

SIEMENS

SIMATIC

Programar con STEP 7

Manual

Prológo	
Índice	
Presentación del producto e instalación	1
Instalación	2
Cómo diseñar una solución de automatización	3
Nociones básicas para diseñar la estructura del programa	4
Cómo arrancar y utilizar STEP 7	5
Elaboración de proyectos	6
Editar proyectos con diferentes versiones de STEP 7	7
Definir símbolos	8
Crear bloques y librerías	9
Crear bloques lógicos	10
Crear bloques de datos	11
Parametrizar bloques de datos	12
Crear fuentes AWL	13
Mostrar datos de referencia	14
Comprobar coherencia del bloque y fecha y hora como propiedad del bloque	15
Configurar avisos	16
Configurar variables para manejo y visualización	17
Establecer enlaces online y ajustar la CPU	18
Cargar	19
Test	20
Test con el estado del programa	21
Test con el programa de simulación S7-PLCSIM (software opcional)	22
Diagnóstico	23
Imprimir y archivar	24
Trabajar con sistemas de automatización M7	25
Consejos y trucos	26
Anexos	A
Índice alfabético	

Este manual forma parte del paquete de documentación con la referencia:
6ES7810-4CA08-8DW0

Consignas de seguridad

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



Peligro

Significa que, si no se adoptan las medidas preventivas adecuadas se producirá la muerte, o bien lesiones corporales graves.



Advertencia

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



Precaución

Con triángulo de advertencia significa que si no se adoptan las medidas preventivas adecuadas, pueden producirse lesiones corporales.

Precaución

Sin triángulo de advertencia significa que si no se adoptan las medidas preventivas adecuadas, pueden producirse daños materiales.

Atención

Significa que puede producirse un resultado o estado no deseado si no se respeta la consigna de seguridad correspondiente.

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

Personal cualificado

El equipo/sistema correspondiente sólo deberá instalarse y operarse respetando lo especificado en este documento. Sólo está autorizado a intervenir en este equipo el 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:



Advertencia

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 asociado a los equipos y componentes de Siemens y de tercera que han sido 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 un manejo y un mantenimiento rigurosos.

Marcas registradas

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

Exención de responsabilidad

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

Prológo

Objetivo del manual

Este manual pretende dar al lector una visión general de cómo programar con **STEP 7**.

Entre otras cosas explica cómo instalar y arrancar el software, cómo crear programas y cuáles son los principales componentes de los programas de usuario.

Este manual está dirigido a aquellas personas que realizan soluciones de automatización con STEP 7 para los sistemas de automatización SIMATIC S7.

Recomendamos familiarizarse primero con los ejemplos descritos en el Getting Started de STEP 7 "Introducción y ejercicios prácticos". Estos ejemplos ayudan a iniciarse en el tema de la "Programar con STEP 7".

Nociones básicas

Se requieren conocimientos básicos en el campo de la automatización Además es necesario estar familiarizado con el uso de ordenadores o equipos similares a un PC (p. ej. unidades de programación) que funcionen con los sistemas operativos MS Windows 2000 Professional, MS Windows XP Profesional o MS Windows Server 2003.

Objeto del manual

El software en el que se basan las indicaciones del manual es STEP 7 V5.4.

Encontrará información sobre los Servicepacks que salgan al mercado después de la fecha de publicación del manual

- en el archivo "Léame.wri"
- en las ayudas actualizadas de STEP 7

El tema "Novedades" de la ayuda en pantalla le ayudará a familiarizarse con las nuevas funciones de STEP 7

Paquetes de documentación de STEP 7

Este manual forma parte del paquete de documentación "STEP 7 Información básica".

La tabla siguiente le proporcionará una visión de conjunto de la documentación de STEP 7:

Manuales	Tema	Referencia
Información básica de STEP 7 compuesta por: <ul style="list-style-type: none"> • STEP 7: Introducción y ejercicios prácticos • Programar con STEP 7 • Configurar el hardware y la comunicación con STEP 7 • De S5 a S7, Guía para facilitar la transición 	Nociones básicas para el personal técnico. Describe cómo realizar soluciones de control con el software STEP 7 para los sistemas S7-300/400.	6ES7810-4CA08-8DW0
Información de referencia para STEP 7, compuesta por <ul style="list-style-type: none"> • Manuales KOP/FUP/AWL para S7-300/400 • Funciones estándar y funciones de sistema para S7-300/400 Tomo 1 y Tomo 2 	Esta obra de consulta describe los lenguajes de programación KOP, FUP y AWL así como las funciones estándar y las funciones de sistema como complemento a la 'Información básica de STEP'.	6ES7810-4CA08-8DW1

Ayudas en pantalla	Tema	Referencia
Ayuda de STEP 7	Nociones básicas para diseñar programas y configurar el hardware con STEP 7. Disponible en forma de Ayuda en pantalla.	Componente del paquete de software STEP 7
Ayudas de referencia para AWL/KOP/FUP Ayudas de referencia para SFBs/SFCs Ayudas de referencia para los bloques de organización	Información de referencia sensible al contexto	Componente del paquete de software STEP 7

Ayuda en pantalla

Como complemento al manual puede recurrir a la Ayuda en pantalla integrada en el software.

A la Ayuda que está integrada en el software se accede de distinta manera:

- El menú **Ayuda** ofrece varios comandos de menú: **Temas de Ayuda** abre el índice de la Ayuda de STEP 7.
- **Uso de la Ayuda** explica detalladamente cómo utilizar la Ayuda en pantalla.
- La Ayuda sensible al contexto ofrece información sobre el contexto actual, p. ej. sobre el cuadro de diálogo que esté abierto o sobre la ventana activa. Para acceder a esta ayuda pulse el botón de comando "Ayuda" o bien la tecla F1.
- También la barra de estado ofrece ayuda sensible al contexto. Si sitúa el puntero del ratón en un comando de menú aparecerá una breve explicación sobre dicho comando.
- Situando el puntero del ratón unos segundos sobre uno de los botones de la barra de herramientas aparecerá una breve explicación al respecto.

