques” contiene tablas de variables.

**Tabla de símbolos**

La tabla de símbolos contiene los nombres (=símbolos) que se asignan, p. ej., a las entradas, salidas, marcas y bloques.



### 3.3 Elaborar un proyecto con el Administrador SIMATIC

#### 3.3.1 Crear el proyecto

##### Nuevo proyecto

Para crear un proyecto, proceda como sigue:

1. Elija el comando de menú **Archivo ► Nuevo** en el Administrador SIMATIC.
2. Elija la opción “Nuevo proyecto” en el cuadro de diálogo “Nuevo”.
3. Introduzca el nombre que desea asignar a su proyecto y confirme con “Aceptar”.

##### Alternativa para proseguir

Para proseguir con la elaboración del proyecto dispone de la máxima flexibilidad, es decir, no tiene que seguir un orden establecido. Una vez creado el proyecto puede:

- configurar el hardware y crear después el software, o bien
- puede empezar a crear el software independientemente de haber configurado o no el hardware. Para introducir programas no es necesario haber configurado el hardware del equipo.

Tabla 3-1 Alternativas para proseguir con la elaboración de un proyecto

Alternativa 1	Alternativa 2
<b>Configurar primero el hardware</b> (v. apt. 3.4)	<b>Crear primero el software</b>
Configure su hardware (v. apt. 3.4).	
Una vez configurado el hardware, dispondrá de los contenedores necesarios para crear el software "Programa S7".	Inserte en su proyecto los contenedores necesarios para crear el software (Programas S7). (v. apt. 3.6).
A continuación, cree el software que cargará posteriormente en los módulos programables. (v. apt. 3.6).	A continuación, cree el software que cargará posteriormente en los módulos programables. (v. apt. 3.6).
	Configure el hardware (v. apt. 3.4).
	Asigne el programa S7 a una CPU después de configurar el hardware.

En el manual del usuario **/231/** encontrará descrito cómo cargar y comprobar sus programas sin necesidad de configurar el hardware.

### 3.3.2 Crear copias de seguridad de los proyectos

**Resumen** Para crear una copia de seguridad de un proyecto puede guardar una copia del proyecto con otro nombre, o bien archivar el proyecto de forma comprimida.

**Guardar como ...** Proceda como sigue:

1. Abra el proyecto.
2. Elija el comando de menú **Archivo ► Guardar como**. Aparecerá el cuadro de diálogo "Guardar como".
3. Elija "Guardar sin reorganizar" o "Guardar reorganizando" y cierre el cuadro de diálogo con "Aceptar". Aparecerá el cuadro de diálogo "Guardar proyecto como".
4. Indique bajo "Guardar como" el directorio en el que desea guardar su proyecto.
5. Introduza en el campo "Nombre del archivo" el nombre que desea dar al archivo en lugar del asterisco (\*). No modifique la extensión del nombre.
6. Cierre el cuadro de diálogo con "Aceptar".

Asegúrese de que haya suficiente memoria libre en la unidad de disco elegida. Ya que, por lo general, la capacidad de memoria de un disquete es insuficiente para guardar un proyecto, no es razonable elegir una unidad de disquete. Para guardar proyectos en disquetes primero hay que comprimirlos. Los archivadores que contienen los proyectos comprimidos se pueden copiar luego repartidos entre varios disquetes.

**Archivar proyectos o librerías** Los proyectos o librerías se pueden guardar en un archivo comprimido, tanto en el disco duro como en un soporte transportable (disquete).

Para acceder a componentes de un proyecto o librería que esté archivado es preciso descomprimir primero el proyecto o la librería en cuestión. Si desea obtener información más detallada al respecto, consulte el manual del usuario **/231/**.

### 3.4 Configurar el hardware en STEP 7

En SIMATIC S5 no era posible configurar el hardware mediante software. S7 ofrece una herramienta especial para direccionar y parametrizar los módulos, así como para configurar la comunicación. La ventaja de este procedimiento consiste en que ahora el usuario ya no tiene que realizar ajustes en los propios módulos, ya que puede despachar las tareas de configuración y parametrización de forma centralizada desde la unidad de programación (PG).

#### Requisitos

Se tiene que haber creado un proyecto.

#### Insertar un equipo

Para crear un nuevo equipo en un proyecto, abra el proyecto de forma que aparezca la ventana del proyecto (si es que aún no está visible en la pantalla).

- Seleccione el proyecto.
- Inserte el nuevo objeto para el hardware deseado con el comando de menú **Insertar > Equipo**.

En el menú que se despliega a continuación puede optar por uno de los siguientes comandos:

- Equipo SIMATIC 300
- Equipo SIMATIC 400
- PC/PG
- SIMATIC S5
- Otros equipos (es decir, cualquiera que no sea un SIMATIC S7/M7, SIMATIC S5)

Los equipos “PC/PG”, “SIMATIC S5” y “Otros equipos” se indican solamente para configurar enlaces de comunicación, por lo que no permiten configurar y programar equipos S5.

En la ventana del proyecto haga clic en el signo “+” a la izquierda del símbolo del proyecto, si es que el equipo todavía no está visible bajo el proyecto.

### **Configurar el hardware**

Proceda como sigue:

- Haga clic en el equipo recién insertado, el cual contiene el objeto "Hardware".
- Abra el objeto "Hardware". Aparecerá la ventana de la herramienta "HW-Config".
- Defina la configuración del equipo en la ventana "Configurar hardware". Para ello dispone del catálogo de módulos, que se abre con el comando de menú **Ver > Catálogo**, si es que aún no está visible en la pantalla.
- Inserte primero un bastidor (rack) del catálogo de módulos en la ventana vacía. Seguidamente vaya seleccionando los módulos y colóquelos sucesivamente en los slots de la tabla de configuración. Para cada equipo se tiene que configurar como mínimo una CPU. Todas las entradas realizadas se comprueban y, si son incorrectas, se rechazan.

Para más información sobre la configuración del hardware, consulte el manual del usuario **/231/**.

### **Resultado de la configuración**

Después de guardar y salir de la herramienta de configuración del hardware se crea automáticamente una tabla de enlaces y un programa S7 para cada una de las CPUs que haya creado en la configuración. El programa S7 contiene los objetos "Fuentes" y "Bloques", que son contenedores de software, así como la tabla de símbolos.

En el contenedor "Bloques" se encuentran el objeto para el OB 1 y los "Datos de sistema" con los datos de configuración compilados.

### 3.5 Configurar enlaces en la tabla de enlaces

En S5 se configuran los enlaces con el software COM NCM. A cada CP le corresponde un paquete COM. Por el contrario, en S7 se configuran todos los enlaces en la tabla de enlaces.

#### Resumen

La configuración de enlaces es una tarea imprescindible para poder utilizar funciones de comunicación SFB en el programa de usuario.

Los enlaces definen lo siguiente:

- los interlocutores del proyecto S7 que intervienen en la comunicación,
- el tipo de enlace (p. ej.: enlace S7, enlace FDL) y
- propiedades especiales, como p. ej. el establecimiento activo o pasivo del enlace (iniciativa local o remota), y si se han de enviar o no mensajes sobre el estado operativo del módulo.

En la configuración de enlaces se asigna a cada enlace un identificador local unívoco, el llamado ID local. Para parametrizar la función de comunicación se requiere solamente este ID local.

Todas aquellas CPUs que pueden ser punto final de un enlace disponen de una tabla de enlaces propia.

#### Particularidad

Si ambos interlocutores son equipos S7-400, se asignará automáticamente un ID local a ambos puntos finales del enlace. Para los enlaces que conducen a un equipo S7-300 se genera un solo ID local en el equipo S7-400.

#### Cargar los datos de configuración

Los datos locales de configuración de los puntos finales del enlace de un equipo S7 tienen que cargarse manualmente en cada uno de los equipos de destino.

Para cada CPU se crea automáticamente una tabla de enlaces (vacía) (objeto "Enlaces"). Esta tabla de enlaces se utiliza para definir los enlaces de comunicación entre las CPUs que están conectadas a una red. Después de abrirla aparece una ventana con una tabla en la que se definen los enlaces entre los módulos programables (para más información sobre cómo definir los enlaces, consulte el manual del usuario **/231/**).

**Ejemplo:  
enlace con un S5**

El ejemplo muestra cómo configurar un enlace con un equipo SIMATIC S5. En el ejemplo ya se ha insertado un equipo SIMATIC 400 en el proyecto.

- Inserte un equipo SIMATIC S5 en su proyecto y ajuste las propiedades del equipo.
- Abra la tabla de enlaces del equipo S7 e inserte un enlace con el comando de menú **Insertar > Enlaces**. Aparecerá un cuadro de diálogo en el que puede introducir los interlocutores - el equipo SIMATIC S5 - y el tipo de enlace.
- Una vez introducidos estos datos aparecerá el enlace en la tabla de enlaces. Las propiedades del enlace tienen que introducirse en el S5 en los COM NCM que correspondan al equipo S5.

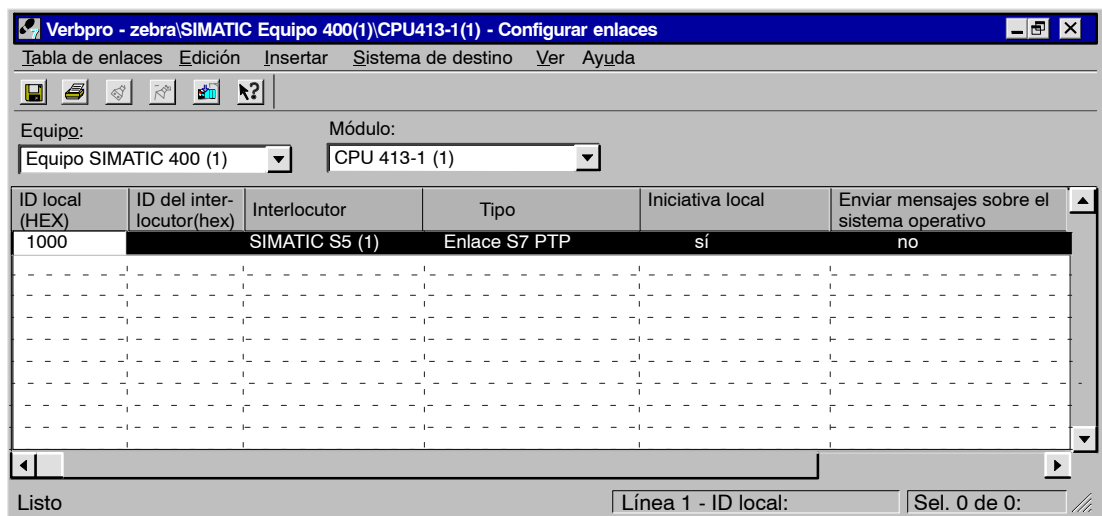


Figura 3-3 Tabla de enlaces

## 3.6 Insertar y elaborar un programa

El procedimiento descrito en este apartado explica cómo crear nuevos programas.

### 3.6.1 Procedimiento general para crear software

#### Resumen

El software para las CPUs se guarda en contenedores de programas. Para los módulos SIMATIC S7 los objetos de este tipo se denominan "Programa S7".

La figura muestra, por ejemplo, un programa S7 en una CPU de un equipo SIMATIC 300.

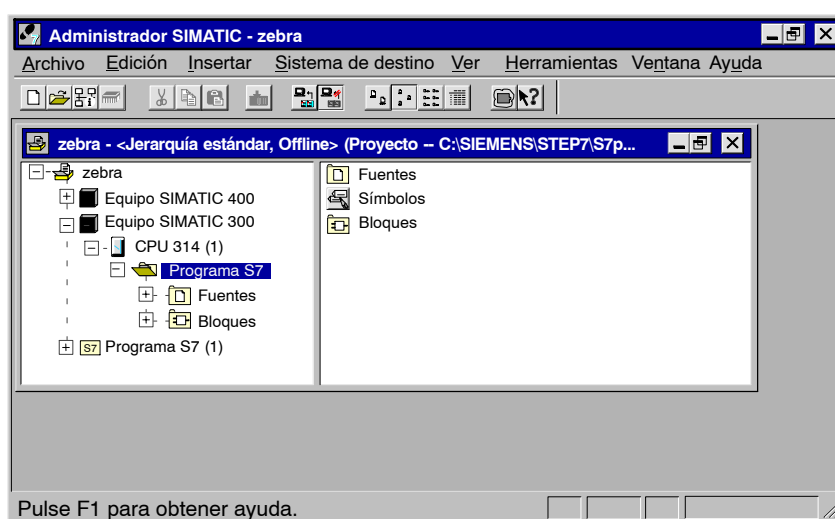


Figura 3-4 Programa S7 abierto en el Administrador SIMATIC

#### Procedimiento

Para crear el software de su proyecto proceda como sigue:

- Abra el programa S7.
- Abra el objeto "Símbolos" en el programa S7 y defina los símbolos. (Este paso también puede realizarse posteriormente.) Para más información al respecto consulte el apartado 3.13.2.
- Abra el contenedor "Bloques" si desea crear bloques, o bien abra el contenedor "Fuentes", si desea crear un programa fuente.
- Inserte un bloque o una fuente (para más detalles lea el apt. 3.6.2 ). Los comandos de menú disponibles son:
  - **Insertar > Software S7 > Bloque...**
  - **Insertar > Software S7 > Fuente**
- Abra el bloque o la fuente e introduzca un programa. Para más información al respecto consulte los manuales de programación /232/-/236/.

Dependiendo de la aplicación, no será necesario ejecutar todos los puntos.

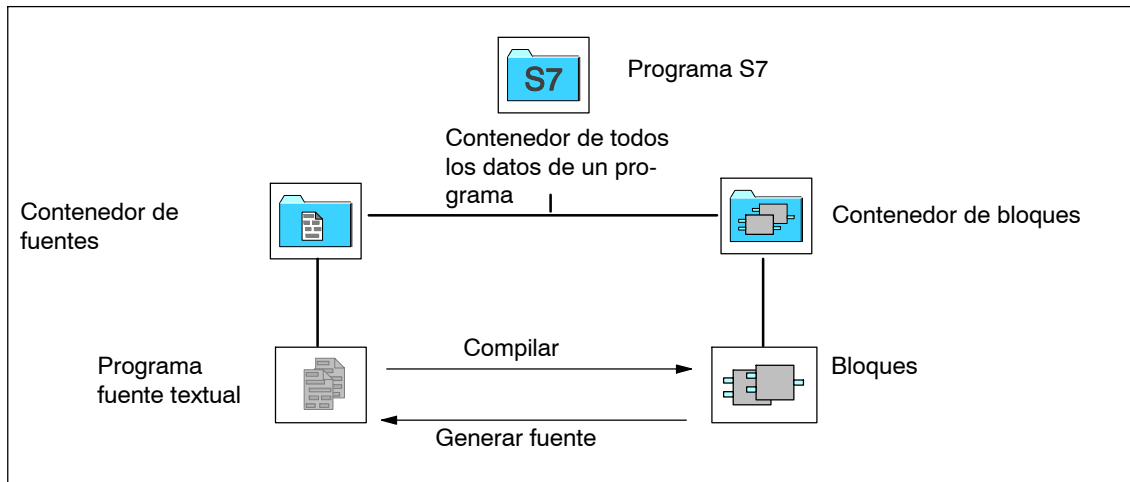


Figura 3-5 Objetos principales de un proyecto de STEP 7 y su estructura jerárquica



### 3.6.2 Insertar componentes para crear software en programas S7/M7

<b>Componentes ya creados</b>	<p>Para cada módulo programable se crea automáticamente un programa S7/M7 que contendrá el software correspondiente.</p> <p>El programa S7 ya contiene los objetos siguientes:</p> <ul style="list-style-type: none"><li>• la tabla de símbolos (objeto "Símbolos"),</li><li>• el contenedor "Bloques" con el primer bloque OB1,</li><li>• el contenedor "Fuentes" para programas fuente.</li></ul>
<b>Crear bloques S7</b>	<p>Para crear programas en los lenguajes AWL, KOP o FUP, seleccione el objeto ya creado ("Bloques") y ejecute el comando de menú <b>Insertar &gt; Software S7 &gt; Bloque</b>. En el menú que se despliega a continuación seleccione entonces el tipo de bloque deseado (bloque de datos, tipo de datos (UDT), función, bloque de función, bloque de organización, tabla de variables (VAT)).</p> <p>Después de abrir el bloque (vacío) proceda a introducir el programa en AWL, KOP o FUP. Para más información al respecto consulte los manuales AWL /232/, KOP /233/ o FUP /236/.</p> <p>El objeto "Datos de sistema" (SDB), que aparece a veces en los programas de usuario, lo crea el sistema. Si bien es posible abrir este objeto, no se debe modificar por motivos de coherencia, ya que sirve para cargar a posteriori las modificaciones que se realicen en la configuración después de cargarla en el sistema de destino.</p>
<b>Utilizar bloques de librerías estándar</b>	<p>Para crear programas de usuario también puede hacer uso de los bloques de las librerías estándar suministradas con el paquete de software. A las librerías se accede con el comando de menú <b>Archivo &gt; Abrir</b>. Para más información sobre el uso de las librerías estándar y sobre cómo crear librerías propias consulte la Ayuda en pantalla.</p>
<b>Crear fuentes</b>	<p>Para crear una fuente en AWL, seleccione el objeto "Fuentes" o "Planos" ("Esquemas") en el programa S7. Para ello marque en el programa S7 el objeto "Fuentes" o "Planos" ("Esquemas") y luego haga clic en el comando de menú <b>Insertar &gt; Software S7 &gt; Fuente</b>. Elija en el menú que se despliega a continuación la fuente apropiada para el lenguaje de programación elegido. Después de abrir la fuente vacía puede proceder a introducir el programa.</p>
<b>Crear la tabla de símbolos</b>	<p>Al crear un programa S7 se crea automáticamente la tabla de símbolos (vacía) (objeto "Símbolos"). Una vez abierta, aparece la ventana "Editor de símbolos" con la tabla de símbolos (v. apt. 3.13.2, <i>Direccionamiento simbólico</i>).</p>

### **Insertar fuentes externas**

Los archivos fuente se pueden crear y elaborar con cualquier editor ASCII. Después se pueden importar y compilar en bloques ejecutables. Para ello proceda como sigue:

- Seleccione el contenedor "Fuentes" al que desee importar el archivo fuente.
- Elija el comando de menú **Insertar > Fuente externa**.
- Introduzca el archivo fuente en el cuadro de diálogo que aparece a continuación.

Los bloques que se crean al compilar una fuente importada se depositan en el contenedor "Bloques".

## 3.7 Bloques

### 3.7.1 Comparativa

La tabla siguiente muestra una comparativa de los bloques de STEP 5 y STEP 7. La tabla responde a la pregunta "¿Qué bloque de STEP 7 debo utilizar para sustituir a un bloque determinado de STEP 5?"

#### Asignación no estricta

La tabla no ha de interpretarse como una asignación estricta de 1 a 1, ya que el nuevo mundo de bloques abre nuevos caminos adicionales para la programación. Se trata de una recomendación para familiarizarse con el software STEP 7.

Tabla 3-2 Comparativa de los bloques STEP 5 y STEP 7

Bloque STEP 5	Bloque STEP 7	Explicación
Bloques de organización (OB)	Bloques de organización (OB)	Interface con el sistema operativo
OBs especiales integrados	Funciones de sistema (SFC) Bloques de función de sistema (SFB)	Las funciones de sistema de STEP 7 sustituyen a los bloques de organización especiales (STEP 5) que se pueden llamar desde el programa de usuario.
Bloque de función (FB, FX)	Función (FC)	Las funciones (FC) de STEP 7 tienen las mismas características que los bloques de función de STEP 5.
Bloque de programa (PB)	Bloque de función (FB)	Los bloques de programa corresponden a los bloques de función de STEP 7. Los bloques de función de STEP 7 tienen unas características completamente distintas de los bloques que llevan el mismo nombre en STEP 5. Con ello se abren nuevas posibilidades para la programación. Atención: Los bloques de programa se convierten en funciones (FCs) al realizar la conversión de S5 a S7.
Bloque de paso (SB)	-	En STEP 7 ya no existen los bloques de paso.
Bloque de datos (DB, DX)	Bloque de datos (DB)	Los bloques de datos de STEP 7 son más largos que los de STEP 5 (en el S7-300 pueden tener una longitud de hasta 8 kbytes, en el S7-400 de hasta 64 kbytes)
Bloque de datos DX0, DB1 en su función especial	Bloques de datos de sistema (SDB) (Parametrización de la CPU)	Los nuevos bloques de datos del sistema contienen todos los datos de configuración del hardware, es decir, también los parámetros de la CPU que definen la ejecución del programa.
Bloques de comentario DK, DKX, FK, FKX, PK	-	En STEP 7 ya no existen los bloques de comentario. Los comentarios figuran en los bloques correspondientes de la gestión de datos offline.

### 3.7.2 Funciones y bloques de función

#### **Función (FC)**

Una función (FC) es un bloque lógico “sin memoria”. Los parámetros de salida contienen los valores que se obtienen tras ejecutar la FC. El uso posterior y la memorización de los parámetros actuales después de llamar a una FC está en manos del usuario.

No confunda las funciones con los bloques de función, ya que en STEP 7 se trata de dos tipos de bloques distintos.

#### **Bloques de función (FB)**

Un bloque de función (FB) es un bloque lógico “con memoria”. Como memoria se utiliza un bloque de datos de instancia que sirve para almacenar los parámetros actuales y los datos estáticos de bloques de función.

Los bloques de función se aplican, p. ej., en la programación de reguladores.

### 3.7.3 Bloques de datos

Los bloques de datos guardan los datos del programa de usuario. Cabe distinguir dos tipos de bloques de datos diferentes: los bloques de datos globales y los bloques de datos de instancia:

- Los bloques de datos globales no están asignados a ningún bloque (como en STEP 5).
- Los bloques de datos de instancia están asignados a un bloque de función y contienen, además de los datos del FB, las multiinstancias que se definan.

Todo bloque de datos puede ser un bloque de datos global o un bloque de datos de instancia.

### 3.7.4 Bloques de sistema

#### **Funciones de sistema (SFC) y bloques de función de sistema (SFB)**

No todas las funciones tienen que ser programadas por el usuario. Existen bloques preconfeccionados que residen en el sistema operativo de los módulos centrales, por ejemplo para programar funciones de comunicación. En particular se trata de los bloques siguientes:

- **funciones de sistema (SFC)** con las características de una función (FC) y
- **bloques de función de sistema (SFB)** con las características de un bloque de función (FB).

#### **Bloques de datos de sistema (SDB)**

Hasta ahora hemos hablado de bloques que contienen el programa o los datos del programa de usuario. Además de estos bloques existen otros que contienen los ajustes, como pueden ser los parámetros de módulos o también direcciones. Estos bloques se denominan **bloques de datos de sistema (SDB)**. Los bloques de datos de sistema se crean con herramientas especiales de STEP 7, p. ej., al introducir los datos de configuración del hardware o al elaborar tablas de enlaces.

### 3.7.5 Bloques de organización

Los bloques de organización (OBs) constituyen el interface entre el sistema operativo y el programa de usuario. Los distintos bloques de organización se reparten diferentes tareas.

#### Clasificación de los bloques de organización

El programa de usuario AWL para la CPU S7 se compondrá de los bloques de organización (OBs) que requiera la solución de automatización.

Tabla 3-3 Comparativa de los OBs en S5 y S7

Función		S5	S7
Programa principal	Ciclo libre	OB 1	OB 1
Alarmas	Alarma de retardo	OB 6	OB 20 a OB 23
	Alarma horaria	OB 9	OB 10 a OB 17
	Interrupciones de hardware	OB 2 a OB 5	OB 40 a OB 47
	Alarmas de proceso	OB 2 a OB 9 (EB 0)	Se sustituyen por interrupciones
	Alarmas cíclicas	OB 10 a OB 18	OB 30 a OB 38
	Alarma de multiprocesamiento	-	OB 60
Arranque	Nuevo arranque manual	OB 21 (S5-115U) OB 20 (a partir del S5-135U)	OB 100
	Rearranque manual	OB 21 (a partir del S5-135U)	OB 101
	Rearranque automático	OB 22	OB 101
Error	Error	OB 19 a OB 35	OB 121, OB 122, OB 80 a OB 87
Otras	Ejecución en STOP	OB 39	no existe
	Tarea no prioritaria	-	OB 90

## Tratamiento de errores

### OBs de error

Los OBs de error se llaman cuando aparece un error durante la ejecución del programa. Estos OBs sirven para programar reacciones a determinados errores. Si no existe ningún OB de error para un determinado tipo de error, la CPU cambia al estado STOP.

Tabla 3-4 Comparativa de los OBs de error en S5 y S7

Función	S5	S7
Llamada de un bloque no cargado	OB 19	OB 121
Retardo en acuse de recibo al acceder directamente a los módulos de la periferia	OB 23	OB 122
Retardo en acuse al actualizar la imagen del proceso y las marcas de acoplamiento	OB 24	OB 122
Errores de direccionamiento	OB 25	OB 122
Tiempo de ciclo excedido	OB 26	OB 80
Error de sustitución	OB 27	no existe
STOP por intervención del operador	OB 28 (S5-135U)	no existe
Retardo en acuse de recibo en el byte de entrada EB 0	OB 28 (S5-155U)	OB 85
Código de operación no permitido	OB 29 (S5-135U)	STOP
Retardo en acuse al acceder directamente a la periferia en el área de direccionamiento ampliada	OB 29 (S5-155U)	OB 122
Parámetro no permitido	OB 30 (S5-135U)	no existe
Error de paridad o retardo en acuse de recibo al acceder a la memoria de usuario	OB 30 (S5-155U)	OB 122
Error colectivo de funciones especiales	OB 31	no existe
Error de transferencia de un bloque de datos	OB 32	OB 121
Colisión de alarmas temporizadas	OB 33	OB 80
Error de regulador	OB 34 (S5-135U)	no existe
Error al crear un bloque de datos	OB 34 (S5-155U)	aviso de la SFC
Error de interface	OB 35	OB 84

## Eliminación de errores en S5 y S7

### Rebase de límites

Al igual que en el S5, el S7 también permite evaluar una señalización de rebase de límites a través de los bits de estado OV y OS. El comportamiento de los bits no presenta grandes diferencias.

En el manual AWL /232/ puede consultar el comportamiento de los bits de estado en relación con las distintas instrucciones.

### Funciones especiales integradas

En las CPUs S5 el interface entre el programa de usuario y el programa de sistema se realiza mediante accesos al sistema operativo y mediante OBs especiales.

Las CPUs S7 ofrecen, además de los bloques de organización, los nuevos tipos de bloques “Funciones de sistema” y “Bloques de función de sistema”.

### Funciones de sistema / Bloques de función de sistema

Las funciones de sistema (SFCs) y los bloques de función de sistema (SFBs) son bloques integrados en el sistema operativo de la CPU que se pueden llamar en el programa de usuario STEP 7 cuando se requieran. Los errores que aparezcan al ejecutar una función de sistema SFC se pueden evaluar en el programa de usuario con ayuda del valor de retorno RET\_VAL.

Tabla 3-5 Funciones especiales en S5 y S7

Función	Bloque S5	Sustituto en S7
Disparo del tiempo de ciclo	OB 31	SFC 43 RE_TRIGR
Fallo de la pila	OB34	OB 81 (la reacción al error puede ser programada por el usuario)
Acceso al byte indicador	OB 110	Instrucción STEP 7: L STW/T STW
Borrar AKKU 1 - 4	OB 111	Secuencia de instrucciones STEP 7: L 0; PUSH; PUSH; PUSH
AKKU Roll Up	OB 112	Con función diferente: Instrucción STEP 7: PUSH
AKKU Roll Down	OB 113	Con función diferente: Instrucción STEP 7: POP
Activar/desactivar bloqueo conjunto de alarmas	OB 120	SFC 41 DIS_AIRT SFC 42 EN_AIRT
Activar/desactivar el bloqueo individual de alarmas cíclicas	OB 121	SFC 39 DIS_IRT SFC 40 EN_IRT
Activar/desactivar el retardo conjunto de alarmas	OB 122	SFC 41 DIS_AIRT SFC 42 EN_AIRT
Activar/desactivar el retardo individual de alarmas cíclicas	OB 123	SFC 39 DIS_IRT SFC 40 EN_IRT



Tabla 3-5 Funciones especiales en S5 y S7, continuación

<b>Función</b>	<b>Bloque S5</b>	<b>Sustituto en S7</b>
Leer/ajustar la hora de la CPU	OB 150	SFC 0 SET_CLK SFC 1 READ_CLK
Ajustar/leer alarma cíclica controlada por reloj	OB 151	SFC 28 SET_TINT SFC 30 ACT_TINT SFC 31 QRY_TINT
Estadísticas del ciclo	OB 152	Datos locales del OB 1
Bucles contadores	OB 160 - 163 (S5-135U)	Instrucciones STEP 7: LOOP
Bucle de tiempo variable	OB 160 (S5-115U)	SFC 47 WAIT
Leer pila de bloques (BSTACK)	OB 170	no existe
Acceso variable a bloques de datos	OB 180	no existe
Comprobar bloques de datos	OB 181	SFC 24 TEST_DB
Copiar área de datos	OB 182	SFC 20 BLKMOV
Transferir marcas a bloques de datos	OB 190, 192	SFC 20 BLKMOV
Transferir bloques de datos a áreas de marcas	OB 191, 193	SFC 20 BLKMOV
Funciones de comunicación en modo multiprocesador	OB 200 - 205	no existe
Acceso a páginas de memoria	OB 216 - 218	En S7 no existe el direccionamiento vía páginas de memoria
Ampliación del signo	OB 220	Instrucción S7: ITD
Ajustar el tiempo de vigilancia del ciclo	OB 221	Parametrización con S7
Rearranca el tiempo de vigilancia del ciclo	OB 222	SFC 43 RE_TRIGR
Comparar modos de arranque	OB 223	Arranque en modo multiprocesador sólo si los modos de arranque son idénticos
Transferir marcas de acoplamiento en bloque	OB 224	no existe
Leer palabra del programa de sistema	OB 226	no existe
Leer suma de verificación del programa de sistema	OB 227	no existe
Leer información de estado de un nivel de ejecución del programa	OB 228	SFC 51 RDSYSST SFC 6 RD_SINFO
Funciones para bloques de manejo	OB 230 - 237	Comunicación con SFBs

Tabla 3-5 Funciones especiales en S5 y S7, continuación

Función	Bloque S5	Sustituto en S7
Inicializar registro de desplazamiento	OB 240	no existe
Ejecutar registro de desplazamiento	OB 241	no existe
Borrar registro de desplazamiento	OB 242	no existe
Regulación: inicializar algoritmo PID Regulación: modificar algoritmo PID	OB 250 OB 251	FB de regulación: FB 41 - FB 43 o SFB 41 - SFB 43
Transferir bloque de datos (DB/DX) a la RAM DB	OB 254, 255	no existe

### 3.7.6 Conversión de los bloques

#### Asignación de bloques

La estructura de los bloques ha sido modificada en S7. La figura siguiente muestra de forma simplificada cómo se asignan los bloques en la conversión de STEP 5 a STEP 7.

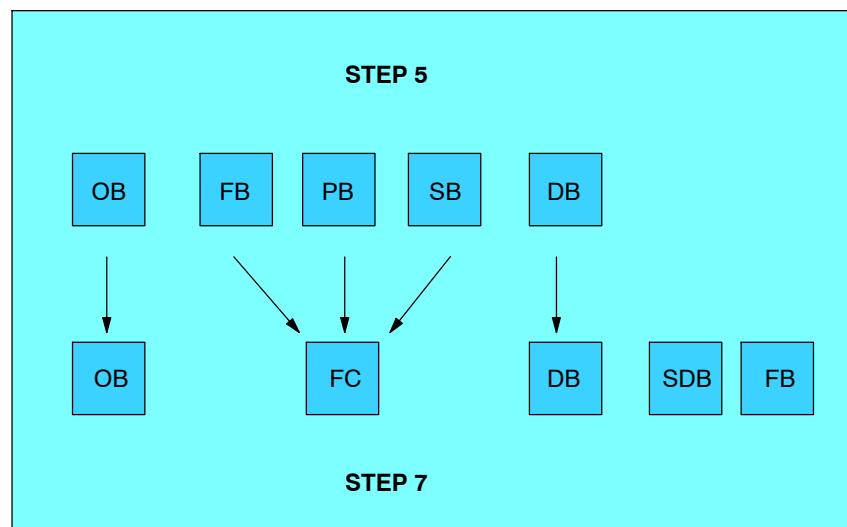


Figura 3-6 Bloques con funciones comparables en STEP 5 y STEP 7

La tabla 3-6 de la página 3-25 muestra cómo se traducen las llamadas de bloques en la conversión.

Tabla 3-6 Tipos de bloques en S5 y S7

S5			S7	
OB	Números fijos	Programa de usuario	OB S7 correspondiente	Números fijos
OB	Números fijos	Funciones especiales	No convertible; se tiene que volver a programar en S7.	
PB	0 a 255	Programa de usuario	Bloques FC sin parámetros	El número propuesto es opcional.
FB/FX	0 a 255	Programa de usuario	Bloques FC con parámetros cuyo nombre no cambia	El número propuesto es opcional.
FB	Números fijos	Bloques de función integrados	FCs cargables incluidos en la librería FBLib1, y que tienen que cargarse en el archivo convertido antes de compilarlo	Números fijos
FB/FX	Números fijos	Bloques de función estándar	FCs cargables incluidos en la librería FBLib1, y que tienen que cargarse en el archivo convertido antes de compilarlo	Números fijos
SB	0 a 255	Programa de usuario	Bloques FC sin parámetros Las cadenas secuenciales no se pueden convertir, por lo que tienen que crearse nuevamente con GRAPH para S7.	El número propuesto es opcional.
DB	2 a 255	Datos de usuario	Bloques de datos globales DB	Se adopta el número del S5.
DX	1 a 255	Datos de usuario	Bloques de datos globales DB	Se propone un número a partir de 256.
DB 1/ DX 0		Bloques de datos con ajustes del sistema	Si los bloques contienen entradas específicas de la CPU habrá que ajustar los parámetros correspondientes con la herramienta de configuración de STEP 7. El contenido del DB 1 y del DX 0 que se convierte no es relevante y puede borrarse.	

### 3.8 Ajustes del sistema

**Convertir el DB 1 y el DX 0** Las tablas siguientes muestran cómo se han realizado las funciones de los parámetros del DB 1 y del DX 0 (ajustes del sistema) en S7:

Tabla 3-7 Realización de los ajustes de sistema del DB 1

Bloque de parámetros S5	Realización en el S7
Retardo del arranque	Llamada de la SFC 47 WAIT
Marcas de acoplamiento	Se ajusta en la herramienta de comunicación de datos globales; llamada de: SFC 60 GD_SND SFC 61 GD_RCV
Posición del código de error	El sistema deposita mensajes de error en el búfer de diagnóstico. Ya no existe el dato "Posición del código de error".
Sustitución del número de los FBs integrados	Ya no existe
Entradas analógicas integradas	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Interrupciones integradas	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Contadores integrados	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Cambiar las prioridades de los OBs	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Emitir/bloquear imagen del proceso	Llamada de la SFC 27 UPDAT_PO
Leer/bloquear imagen del proceso	Llamada de la SFC 26 UPDAT_PI
Marcas remanentes	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Temporizadores remanentes	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Contadores remanentes	Se ajustan en el diálogo de propiedades de la CPU con la herramienta HW-Config
SINEC L1	Ha sido sustituido por el bus MPI (comunicación de datos globales)
SINEC L2	Se ajusta con la herramienta HW-Config
Protección del software	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Parámetros de la hora	Se ajustan con la herramienta HW-Config en el diálogo de propiedades de la CPU o llamando a la SFC 28 SET_TINT
Parametrizar OBs de alarma cíclica	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Vigilancia del tiempo de ciclo (watchdog)	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config

Tabla 3-8 Realización de los ajustes de sistema del DX 0

<b>Bloque de parámetros S5</b>	<b>Realización en S7</b>
Vigilancia de errores de direccionamiento	Llamada del OB 121
Actualización de las marcas de acoplamiento	Comunicación de datos globales
Modo de arranque tras CONEXION a la red eléctrica	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Sincronización del arranque en modo multiprocesador	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Número de temporizadores	Número fijo que depende de la CPU utilizada (en S7-300) o número ajustable en el diálogo de propiedades CPU con la herramienta HW-Config (en S7-400)
Tratamiento de errores	Llamada de: SFC 36 MSK_FLT SFC 37 DMSK_FLT
Aritmética en coma flotante	disponible
Disparo de alarmas de proceso	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config
Modo de procesamiento de alarmas cíclicas	Llamada de la SFC 28 SET_TINT
Vigilancia del tiempo de ciclo	Se ajusta en el diálogo de propiedades de la CPU con la herramienta HW-Config

### 3.9 Funciones estándar

Las funciones estándar de S5 se convierten automáticamente en funciones S7 que ofrecen la misma funcionalidad. Estas funciones se suelen sustituir en S7 por secuencias de instrucciones simples, lo que ahorra espacio de memoria y tiempo de ciclo.

Las funciones estándar están incluidas en la librería S7 “StdLib30” que se encuentra en el contenedor de programas FBLib1.

Para más información sobre el uso de las librerías consulte la Ayuda en pantalla.

#### 3.9.1 Aritmética en coma flotante

STEP 5	STEP 7		STEP 5	STEP 7	
Nombre del FB	Número	Nombre	Nombre del FB	Número	Nombre
GP:FPGP	FC 61	GP_FPGP	GP:MUL	FC 65	GP_MUL
GP:GFPF	FC 62	GP_GFPF	GP:DIV	FC 66	GP_DIV
GP:ADD	FC 63	GP_ADD	GP:VGL	FC 67	GP_VGL
GP:SUB	FC 64	GP_SUB	RAD:GP	FC 68	RAD_GP

#### 3.9.2 Funciones de señales

STEP 5	STEP 7		STEP 5	STEP 7	
Nombre del FB	Número	Nombre	Nombre del FB	Número	Nombre
MLD:TG	FC 69	MLD_TG	MLD:EZ	FC 75	MLD_EZ
MELD:TGZ	FC 70	MELD_TGZ	MLD:ED	FC 76	MLD_ED
MLD:EZW	FC 71	MLD_EZW	MLD:EZWK	FC 77	MLD_EZWK
MLD:EDW	FC 72	MLD_EDW	MLD:EDWK	FC 78	MLD_EDWK
MLD:SAMW	FC 73	MLD_SAMW	MLD:EZK	FC 79	MLD_EZK
MLD:SAM	FC 74	MLD_SAM	MLD:EDK	FC 80	MLD_EDK

#### 3.9.3 Funciones integradas

STEP 5	STEP 7	
Nombre del FB	Número	Nombre
COD:B4	FC 81	COD_B4
COD:16	FC 82	COD_16
MUL:16	FC 83	MUL_16
DIV:16	FC 84	DIV_16

### 3.9.4 Funciones básicas

STEP 5	STEP 7		STEP 5	STEP 7	
Nombre del FB	Número	Nombre	Nombre del FB	Número	Nombre
ADD:32	FC 85	ADD_32	REG:LIFO	FC 93	REG_LIFO
SUB:32	FC 86	SUB_32	DB:COPY	FC 94	DB_COPY
MUL:32	FC 87	MUL_32	DB:COPY	FC 95	DB_COPY
DIV:32	FC 88	DIV_32	RETTEN	FC 96	RETTEN
RAD:16	FC 89	RAD_16	LADEN	FC 97	LADEN
REG:SCHB	FC 90	REG_SCHB	COD:B8	FC 98	COD_B8
REG:SCHW	FC 91	REG_SCHW	COD:32	FC 99	COD_32
REG:FIFO	FC 92	REG_FIFO			

### 3.9.5 Funciones analógicas

STEP 5	STEP 7		STEP 5	STEP 7	
Nombre del FB	Número	Nombre	Nombre del FB	Número	Nombre
AE:460	FC 100	AE_460_1	AE:466	FC 106	AE_466_1
AE:460	FC 101	AE_460_2	AE:466	FC 107	AE_466_2
AE:463	FC 102	AE_463_1	RLG:AA	FC 108	RLG_AA1
AE:463	FC 103	AE_463_2	RLG:AA	FC 109	RLG_AA2
AE:464	FC 104	AE_464_1	PER:ET	FC 110	PER_ET1
AE:464	FC 105	AE_464_2	PER:ET	FC 111	PER_ET2

### 3.9.6 Funciones matemáticas

STEP 5	STEP 7		STEP 5	STEP 7	
Nombre del FB	Número	Nombre	Nombre del FB	Número	Nombre
SINUS	FC 112	SINUS	ARCCOT	FC 119	ARCCOT
COSINUS	FC 113	COSINUS	LN X	FC 120	LN_X
TANGENS	FC 114	TANGENS	LG X	FC 121	LG_X
COTANG	FC 115	COTANG	B LOG X	FC 122	B_LOG_X
ARCSIN	FC 116	ARCSIN	E^X	FC 123	E_H_N
ARCCOS	FC 117	ARCCOS	ZEHN^X	FC 124	ZEHN_H_N
ARCTAN	FC 118	ARCTAN	A2^A1	FC 125	A2_H_A1

### 3.10 Tipos de datos

STEP 7 utiliza nuevos formatos para los datos. La tabla siguiente muestra los diferentes tipos de datos en S5 y S7.

Tabla 3-9 Tipos de datos en S5 y S7

Tipos de datos en S5	Tipos de datos en S7	Tipos de datos
BOOL BYTE WORD DWORD Coma fija 16 bits Coma fija 32 bits Coma flotante Tiempo - (Caracteres ASCII)	BOOL, BYTE, WORD, DWORD, INT, DINT, REAL, S5TIME, TIME, DATE; TIME_OF_DAY, CHAR	Tipos de datos simples
-	DATE_AND_TIME, STRING, ARRAY, STRUCT	Tipos de datos compuestos
Temporizadores Contadores Bloques - -	TIMER, COUNTER, BLOCK_FC, BLOCK_FB, BLOCK_DB, BLOCK_SDB, POINTER, ANY	Parámetros



Tabla 3-10 Distintos formatos de constantes en S5 y en S7

Formatos de S5	Ejemplo	Formatos de S7	Ejemplo																							
KB	L KB 10	3#16#	L B#16# A																							
KF	L KF 10	-	L 10																							
KH	L KH FFFF	W#16#	L W#16# FFFF																							
KM	L KM 1111111111111111	2#	L 2# 11111111_11111111																							
KY	L KY 10,12	B#	L B# (10,12)																							
KT	L KT 10.0	S5TIME# (S5T#)	L S5TIME# 100ms																							
KZ	L KZ 30	C#	L C#30																							
DH	L DH FFFF FFFF	DW#16#	L DW#16# FFFF_FFFF																							
KC	L KC WW	' xx '	L ' WW '																							
KG	L KG +234 +09	REAL	L +2.34 E+08																							
<b>Representación: Formato S5</b>		<b>Representación: Formato simple según ANSI/IEEE</b>																								
← Exponente → ← Mantisa →		V ← Exponente → ← Mantisa →																								
<table border="1"> <tr> <td>31</td><td>30</td><td>24</td><td>23</td><td>22</td><td>0</td> </tr> <tr> <td>SE</td><td>2<sup>6</sup>..</td><td>...</td><td>...</td><td>2<sup>0</sup></td><td>SM 2<sup>-1</sup>..... 2<sup>-23</sup></td> </tr> </table>		31	30	24	23	22	0	SE	2 <sup>6</sup> ..	...	...	2 <sup>0</sup>	SM 2 <sup>-1</sup> ..... 2 <sup>-23</sup>	<table border="1"> <tr> <td>31</td><td>30</td><td>23</td><td>22</td><td>0</td> </tr> <tr> <td>V</td><td>2<sup>7</sup>..</td><td>...</td><td>...</td><td>2<sup>0</sup></td><td>2<sup>-1</sup>.. 2<sup>-23</sup></td> </tr> </table>		31	30	23	22	0	V	2 <sup>7</sup> ..	...	...	2 <sup>0</sup>	2 <sup>-1</sup> .. 2 <sup>-23</sup>
31	30	24	23	22	0																					
SE	2 <sup>6</sup> ..	...	...	2 <sup>0</sup>	SM 2 <sup>-1</sup> ..... 2 <sup>-23</sup>																					
31	30	23	22	0																						
V	2 <sup>7</sup> ..	...	...	2 <sup>0</sup>	2 <sup>-1</sup> .. 2 <sup>-23</sup>																					
Exponente = Valor del exponente		Exponente = exponente efectivo + Bias* (+127)																								
SE = Signo del exponente		V = Signo de la mantisa																								
SM = Signo de la mantisa																										
Margen de valores: de 1,5 x 10 <sup>-39</sup> a 1,7 x 10 <sup>38</sup>		Margen de valores: aprox. de 1,18 x 10 <sup>-38</sup> a 3,40 x 10 <sup>+38</sup>																								

\* Bias: se trata de un offset que divide el exponente en un margen positivo y uno negativo. El valor 127 en el margen del exponente corresponde absolutamente al valor 0.

Si desea obtener más información sobre los tipos de datos, consulte el manual de AWL /232/.

## 3.11 Áreas de operandos

### 3.11.1 Panorámica

Tabla 3-11 Operandos de S5 y S7

Áreas de operandos	Operandos de S5	Operandos de S7	Observación
Entradas	E	E	
Salidas	A	A	
Periferia	P, Q, G	PE → en instrucciones de carga	la periferia global no se convierte
		PA → en instrucciones de transferencia	
Area de marcas	M	M	
	S	M	a partir de M 256.0 (convertidor)
	“Marcas de trabajo”	L	se convierten igual que las marcas
Temporizadores	T	T	
Contadores	Z	Z	
Area de datos	D...	DB...	se convierten en operandos de datos globales
Datos de sistema	BS, BT, BA, BB	-	no se convierten
Area de páginas	C	-	

#### Observación acerca de los operandos de datos

En S7 existen dos registros de bloques de datos: el registro DB, que se utiliza principalmente para bloques de datos globales y el registro DI, que se utiliza en particular para los DBs de instancia. Por ello existen dos tipos de operandos de datos. Los operandos DBX, DBB, DBW, DBD son operandos de bloques de datos globales, mientras que los operandos DIX, DIB, DIW, DID son operandos de DBs de instancia. En la conversión se utilizan operandos de bloques de datos globales para los operandos de bloques de datos D, DB, DW, DD.

En lo que respecta a la conversión de los bloques de datos, v. apt. 3.7.6.

#### Nota

Tenga en cuenta que en S7 el tamaño y los márgenes numéricos de las áreas de operandos, así como el número y la longitud de los bloques, dependen de la CPU utilizada. Las características de las distintas CPUs figuran en el apt. 2.2.1.

### 3.11.2 Nuevos operandos de S7: Datos locales

#### Datos locales de STEP 7

Los datos locales de STEP 7 son datos que se asignan a un bloque lógico y que se declaran en el área de declaración de variables del mismo. Dependiendo del bloque del que se trate, pueden ser parámetros de bloques, datos estáticos o datos temporales. Los datos locales se direccionan generalmente con nombres simbólicos.

#### Parámetros de bloques

Los parámetros de las funciones (FC) reciben el mismo tratamiento que los parámetros de los bloques S5: los parámetros actúan de punteros que señalan a sus parámetros actuales (reales).

Los parámetros de los bloques de función (FB) se depositan - al igual que los datos locales estáticos - en el bloque de datos de instancia.

#### Datos locales

Los datos locales estáticos pueden ser utilizados en cualquier bloque de función. Se definen en el área de declaración y se depositan en el DB de instancia.

Al igual que los operandos de datos de los bloques de datos globales, los datos locales estáticos mantienen su valor hasta ser sobrescritos por el programa.

Generalmente, los datos locales estáticos se procesan solamente en el bloque de función. Sin embargo, como se guardan en un bloque de datos, el programa de usuario puede acceder a ellos en cualquier momento del mismo modo que a las variables de un bloque de datos globales.

#### Datos locales temporales

##### Marcas de trabajo de STEP 5

En STEP 5 se utilizaban áreas de marcas para guardar datos de forma intermedia en bloques. Las marcas 200 ... 255 están reservadas para utilizarlas de memoria intermedia. La gestión de las marcas de trabajo es tarea exclusiva del usuario.

##### Datos locales temporales de STEP 7

Los datos locales temporales sirven de memoria para aquellos datos que solamente son válidos durante la ejecución del bloque. El espacio de memoria que ocupan se libera en cuanto se termina de ejecutar el bloque. Todo nivel de ejecución dispone de una pila propia de datos locales. Con ello queda excluida la posibilidad de que los programas de alarmas sobrescriban resultados intermedios accidentalmente.

**Uso de datos locales temporales en STEP 7**

En STEP 7, las variables temporales se utilizan para tres fines distintos:

- de memoria intermedia para datos del programa de usuario.

Este uso, que ya hemos descrito anteriormente, es aplicable a los bloques "Funciones" (FC), "Bloques de función" (FB) y "Bloques de organización" (OB).

- de memoria para transferir información del sistema operativo al programa de usuario.

Las informaciones que el sistema operativo proporciona al programa de usuario se denominan "Informaciones de arranque". Las informaciones de arranque las reciben exclusivamente los bloques de organización (OB) en calidad de interface entre el sistema operativo y el programa de usuario.

- para transferir parámetros en las FCs.

**¿Dónde se declaran los datos locales temporales?**

Los datos locales temporales se declaran dentro del propio bloque. Cuando cree un bloque, declare primero símbolos para las variables temporales y utilícelos luego en el área de instrucciones del bloque. En el S7-300 dispone de 256 bytes por cada nivel de ejecución. En el S7-400 dispone de hasta un total de 16 kbytes que puede repartir entre los distintos niveles de ejecución al parametrizar la CPU.

### 3.12 Operaciones

La tabla siguiente ofrece una vista general de las operaciones disponibles. La tabla indica las operaciones que se pueden convertir y, caso de no ser convertibles, se ofrecen alternativas para conseguir el mismo efecto.

Tabla 3-12 Operaciones de S5 y S7

Tipo de operación	Operaciones de S5	Operaciones de S7	Convertible	Alternativa recomendada
Operaciones con acumuladores	TAK, ENT, I, D, ADDBF, ADDKF, ADDDH	TAK, ENT, INC, DEC, +,  Operaciones nuevas en S7: TAW, TAD, PUSH, POP, LEAVE	sí	-
Operaciones con registros de direcciones u otros registros	MA1, MBR, ABR, MAS, MAB, MSB, MSA, MBA, MBS; TSG, LRB, LRW, LRD, TRB, TRW, TRD	Operaciones nuevas en S7: LAR1, LAR2, TAR1, TAR2, +AR1, +AR2, TAR	no	Utilizar el registro de direcciones (AR1, AR2)
Operaciones lógicas con bits	U, UN, O, ON, U(, O(, ), O, S, R, RB, RD, = P, PN, SU, RU	U, UN, O, ON, U(, O(, ), O, S, R, =  SET; U, SET; UN, SET; S, SET; R  Operaciones nuevas en S7: X, XN, X(, XN(,FP, FN, NOT, SET, CLR, SAVE	sí	-
Operaciones de temporización	SI, SV, SE, SS/SSV, SA/SAR, FR, SVZ	SI, SV, SE, SS, SA, FR, S T	sí	-
Operaciones de contaje	ZV/SSV, ZR/SAR, FR, SVZ	ZV, ZR, FR, S Z	sí	-

Tabla 3-12 Operaciones de S5 y S7, continuación

Tipo de operación	Operaciones de S5	Operaciones de S7	Convertible	Alternativa recomendada
Operaciones de carga y transferencia	L, LC, LW, LD, T L PB, L QB, L PW, L QW, T PB, T QB, T PW, T QW	L, LC, T L PEB, L PEW, T PAB, T PAW	sí	-
	LB GB / GW / GD / CB / CW / CD, LW GW / GD / CW / CD, TB GB / GW / GD / CB / CW / CD, TW GW / GD / CW / CD		no	Sustituir por un acceso al área de la periferia
Aritmética en coma fija	+F, -F, XF, :F, +D, -D	+I, -I, *I, /I, +D, -D, *D, /D  Operación nueva en S7: MOD	sí	-
Aritmética en coma flotante	+G, -G, XG, :G	+R, -R, *R, /R	sí	-
Operaciones de comparación	!=F, ><F, >F, <F, >=F, <=F, !=D, ><D, D, <D, >=D, <=D, !=G, ><G, >G, <G, >=G, <=G	==I, <>I, >I, <I; >=I, <=I, ==D, <>D, >D, <D, >=D, <=D, ==R, <>R, >R, <R, >=R, <=R	sí	-
Operaciones de conversión	KEW, KZW, KZD DEF, DED, DUF, DUD, GFD, FDG	INVI, NEGI, NEGD, BTI, BTD, DTB, ITB, RND, DTR  Operaciones nuevas en S7: ITD, RND+, RND-, TRUNC, INVD, NEGR	sí	-
Operaciones lógicas con palabras	UW, OW, XOW	UW, OW, XOW  Operaciones nuevas en S7: UD, OD, XOD	sí	-

Tabla 3-12 Operaciones de S5 y S7, continuación

Tipo de operación	Operaciones de S5	Operaciones de S7	Convertible	Alternativa recomendada
Operaciones de desplazamiento y rotación	SLW, SLD, SRW, SRD, SVW, SVD, RLD, RRD	SLW, SLD, SRW, SRD, SSI, SSD, RLD, RRD  Operaciones nuevas en S7: RLDA, RRDA	sí	-
Operaciones con bloques de datos	A, AX	AUF	sí	
	E, EX	SFC 22	no	Sustituir por una llamada al SFC 22 CREATE_DB
		Operaciones nuevas en S7: TDB L DBLG, L DBNO, L DILG, L DINO		
Operaciones de salto	SPA, SPB, SPN, SPZ, SPP, SPM, SPO, SPS, SPR	SPA, SPB, SPN, SPZ, SPP, SPM, SPO, SPS  Operaciones nuevas en S7: SPBN, SPBB, SPBNB, SPBI, SPBIN, SPMZ, SPPZ, SPU, LOOP, SPL	sí	-
Operaciones con bloques	SPA, SPB, BA, BAB, BE, BEA, BEB	CALL, BE, BEA, BEB	sí	-
Operaciones de habilitación de salidas / Operaciones Master Control Relay	BAS, BAF	Operaciones nuevas en S7: MCRA, MCRD, MCR(, )MCR	no	Sustituir por llamada a la SFC 26, SFC 27 u operaciones Master Control-Relay
Operaciones de Stop	STP, STS, STW	SFC 46	no	Sustituir por llamada a la SFC 46 STP

Tabla 3-12 Operaciones de S5 y S7, continuación

Tipo de operación	Operaciones de S5	Operaciones de S7	Convertible	Alternativa recomendada
Operaciones de procesamiento	B <parámetro formal>	-	no	Volver a programar la llamada del DB / bloque lógico
	B MW, B DW	Direccionamiento indirecto por memoria	sí	Recomendación: sustituir por direccionamiento indirecto por registro
	B BS	Direccionamiento interárea indirecto por registro	no	Sustituir por direccionamiento indirecto (v. apt. 3.13.4)
Direccionamiento absoluto por memoria	LIR, TIR, LDI, TDI	-	no	Sustituir por direccionamiento indirecto (v. apt. 3.13.4)
Transferencia en bloque	TNB, TNW, TXB, TXW	SFC 20	no	Sustituir por llamada a SFC 20 BLKMOV
Instrucciones de interrupción	LIM, SIM, AFS, AFF, AS, AF	SFC 39 -42	no	Sustituir por llamada a SFC 39 - 42
Instrucciones para páginas de memoria	ACR, TSC, TSG	-	no	En S7 ya no existen los accesos a páginas.
Funciones matemáticas	-	ABS, COS, SIN, TAN , ACOS, ASIN, ATAN, EXP, LN	-	-
Operaciones nulas	BLD xxx NOP 0, NOP 1	BLD xxx NOP 0, NOP 1	sí	-



### 3.13 Direccionamiento

#### 3.13.1 Direccionamiento absoluto

El direccionamiento absoluto es idéntico en S5 y S7 salvo en una excepción:

En S7 los datos de los bloques de datos se direccionan **byte a byte**, es decir, las direcciones de palabras en S5 se convierten en direcciones de bytes (multiplicándolas por 2).

La tabla siguiente muestra cómo se asignan las direcciones durante la conversión (direccionamiento de las áreas de datos):

S5	S7
DL 0, 1, 2, 3, ...255	DBB 0, 2, 4, 6, ...510
DR 0, 1, 2, 3, ...255	DBB 1, 3, 5, 7, ...511
DW 0, 1, 2, 3, ...255	DBW 0, 2, 4, 6, ...510
DD 0, 1, 2, 3, ...254	DBD 0, 2, 4, 6, ...508
D x.y	DBX 2 x.y para $8 \leq y \leq 15$ DBX (2 x+1).y para $0 \leq y \leq 7$

#### 3.13.2 Direccionamiento simbólico

En S7 se ha adoptado el direccionamiento simbólico de S5. Sin embargo, existen nuevas posibilidades para crear y utilizar los símbolos. En lo que respecta a la programación, no ha cambiado nada.

##### Símbolos de STEP 5

En los programas de STEP 5 se declaraban los símbolos con el editor de símbolos. El editor de símbolos crea una lista de asignación que permite utilizar los símbolos definidos en la misma (también llamada lista de correspondencia) en lugar de direcciones absolutas.

##### Símbolos de STEP 7

Los símbolos de STEP 7 pueden tener una longitud de hasta 24 caracteres.

##### Símbolos globales

STEP 7 también ofrece un editor de símbolos. La lista de asignación se denomina ahora "Tabla de símbolos". En ella se declaran todos los símbolos globales (p. ej.: entradas, salidas, marcas, bloques).

Si asigna símbolos con el editor de símbolos, éstos valdrán para un programa S7.

**Símbolos locales**

Además de poder declarar símbolos con el editor de símbolos, STEP 7 ofrece la posibilidad de definir símbolos locales para operandos de datos y para el área de datos locales al programar bloques.

En caso de no asignar los símbolos con el editor de símbolos, sino de declararlos en el bloque mismo, estos símbolos solamente serán válidos dentro del bloque en cuestión. Los símbolos así definidos se denominan símbolos locales del bloque.

**Cuándo definir símbolos**

STEP 7 no impone cuándo se tienen que definir los símbolos. Existen dos posibilidades:

- antes de empezar a programar  
(si va a introducir el programa de usuario de forma incremental, es decir, comprobándose la sintaxis cada vez que se introduce una línea),
- al terminar de escribir el programa de usuario, pero antes de compilarlo  
(si va a introducir el programa en una fuente, es decir, el programa creado es un archivo ASCII (fuente)).

**Importar una tabla de símbolos**

En S7 existe la posibilidad de crear y elaborar la tabla de símbolos con un editor de su elección.

También puede importar y seguir elaborando las tablas que haya creado con otras herramientas. La función de importación se puede utilizar, por ejemplo, para integrar en la tabla de símbolos las listas de asignación creadas con STEP5/ST después de la conversión.

Los tipos de archivos disponibles son \*.SDF, \*.ASC, \*.DIF y \*.SEQ.

Para importar una tabla de símbolos proceda como sigue:

1. Abra en la ventana del proyecto el programa S7 en el que se encuentra la tabla de símbolos.
2. Abra la tabla de símbolos haciendo doble clic en el contenedor “Símbolos”.
3. Elija en la ventana de la tabla de símbolos el comando de menú **Tabla > Importar**. Aparecerá un cuadro de diálogo.
4. Seleccione en el cuadro de diálogo la tabla de símbolos que desea importar y haga clic en el botón de comando “Abrir”.
5. Compruebe los registros de la tabla de símbolos y, si es preciso, corríjalos.
6. Guarde y cierre la tabla de símbolos.

---

**Nota**

Las tablas de símbolos del tipo \*.SEQ que se hayan convertido de S5 a S7 no se pueden reimportar a S5. Para intercambiar tablas de símbolos entre S5 y S7 se recomienda utilizar el tipo de archivo \*.DIF.

---

Para más información sobre las tablas de símbolos consulte el manual del usuario **/231/**.

### 3.13.3 Novedad: Direccionamiento completo de operandos de datos

Por direccionamiento completo de operandos de datos se entiende la indicación conjunta del bloque de datos y del operando. Esto no era posible en S5.

El direccionamiento completo sólo puede realizarse de forma absoluta o de forma simbólica, lo que significa que no es posible mezclar ambos tipos de direccionamiento en una misma instrucción.

#### Ejemplo

```
L DB100.DBW6  
L DB_MOTOR.REVOLUCIONES
```

DB\_MOTOR es el símbolo o nombre simbólico del bloque de datos DB 100 que está definido en la tabla de símbolos; REVOLUCIONES es un operando de datos que está declarado en el bloque de datos, con lo cual la indicación simbólica del operando de datos (DB\_MOTOR.REVOLUCIONES) es igual de unívoca que la indicación absoluta (DB100.DBW6).

El acceso a datos mediante direccionamiento completo sólo se puede realizar a través del registro de bloques de datos globales (registro DB). El editor de AWL crea dos instrucciones cuando se utiliza el direccionamiento completo de operandos de datos:

1. Abrir el bloque de datos a través del registro DB (p. ej.: AUF DB 100)
2. Acceso al operando de datos (p. ej.: L DBW 6)

#### Operaciones disponibles para acceder a datos con direccionamiento completo

Existe la posibilidad de realizar accesos mediante direccionamiento completo con todas las operaciones disponibles para el tipo de datos del operando direccionado.

Al asignar operandos de datos a los parámetros del bloque en secuencias de llamadas se puede indicar la dirección completa, lo que se recomienda encarecidamente, ya que al efectuar llamadas puede cambiar el bloque de datos. Indicando la dirección completa evitará que el operando de datos direccionado no sea el correcto, es decir, que pertenezca a otro bloque de datos que el deseado.

**Riesgos del  
“direccionamiento  
incompleto”**

En principio se puede acceder a los operandos de datos de la misma manera que en STEP 5 (“direccionamiento incompleto”).

Ejemplo:

```
L DBW 6  
L REVOLUCIONES
```

Sin embargo, en STEP 7 el direccionamiento incompleto puede acarrear problemas, ya que STEP 7 modifica los registros de las CPU S7-300/S7-400 con determinadas operaciones. En algunos casos se sobrescribe el número del DB en el registro DB.

En las situaciones listadas a continuación existe riesgo de que se sobrescriba el registro DB, por lo que recomendamos prestar particular atención:

- El registro DB se sobrescribe cuando se accede a los datos con direccionamiento completo.
- Cuando se llama a un FB, se sobrescribe el registro DB del bloque invocante.
- Después de una instrucción de llamada a una FC que transfiera un parámetro de un tipo de datos compuesto (p. ej.: STRING, DATE\_AND\_TIME, ARRAY, STRUCT o UDT), se sobrescribe el contenido del registro DB del bloque invocante.
- Después de asignar a una FC un parámetro actual que esté depositado en un DB (p. ej.: DB100.DBX0.1), STEP 7 abrirá el DB (DB 100) sobrescribiendo el contenido del registro DB.
- Después de que un FB haya direccionado un parámetro de entrada/salida de un tipo de datos compuesto (p. ej.: STRING, DATE\_AND\_TIME, ARRAY, STRUCT o UDT), STEP 7 utilizará el registro DB para acceder a los datos, con lo cual se sobrescribe el contenido del registro DB.
- Después de que una FC haya direccionado un parámetro (entrada, salida o entrada/salida) de un tipo de datos compuesto (p. ej.: STRING, DATE\_AND\_TIME, ARRAY, STRUCT o UDT), STEP 7 utiliza el registro DB para acceder a los datos, con lo cual se sobrescribe el registro DB.

### 3.13.4 Direccionamiento indirecto

El direccionamiento indirecto con ayuda de la función de elaboración de S5 ha sido sustituido en S7 por operaciones que utilizan un direccionamiento indirecto por memoria e indirecto por registro.

#### Formato de los punteros en STEP 5

En S5 el puntero para la operación indizada de elaboración ocupa una palabra. La figura 3-7 muestra la estructura del puntero:

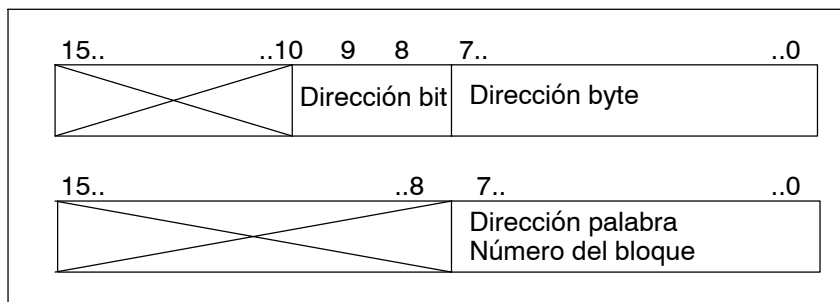


Figura 3-7 Estructura de los punteros S5

#### Formato de los punteros en STEP 7

En S7, los punteros pueden tener dos formatos distintos: palabra y palabra doble.

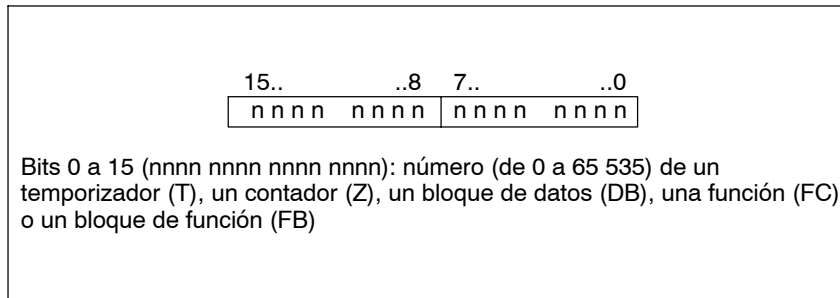


Figura 3-8 Puntero en formato de palabra para el direccionamiento indirecto por memoria

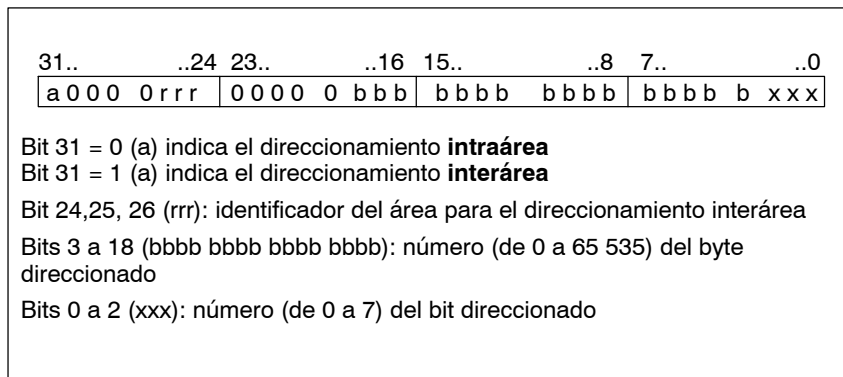


Figura 3-9 Puntero en formato de palabra doble para el direccionamiento indirecto por memoria e indirecto por registro

**Direccionamiento indirecto por memoria**

El direccionamiento indirecto por memoria corresponde al direccionamiento indirecto de S5. El operando indica la dirección del valor que deberá procesar la operación. El operando se compone de los elementos siguientes:

- el identificador del operando (p. ej.: “EB” para ”byte de entrada”) y
- una palabra que contenga el número de un temporizador (T), un contador (Z), un bloque de datos (DB), una función (FC) o un bloque de función (FB) o
- una palabra doble que contenga la dirección exacta de un valor del área de memoria indicada por el identificador.

En este tipo de direccionamiento, el operando indica la dirección del valor o del número indirectamente, es decir, a través del puntero. Esta palabra o palabra doble se puede encontrar en una de las áreas siguientes:

- Marcas (M)
- Bloque de datos (DB)
- Bloque de datos de instancia (DI)
- Datos locales (L)

Una ventaja del direccionamiento indirecto por memoria es que permite modificar el operando de la instrucción dinámicamente durante la ejecución del programa.

**Ejemplo**

El ejemplo siguiente muestra cómo utilizar un puntero en formato de palabra:

AWL S5	AWL S7	Explicación
L KB 5	L +5	Cargar el valor 5 en formato de número entero en el ACU 1.
T MW 2	T MW 2	Transferir el contenido del ACU 1 a la palabra de marcas MW 2.
B MW 2		
L T 0	L T [MW 2 ]]	Cargar el valor del temporizador T 5.

Los dos ejemplos siguientes muestran cómo utilizar un puntero en formato de palabra doble:

AWL S5	AWL S7	Explicación
L KB 8 T MB 3 L KB 7 T MB 2 B MW 2 U E 0.0 B MW 2 = A 0.0	L P#8.7 T MD 2    U E [MD 2]  = A [MD 2]	Cargar 2#0000 0000 0000 0000 0000 0000 0100 0111 (valor binario) en el ACU 1 (S7). Guardar la dirección 8.7 en la palabra de marcas MW 2 (S5) / palabra doble de marcas MD 2 (S7).  El autómata lee la entrada E 8.7 y escribe su estado de señal en la salida A 8.7.

AWL S5	AWL S7	Explicación
L KB 8 T MW 2 B MW 2 L EB 0 B MW 2 T MW 0	L P#8.0 T MD2   L EB [MD2]  T MW [MD2]	Cargar 2#0000 0000 0000 0000 0000 0000 0100 0000 (valor binario) en el ACU 1 (S7). Guardar la dirección 8 en la palabra de marcas MW 2 (S5) / palabra doble de marcas MD 2 (S7). El autómata carga el byte de entrada EB 8 y transfiere su contenido a la palabra de marcas MW 8.

### Uso de la sintaxis correcta

En caso de utilizar un operando indirecto por memoria que esté guardado en el área de memoria del bloque de datos, deberá abrir primero el bloque de datos con la operación “Abrir bloque de datos”. A continuación puede utilizar la palabra de datos o la palabra doble de datos como operando indirecto, como muestra el ejemplo:

```
AUF          DB 10
L            EB [DBD 20]
```

Antes de acceder a un byte, a una palabra o a una palabra doble, asegúrese de que el número del bit del puntero sea “0”.

### Direccionamiento indirecto por registro

En STEP 7 los registros de direccionamiento AR 1 y AR 2 se utilizan para el direccionamiento indirecto por registro.

En este tipo de direccionamiento, el operando indica la dirección del valor que procesará la operación. El operando se compone de los elementos siguientes:

- un identificador del operando,
- un registro de direcciones y un puntero para indicar un offset o desplazamiento que se sumará al contenido del registro de direccionamiento con el fin de determinar la dirección exacta que deberá procesar la dirección. El puntero se indica de la manera siguiente P#Byte.Bit.

El operando señala a la dirección del valor indirectamente, es decir, a través del registro de direccionamiento más el offset.

Una operación que utiliza el direccionamiento intraárea e indirecto por registro, no modifica el valor del registro de direccionamiento.

Para más información al respecto consulte el manual AWL /232/.





## 2ª parte: Conversión de programas

---

Procedimiento

---

**4**

---

Preparando la conversión

---

**5**

---

Conversión

---

**6**

---

Retocar el programa  
convertido

---

**7**

---

Compilación

---

**8**

---

Ejemplo de aplicación

---

**9**



# Procedimiento

# 4

La programación con el lenguaje AWL de S7 es altamente compatible con el AWL de S5, como el KOP de S7 con el KOP de S5 o el FUP de S7 con el FUP de S5. Si usted es usuario de S5 y desea utilizar sus programas en S7 lo tiene muy fácil, puesto que puede seguir utilizando sus programas y convertirlos en programas de S7.

## ¿Cómo proceder?

La lista siguiente muestra cómo proceder al convertir sus programas e indica los capítulos del manual donde encontrará la información que necesita.

La lista hay que entenderla a modo de ejemplo (algunos pasos no deben realizarse necesariamente).

## 4.1 Analizar el sistema S5

Antes de convertir el programa S5 debería asegurarse de que cumple las condiciones necesarias para poderlo convertir.

### **Funcionalidad de los módulos (ver capítulo 2)**

¿Cómo se realiza la funcionalidad de los módulos S5 utilizados? ¿Se pueden utilizar módulos S5 con cápsulas de adaptación o interfases en el S7? ¿Se pueden sustituir los módulos S5 por módulos S7?

### **Ajustes del sistema (v. apt. 3.8)**

¿Cómo se realizan los ajustes de sistema en S7?

### **Juego de operaciones (v. apt. 3.12)**

¿Cómo se aplica a la CPU S7 el juego de operaciones de la CPU S5?

Cuando hay instrucciones que no se pueden convertir aparece un mensaje que indica la posición de la instrucción en el programa. Estas instrucciones deben ser reprogramadas por el usuario.

### **Software estándar (v. apt. 3.9)**

¿Están disponibles también como funciones S7 los bloques de función estándar S5 del programa a convertir?

El paquete básico del software S7 incluye asimismo los paquetes de software estándar para aritmética en coma flotante, funciones de señales, funciones integradas, funciones básicas y funciones matemáticas.

### **Funciones estándar (v. tabla a partir de la pág. 3-22)**

¿Se pueden sustituir las funciones especiales integradas que se utilicen en el programa S5?

**¿Qué partes del programa S5 hay que reprogramar en S7?**

Por lo general no es posible convertir el programa completo. Los puntos siguientes le ayudarán a transcribir el programa S5 con el convertidor o a crear su programa S5 con S7.

- Los programas que solamente contienen combinaciones digitales y binarias no necesitan ser retocados.
- S7 no permite direccionar operandos con direcciones absolutas, por lo que no es posible convertir las instrucciones correspondientes (p. ej.: LIR, TIR, ...). En caso de que el programa emplee muchas direcciones absolutas habrá que reescribir estas secciones del programa y, si conviene, reescribirlo todo.
- Si bien algunas funciones de elaboración (o procesamiento) (p. ej.: B MW, B DW) se pueden convertir, ahorrará espacio de memoria si reprograma estas funciones en S7. Obtendrá el mismo resultado que en S5 si utiliza el direccionamiento indirecto.
- Al realizar llamadas a bloques deberá comprobar y adaptar la ocupación de los parámetros, ya que los parámetros actuales se adoptan sin cambios durante la conversión.

## 4.2 Crear un proyecto S7

STEP 7 ofrece dos métodos distintos para crear un proyecto:

### **Crear un proyecto con el asistente de STEP 7**

El asistente de STEP 7 le ayudará a crear rápidamente un proyecto de STEP 7 con la CPU que vaya a utilizar. Seguidamente puede proceder a escribir su programa.

### **Crear un proyecto manualmente**

Además existe la posibilidad de crear el proyecto manualmente. Este procedimiento se explica en el apartado 3.3.1.

## 4.3 Configurar el hardware

Ahora conviene configurar el hardware, ya que con la herramienta de configuración HW-Config se definen datos que ya se pueden utilizar para preparar la conversión.

No obstante, si aún no desea decidirse por una configuración de hardware en particular, puede realizar la configuración en otro momento.

### **Definir el hardware**

Sirviéndose de las informaciones contenidas en el capítulo 2 (hardware) puede seleccionar los módulos S7 o S5 que necesite para su configuración y rellenar la tabla de configuración del hardware (v. apt. 3.4).

### **Asignar direcciones**

Las direcciones de los módulos se asignan en HW-Config automáticamente. Estas direcciones ya las puede adaptar en la conversión.

### **Realizar los ajustes del sistema**

Al parametrizar la CPU con la herramienta HW-Config puede llevar a cabo los ajustes del sistema que se realizaron en el S5 con el DB 1/DX 0 o con los servicios del sistema (v. apt. 3.4).

### **Ajustar la remanencia**

En los datos de parametrización de la CPU también puede ajustar las áreas remanentes. La capacidad de remanencia depende, sin embargo, de la pila de respaldo (v. apt. 3.4).

# 5

## Preparando la conversión

### Resumen

Preparar los datos necesarios (v. apt. 5.1)	<ul style="list-style-type: none"><li>• Archivo de programa &lt;nombre&gt;ST.S5D</li><li>• Lista de referencias cruzadas &lt;nombre&gt;XR.INI</li><li>• Lista de asignación opcional &lt;nombre&gt;Z0.SEQ</li></ul>
Comprobar los operandos (v. apt. 5.2)	<ul style="list-style-type: none"><li>• Número de operandos</li><li>• Número de bloques</li></ul>
Preparar el programa S5 (v. apt. 5.3)	<ul style="list-style-type: none"><li>• Evaluar y borrar los bloques de datos DB 1 / DX 0</li><li>• Eliminar las llamadas a bloques integrados</li><li>• Eliminar los accesos al área de datos del sistema</li><li>• Adaptar las áreas de operandos</li><li>• Asignar macros a secciones no convertibles del programa</li><li>• Eliminar bloques de datos sin estructura hasta una palabra de datos</li></ul>
Crear macros (v. apt. 5.4)	<ul style="list-style-type: none"><li>• Macros de instrucciones</li><li>• Macros de OBs</li></ul>

## 5.1 Preparar los archivos necesarios

Como punto de partida para la transcripción del programa S5 se requieren los datos siguientes:

- el archivo de programa <nombre> ST.S5D y
- la lista de referencias cruzadas <nombre> XR.INI.

La lista de referencias cruzadas es necesaria para mantener la estructura del programa y la jerarquía de llamadas del programa S5.

### **Indicaciones opcionales**

Si desea utilizar nombres simbólicos en lugar de operandos absolutos en su programa, necesitará además la lista de asignación S5 <nombre> Z0.SEQ para poder generar la lista de asignación convertida.

### **Procedimiento**

Prepare la conversión como sigue:

1. Actualice la lista de referencias cruzadas de su programa S5 con ayuda del software S5.
2. Copie su archivo de programa STEP 5, la lista de referencias cruzadas y, dado el caso, la lista de asignación en un directorio del DOS.



## 5.2 Comprobar los operandos

### Funcionalidad de la CPU

En determinados casos es necesario adaptar el programa a convertir a la CPU S7 que vaya a utilizar.

Para hacerse una idea de la funcionalidad que ofrece la CPU S7, proceda como sigue:

1. Decida qué CPU S7 desea utilizar.
2. Busque la CPU S7 en las tablas de datos característicos que figuran en el apartado 2.2.1 y compare
  - el número de operandos y
  - el número de bloques

con los operandos y bloques utilizados;

o bien:

1. Abra el Administrador SIMATIC.
2. Elija la CPU S7 en la vista online de la estructura del proyecto.
3. Abra con el comando de menú **Sistema de destino > Información del módulo** un diálogo con fichas, las cuales le proporcionarán las siguientes informaciones:
  - en la ficha **General** puede indentificar el tipo de CPU, consultar su capacidad de memoria y averiguar el tamaño de las áreas de operandos disponibles;
  - en la ficha **Bloques** puede ver los bloques disponibles. Además se indica el número máximo de tipos de datos, su longitud y una lista de los OBs, SFBs y SFCs disponibles en la CPU.

### Adaptar el programa a convertir

Para poder ejecutar en la CPU S7 el programa AWL que desea convertir, tiene que comprobar si éste respeta el número de bloques y operandos permitido y, en caso negativo, modificarlo.

### 5.3 Preparar el programa S5

Antes de convertir el programa STEP 5, puede prepararlo para su futuro empleo como programa STEP 7 (si bien no es obligatorio; todas las correcciones las puede realizar en el archivo fuente de STEP 7 después de la conversión). Con esta adaptación se reduce el número de mensajes de error y de advertencias.

Antes de la conversión puede realizar, por ejemplo, las siguientes adaptaciones:

- Evaluar los ajustes de sistema en los bloques de datos con propiedades de programa DB 1 o DX 0 y borrar después el DB 1 o DX 0.
- Eliminar todas las llamadas a bloques integrados o los accesos al área de datos del sistema operativo cuya funcionalidad se pueda conseguir parametrizando la CPU S7.
- Adaptar las áreas de operandos “Entradas”, “Salidas” y “Periferia” a las direcciones (nuevas) de los módulos con la función de STEP 5 “Reasignar” (no exceder el margen de direcciones de STEP 5, pues de lo contrario ya se señalará un error durante la primera fase de la conversión, con lo cual no se convertirán estas instrucciones).
- Aquellas secciones del programa que no sean convertibles y que se repitan a lo largo del programa se pueden borrar hasta dejar una instrucción STEP 5 “unívoca” por cada sección del programa. Asigne entonces esta instrucción “unívoca” a una macro (v. apt. 5.4), que reemplace esta sección.
- Si su programa contiene demasiados bloques de datos (y muy extensos) que no presenten estructura alguna (p. ej.: los que se utilizan de búfer de datos), puede borrar las palabras de datos de estos bloques de datos hasta dejar una sola palabra. Después de la conversión (y antes de la compilación) programe el contenido de estos bloques en el archivo fuente declarando un array, p. ej.: un búfer (ARRAY [1..256] of WORD).

El programa de conversión no sólo se puede utilizar para convertir programas enteros, sino también para convertir bloques.

## 5.4 Crear macros

**Objetivo** Para la conversión se pueden definir macros en el programa de conversión S5/S7 para

- las instrucciones de S5 que no se convierten automáticamente y
- para las instrucciones S5 que desea convertir de forma distinta de la estándar.

Las macros resultan muy útiles cuando el programa S5 contiene instrucciones S5 de las características arriba mencionadas.

**Función de macros** Las macros pueden sustituir:

- instrucciones S5 (operadores) y
- bloques de organización S5 (OB).

Las macros se depositan en el archivo S7S5CAPA.MAC para el juego de instrucciones SIMATIC, y en el archivo S7S5CAPB.MAC para el juego de instrucciones internacional. Si su programa utiliza ambos juegos de instrucciones, deberá indicar las macros en ambos archivos. Cabe distinguir las macros de instrucciones y las macros de OBs. Se pueden crear 256 macros de instrucciones y de OBs, respectivamente.

### 5.4.1 Macros de instrucciones

Las macros de instrucciones tienen que tener la estructura siguiente:

```
$MAKRO: <Instrucción S5>
```

```
Secuencia de instrucciones S7
```

```
$ENDMAKRO
```

Al definir la macro, en <Instrucción S5> tiene que indicar la instrucción completa (operador y operando absoluto).

La tabla muestra una macro para la instrucción E DB 0, con la que se crean bloques de datos en S5. La longitud (en palabras) del bloque de datos a crear figura en el ACU 1. Esta función se realiza en S7 con la función de sistema SFC 22 CREAT\_DB. La longitud del bloque de datos se tiene que indicar en número de bytes.

Tabla 5-1 Ejemplo de una macro de instrucciones

Macro	S5	S7
\$MAKRO: E DB 0 //Sustituye la instrucc //ción para crear un DB	L Constante B MW 100	L Constante;
SLW 1 //Convertir número de pa- //labras en número de //bytes	E DB 0	SLW 1; T MW 102; CALL SFC 22( LOW_LIMIT := MW 100, UP_LIMIT := MW 100, COUNT := MW 102, RET_VAL := MW 106, DB_NUMBER := MW 104);
T MW 102		
CALL SFC 22( // Llamada a SFC CREAT_DB LOW_LIMIT := MW 100, UP_LIMIT := MW 100, COUNT := MW 102, RET_VAL := MW 106, DB_NUMBER := MW 104);		
\$ENDMAKRO		

## 5.4.2 Macros de OBs

Debido a las diferencias que presentan los bloques de organización del S5 y los del S7 conviene convertir a mano los OBs del programa S5. Las macros de OBs tienen que tener la siguiente estructura:

```

$OBCALL: <Número del OB>
CALL <Función de sistema S7>;
$ENDMAKRO

```

Si el archivo fuente S5 contiene una instrucción con el operando OB x, ésta se sustituye por la macro definida, a excepción de las llamadas a FBs que utilizan OBs como parámetros formales.

Tabla 5-2 Ejemplo de una macro de OB

Macro	S5	S7
<pre> \$OBCALL: 31 //Sustituye instrucciones            //con OB 31 CALL SFC 43; \$ENDMAKRO </pre>	<pre> SPA OB 31 </pre>	<pre> CALL SFC 43; </pre>

### Indicaciones para crear macros de OBs

Las funciones de los bloques de organización del S5 se distinguen de las funciones de los OBs del S7. Los OBs que no se convierten automáticamente tienen que ser retocados a posteriori, es decir, sustituidos por:

- OBs de funcionalidad distinta,
- nuevas instrucciones de S7 o
- ajustes de sistema que se definen al parametrizar el hardware.

Para más información sobre cómo sustituir los OBs del S5, consulte el apt. 3.7.5.

### Nota

Es importante señalar que no se comprueba si una macro ha sido definida más de una vez. Pero, si se da el caso, se utiliza la primera macro definida. Tampoco se comprueba si la secuencia de instrucciones S7 indicada es correcta. No cometa errores de sintaxis en las palabras clave y en los caracteres especiales (p. ej.: no se olvide los dos puntos).

### 5.4.3 Editar macros

Las macros se crean de la manera siguiente:

- Arranque el programa de conversión S5/S7 haciendo clic en el botón “Inicio” de la barra de tareas de Windows 95 y luego en ”Simatic/STEP 7/Convertir archivo S5”.
- Elija el comando de menú **Edición > Macro de reemplazo** (¡no hay ningún archivo de programa abierto!).

**Resultado:** se abre el archivo S7S5CAPA.MAC.

- Introduzca las macros como se indica más arriba y guarde el archivo con el comando de menú **Archivo > Guardar**.
- Cierre el archivo con el comando de menú **Archivo > Cerrar**.

**Resultado:** se cierra el archivo S7S5CAPA.MAC. Las macros definidas tendrán efecto a partir de la siguiente fase de la conversión.

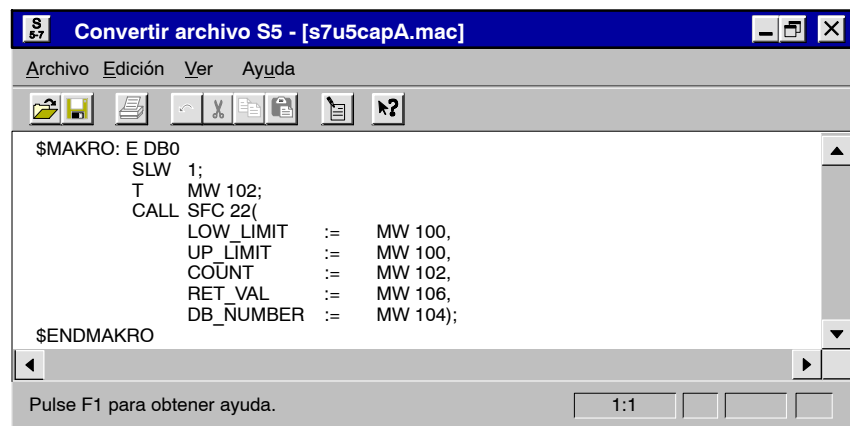


Figura 5-1 Macro en la ventana de "Convertir archivo S5"

# 6

## Conversión

### 6.1 Iniciar la conversión

#### Requisitos

Antes de iniciar la conversión tiene que asegurarse de que el archivo S5 a convertir, así como la lista de referencias cruzadas y, dado el caso, la lista de asignación, se encuentren en el mismo directorio (v. apt. 5.1: *Preparar los archivos necesarios*).

#### Arrancar el convertidor de archivos S5/S7

Después de instalar el software STEP 7 en la PG, arranque el programa “Convertir archivo S5” partiendo del botón de comando “Inicio” de la barra de tareas de Windows 95:

- la vía de acceso al comando de menú es ”Simatic/STEP 7/Convertir archivo S5”.

El programa de conversión S5/S7 visualiza la siguiente pantalla inicial:

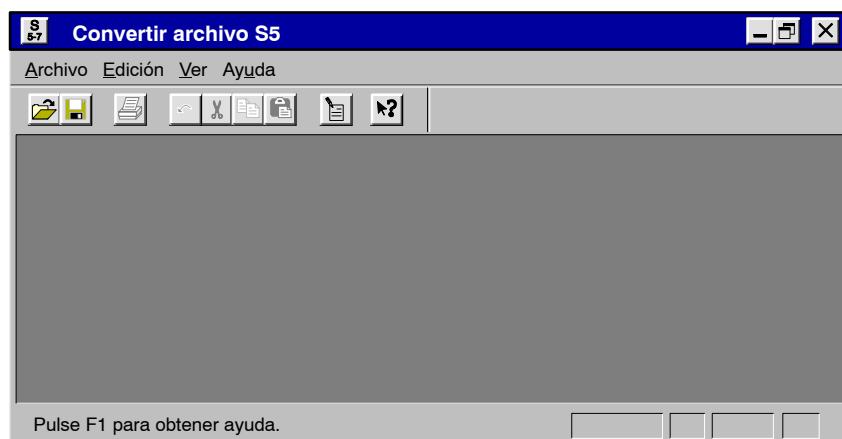


Figura 6-1 Pantalla inicial del programa de conversión S5/S7

**Seleccionar un archivo de programa**

Para seleccionar un archivo de programa, proceda como sigue:

1. Elija el comando de menú **Archivo > Abrir**.
2. Elija la unidad y el directorio donde se encuentren los archivos a convertir.
3. Seleccione el archivo a convertir y pulse el botón de comando “Aceptar” para confirmar la selección.

**Resultado:** el programa de conversión S5/S7 muestra los archivos fuente y los archivos de destino y una lista de asignación de los números viejos y nuevos de los bloques.

La figura muestra el cuadro de diálogo ”Convertir archivo S5 [<nombre>ST.S5D]”.

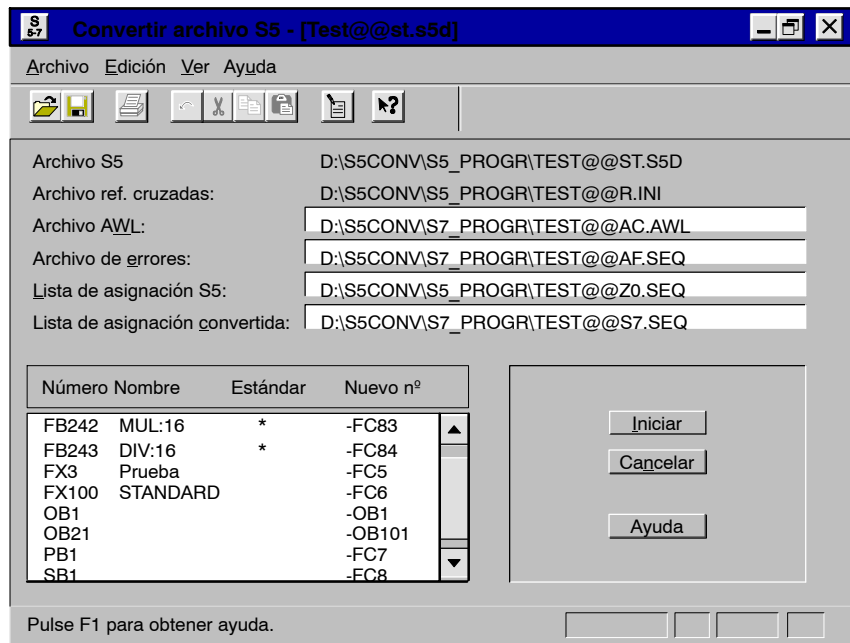


Figura 6-2 Cuadro de diálogo ”Convertir archivo S5 [<nombre>ST.S5D]”

**Cambiar el nombre del archivo de destino**

En caso necesario, puede modificar el nombre de los archivos de destino ”Archivo AWL”, ”Archivo de errores” y ”Lista de asignación convertida” que propone el software. Esto puede ser necesario cuando el editor con el que desea retocar el archivo convertido le exija respetar determinadas convenciones para el nombre (p. ej.: NOMBRE.TXT).

Proceda como sigue:

- Haga clic en el campo de texto que indica la ruta del archivo de destino que desea cambiar.
- Cambie el texto a voluntad.



**Asignación  
Número ->  
Nuevo nº**

El software propone nuevos números para los bloques a convertir y los visualiza en el cuadro de diálogo "Convertir archivo S5 [<nombre>ST.S5D]". Si desea asignar otros nombres, proceda como sigue:

1. Haga doble clic en el número del bloque que desea modificar.
2. Introduzca el nuevo número en el cuadro de diálogo "Nuevo número de bloque" y haga clic en el botón de comando "Aceptar" para confirmar la modificación realizada.

**Bloques de función estándar S5**

Si su programa S5 contiene bloques de función estándar, éstos aparecerán marcados con un asterisco en la columna "Estándar".

**Iniciar la  
conversión**

Haciendo clic en el botón de comando "Iniciar" se da inicio al proceso de conversión. Este proceso se divide en dos fases de ejecución más la fase de conversión de la lista de asignación.

En la primera fase de la conversión, el programa S5 se convierte en una fuente S5 con todos los bloques y comentarios.

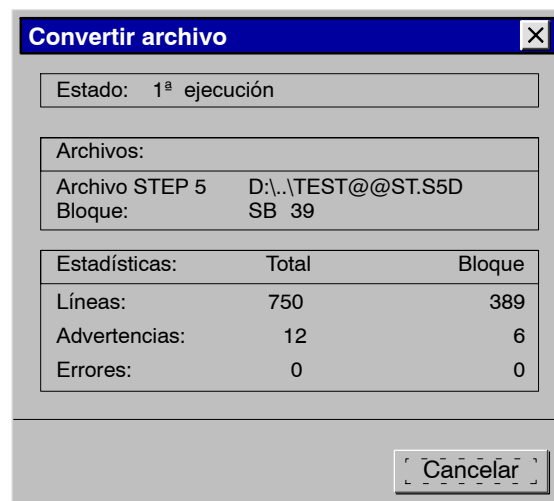


Figura 6-3 Primera fase de la conversión

En la segunda fase, la fuente S5 se convierte en un archivo fuente AWL con los nuevos tipos y números de bloques y la sintaxis S7.

**Convertir la lista de asignación**

Al convertir la lista de asignación S5, los símbolos de la lista se convierten a un formato importable por el editor de símbolos.



Figura 6-4 Conversión de la lista de asignación

## 6.2 Archivos generados

El programa de conversión S5/S7 genera los siguientes archivos:

- Archivo <Nombre>A0.SEQ:

Este archivo se genera en la primera fase de la conversión y contiene el archivo <Nombre>ST.S5D en formato ASCII.

- Archivo <Nombre>AC.AWL:

Este archivo se genera en la segunda fase de la conversión y contiene el programa AWL. Si se han cometido errores al definir una macro, los mensajes de error correspondientes se generarán también en esta fase.

- Archivo <Nombre>S7.SEQ:

Este archivo se genera al transcribir la lista de asignación y contiene la lista de asignación convertida en un formato importable por el editor de símbolos.

- Archivo de errores ”<Nombre>AF.SEQ”:

Se visualiza en el cuadro de lista en la parte superior de la ventana ”Convertir archivo S5” y contiene los errores y advertencias del programa convertido. Estos mensajes se generan en la primera y segunda fase de conversión y al transcribir la lista de asignación.

Al terminar la fase de conversión se indica el número de errores y advertencias en un cuadro de diálogo.

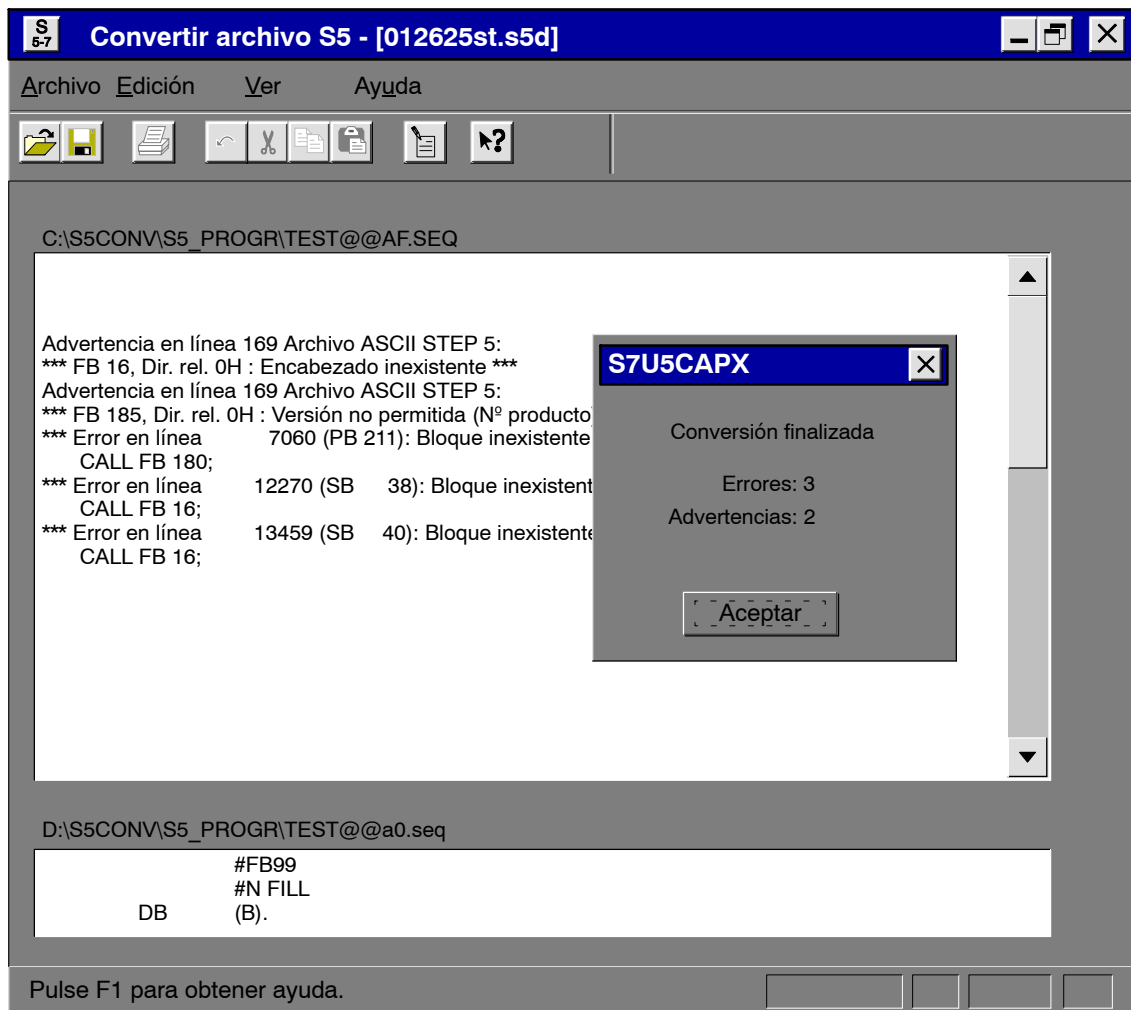


Figura 6-5 Mensajes del programa de conversión

**Localización de errores**

En el cuadro de lista inferior de la ventana puede hacerse mostrar la posición exacta del archivo en el que ha aparecido el error en cuestión.

En el archivo AWL se visualizan mensajes en aquellos puntos del programa en los que se detectaron errores. El archivo contiene además advertencias o indicaciones para los problemas que puedan surgir (p. ej.: al cambiar la semántica de las instrucciones).

**Imprimir mensajes**

Con el comando de menú **Archivo > Imprimir** puede imprimir los archivos creados.

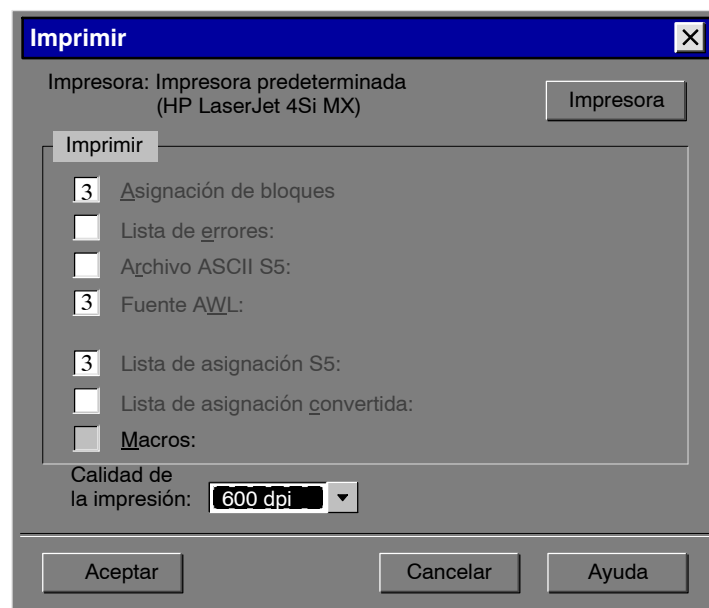


Figura 6-6 Cuadro de diálogo "Imprimir"

### 6.3 Evaluar mensajes de error

#### Analizar los mensajes

Los mensajes que se visualizan durante la conversión pueden ser mensajes de error o advertencias. Para analizar los mensajes proceda como sigue:

- Hágase mostrar en el cuadro de lista inferior de la ventana "Mensajes" el archivo en el que ha aparecido el error.
- Consulte el significado del mensaje en la Ayuda en pantalla.
- Corrija el error ateniéndose a la solución propuesta.

#### Mensajes de error

Los mensajes de error aparecen cuando existen secciones del programa S5 que no se pueden convertir y que solamente se incluyen en el programa S7 como comentarios. La tabla siguiente muestra todos los mensajes de error, su significado y su posible solución.

#### Observaciones sobre las reglas de conversión

El capítulo 3 (Software) contiene las reglas de conversión de programas S5 a S7. Aquí encontrará más indicaciones sobre posibles causas de error, así como ayudas para retocar el programa AWL convertido.

Tabla 6-1 Mensajes de error, su significado y su solución

Mensaje de error	Origen	Significado	Solución
El parámetro absoluto no se corresponde con el identificador del operando	1ª fase	El identificador del operando es incorrecto	Compruebe la instrucción y, dado el caso, corrijala.
Bloque inexistente	1ª fase	Falta el bloque llamado (FB, FX) o bien el bloque figura en la lista de bloques, pero no figura en el archivo de programa.	Compruebe la estructura del programa.
	2ª fase	Se ha llamado a un bloque que no figura en el archivo de programa.	Compruebe si en la conversión se indicó la lista de referencias cruzadas o compruebe la estructura del programa.
Instrucción no permitida en el bloque	1ª fase	P. ej.: salto dentro de un bloque de programa	Compruebe la instrucción.
Instrucción no definida	1ª fase	Instrucción MC5/AWL no válida	Corrija el archivo de programa S5.
	2ª fase	Instrucción inexistente en S7	Edite una macro o sustituya la instrucción por la secuencia de instrucciones S7 que corresponda.
Ya no se pueden realizar accesos de bit a T/Z (¡comprobar!)	2. fase	El programa S5 contiene accesos de bit a temporizadores y contadores.	Compruebe el programa AWL.
CALL OB no está permitido	2. fase	S7 no permite efectuar llamadas a OBs.	En caso necesario, utilice la instrucción CALL SFC .
CALL SFC xy generado, completar lista de parámetros	2. fase	Faltan parámetros para la SFC.	Complete la lista de parámetros SFC.
Archivo inexistente (Continúa en la página siguiente)	general	El archivo seleccionado no existe.	Compruebe el archivo de programa.

Tabla 6-1 Mensajes de error, su significado y su solución

Mensaje de error	Origen	Significado	Solución
Nivel de anidamiento no permitido	1ª fase	No se ha cerrado un paréntesis, es decir, hay más paréntesis abiertos que cerrados.	Observe los niveles de anidamiento (paréntesis) y corrija el error de programación.
Operando no válido	1ª fase	El operando no es compatible con la instrucción.	Compruebe la fuente S5.
	2ª fase	El operando no es compatible con la instrucción.	Corrija el archivo AWL.
Error de conversión	2ª fase	BI sin constante	Indique la constante en la instrucción de carga.
Error en archivo de macros, se ignora macro xy	2ª fase	Error en macro	Compruebe la macro de instrucción
Parámetro formal no definido	1ª fase	Existen más parámetros que en el bloque llamante	Compruebe el archivo de programa S5.
Falta el índice	1ª fase	El archivo de programa no contiene bloques.	Compruebe el archivo de programa.
La longitud del comentario es incorrecta	1ª fase	Error en el archivo S5	Compruebe el archivo de programa.
Comentario demasiado largo	1ª fase	Error en el archivo S5	Compruebe el archivo de programa.
No se ha indicado ningún nombre de bloque	1ª fase	El nombre del bloque está formado por espacios en blanco.	Introduzca el nombre del bloque.
Faltan los derechos de acceso	general	El archivo está protegido contra sobreescritura.	Desactive la protección contra sobreescritura.
Marca no definida	1ª fase	La marca de salto no está definida en el encabezado	Compruebe el archivo S5.
Marca no válida	1ª fase	La marca de salto contiene caracteres no válidos.	Compruebe el archivo S5.
Operador no válido	1ª fase	Operador desconocido en el archivo S5 o no convertible	Sustituya el operador por la instrucción S7 que corresponda.
Operador no válido; quizá pueda sustituirse por la instrucción <code>\*L P# Parámetro formal\*</code>	2ª fase	El operador no se puede cargar así en S7.	Utilice, dado el caso, la instrucción indicada.
Número de parámetros incorrecto	1ª fase	Error en el programa S5	Compruebe el archivo de programa.
Parámetro erróneo	1ª fase	Error en el programa S5	Compruebe el archivo de programa.
Tipo de parámetro erróneo	1ª fase	Error en el programa S5	Compruebe el archivo de programa.
Error de escritura en disquete	general	El archivo está protegido contra escritura o ya no queda memoria libre en el disquete.	Desactive la protección contra escritura o borre los datos que no necesite.
Desbordamiento de la memoria en la PG (problemas de espacio)	1ª fase	El espacio de memoria principal es insuficiente	Borre los archivos que no necesite de la memoria principal.
No se pudo generar la marca de salto	2ª fase	Instrucción SPR más allá del límite del bloque	Elimine el error en el programa S5.
Se ha convertido un código MC5 no válido	1ª fase	Conversión de una instrucción S5 de una versión antigua.	ninguna

**Advertencias**

Las advertencias aparecen cuando se convierten partes del programa S5 que deben comprobarse.

Tabla 6-2 Advertencias, su significado y su solución

Advertencia	Origen	Significado	Solución
Versión no permitida (nº de producto)	1ª fase	Sustituir el bloque de función estándar S5 por un bloque FC S7.	ninguna
Versión no permitida (bloque de GRAPH5)	1ª fase	Los bloques de GRAPH5 no son convertibles.	Dado el caso, utilice un bloque creado con GRAPH para S7.
Compruebe los ajustes de la base de tiempo	2ª fase	S7 permite ajustar una base de tiempo menor que en S5.	Ajuste la base de tiempo con la función "Configurar hardware" (HW-Config).
I/D solamente influye en el ACU1-L, ahora en todo el ACU1	2ª fase	Los acumuladores de S7 han sido ampliados a 32 bits.	Compruebe las consecuencias que puede tener una instrucción indirecta INCREMENTAR/DECREMENTAR sobre el programa AWL
Tenga en cuenta la nueva numeración de los bloques	2ª fase	Una llamada indirecta a un bloque no considera los nuevos números de los bloques (el número se extrae de la marca o palabra de datos correspondiente)	Cambie la lógica en S5 o utilice llamadas fijas.
Los OB 23 y OB 24 se convierten en el OB 122	2ª fase	Ambos bloques OB 23 y OB 24 se sustituyen en S7 por el OB 122.	Resuma el contenido del OB 23 y del OB 24 en un OB 122 y borre el otro OB 122.
El OB ha sido interpretado como OB 34 del S5-115U	2ª fase	Dependiendo de la CPU utilizada puede variar el significado del OB 34.	Compruebe si este OB es apropiado para su programa.
La máscara DB S5 no se utiliza para parametrizar en S7.	1ª fase	MASK figura en DW0 y DW1.	Parametrice el PLC con STEP 7.
La instrucción de salto que sigue a B no es convertible	2ª fase	La instrucción de elaboración/procesamiento B con SPA no se puede convertir automáticamente.	Sustituya la instrucción en el archivo AWL por SPL y compruebe el salto.
El convertidor S5/S7 no realiza los ajustes de sistema.	2ª fase	Si bien el DB y el DX son convertidos, no significan lo mismo que en S5.	Los ajustes de sistema se tienen que efectuar a mano en la tabla de configuración.
Considerar las distintas instrucciones de STOP	2ª fase	No se hace distinción entre las instrucciones STP, STS y STW.	Compruebe el archivo de programa.
El RLO se pone a 1 (antes VKE)	2ª fase	En S7 se pone a 1 el RLO en las instrucciones S5 SU y RU.	Dado el caso, inserte la instrucción CLEAR.
Falta el preencabezado	1ª fase	Faltan los nombres de las marcas de salto para el FB y el FX. Para el DB y el DX falta el formato de los datos.	Compruebe si los preencabezados se encuentran en otro archivo.
Tratándose de un S5-115U, cambiar a OB 100	2ª fase	El OB de arranque 21 de S5 se convierte automáticamente en el OB 101.	Si el programa S5 se ejecutaba en un S5-115U hay que sustituir el OB 101 por el OB 100.



## Retocar el programa convertido

### Preparativos para retocar el programa

Para retocar el archivo fuente AWL generado es necesario realizar determinados pasos preparativos:

- Imprimir los mensajes.
- Crear primero un programa S7 en un proyecto del Administrador SIMATIC, si es que aún no ha creado ningún programa S7 en un proyecto.
- Importar con el comando de menú **Insertar > Fuente externa** el programa fuente AWL que ha creado en el contenedor “Fuentes” del programa S7 creado.
- Abrir el archivo convertido.

### Retocar el programa

Para retocar el archivo fuente AWL que ha creado, recomendamos proceder de la siguiente manera:

- Recorra el programa en modo interactivo y modifique/complete las instrucciones S5 y los bloques de organización que no se puedan convertir sirviéndose de las advertencias visualizadas (ver 1ª parte del manual).

## 7.1 Cambiar direcciones

Las direcciones que se tienen que cambiar suelen pertenecer a los módulos de entrada/salida. Las direcciones de los módulos se pueden consultar en la tabla de configuración del hardware (HW-Config).

### 7.1.1 Métodos para cambiar las direcciones

#### Reasignar direcciones en S5

Las direcciones de los distintos operandos se pueden cambiar en S5 antes de la conversión mediante la función “Reasignar”, adaptándolas a las nuevas direcciones de S7.

#### Reasignar direcciones en S7

El Administrador SIMATIC ofrece una función de reasignación automática (=recablear) para los bloques generados a partir del archivo fuente.

Procedimiento:

1. Seleccione en el Administrador SIMATIC los bloques del programa en los que desee reasignar direcciones (recablear).
2. Abra la tabla de reasignación con el comando de menú **Herramientas > Recablear**.
3. Introduzca en la tabla tanto las direcciones antiguas como las nuevas de los operandos que deba modificar y guárdela.

Los bloques contendrán a partir de ahora las direcciones modificadas.

#### Cambiar direcciones en el archivo fuente S7

Adapte en su programa los accesos a entradas y salidas, así como los accesos directos a la periferia, a las nuevas direcciones de los módulos S7.

En el archivo fuente S7 puede cambiar las direcciones absolutas simplemente con el comando de menú **Edición > Reemplazar**.

Atención: si la nueva área de direcciones se solapa con la vieja pueden obtenerse resultados indeseados.

#### Crear un archivo fuente S7 (con direccionamiento simbólico)

Si desea utilizar el direccionamiento simbólico puede reasignar las direcciones directamente en la tabla de símbolos.

#### Requisitos

Se tiene que haber compilado el programa sin errores y haber creado una tabla de símbolos que contenga todos los símbolos de las direcciones absolutas que deben ser modificadas.

**Procedimiento**

Para cambiar las direcciones en la tabla de símbolos, proceda como sigue:

- Abra un bloque que contenga las direcciones a modificar y ajuste - con el comando de menú **Herramientas > Preferencias** - la opción **Representación simbólica** en la ficha **Editor**.  
Repita este procedimiento para todos los bloques que contengan direcciones que desee modificar.
- Genere una fuente a partir de los bloques con el comando de menú **Archivo > Generar fuente**. Los bloques se pueden seleccionar en un cuadro de diálogo una vez introducido el nombre de la fuente.

Tenga presente la jerarquía de llamadas al establecer la secuencia de tratamiento de los bloques. Por regla general, los bloques llamados ya tienen que existir, es decir, tienen que insertarse en la fuente delante de los bloques desde los que son llamados.

**Resultado:** en la fuente generada aparecen las instrucciones con direcciones simbólicas.

- Ahora puede realizar el recableado o reasignación en la tabla de símbolos: sustituya las direcciones S5 que han cambiado por las nuevas direcciones S7.
- Después de compilar el archivo fuente, los bloques contendrán las nuevas direcciones.

**7.2 Funciones no convertibles**

Los operandos y las operaciones que no se pueden convertir se incluyen en el programa S7 generado en calidad de comentario y tienen que ser retocados o adaptados por el usuario.

Existen dos métodos distintos:

- Defina para estos operandos y operaciones (si es que aparecen en el programa de usuario) secuencias de instrucciones AWL S7 (macros) que se utilizarán en la conversión.
- Edite las secuencias de instrucciones apropiadas en el programa S7 resultante.

Qué método resulta más adecuado depende, entre otras cosas, de la frecuencia con que aparezcan instrucciones no convertibles en el programa de usuario.

En los apartados 3.11 y 3.12 puede consultar los operandos y las operaciones no convertibles y además se le propondrán distintos métodos para transcribir funciones no convertibles en S7.

### 7.3 Direccionamiento indirecto: conversión

El programa de conversión S5/S7 convierte el direccionamiento indirecto con instrucciones del tipo B MW y B DW mediante instrucciones de STEP 7. La secuencia de instrucciones generada suele ser muy extensa, ya que el puntero de STEP 5 se ha de convertir al formato de STEP 7, siendo necesario memorizar de forma intermedia el contenido de los acumuladores y de la palabra de estado.

Si su programa utiliza con frecuencia el direccionamiento indirecto, conviene adaptarlo al direccionamiento indirecto de STEP 7. Reprogramándolo adecuadamente se ahorra mucho espacio de memoria.

A continuación se explica cómo convierte el programa de conversión S5/S7 el direccionamiento indirecto en los diferentes casos:

#### **Temporizadores y contadores**

El direccionamiento indirecto de temporizadores y contadores se convierte en direccionamiento indirecto por memoria utilizando una palabra de datos local y temporal.

#### **Bloques**

El direccionamiento indirecto de bloques se convierte en direccionamiento indirecto por memoria utilizando una palabra de datos local y temporal.

Los nuevos números de los bloques no se tienen en cuenta en la conversión, por lo que deben ser corregidos.

#### **Operandos**

El direccionamiento indirecto de operandos se convierte bit por bit y palabra por palabra en direccionamiento indirecto por registro utilizando el registro de direccionamiento AR1 y datos locales temporales para almacenar de forma intermedia el contenido de la palabra de estado, así como el ACU 1 y el ACU 2.

#### **Direccionamiento indirecto a través del registro RB**

Las instrucciones no se convierten, por lo que habrá que reprogramar el direccionamiento indirecto en S7.

#### **Otros tipos de direccionamiento indirecto**

Reprogramar las instrucciones en S7.

Para más información sobre el direccionamiento indirecto v. apt. 3.13.4.

## 7.4 Accesos directos a la memoria

En STEP 5 se utilizaban accesos a direcciones absolutas de la memoria para determinadas funciones; este tipo de accesos ya no existe en STEP 7.

STEP 5	STEP 7
Direccionamiento de operandos de datos en bloques de datos “extralargos”	El direccionamiento de operandos de datos cuya dirección sea superior a 255 se puede realizar ahora con instrucciones normales (L, T, ...).
Direccionamiento indirecto con el registro RB	El direccionamiento indirecto se puede realizar con el direccionamiento indirecto por registro (v. <i>Direccionamiento indirecto</i> en el apt. 3.13.4 y el <i>Manual de AWL [232]</i> ).
Transferencia en bloque	Para la transferencia en bloque existe ahora una función de sistema: la SFC 20 BLKMOV. Las áreas de memoria a copiar se indican en los parámetros del bloque. Si las áreas de memoria son variables es posible asignarles punteros ANY a los parámetros, y ocuparlos luego en el programa de usuario.

## 7.5 Asignación de parámetros

### Instrucción S5 B<parámetro del bloque>

Dependiendo del tipo de bloque transferido, la instrucción B <parámetro formal del tipo “B”> se ejecuta en el S5 como sigue:

- “SPA bloque lógico” o
- “A DB bloque de datos”.

Como en el parámetro formal no se indica el tipo de datos no se puede realizar una conversión automática. Analice, pues, si su programa contiene instrucciones X con parámetros del tipo “B” y, en caso afirmativo, transcriba las instrucciones a mano.

### Parámetros actuales

El programa de conversión S5/S7 adopta los parámetros actuales de los bloques de función parametrizados en las llamadas a bloques sin modificación alguna. Si ha asignado direcciones con un parámetro actual, tendrá que comprobar esta asignación y, dado el caso, modificarla.

Ejemplo:

- Indicación del número de una palabra de datos:  
se tiene que convertir a direccionamiento por bytes.
- Indicación de una dirección de la periferia:  
se tiene que utilizar la nueva dirección del módulo.
- Transferencia de un bloque:  
se le tiene que dar el nuevo número de bloque.

## 7.6 Funciones estándar

### **Bloques de función estándar S5**

Si su programa S5 contiene bloques de función estándar, éstos se indicarán

- antes de la conversión mediante un asterisco en la columna “Estándar” del cuadro de diálogo “Convertir archivo S5 [<nombre>ST.S5D]” y
- después de la conversión mediante el mensaje “Versión no permitida (nº de producto)”.

El paquete de software estándar S7 incluye funciones S7 ya convertidas (ex-bloques de función estándar S5) para aritmética en coma flotante, funciones de señales, funciones integradas, funciones básicas y funciones aritméticas que van de la FC 61 a la FC 125 (v. apt. 3.9).

### **Insertar las FCs**

Para integrar las funciones S7 en el programa S7 proceda así:

1. Abra el proyecto en el que desee insertar las funciones.
2. Abra la librería estándar del Administrador SIMATIC que contiene las funciones S5 convertidas (StdLib30).
3. Copie en su programa S7 las funciones S7 que necesite de la librería estándar.

# Compilar

# 8

Para poder ejecutar el programa convertido y quizás ya retocado hay que compilarlo primero con el compilador AWL. Proceda de la misma manera que al compilar un archivo de texto recién creado.

## Comprobar la coherencia

Con el comando de menú **Archivo > Comprobar coherencia** puede comprobar en cualquier momento la sintaxis y la coherencia del archivo fuente sin que se creen bloques. Al comprobar la coherencia se analiza:

- la sintaxis
- los nombres simbólicos y
- la existencia de los bloques llamados en el programa.

Después de esta comprobación se obtiene un informe de compilación que indica el nombre del archivo compilado y el número de líneas compiladas, así como el número de errores y advertencias aparecidas.

## Compilar el archivo fuente

Con el comando de menú **Archivo > Compilar** se convierte el archivo fuente en uno o varios bloques ejecutables.

Después de la compilación aparece un informe con los resultados obtenidos. Los errores se indican de la misma manera que al comprobar la coherencia. En caso de que un archivo fuente contenga varios bloques, solamente se compilarán y guardarán los que no presenten errores.

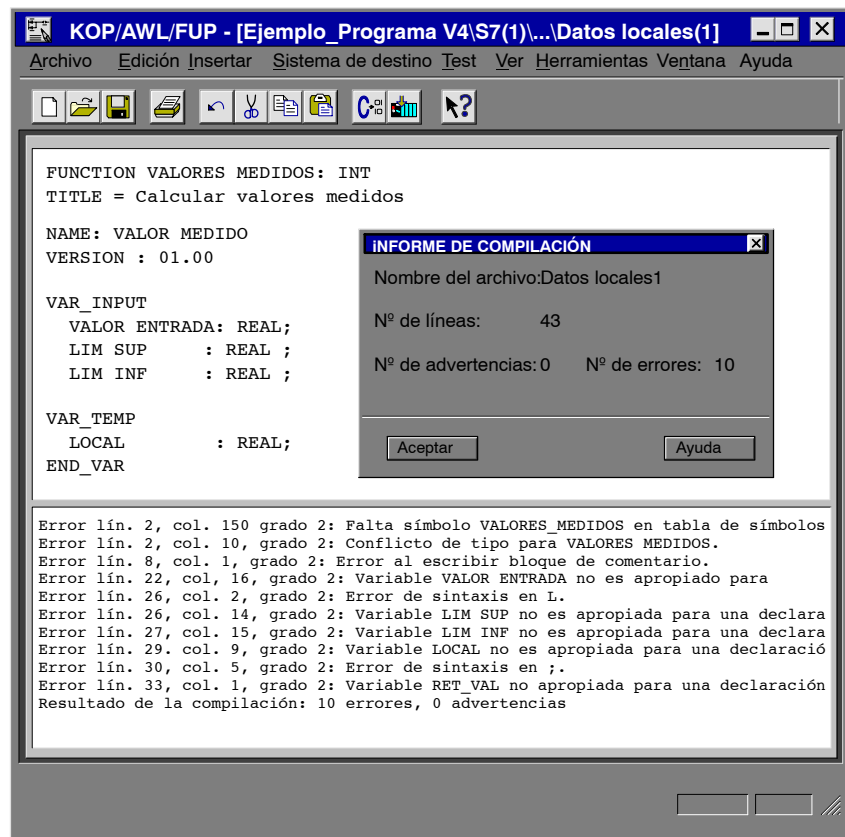


Figura 8-1 Comprobación de la coherencia y compilación de archivos fuente

### Eliminación de errores

Si el programa convertido presenta errores y/o advertencias, éstos se visualizarán - después de comprobar la coherencia o de compilar el programa - al final del archivo fuente en una segunda ventana, indicándose la causa del error. Si selecciona uno de dichos mensajes de error, el programa de conversión le indicará la posición exacta del error en el archivo fuente. Este vínculo entre el mensaje de error y la posición del mismo le permitirá eliminar rápidamente el error en cuestión.

Las correcciones o modificaciones se pueden realizar en el modo de edición "Sobreescribir". Para cambiar del modo "Sobreescribir" al modo "Insertar", y viceversa, utilice la tecla INTRO.



## Ejemplo de aplicación

El presente capítulo explica mediante un ejemplo cuatro aspectos de S7 que o son nuevos o se han realizado de manera distinta que en S5:

- Tratamiento de valores analógicos
- Datos locales
- Evaluación de la información de arranque de los bloques de organización
- Transferencia en bloque

En nuestro ejemplo se controla un motor de giro a la izquierda/derecha mediante un módulo de entradas/salidas digitales. Las revoluciones se leen con un módulo de entradas analógicas y se emiten a un módulo de salidas analógicas. Los módulos digitales y analógicos del ejemplo tienen funciones de diagnóstico.

### Configuración

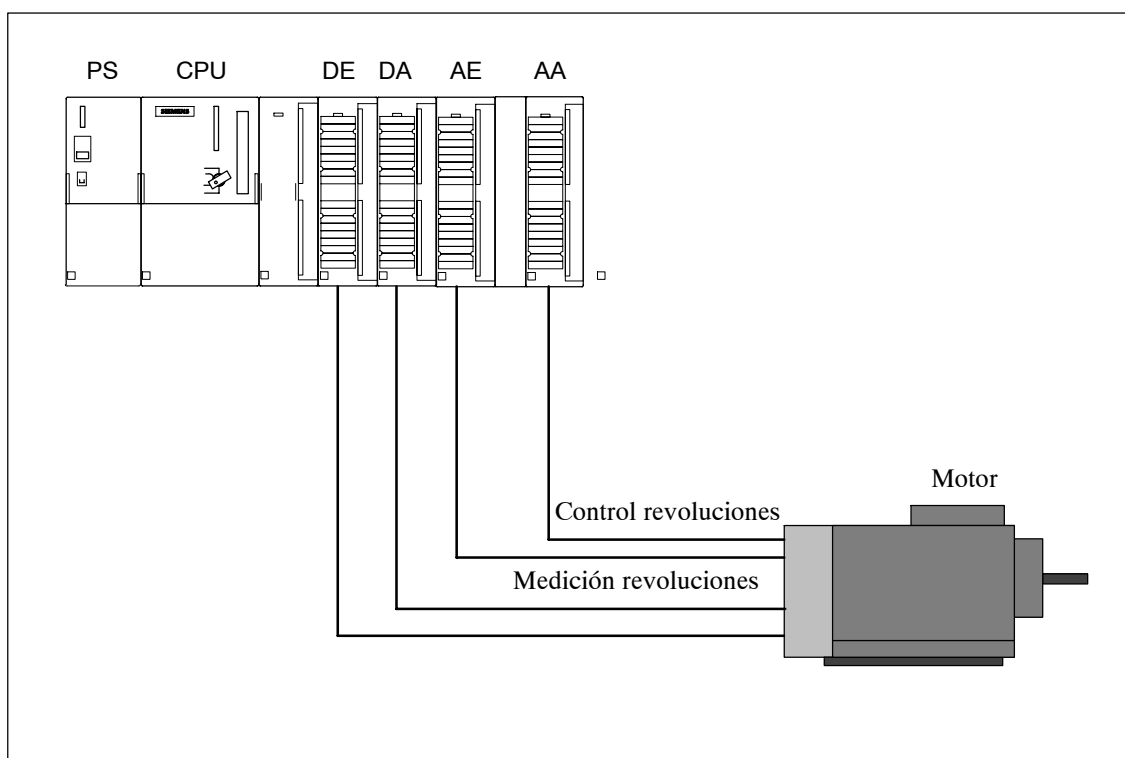


Figura 9-1 Configuración del ejemplo

## 9.1 Tratamiento de valores analógicos

### Conversión de valores analógicos

Los valores analógicos solamente pueden ser procesados por la CPU de forma digital.

Los módulos de entradas analógicas convierten las señales analógicas del proceso en señales digitales.

Los módulos de salidas analógicas convierten los valores de las salidas digitales en señales analógicas.

### Representación de valores analógicos en S5

Tabla 9-1 Ejemplo del módulo de entradas analógicas 6ES5 460-7LA13

Resolución	Valor analógico															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Significancia de los bits	Sg	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	A	E	D

En los módulos de salidas analógicas los valores se representan como complementos a 2 de 12 bits.

Los módulos de entradas analógicas pueden evaluar el valor opcionalmente como número de 12 bits con signo o como complemento a 2 de 13 bits.

El bit "D" sirve para indicar un desbordamiento.

El bit "E" es el bit de error que se activa cuando aparece un error (p. ej. rotura de hilo, si es que ha sido parametrizada).

El bit "A" es el bit de actividad. El valor indicado es válido cuando este bit está a "0".

### Representación de valores analógicos

El valor analógico digitalizado para los valores de entrada y salida que tengan el mismo margen nominal será el mismo.

Los valores analógicos se representan como complementos a 2.

Tabla 9-2 Ejemplo de los módulos de entradas analógicas en S7

Resolución	Valor analógico															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Significancia de los bits	S	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>

El signo (S) del valor analógico se representa siempre en el bit 15: "0" indica un valor positivo, y "1" un valor negativo.

En S7 ya no existen los bits de error.

Cuando aparece un error se emite el valor W#16#7FFF.

Si los módulos tienen funciones de diagnóstico, es posible que se dispare una alarma de diagnóstico en caso de error. Dicha alarma de diagnóstico se ajusta con la herramienta de configuración HW-Config.

Cuando la resolución de un módulo analógico es inferior a 15 bits, el valor analógico figura en los datos útiles justificado a la izquierda. Los bits menos significativos y no ocupados están a "0".

## Ejemplo

En el ejemplo, las revoluciones del motor son leídas por un módulo de entrada analógica. El módulo utilizado tiene una resolución de 14 bits y el valor medido es un valor bipolar (p. ej.: margen de medida +/-10V).

Los límites superior e inferior se transfieren en forma de parámetros.

Se comprueban los límites inferior y superior del valor analógico. Si el valor leído se encuentra fuera del margen permitido, se señala un error en el resultado binario (bit RB = "0") y se emite el valor "0". Si el valor es correcto, se emite el valor leído.

El valor analógico se emite al valor de respuesta RET\_VAL de la función, también llamado valor de retorno. RET\_VAL representa un valor funcional, lo cual es una novedad con respecto al S5.

```

FUNCTION FC1: REAL
TITLE = Tratamiento de valores analógicos
NAME:          ANALOG
VERSION:       01.00

VAR_INPUT
  VALOR_ENTRADA : INT;          // Valor de entrada
  LIM_SUP        : REAL;        // Límite superior para el valor analógico
  LIM_INF        : REAL;        // Límite inferior para el valor analógico
END_VAR

BEGIN
NETWORK
TITLE = Comprobar límites superior e inferior

  O(
    L   VALOR_ENTRADA;          // Valor entrada > límite superior
    L   +27648;
  >I;
  );
  O(
    L   VALOR_ENTRADA;          // o
    L   -27648;
  <I;
  );
  NOT;
  L   0;
  SPBNB FIN;                    // si se excede el límite superior o inferior
                                  // no seguir procesando, valor de respuesta = 0
                                  // y RB = "0"
                                  // si no se exceden los límites => RB = "1"

NETWORK
TITLE = Convertir valor digital en revoluciones

  L   LIM_SUP;                  // Fórmula para convertir VALOR_ENTRADA en
                                  // revoluciones:
  L   LIM_INF;                  // Valor analógico = (LIM_SUP - LIM_INF)
                                  // * VALOR ENTRADA
  -R;                            // / (55296 (número de unidades))
  L   VALOR_ENTRADA;
  ITD;                            // Convertir valor en número en coma flotante
  DTR;
  *R;
  L   55296.0;
  /R;
ENDE:   T      RET_VAL;

  BE;

END_FUNCTION

```

Figura 9-2 Tratamiento de valores analógicos

## 9.2 Datos locales temporales

Los datos locales temporales sirven de memoria intermedia, con lo cual sustituyen a las marcas de trabajo del S5. Estos datos temporales se pueden utilizar en cualquier bloque lógico y se pierden después de ejecutar el bloque. Dichos datos figuran en la pila de datos locales (LSTACK).

### Ejemplo 1

En el ejemplo 1 se utilizan los datos locales temporales de memoria intermedia con direccionamiento **simbólico**. El número de revoluciones dado se convierte en un valor medido digitalizado para el módulo de salidas analógicas. El módulo empleado tiene una resolución de 14 bits. El valor medido es un valor bipolar (p. ej. margen de medida +/-10V).

Los límites superior e inferior se transfieren en forma de parámetros.

El valor medido se obtiene en el valor de respuesta de la función (RET\_VAL). Todas las funciones pueden suministrar opcionalmente un valor de respuesta. El tipo de datos del valor de respuesta se indica detrás del nombre de la función. Si no se requiere ningún valor de respuesta, se escribirá VOID en lugar del tipo de datos.

```

FUNCTION FC2: INT
TITLE = Convertir el valor medido
NAME:          VALOR MEDIDO
VERSION:      01.00

VAR_INPUT
  VALOR_ENTRADA : REAL;          // Valor de entrada (intensidad)
  LIM_SUP       : REAL;          // Límite superior
  LIM_INF       : REAL;          // Límite inferior
END_VAR

VAR_TEMP
  LOCAL         : REAL;          // Datos locales para resultado intermedio
END_VAR

BEGIN
NETWORK
TITLE = Convertir el valor medido

  L   VALOR_ENTRADA;             // Fórmula para calcular las unidades:
  L   55296.0;                   // Valor medido = VALOR_ENTRADA
  *R;                             // * 55296 (número de unidades)
                                   // / (LIM_SUP - LIM_INF)
  T   LOCAL;                     // Guardar resultado intermedio en
  L   LIM_SUP;                   // los datos locales
  L   LIM_INF;
  -R;
  L   LOCAL;
  TAK;
  /R;
  RND;                             // Convertir el número en coma flotante en un
                                   // número entero
  T   RET_VAL;

END_FUNCTION

```

Figura 9-3 Convertir el valor medido

**Ejemplo 2**

En el ejemplo 2 se utilizan los datos locales como si fueran marcas de trabajo de S5 que se direccionan de forma **absoluta**. El programa de ejemplo muestra el control de un motor que gira a izquierda y derecha. El byte de entrada y el byte de salida se copian en el área de datos locales. Para poder utilizar los datos locales temporales es preciso reservar un área de la pila de datos locales (LSTACK), puesto que esta pila también es utilizada por el editor de programas KOP/FUP/AWL. Las direcciones de los datos locales se pueden consultar en el área de declaración del bloque. Los bits de datos locales se combinan entre sí mediante combinaciones lógicas, suministrando así las señales de salida que vuelven a escribir los datos locales en el byte de salida al final del bloque. Las direcciones del byte de entrada y del de salida pueden ser parametrizadas.

**Nota**

Si introduce una variable adicional antes de los datos locales ya existentes, se desplazarán las direcciones de los datos locales siguientes.

Tabla 9-3 Ocupación de las entradas y salidas / datos locales

Dirección	Datos locales	Denominación	Descripción
E n.0	L 0.0	CONEXION	Interruptor de conexión
E n.1	L 0.1	PARO	Parar motor
E n.2	L 0.2	PARO_EMERG	Pulsador de emergencia
E n.3	L 0.3	MOTOR_DERECHA	Girar motor a la derecha
E n.4	L 0.4	MOTOR_IZQUIERDA	Girar motor a la izquierda
E n.5	L 0.5	FINAL_CARRERA_DERECHA	Final de carrera a la derecha
E n.6	L 0.6	FINAL_CARRERA_IZQUIERDA	Final de carrera a la izquierda
E n.7	L 0.7	-	libre
A m.0	L 1.0	LISTO	Motor listo
A m.1	L 1.1	GIRAR_DERECHA	Giro a la derecha está activo
A m.2	L 1.2	GIRAR_IZQUIERDA	Giro a la izquierda está activo
A m.3	L 1.3	POSICION_ALCANZADA	Posición alcanzada

**Funcionamiento**

La tensión se aplica accionando el interruptor de conexión. Seguidamente el motor se pone en marcha, lo cual se señala con la salida LISTO. Con los pulsadores MOTOR\_DERECHA y MOTOR\_IZQUIERDA se gira el motor en el sentido deseado. Sólo se puede girar en un sentido. Si desea cambiar el sentido de giro, debe parar primero el motor con PARO. Cuando se alcanza un final de carrera, se apaga el motor, así como en caso de accionar PARO\_EMERG, y sólo se puede volver a girar una vez que se haya soltado el interruptor PARO\_EMERG.

```

FUNCTION FC3: VOID
TITLE = Control del motor
NAME:     MOTOR
VERSION:  01.00

VAR_INPUT
    BYTE_ENTRADA      : BYTE; // Byte de entrada
END_VAR

VAR_IN_OUT
    BYTE_SALIDA       : BYTE; // Byte de salida
END_VAR

VAR_TEMP
    IMAGEN_BYTE_ENTRADA : BYTE; // Imagen del byte de entrada
    IMAGEN_BYTE_SALIDA  : BYTE; // Imagen del byte de salida
END_VAR

BEGIN
NETWORK
TITLE = Control del motor

    L    BYTE_ENTRADA; // Copiar byte de entrada en área de datos locales
    T    IMAGEN_BYTE_ENTRADA;
    L    BYTE_SALIDA; // Copiar byte de salida en área de datos locales
    T    IMAGEN_BYTE_SALIDA;

    ON   L0.0;        // Motor desconectado (sin tensión de alimentación)
    ON   L0.2;        // o pulsador de emergencia PARO_EMERG accionado
    R    L1.0;        // => Resetear motor listo
    R    L1.1;        // => Resetear control motor
    R    L1.2;
    R    L1.3;        // => Resetear posición alcanzada
    SPB  ENDE;        // => No se evalúan más señales

    U    L0.0;        // Motor conectado
    S    L1.0;        // => Setear motor listo

    U    L0.3;        // Control del motor a la derecha
    UN   L0.4;        // Bloqueo: control a la izquierda bloqueado
    UN   L1.2;        // y giro a la izquierda inactivo
    FP   M0.0;        // Generar flanco positivo
    S    L1.1;        // Después: conectar giro a la derecha
    R    L1.3;        // Resetear posición alcanzada

    U    L0.4;        // Control del motor a la izquierda
    UN   L0.3;        // Bloqueo: control a la derecha bloqueado
    UN   L1.1;        // y giro a la derecha inactivo
    FP   M0.1;        // generar flanco positivo
    S    L1.2;        // Después: conectar giro a la izquierda
    R    L1.3;        // Resetear posición alcanzada

```

*Continúa en la página siguiente*

Figura 9-4 Función para controlar el motor

```
O(;  
U    L0.5;          // Final de carrera alcanzado a la derecha y  
U    L1.1;          // giro a la derecha activo  
);  
O(;  
U    L0.6;          // final de carrera alcanzado a la izquierda y  
U    L1.2;          // giro a la izquierda activo  
);  
S    L1.3;          // => Setear posición alcanzada  
O    L0.1;          // Paro motor accionado o  
O    L1.3;          // posición alcanzada  
R    L1.1;          // => Resetear control del motor  
R    L1.2;  
  
FIN: L    IMAGEN_BYTE_SALIDA;    // Copiar datos locales en el byte de salida  
T    BYTE_SALIDA;  
  
END_FUNCTION
```

Figura 9-5 Función para controlar el motor, continuación



### 9.3 Evaluar la información de arranque del OB de alarma de diagnóstico (OB 82)

#### Información de arranque

Cuando el sistema operativo llama a los bloques de organización, el usuario obtiene en la pila de datos locales una información de arranque para todo el sistema. La información de arranque tiene una longitud de 20 bytes y está disponible tras iniciar la ejecución del OB.

#### Información de arranque del OB 82

La información de arranque del OB de alarma de diagnóstico contiene la dirección base lógica y una información de diagnóstico de cuatro bytes. El manual de referencia **/235/** describe la estructura exacta de esta información. En la librería estándar “StdLib30” bajo “StdOBs” encontrará plantillas para la tabla de declaración de variables correspondiente.

Los módulos digitales piden una alarma de diagnóstico a la CPU (tanto con un evento entrante como con uno saliente), siempre y cuando se haya habilitado la alarma de diagnóstico al configurar el hardware. Seguidamente, el sistema operativo llama al OB 82.

Las SFCs 39 a 42 permiten bloquear o retardar y volver a habilitar la llamada del OB de alarma de diagnóstico. Para más información al respecto consulte el manual de referencia **/235/**.

#### Ejemplo

En el programa de ejemplo se evalúa la tensión auxiliar externa. En caso de que ésta quede interrumpida, se activa el bit `FALTA_TENSION_EXT` en el bloque de datos DB 82 “DB\_DIAG”. Además se guarda la dirección del módulo y la hora exacta en que se cortó la alimentación. Esta información puede ser procesada en el resto del programa.

Antes de compilar la fuente AWL hay que introducir el símbolo del bloque de datos DB 82 “DB\_DIAG” en la tabla de símbolos.

```

DATA_BLOCK DB_DIAG
TITLE = Datos de diagnóstico
NAME:      DB_DIAG
VERSION:   01.00

STRUCT
    MDL_ADDR      : INT;           // Dirección del módulo
    EXT_VOLTAGE_FAULT : BOOL;      // Bit de error Falta tensión externa
    DATE_TIME     : DATE_AND_TIME; // Fecha y hora en que se disparó la
                                     // alarma de diagnóstico
    SFC_RET_VAL   : INT;           // Código de respuesta SFC BLKMOV
END_STRUCT;

BEGIN
END_DATA_BLOCK

ORGANIZATION_BLOCK OB82
TITLE = Alarma de diagnóstico
NAME:      Diagnóstico
VERSION:   01.00

VAR_TEMP
    OB82_EV_CLASS      : BYTE; // Clase de evento e identificaciones:
                                     // B#16#38: evento saliente
                                     // B#16#39: evento entrante

    OB82_FLT_ID        : BYTE; // Código de error (B#16#42)
    OB82_PRIORITY      : BYTE; // Prioridad 26 o 28
    OB82_OB_NUMBR      : BYTE; // Número del OB
    OB82_RESERVED_1    : BYTE; // Reservado
    OB82_IO_FLAG       : BYTE; // Módulo de entrada: B#16#54
                                     // Módulo de salida: B#16#55

    OB82_MDL_ADDR      : INT; // Dirección base lógica del módulo
                                     // en el que ha aparecido el error

    OB82_MDL_DEFECT    : BOOL; // Fallo del módulo
    OB82_INT_FAULT     : BOOL; // Fallo interno
    OB82_EXT_FAULT     : BOOL; // Fallo externo
    OB82_PNT_INFO      : BOOL; // Se ha verificado un error de canal
    OB82_EXT_VOLTAGE   : BOOL; // Falta la tensión auxiliar externa
    OB82_FLD_CONNCTR   : BOOL; // Falta el conector frontal
    OB82_NO_CONFIG     : BOOL; // Módulo no parametrizado
    OB82_CONFIG_ERR    : BOOL; // Parámetro erróneo en módulo
    OB82_MDL_TYPE      : BYTE; // Bit0-3: clase de módulo
                                     // Bit4: falta información del canal
                                     // Bit5: hay información de usuario
                                     // Bit6: alarma de diagnóstico del dummy
                                     // Bit7: Reservado

    OB82_SUB_MDL_ERR   : BOOL; // Módulo de usuario defectuoso o inexistente
    OB82_COMM_FAULT    : BOOL; // Fallo de comunicación
    OB82_MDL_STOP      : BOOL; // Estado operativo (0: RUN, 1: STOP)
    OB82_WTCH_DOG_FLT  : BOOL; // El watchdog ha reaccionado
    OB82_INT_PS_FLT    : BOOL; // Fallo de la fuente de alimentación interna
    OB82_PRIM_BATT_FLT : BOOL; // Pila vacía
    OB82_BCKUP_BATT_FLT : BOOL; // Falla todo el respaldo de la memoria
    OB82_RESERVED_2    : BOOL; // Reservado
    OB82_RACK_FLT      : BOOL; // Fallo del bastidor
    OB82_PROC_FLT      : BOOL; // Fallo del procesador
    OB82_EPROM_FLT     : BOOL; // Fallo del EPROM
    OB82_RAM_FLT       : BOOL; // Fallo de la RAM

```

*Continúa en la página siguiente*

Figura 9-6 Evaluación de los datos de diagnóstico

```

OB82_ADU_FLT      : BOOL;          // Fallo ADU/DAU
OB82_FUSE_FLT     : BOOL;          // Actuación del fusible
OB82_HW_INTR_FLT  : BOOL;          // Se ha perdido la alarma de proceso
OB82_RESERVED_3   : BOOL;          // Reservado
OB82_DATE_TIME    : DATE_AND_TIME; // Fecha y hora en que se pidió
                                           // el OB

END_VAR

BEGIN
NETWORK
TITLE = Alarma de diagnóstico

    L    OB82_MDL_ADDR;              // Guardar dirección del módulo
    T    DB_DIAG.MDL_ADDR;

    L    OB82_EV_CLASS;              // Clase de evento = B#16#38:
    L    B#16#38;                    // Evento saliente
    ==I;
    SPB  IR_A;

    U    OB82_EXT_VOLTAGE;           // Evento entrante:
                                           // Comprobar si falta la
                                           // tensión auxiliar

    S    DB_DIAG.EXT_VOLTAGE_FAULT; // Activar bit
    SPA  ;TIME

    IR A: U    OB82_EXT_VOLTAGE;     // Evento saliente:
    R    DB_DIAG.EXT_VOLTAGE_FAULT; // Retorno de la tensión auxiliar externa
                                           // Desactivar bit

NETWORK

TITLE = Memorizar la fecha y la hora
TIME:CALL SFC 20( // SFC BLKMOV
    SRCBLK :=OB82_DATE_TIME, // Memorizar la fecha y la hora
    RET_VAL:=DB_DIAG.SFC_RET_VAL, // en que se pidió la alarma de
    DSTBLK :=DB_DIAG.DATE_TIME); // diagnóstico

END_ORGANIZATION_BLOCK

```

Figura 9-7 Evaluación de los datos de diagnóstico; continuación

## 9.4 Transferencia en bloque

La función de sistema SFC 20 “BLKMOV” (block move) sirve para copiar el contenido de un área de memoria (=fuente) a otra área de memoria (=destino).

Con la SFC 20 “BLKMOV” se pueden copiar todas las entradas, salidas, marcas y datos.

### Parámetros

Parámetro	Declaración	Tipo de datos	Area de memoria	Descripción
SRCBLK	INPUT	ANY	E, A, M, D, L	Indica el área de memoria a copiar (fuente).
RET_VAL	OUTPUT	INT	E, A, M, D, L	Si aparece un error mientras se ejecuta la función, el valor de retorno suministrará el código de error.
DSTBLK	OUTPUT	ANY	E, A, M, D, L	Indica el área de memoria en la que se copia (destino).

---

### Nota

La fuente y el destino no se pueden solapar. Si el destino es más grande que la fuente, solamente se copiarán tantos datos en el destino como contenga la fuente.

Si el destino indicado es menor que la fuente, solamente se copiarán tantos datos como quepan en el destino.

Si no desea ocupar los parámetros para la fuente y el destino de la SFC 20 “BLKMOV” con punteros constantes, sino que desea asignar áreas variables, puede hacerlo con variables temporales del tipo ANY.

---

## Puntero ANY para tipos de datos

Las tablas siguientes muestran la estructura del puntero ANY.

Tabla 9-4 Puntero ANY

Byte n	Byte n+1	Byte n+2	Byte n+3	Byte n+4	Byte n+5	Byte n+6	Byte n+7	Byte n+8	Byte n+9
B#16#10	Tipo (v. tabla 9-5)	Longitud		Número del bloque de datos en el bloque de datos		Puntero de área (v. figura 9-8)			

Tabla 9-5 Tipo (byte n+1)

<b>Valor:</b>	<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>	<b>06</b>	<b>07</b>
<b>Tipo:</b>	BOOL	BYTE	CHAR	WORD	INT	DWORD	DINT
<b>Valor:</b>	<b>08</b>	<b>09</b>	<b>0A</b>	<b>0B</b>	<b>0C</b>	<b>0E</b>	<b>13</b>
<b>Tipo:</b>	REAL	DATE	TOD	TIME	S5TIME	DT	String

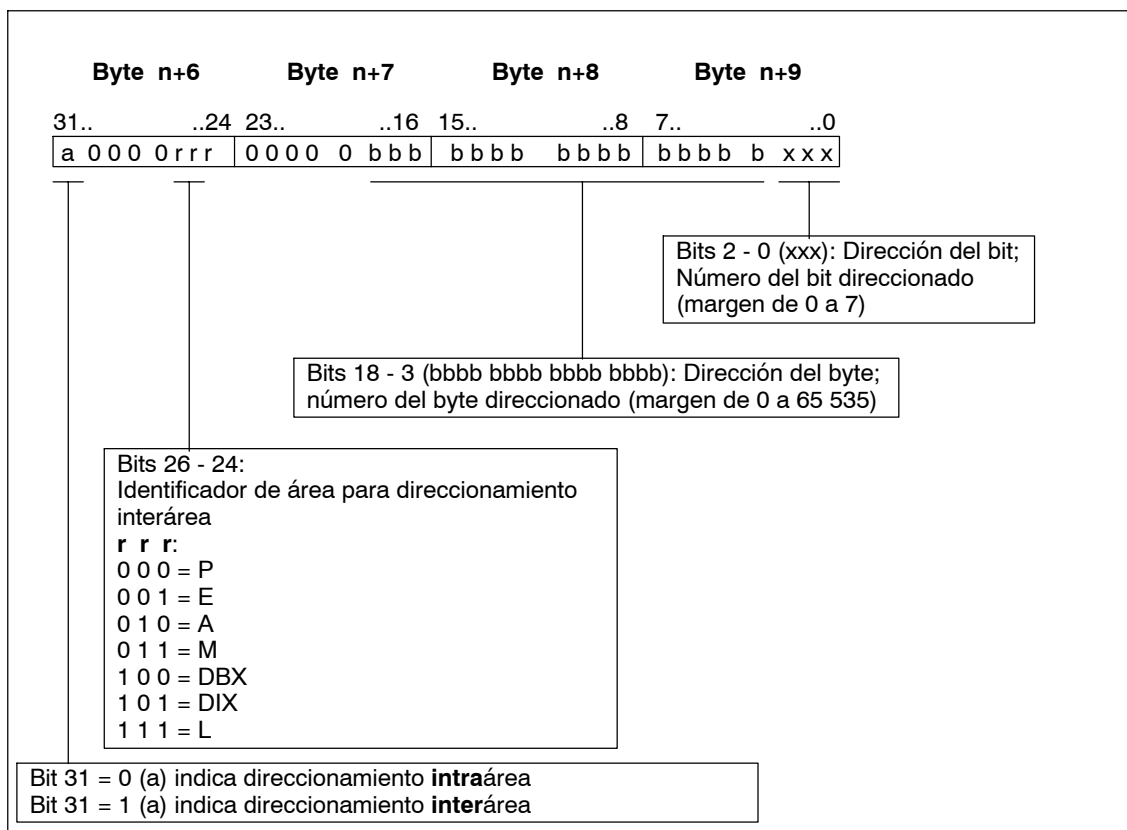


Figura 9-8 Puntero de área (byte n+6 a byte n+9)

**Ejemplo**

El ejemplo muestra una función que permite copiar áreas de datos mediante la función de sistema SFC 20 “BLKMOV” (en bloques de datos). El área de origen y el área de destino se pueden asignar a los parámetros en forma de variables.

**Principio**

La función contiene dos punteros ANY en el área de datos locales: uno para el área de origen y otro para el área de destino. Por regla general, el tipo de datos ANY sólo está permitido para variables del área de datos locales.

Los punteros ANY se ocupan en la función según la estructura que acabamos de describir y se asignan a los parámetros al llamar a la función SFC 20 “BLKMOV”.

```

FUNCTION FC4: INT
TITLE = Copiar áreas de datos
NAME:     COPY
VERSION:  01.00

VAR_INPUT
  ORIGEN_DBNR   : INT;           // Número del DB del área de origen
  ORIGEN_INICIO : INT;           // Nº de la palabra de datos del inicio del
                                  // área de origen
  ORIGEN_LONG   : INT;           // Longitud del área de origen en bytes
  DESTINO_NDB   : INT;           // Número del DB del área de destino
  DESTINO_INICIO : INT;           // Nº de la palabra de datos del inicio del
                                  // área de destino
  DESTINO_LONG  : INT;           // Longitud del área de destino en bytes
END_VAR

VAR_TEMP
  PUNTERO_ORIGEN : ANY;           // Puntero ANY para el área de origen
  PUNTERO_DESTINO : ANY;           // Puntero ANY para el área de destino
END_VAR

BEGIN
NETWORK
TITLE = Programar el puntero del origen

  L    P##PUNTERO_ORIGEN;         // Cargar dirección del puntero que señala al
  LAR1;                           // área de origen en el registro de dir. 1
  L    W#16#1002;                 // Escribir identificador del área de datos
  T    LW[AR1, P#0.0];            // en el puntero ANY del origen
  L    ORIGEN_NDB;                // Escribir nº DB en puntero ANY del origen
  T    LW[AR1, P#4.0];
  L    ORIGEN_INICIO;             // Convertir inicio del área de datos al for-
  SLD  3;                          // mato de puntero,
  OD   DW#16#84000000;            // Combinar lóg. con identificador de área
  T    LD[AR1, P#6.0];            // y escribir en puntero ANY del origen
  L    ORIGEN_LONG;               // Escribir longitud del área de datos en pun-
  T    LW[AR1, P#2.0];            // tero ANY para origen

```

*Continúa en la página siguiente*

Figura 9-9 Copiar áreas de datos

```

NETWORK
TITLE = Programar el puntero del destino

L    P##PUNTERO_DESTINO;    // Cargar dirección del puntero que señala al
LAR1;                        // área de destino en el registro de dir. 1
L    W#16#1002;             // Escribir identificador del área de datos
T    LW[AR1, P#0.0];        // en el puntero ANY del destino
L    DESTINO_NDB;           // Escribir nº DB en puntero ANY del destino
T    LW[AR1, P#4.0];        //
L    DESTINO_INICIO;        // Convertir el inicio del área de datos al
SLD  3;                      // formato de puntero,
OD   DW#16#84000000;        // Combinar lóg. con identificador del área
T    LD[AR1, P#6.0];        // y escribir en puntero ANY del destino
L    DESTINO_LONGITUD;     // Escribir longitud del área de datos
T    LW[AR1, P#2.0];        // en el puntero ANY del destino

NETWORK
TITLE = Copiar los datos
CALL  SFC 20(                // Copiar datos con SFC BLKMOV (transf. bloque)
SRCBLK := PUNTERO_ORIGEN,    // Puntero al área de origen
RET_VAL:= RET_VAL,           // Código de respuesta de la SFC BLKMOV
DSTBLK := PUNTERO_DESTINO);  // Puntero al área de destino
END_FUNCTION

```

Figura 9-10 Copiar áreas de datos, continuación

## 9.5 Llamada de los ejemplos

Este apartado muestra la tabla de símbolos, los bloques de datos necesarios para ocupar los parámetros del bloque y el bloque de organización OB 1 con las llamadas de las funciones antes descritas.

Tabla 9-6 Tabla de símbolos

Símbolo	Dirección	Tipo de datos	Comentario
DB_DIAG	DB 82	DB 82	Bloque de datos de diagnóstico
DB_VAL MED	DB 100	DB 100	Bloque de datos para valores medidos
DB_MOTOR_1	DB 110	DB 110	Bloque de datos para motor 1
ERROR	MW 100	WORD	Valor de retorno de la función FC 4 para transferencia en bloque

```

DATA_BLOCK DB_VALORES_MEDIDOS
TITLE = Valores medidos
NAME:      DB_MED
VERSION:   01.00
STRUCT
    VAL_ANALOG_1  : REAL;          // Valor analógico 1 de FC 1
    VAL_ANALOG_2  : REAL;          // Valor analógico 2 de FC 2
    VAL_DIGITAL_2 : INT;           // Valor medido digitalizado de FC 2
END_STRUCT;
BEGIN
END_DATA_BLOCK

DATA_BLOCK DB_MOTOR_1
TITLE = Datos del motor
NAME:      DB_MOT_1
VERSION:   01.00
STRUCT
    PALABRA_CONTROL: WORD;        // Control del motor 1
    REVOLUCIONES   : REAL;        // Revoluciones del motor 1
    TEMPERATURA    : REAL;        // Temperatura del motor 1
    CORRIENTE      : REAL;        // Consumo de corriente del motor 1
END_STRUCT;
BEGIN
END_DATA_BLOCK

ORGANIZATION_BLOCK OB1
TITLE = Llamada en el ciclo
NAME:      CICLO
VERSION:   01.00
VAR_TEMP
    INFORM_ARRANQ: ARRAY [1..20] OF BYTE;
END_VAR
BEGIN
NETWORK
TITLE = Llamada de las funciones
CALL FC 1(
    VALOR_ENTRADA := EW 0,          // Llamada de la función para
    LIM_SUP       := +10.0,        // procesamiento valores analógicos
    LIM_INF       := -10.0,        // Margen de medida: +/-10V
    RET_VAL       := DB_VAL_MED.VAL_ANALOG_1);
    // RET_VAL = valor analógico
CALL FC 2(
    VALOR_ENTRADA := DB_VAL_MED.VALOR_ANALOG_2, // Llamada de la función para calcular
    LIM_SUP       := +10.0,        // el valor med. digitalizado
    LIM_INF       := -10.0,        // Margen de medida: +/-10V
    RET_VAL       := DB_VAL_MED.VAL_DIGITAL_2);
    // RET_VAL = valor med. digitalizado
CALL FC 3(
    BYTE_ENTRADA := EB 4,          // Llamada de la función para control motor
    BYTE_SALIDA  := AB 8);
CALL FC 4(
    ORIGEN_NDB := 100,             // Llamada de función para transf. en bloque
    ORIGEN_INICIO := 0,           // Origen: DB 100
    ORIGEN_LONG := 8,             // a partir del byte de datos DBB 0
    DESTINO_NDB := 110,           // Longitud: 4 bytes
    DESTINO_INICIO := 2,         // Destino: DB 110
    DESTINO_LONG := 8,           // a partir del byte de datos DBB 6
    RET_VAL      := ERROR);      // Longitud: 4 bytes
    // RET_VAL = Código de error de SFC 20 BLKMOV
END_ORGANIZATION_BLOCK

```

Figura 9-11 OB 1



## Anexos

---

Listas de operandos y de  
operaciones

---

**A**

---

Indice alfabético

---

**B**



# Listas de operandos y operaciones

# A

## A.1 Operandos

### Operandos convertibles

Se convierten los operandos siguientes:

Tabla A-1 Operandos convertibles

S5-AWL (alemán)	S5-AWL (internacional)	S7-AWL (alemán)	S7-AWL (internacional)
"A"	"Q"	"A"	"Q"
"AB"	"QB"	"AB"	"QB"
"AD"	"QD"	"AD"	"QD"
"AW"	"QW"	"AW"	"QW"
"BF"	"BN"	""	""
"D"	"D"	"DBX"	"DBX"
"DW"	"DW"	"DBW"	"DBW"
"DD"	"DD"	"DBD"	"DBD"
"DR"	"DR"	"DBB"	"DBB"
"DL"	"DL"	"DBB"	"DBB"
"E"	"I"	"E"	"I"
"EB"	"IB"	"EB"	"IB"
"ED"	"ID"	"ED"	"ID"
"EW"	"IW"	"EW"	"IW"
"M"	"F"	"M"	"M"
"MB"	"FY"	"MB"	"MB"
"MD"	"FD"	"MD"	"MD"
"MW"	"FW"	"MW"	"MW"
"PW"	"PW"	"PEW/PAW"	"PIW/PQW"
"PY"	"PY"	"PEB/PAB"	"PIB/PQB"
"QB"	"OY"	"PEB/PAB"	"PIB/PQB"
"QW"	"OW"	"PEW/PAW"	"PIW/PQW"
"S"	"S"	"M"	"M"
"SD"	"SD"	"MD"	"MD"

Tabla A-1 Operandos convertibles

S5-AWL (alemán)	S5-AWL (internacional)	S7-AWL (alemán)	S7-AWL (internacional)
"SW"	"SW"	"MW"	"MW"
"SY"	"SY"	"MB"	"MB"
"T"	"T"	"T"	"T"
"Z"	"C"	"Z"	"C"
"= <parámetro formal>"	"= <parámetro formal>"	"# <parámetro formal>"	"# <parámetro formal>"

**Operandos no convertibles**

La tabla A-2 muestra los operandos que **no son convertibles**.

Tabla A-2 Operandos no convertibles

S5-AWL (alemán)	S5-AWL (internacional)
"A1"	"A1"
"A2"	"A2"
"BA"	"RI"
"BB"	"RJ"
"BR"	"BR"
"BS"	"RS"
"BT"	"RT"
"CB"	"CY"
"CD"	"CD"
"CW"	"CW"
"GB"	"GY"
"GD"	"GD"
"GW"	"GW"
"SA"	"SA"

## A.2 Operaciones

### Operandos y operaciones no convertibles

La tabla A-3 muestra todas las operaciones S5 (sin operandos) en AWL que se convierten automáticamente en S7-AWL:

Tabla A-3 Operaciones convertibles (sin operandos)

S5-AWL (alemán)	S5-AWL (internacional)	S7-AWL (alemán)	S7-AWL (internacional)
"AF"	"RA"	"CALL SFC 42"	"CALL SFC 42"
"AS"	"IA"	"CALL SFC 41"	"CALL SFC 41"
"BEA"	"BEU"	"BEA"	"BEU"
"BEB"	"BEC"	"BEB"	"BEC"
" +D"	" +D"	" +D"	" +D"
" -D"	" -D"	" -D"	" -D"
" !=D"	" !=D"	" ==D"	" ==D"
" ><D"	" ><D"	" <>D"	" <>D"
" >D"	" >D"	" >D"	" >D"
" >=D"	" >=D"	" >=D"	" >=D"
" <D"	" <D"	" <D"	" <D"
" <=D"	" <=D"	" <=D"	" <=D"
"DED"	"DED"	"BTD"	"BTD"
"DEF"	"DEF"	"BTI"	"BTI"
"DUD"	"DUD"	"DTB"	"DTB"
"DUF"	"DUF"	"ITB"	"ITB"
"ENT"	"ENT"	"ENT"	"ENT"
" +F"	" +F"	" +I"	" +I"
" -F"	" -F"	" -I"	" -I"
".F"	".F"	".I"	".I"
"x F"	"x F"	"*I"	"*I"
" !=F"	" !=F"	" ==I"	" ==I"
" ><F"	" ><F"	" <>I"	" <>I"
" >F"	" >F"	" >I"	" >I"
" >=F"	" >=F"	" >=I"	" >=I"
" <F"	" <F"	" <I"	" <I"
" <=F"	" <=F"	" <=I"	" <=I"
"FDG"	"FDG"	"DTR"	"DTR"
" +G"	" +G"	" +R"	" +R"
" -G"	" -G"	" -R"	" -R"

Tabla A-3 Operaciones convertibles (sin operandos), continuación

S5-AWL (alemán)	S5-AWL (internacional)	S7-AWL (alemán)	S7-AWL (internacional)
":G"	":G"	"/R"	"/R"
"xG"	"xG"	"*R"	"*R"
"! =G"	"! =G"	"==R"	"==R"
"><G"	"><G"	"<>R"	"<>R"
">G"	">G"	">R"	">R"
">=G"	">=G"	">=R"	">=R"
"<G"	"<G"	"<R"	"<R"
"<=G"	"<=G"	"<=R"	"<=R"
"GFD"	"GFD"	"RND"	"RND"
"KEW"	"CFW"	"INVI"	"INVI"
"KZD"	"CSD"	"NEGD"	"NEGD"
"KZW"	"CSW"	"NEGI"	"NEGI"
"O"	"O"	"O"	"O"
"O("	"O("	"O("	"O("
"OW"	"OW"	"OW"	"OW"
"STP"	"STP"	"CALL SFC 46"	"CALL SFC 46"
"STS"	"STS"	"CALL SFC 46"	"CALL SFC 46"
"STW"	"STW"	"CALL SFC 46"	"CALL SFC 46"
"TAK"	"TAK"	"TAK"	"TAK"
"U("	"A("	"U("	"A("
"UW"	"AW"	"UW"	"AW"
"XOW"	"XOW"	"XOW"	"XOW"
)"	)"	)"	)"
"**"	"**"	"NETWORK"	"NETWORK"

### Operaciones convertibles con operandos

La tabla A-4 muestra todas las operaciones S5 (con operandos) en AWL que se convierten automáticamente en S7-AWL:

Tabla A-4 Operaciones convertibles (con operandos)

S5-AWL (alemán)	S5-AWL (internacional)	S7-AWL (alemán)	S7-AWL (internacional)
"A"	"C"	"AUF"	"OPN"
"ADD BF"	"ADD BF"	"+"	"+"
"ADD DH"	"ADD DH"	"+"	"+"
"ADD KF"	"ADD KF"	"+"	"+"
"AX"	"CX"	"AUF"	"OPN"

Tabla A-4 Operaciones convertibles (con operandos)

S5-AWL (alemán)	S5-AWL (internacional)	S7-AWL (alemán)	S7-AWL (internacional)
"B"	"DO"	"secuencia de instrucciones para direccionamiento indirecto"	"secuencia de instrucciones para direccionamiento indirecto"
"BA"	"BA"	"	""
"BAB"	"DOC"	"SPB"	"JC"
"D"	"D"	"DEC"	"DEC"
"E"	"G"	"CALL SFC 22"	"CALL SFC 22"
"EX"	"GX"	"CALL SFC 22"	"CALL SFC 22"
"FR"	"FR"	"FR"	"FR"
"I"	"I"	"INC"	"INC"
"L"	"L"	"L"	"L"
"LC"	"LD"	"LC"	"LC"
"NOP"	"NOP"	"NOP"	"NOP"
"O"	"O"	"O"	"O"
"ON"	"ON"	"ON"	"ON"
"P"	"TB"	"SET; U"	"SET; A"
"PN"	"TBN"	"SET; UN"	"SET; AN"
"R"	"R"	"R"	"R"
"RB"	"RB"	"R"	"R"
"RD"	"RD"	"R"	"R"
"RLD"	"RLD"	"RLD"	"RLD"
"RLW"	"RLW"	"RLW"	"RLW"
"RRD"	"RRD"	"RRD"	"RRD"
"RRW"	"RRW"	"RRW"	"RRW"
"RU"	"RU"	"SET; R"	"SET; R"
"S"	"S"	"S"	"S"
"SA"	"SF"	"SA"	"SF"
"SAR"	"SFD"	"SA" Timer "ZR" Contador	"SF" Timer "CD" Counter
"SE"	"SD"	"SE"	"SD"
"SI"	"SP"	"SI"	"SP"
"SLD"	"SLD"	"SLD"	"SLD"
"SLW"	"SLW"	"SLW"	"SLW"
"SPA"	"JU"	"SPA"	"JU"

Tabla A-4 Operaciones convertibles (con operandos)

S5-AWL (alemán)	S5-AWL (internacional)	S7-AWL (alemán)	S7-AWL (internacional)
"SPB"	"JC"	"SPB"	"JC"
"SPM"	"JM"	"SPM"	"JM"
"SPN"	"JN"	"SPN"	"JCN"
"SPO"	"JO"	"SPO"	"JO"
"SPP"	"JP"	"SPP"	"JP"
"SPR"	"JUR"	"SPA"	"JU"
"SPS"	"JOS"	"SPS"	"JOS"
"SPZ"	"JZ"	"SPZ"	"JZ"
"SRD"	"SRD"	"SRD"	"SRD"
"SRW"	"SRW"	"SRW"	"SRW"
"SS"	"SS"	"SS"	"SS"
"SSV"	"SSU"	"SS" Timer "ZV" Contador	"SS" Timer "CU" Counter
"SU"	"SU"	"SET; S"	"SET; S"
"SV"	"SE"	"SV"	"SE"
"SVD"	"SSD"	"SSD"	"SSD"
"SVW"	"SSW"	"SSI"	"SSI"
"SVZ"	"SEC"	"SV" Timer "S" Contador	"SE" Timer "S" Counter
"T"	"T"	"T"	"T"
"TNB"	"TNB"	"CALL SFC 20"	"CALL SFC 20"
"TNW"	"TNW"	"CALL SFC 20"	"CALL SFC 20"
"U"	"A"	"U"	"A"
"UN"	"AN"	"UN"	"AN"
"ZR"	"CD"	"ZR"	"CD"
"ZV"	"CU"	"ZV"	"CU"
"="	"="	"="	"="

**Operaciones no convertibles**

La tabla siguiente muestra las operaciones S5-AWL que no se convierten automáticamente.

Tabla A-5 Operaciones no convertibles

S5-AWL (alemán)	S5-AWL (internacional)
"AAS"	"IAI"
"AAF"	"RAI"



Tabla A-5 Operaciones no convertibles, continuación

S5-AWL (alemán)	S5-AWL (internacional)
"ABR"	"ABR"
"ACR"	"ACR"
"AFF"	"RAE"
"AFS"	"IAE"
"ASM"	"ASM"
"BAF"	"BAF"
"BAS"	"BAS"
"BI" (sólo convertible con el tipo de parámetro D/constante)	"DI" (sólo convertible con el tipo de parámetro D/constante)
"BLD"	"BLD"
"LB"	"LB"
"LD"	"LD"
"LD=<parámetro formal>" (sólo convertible con el tipo de parámetro D/constante)	"LD=<parámetro formal>" (sólo convertible con el tipo de parámetro D/constante)
"LDI"	"LDI"
"LIM"	"LIM"
"LIR"	"LIR"
"LRB"	"LRB"
"LRD"	"LRD"
"LRW"	"LRW"
"LW"	"LW"
"LW=<parámetro formal>" (sólo convertible con el tipo de parámetro D/constante)	"LW=<parámetro formal>" (sólo convertible con el tipo de parámetro D/constante)
"MA1"	"MA1"
"MAB"	"MAB"
"MAS"	"MAS"
"MBA"	"MBA"
"MBR"	"MBR"
"MBS"	"MBS"
"MSA"	"MSA"
"MSB"	"MSB"
"SEF"	"SEE"
"SES"	"SED"
"SIM"	"SIM"
"TB"	"TB"
"TDI"	"TDI"

Tabla A-5 Operaciones no convertibles, continuación

<b>S5-AWL (alemán)</b>	<b>S5-AWL (internacional)</b>
"TIR"	"TIR"
"TSC"	"TSC"
"TSG"	"TSG"
"TRB"	"TRB"
"TRD"	"TRD"
"TRW"	"TRW"
"TW"	"TW"
"TXB"	"TXB"
"TXW"	"TXW"
"UBE"	"UBE"

## Índice bibliográfico

- /21/ Resumen técnico: *Sistemas de automatización S7/M7*,  
Descentralización con PROFIBUS-DP y AS-I
- /30/ ABC de la programación: *Autómata programable S7-300*,  
Iniciación a la configuración y programación
- /70/ Manual: *Autómata programable S7-300*,  
Configuración, instalación y datos de las CPU
- /71/ Manual de referencia: *Sistemas de automatización S7-300 y M7-300*,  
Datos de los módulos
- /72/ Lista de operaciones: *S7-300*,  
*CPU 312 IFM, 314 IFM, 313, 314, 315, 315-2 DP, 316*
- /100/ Manual de instalación: *Sistemas de automatización S7-400, M7-400*,  
Configuración e instalación
- /101/ Manual de referencia: *Sistemas de automatización S7-400, M7-400*,  
Datos de los módulos
- /102/ Lista de operaciones: *Autómata programable S7-400*,  
CPU 412, 413, 414, 416
- /231/ Manual del usuario: *Software estándar para SIMATIC S7 y M7*,  
STEP 7
- /232/ Manual: *AWL para S7-300/400*,  
Programación de bloques
- /233/ Manual: *KOP para S7-300/400*,  
Programación de bloques
- /234/ Manual de programación: *Software de sistema para S7-300/400*,  
Diseño de programas
- /235/ Manual de referencia: *Software de sistema para S7-300/400*,  
Funciones estándar y funciones de sistema
- /236/ Manual: *FUP para S7-300/400*,  
Programación de bloques
- /249/ Manual: *CFC: Continuous Function Chart*,  
Tomo 2: S7/M7
- /250/ Manual: *SCL para SIMATIC S7-300/400*,  
Programación de bloques
- /251/ Manual: *GRAPH para SIMATIC S7-300/400*,  
Programación de controles secuenciales

- /252/ Manual: *HiGraph for S7-300 and S7-400*,  
Programming State Graphs (disponible sólo en inglés)
- /254/ Manual: *CFC: Continuous Function Chart*,  
Tomo 1
- /270/ Manual: *S7 PDIAG para S7-300/400*,  
Configuración del diagnóstico de procesos para KOP, FUP y AWL
- /280/ Programming Manual: *System Software for M7-300/400*,  
Program Design (disponible sólo en inglés)
- /281/ Reference Manual: *System Software for M7-300 and M7-400*,  
System and Standard Functions (disponible sólo en inglés)
- /282/ User Manual: *System Software for M7-300 and M7-400*,  
Installation and Operation (disponible sólo en inglés)
- /290/ User Manual: *ProC/C++ for M7-300 and M7-400*,  
Writing C Programs (disponible sólo en inglés)
- /291/ User Manual: *ProC/C++ for M7-300 and M7-400*,  
Debugging C Programs (disponible sólo en inglés)
- /500/ Manual: *SIMATIC NET NCM S7* para Industrial Ethernet
- /501/ Manual: *SIMATIC NET NCM S7* para PROFIBUS,  
Tomos 1 y 2
- /800/ *DOCPRO*: Confección de documentación normalizada  
(sólo en CD)
- /801/ *Teleservicio para S7, C7 y M7*,  
Asistencia técnica a distancia para sistemas de automatización  
(sólo en CD)
- /802/ *PLC Simulation para S7-300 y S7-400*  
(sólo en CD)
- /803/ Manual de referencia: *Software estándar para S7-300 y S7-400*,  
Funciones estándar, Segunda parte (sólo en CD)

# Glosario

## A

### **Area de declaración**

En el área de declaración se declaran los datos locales de un bloque lógico, siempre y cuando el programa se cree con un editor de textos.

### **Ayuda en pantalla**

STEP 7 ofrece la posibilidad de hacerse mostrar en pantalla textos de ayuda contextual al utilizar el software de programación.

## B

### **Bloque**

Los bloques son partes del programa de usuario que se distinguen unos de otros por su función, su estructura o su finalidad. En STEP 7 existen los bloques siguientes:

- bloques lógicos (FB, FC, OB, SFB, SFC),
- bloques de datos (DB, SDB) y
- tipos de datos de usuario (UDT).

### **Bloque de datos (DB)**

Los bloques de datos son áreas de datos del programa de usuario que contienen datos del usuario. Cabe distinguir entre los bloques de datos globales, a los que pueden acceder todos los bloques lógicos, y los bloques de datos de instancia, que están asignados a una llamada de un FB determinado. A diferencia de los demás bloques, los bloques de datos no contienen instrucciones.

### **Bloque de datos de instancia**

Los bloques de datos de instancia guardan los parámetros formales y los datos estáticos de bloques de funciones. Un bloque de datos de instancia puede estar asignado a la llamada de un FB o a una jerarquía de llamadas de bloques de función.

**Bloque de función (FB)** Según la norma IEC 1131-3 un bloque de función es un bloque lógico que contiene datos estáticos. Los bloques de función ofrecen la posibilidad de transferir parámetros en el programa de usuario. Por consiguiente, son especialmente apropiados para programar funciones complejas de uso frecuente, p. ej. regulaciones, selección del modo de operación etc. Como el FB dispone de memoria (bloque de datos de instancia), es posible acceder a sus parámetros (p. ej. salidas) en cualquier momento y desde cualquier punto del programa.

**Bloque lógico** En SIMATIC S7, los bloques lógicos son aquellos que contienen partes del programa de usuario STEP 7.

**Bloque de organización (OB)** Los bloques de organización constituyen el interface entre el sistema operativo de la CPU y el programa de usuario. En los bloques de organización se establece el orden de ejecución del programa de usuario.

A diferencia de los bloques lógicos, los bloques de datos sólo contienen datos. Existen cinco tipos distintos de bloques lógicos: los bloques de organización (OB), los bloques de función (FB), las funciones (FC), los bloques de función de sistema (SFB) y las funciones de sistema (SFC).

## C

**Compilador** Un compilador es un programa que transcribe un programa escrito en un lenguaje de programación de alto nivel al código máquina con el que opera la CPU.

**Configurar** Configurar significa seleccionar y disponer los distintos componentes de un sistema de automatización, así como instalar el software requerido y adaptarlo al uso específico que se le vaya a dar (p.ej.: parametrizando los módulos).

## D

**Datos estáticos** Los datos estáticos son datos locales de un bloque de función que se guardan en el bloque de datos de instancia y que, por consiguiente, se conservan hasta la próxima ejecución del bloque de función.

**Datos globales** Los datos globales son datos a los que se puede acceder desde cualquier bloque lógico. En particular se trata de marcas (M), entradas (E), salidas (A), temporizadores, contadores y elementos de bloques de datos DB. A los datos globales sólo se puede acceder con direccionamiento absoluto o simbólico.

**Datos locales** Los datos locales son datos asignados a un bloque lógico que se declaran en el área de declaración de variables del mismo. Dependiendo del bloque, los datos locales pueden ser parámetros formales, datos estáticos o datos temporales.

**Datos temporales** Los datos temporales son datos locales de un bloque que se guardan en la pila de datos locales (LSTACK) durante la ejecución del bloque, y que se pierden una vez ejecutado el bloque.

**Dirección** Por “dirección” se entiende el identificador de un operando o de un área de operandos. Ejemplos: entrada E12.1; palabra de marcas MW 25; bloque de datos DB 3.

## F

**Función (FC)** Según la norma IEC 1131-3, una función (FC) es un bloque lógico sin memoria que permite transferir parámetros en el programa de usuario. Por consiguiente, las funciones son especialmente apropiadas para programar funciones complejas de uso frecuente, como p. ej. determinados cálculos. Puesto que no poseen memoria, los valores calculados tienen que seguir procesándose directamente después de la llamada de la FC.

Al llamar al bloque se asigna un parámetro actual (=real) al parámetro formal, con lo cual el bloque llamado opera con su valor real. Los parámetros formales son datos locales del bloque y pueden ser parámetros de entrada, de salida y de entrada/salida.

## I

**Instancia** Por “instancia” se entiende la llamada de un bloque de función que tiene asignado un bloque de datos de instancia.

**Instrucción** La instrucción constituye la menor unidad independiente de un programa creado en un lenguaje textual y representa una orden de ejecución para el procesador.

## L

### **Lenguaje de programación**

Los lenguajes de programación sirven para crear programas ofreciendo un determinado repertorio lingüístico constituido por instrucciones gráficas o textuales. Las instrucciones las introduce el usuario con un editor y luego se compilan en un programa de usuario ejecutable.

### **Lista de instrucciones (AWL)**

La lista de instrucciones es un lenguaje de programación textual orientado a la máquina.

## LL

### **Llamada al bloque**

Por “llamada” al bloque se entiende la derivación de la ejecución del programa al bloque llamado.

## M

### **Macro**

Una macro es una secuencia de instrucciones cuya ejecución está optimizada por estar resumidas en una llamada nemotécnica.

## O

### **Operando**

Un operando es parte de una instrucción de STEP 7 que indica qué debe hacer el procesador y con qué. Al operando se puede acceder con direccionamiento absoluto o simbólico.

### **Operación**

Una operación es parte de una instrucción STEP 7 que indica lo que tiene que hacer el procesador.

## P

### **Parametrizar**

Parametrizar significa ajustar el comportamiento de un módulo.

### **Parámetros actuales**

Los parámetros actuales sustituyen a los parámetros formales al llamar a un bloque de función (FB) o a una función (FC). Por ejemplo, el parámetro formal "START" se sustituye por el parámetro actual "E 3.6".



---

<b>Parámetros de bloques</b>	Los parámetros de bloques son comodines que se usan en bloques multiuso y que reciben valores reales cuando se llama al bloque en cuestión.
<b>Parámetros formales</b>	Un parámetro formal es un comodín para el parámetro “real” (parámetro actual) de un bloque lógico parametrizable. En el caso de los bloques de función (FB) o las funciones (FC) es el usuario quien declara los parámetros formales, mientras que en el caso de los bloques de función de sistema (SFB) y en el de las funciones de sistema (SFC) ya están presentes.
<b>Periferia descentralizada</b>	La periferia descentralizada está formada por módulos analógicos y digitales no ubicados en el bastidor central. Una característica de la periferia descentralizada es su construcción modular, cuyo objetivo es reducir los costes de cableado disponiendo módulos periféricos a pie del proceso.
<b>Programa S7</b>	Se trata de un contenedor de bloques, fuentes y planos (esquemas) de módulos S7 programables, el cual contiene asimismo la tabla de símbolos.
<b>Proyecto</b>	Un proyecto es un contenedor que integra todos los objetos de una solución de automatización, independientemente del número de equipos y módulos, y de cómo estén interconectados en la red.
<b>Puntero</b>	Un puntero es una variable que no contiene un valor determinado sino la dirección de otra variable. En las operaciones que trabajan con punteros hay que vigilar que el tipo indicado a la derecha del operador coincida con el tipo indicado a su izquierda.
<b>R</b>	
<b>Remanencia</b>	Los datos son remanentes cuando tras un corte de alimentación mantienen el mismo valor que tenían antes de cortarse la alimentación. La remanencia se puede realizar con dos tipos de respaldo diferentes: respaldo de la alimentación eléctrica y respaldo de la memoria.
<b>S</b>	
<b>Símbolo</b>	<p>Un símbolo es un nombre definido por el usuario respetando determinadas reglas de sintaxis. Una vez definido su significado (p. ej. una variable, un tipo de datos, una marca de salto o un bloque) se puede utilizar para la programación y para “manejo y observación” (M+V).</p> <p>Ejemplo: operando E 5.0, tipo de datos BOOL, símbolo del pulsador PARO_EMERGENCIA.</p>

**Símbolos globales y símbolos locales**

Los símbolos pueden ser globales o locales. A los símbolos declarados globalmente pueden acceder todas las partes del programa. Cada uno de los símbolos tiene que ser unívoco para todo el programa de usuario. Los símbolos locales del bloque sólo valen para el bloque en que han sido declarados.

**T**

**Tabla de símbolos**

La tabla de símbolos sirve para asignar símbolos a las direcciones de los datos globales y a las direcciones de los bloques. Ejemplos: PARO\_EMERGENCIA (símbolo), E1.7 (dirección) o regulador (símbolo), SFB 24 (bloque)

**Tipos de datos**

El tipo de datos permite definir cómo se utilizará el valor de una variable o de una constante en el programa de usuario. El usuario dispone en SIMATIC S7 de dos clases de tipos de datos según la norma IEC 1131-3: los tipos de datos simples y los tipos de datos compuestos.

**Tipos de datos simples**

Los tipos de datos simples son tipos de datos predefinidos según la norma IEC 1131-3, p.ej.: el tipo de datos BOOL define una variable binaria ("bit"), mientras que el tipo de datos INT define una variable de 16 bits en coma fija.

**Tipos de datos compuestos**

Los tipos de datos compuestos los define el usuario en el área de declaración del tipo de datos. Estos tipos de datos no tienen nombre propio por lo que tampoco son de uso múltiple. Se distingue entre arrays y estructuras, los cuales a su vez pueden ser tipos de datos *String* y *Date and Time*.

**V**

**Variable**

Las variables definen datos de contenido variable que pueden ser utilizados en el programa de usuario STEP 7. Las variables se componen de un operando y un tipo de datos, y se les puede asignar un símbolo para identificarlas.

# Índice alfabético

## A

- Actuator/Sensor-Interface, 2-10
- Administrador SIMATIC, 3-3
  - ventana, 3-13
- Advertencia, mensajes del convertidor, 6-10
- Ajustar / leer la hora, 3-23
- Ajuste del sistema S5, 3-26
- Alarma, 3-20, 3-22
- Alarma cíclica, 3-20
- Alarma de diagnóstico, 2-15, 9-3
- Alarma de multiprocesamiento, 3-20
- Alarma de proceso, 2-15, 3-20
- Alarma de retardo, 3-20
- Alarma horaria, 3-20
- Áreas de operandos, resumen, 3-32
- Aritmética
  - en coma fija, 3-36
  - en coma flotante, 3-36
- Aritmética en coma flotante, 3-28
- Arrancar STEP 7, 3-3
- Arranque, 3-20
- AS-Interface, 2-10
- AS511, 2-3
- Asignación de direcciones, 4-4
- Autorización, 3-2

## B

- Bloque
  - comparativa STEP 5 / STEP 7, 3-17
  - de comentario, 3-17
  - de datos, 3-17
  - de datos de sistema, 3-17, 3-19
  - de función, 3-17, 3-18
  - de función de sistema, 3-17, 3-19
  - de manejo, 2-20
  - de organización, 3-17, 3-20, 5-7
  - de paso, 3-17
  - de programa, 3-17
  - de STEP 5, 3-17

- Bloque, de función estándar S5, 7-6
- Bloques de las CPUs, 2-6
- Bloques S7, crear, 3-15
- Búfer de diagnóstico, 2-15

## C

- Cambiar direcciones, 7-2
- Cambio de flanco, 2-15
- Cápsula de adaptación, 2-13, 4-2
- Catálogo de módulos, 3-10
- CD-ROM, 2-1
- Coherencia, comprobar, 8-1
- Compilador, 8-1
- Compilar, 8-1
- Comunicación
  - controlada por eventos, 2-19
  - función de, 2-18
- Comunicación de datos globales, 2-19
- Comunicación GD, 2-19
- Con operando, operación, convertible, A-4
- Conexión punto a punto, 2-10
  - interface del programa de usuario, 2-20
  - módulos, 2-12
- Configurar el hardware, 3-9
- Configurar enlaces de comunicación, 3-11
- Contadores de las CPUs, 2-6
- Contenedor de bloques, objeto de STEP 7, 3-6
- Conversión, condiciones, 4-2
- Convertibilidad
  - de operaciones
    - con operando, A-4
    - sin operando, A-3
  - de operandos, A-1
- COROS, 2-3
- CPU, 5-3
  - bloques, 2-6
  - contadores, 2-6
  - datos locales, 2-6
  - datos remanentes, 2-6
  - DBs, 2-6
  - entradas analógicas, 2-6

- entradas digitales, 2-6
- FBs, 2-6
- FCs, 2-6
- imagen del proceso, 2-6
- marcas, 2-6
- memoria de carga, 2-6, 2-7
- memoria de trabajo, 2-6
- OBs, 2-6
- S7-400, 2-7
- salidas analógicas, 2-6
- salidas digitales, 2-6
- SFBs, 2-6
- SFCs, 2-6
- temporizadores, 2-6
- CPUs, S7-300, 2-6
- Crear macros, 5-8
- Crear software, 3-13
  - crear componentes, 3-15
  - vista de los componentes, 3-14

**D**

- Datos locales, 3-33
- Datos locales de las CPUs, 2-6
- Datos remanentes de las CPUs, 2-6
- DB 1, 3-26
- DB 1 / DX 0, 5-4
- DB 1 /DX 0, 4-4
- Dirección
  - absoluta, 4-3
  - cambiar, 7-2
- Direccionamiento
  - absoluto, 3-39
  - indirecto, 3-43
    - conversión, 7-4
  - indirecto por memoria, 3-44
  - indirecto por registro, 3-45
  - operandos de datos, 3-41
  - simbólico, 3-39
- Direcciones, asignar, 4-4
- DX 0, 3-26

**E**

- Ejemplo
  - datos locales temporales, 9-5
  - información de arranque, 9-9
  - transferencia en bloque, 9-12
  - tratamiento de valores analógicos, 9-2
- Enlace, configurar un enlace con un equipo S5, 3-12

- Entradas
  - analógicas, 2-6
  - digitales, 2-6
- Equipo, objeto de STEP 7, 3-5
- Esclavo DP, módulos, 2-17
- Esclavos FMS, 2-17
- Espacio de memoria, 4-3
- ET 200, 2-17
- Ethernet, 2-10

**F**

- Fallo de la pila, 3-22
- FDL (SDA), 2-18
- Fichero de proyecto, 3-4
- Formato
  - de constantes, 3-31
  - del puntero, 3-43
- Fuente, objeto de STEP 7, 3-6
- Fuente ASCII, 3-16
- Fuentes de alimentación, 2-8
- Funciones, 3-18
  - analógicas, 3-29
  - básicas, 3-29
  - de elaboración (B MW, B DW), 4-3
  - de procesamiento, 4-3
  - de señales, 3-28
  - de sistema, 3-17, 3-19
  - especiales, 3-22
  - estándar, 3-28
  - integradas, 3-28
  - matemáticas, 3-29, 3-38

**G**

- Gama de prestaciones, 2-2

**H**

- Hardware, objeto de STEP 7, 3-5
- Herramienta
  - de configuración, 2-22
  - para cambiar el hardware de S5 a S7, 2-1
- HMI (Human Machine Interface), 2-3, 2-21

**I**

- IM, módulo interfase, 2-9
- Imagen de proceso de la CPU, 2-6

**Importar**

- fuente ASCII, 3-16
- tabla de símbolos, 3-40
- Industrial Ethernet, 2-10, 2-18
  - interface del programa de usuario, 2-20
  - módulos, 2-11
- Información de arranque, 3-34, 9-9
- Información del módulo, 5-3
- Instalación del software STEP 7, 3-2
- Instrucciones de interrupción, 3-38
- Instrucciones para páginas, 3-38
- Integración Total en Automatización, 1-1
- Interface
  - de PG, 2-10
  - multipunto, 2-3
- Interrupción de hardware, 3-20
- Interruptores DIL, 2-5
- ISO-on-TCP, 2-18

**L**

- Leva electrónica, 2-13
- Librería estándar, 3-15
- Licencia de uso, 3-2
- LIR, 4-3
- Lista, de asignación, 3-39, 6-1, 6-4
- Lista de referencias cruzadas, 6-1

**M**

- Macros, 5-5
  - de instrucciones, 5-6
  - de OBs, 5-7
- Maestro DP, módulos, 2-17
- Maestros FMS, 2-17
- Manejo y visualización, 2-21
- Marcas
  - de acoplamiento, 3-23
  - de las CPUs, 2-6
  - de trabajo, 3-33, 9-6
- Memoria de carga
  - CPU S7-300, 2-6
  - CPU S7-400, 2-7
- Memoria de trabajo de la CPU, 2-6
- Mensaje de error, 6-8
- Micro-PLC, 2-2

**Módulos**

- centrales, CPU S7-400, 2-7
- contadores, 2-13
- CP, 2-10
- de comunicación, 2-10
- de función, 2-13
- de manejo, 2-20
- de posicionamiento, 2-13
- de regulación, 2-13
- de señales, 2-15
- de simulación, 2-16
- dosificadores, 2-13
- FM, 2-13
- IM, 2-9
- interfase, 2-9
- IP, 2-13
- panorámica, 2-4
- preprocesadores de señal, 2-13
- SM, 2-15
- WF, 2-13
- Módulos , centrales, CPU S7-300, 2-6
- Módulos (software)
  - de comentario, 3-17
  - de datos, 3-17
  - de función, 3-17
  - de organización, 3-17
  - de paso, 3-17
  - de programa, 3-17
- MPI, 2-3, 2-10, 2-18

**N**

- No convertibilidad
  - de operaciones, A-6
  - de operandos, A-2
- Nuevo arranque, 3-20

**O**

- OB 1, ejemplo, 9-15
- OB especial, 3-17
- Operación
  - convertible
    - con operando, A-4
    - sin operando, A-3
  - no convertible, A-6

## Operaciones

- con acumuladores, 3-35
- con bloques, 3-37
- con bloques de datos, 3-37
- con registros, 3-35
- de carga, 3-36
- de comparación, 3-36
- de contaje, 3-35
- de conversión, 3-36
- de desplazamiento, 3-37
- de habilitación de salidas, 3-37
- de rotación, 3-37
- de salto, 3-37
- de temporización, 3-35
- de transferencia, 3-36
- lógicas
  - con bits, 3-35
  - con palabras, 3-36
- nulas, 3-38
- panorámica, 3-35

Operaciones de Stop, 3-37

## Operandos

- convertibles, A-1
- no convertibles, A-2

Operator Panel (OP), 2-21

## P

Panel de operador (OP), 2-21

Parametrización de módulos/tarjetas, comparativa S5/S7, 2-5

Periferia descentralizada, 2-17

Pro Tool, 2-22

PROFIBUS, 2-10, 2-18

- interface del programa de usuario, 2-20
- módulos, 2-11

Proyecto, 3-4

Proyecto de STEP 5, 3-4

Proyecto de STEP 7, 3-4

- archivar, 3-8
- crear, 3-7
- crear una copia de seguridad, 3-8

Proyecto S7, crear, 4-4

Proyectos, crear, 3-7

Proyectos de STEP 7, componentes, 3-5

Puerto de programación

- AS511, 2-3
- MPI, 2-3

Puntero ANY, 9-13

## R

Rearranque, 3-20

Reasignar, 5-4

Reasignar direcciones, 7-2

Rebase de límites, 3-22

Red, objeto de STEP 7, 3-5

Registro DB, 3-41, 3-42

Registro de desplazamiento, 3-24

Registro de direccionamiento, 3-45

Registro RB, 7-5

Respaldo de la memoria, 2-7

RET\_VAL, 9-3

## S

### Salidas

analógicas, 2-6

digitales, 2-6

Servicio FMS, 2-19

SIMATIC S7, panorámica, 2-2

Símbolos locales, 3-40

Sin operando, operación, convertible, A-3

SINEC H1, 2-11

SINEC L1, 2-11, 3-26

SINEC L2, 2-11, 3-26

SINEC S1, 2-11

Sistemas de automatización, panorámica, 2-2

STEP 7, instalar, 3-2

Subred, 2-10

Suma de verificación, 3-23

## T

Tabla de enlaces, 3-11

objeto de STEP 7, 3-6

Tabla de símbolos, 3-40

crear, 3-15

ejemplo, 9-15

objeto de STEP 7, 3-6

Tarea no prioritaria, 3-20

Temporizadores de la CPU, 2-6

Tiempo de vigilancia del ciclo, 3-23

Tipos de archivos, 3-40

Tipos de bloques en S5 y S7, 3-25

TIR, 4-3

Transferencia en bloque, 3-38, 7-5

ejemplo, 9-12

Transporte ISO, 2-18

Tratamiento de errores, 3-21  
Tratamiento de valores analógicos, ejemplo, 9-2

## **U**

Universalidad, 1-1

## **V**

Valor de retorno  
de una función, 9-3  
de una función de sistema, 3-22

Visualización, 2-22

## **W**

WinCC, 2-22