Si prefiere leer los textos de la Ayuda en papel, puede imprimir los temas o libros que le interesen o bien imprimir toda la Ayuda.

Al igual que los manuales "Configurar el hardware y la comunicación con STEP 7", "Modificaciones con la instalación en marcha mediante CiR" y "Sistema de automatización S7-400H - Sistemas de alta disponibilidad", el presente manual constituye un extracto de la Ayuda de STEP 7. Si necesita instrucciones más detalladas, consulte la ayuda de STEP 7.

Como la estructura del manual se corresponde a grandes rasgos con la de la Ayuda en pantalla, puede alternar la lectura del manual con la de la Ayuda en pantalla.

Después de instalar STEP 7 encontrará los manuales electrónicos en el menú **Inicio > SIMATIC > Documentación**

Asistencia adicional

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.

Encontrará a su persona de contacto en la página de Internet:

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

Encontrará una guía sobre el conjunto de la información técnica correspondiente a los distintos productos y sistemas SIMATIC en la página de Internet:

<http://www.siemens.com/simatic-tech-doku-portal>

Encontrará el catálogo y el sistema de pedidos on-line en:

<http://mall.automation.siemens.com/>

Centro de formación SIMATIC

Para ofrecer a nuestros clientes un fácil aprendizaje de los sistemas de automatización SIMATIC S7, les ofrecemos distintos cursos de formación. Diríjase a su centro de formación regional o a la central en D 90327 Nuernberg.

Teléfono: +49 (911) 895-3200.

Internet: <http://www.sitrain.com>

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

Índice

1	Presentación del producto e instalación	1-1
1.1	Guía de orientación de STEP 7	1-1
1.2	El software estándar STEP 7	1-6
1.3	Novedades de la versión 5.4 de STEP 7	1-10
1.4	Posibilidades de ampliar el software estándar STEP 7	1-12
1.4.1	Herramientas de ingeniería	1-13
1.4.2	Software Runtime	1-15
1.4.3	Interfaces hombre-máquina	1-17
2	Instalación	2-1
2.1	Automation License Manager	2-1
2.1.1	Autorización de utilización con el Automation License Manager	2-1
2.1.2	Instalar el Automation Licence Manager	2-4
2.1.3	Reglas para la manipulación de claves de licencia	2-5
2.2	Instalar STEP 7	2-6
2.2.1	Procedimiento de instalación	2-8
2.2.2	Ajustar el interface PG/PC	2-11
2.3	Desinstalar STEP 7	2-13
3	Cómo diseñar una solución de automatización	3-1
3.1	Procedimiento básico para planificar una solución de automatización	3-1
3.2	Subdividir el proceso en tareas y áreas	3-2
3.3	Describir el funcionamiento en diversas áreas	3-4
3.4	Listado de entradas y salidas	3-6
3.5	Crear un diagrama de E/S para los motores	3-7
3.6	Crear un diagrama de E/S para las válvulas	3-7
3.7	Definir los requerimientos de seguridad	3-8
3.8	Describir los elementos necesarios para manejo y visualización	3-9
3.9	Crear un esquema de configuración	3-10
4	Nociones básicas para diseñar la estructura del programa	4-1
4.1	Programas de una CPU	4-1
4.2	Bloques del programa de usuario	4-2
4.2.1	Bloques de organización y estructura del programa	4-3
4.2.2	Jerarquía de llamada en el programa de usuario	4-9
4.2.3	Tipos de bloques	4-11
4.2.3.1	Bloque de organización para la ejecución cíclica del programa (OB 1)	4-11
4.2.3.2	Funciones (FC)	4-16
4.2.3.3	Bloques de función (FB)	4-18
4.2.3.4	Bloques de datos de instancia	4-21
4.2.3.5	Bloques de datos globales (DB)	4-24
4.2.3.6	Bloques de función de sistema (SFB) y funciones del sistema (SFC)	4-25

4.2.4	Bloques de organización para la ejecución controlada por alarmas	4-27
4.2.4.1	Bloques de organización de alarma horaria (OB 10 a OB 17)	4-28
4.2.4.2	Bloques de organización de alarma de retardo (OB 20 a OB 23)	4-30
4.2.4.3	Bloques de organización de alarma cíclica (OB 30 a OB 38)	4-31
4.2.4.4	Bloques de organización de alarma de proceso (OB 40 a OB 47)	4-33
4.2.4.5	Bloques de organización de arranque (OB 100 / OB 101 / OB 102)	4-34
4.2.4.6	Bloque de organización de tarea no prioritaria (OB 90)	4-36
4.2.4.7	Bloques de organización de tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)	4-38
5	Cómo arrancar y utilizar STEP 7	5-1
5.1	Cómo iniciar STEP 7	5-1
5.2	Arrancar STEP 7 con los parámetros de arranque estándar	5-2
5.3	Cómo acceder a las funciones de ayuda	5-4
5.4	Objetos y su jerarquía	5-4
5.4.1	Objeto "Proyecto"	5-6
5.4.2	Objeto "Librería"	5-8
5.4.3	Objeto "Equipo"	5-9
5.4.4	Objeto "Módulo programable"	5-10
5.4.5	Objeto "Programa S7/M7"	5-12
5.4.6	Objeto "Carpeta de bloques"	5-14
5.4.7	Objeto "Carpeta de fuentes"	5-18
5.4.8	Programa S7/M7 sin equipo ni CPU	5-19
5.5	Interface de usuario	5-20
5.5.1	Filosofía de manejo	5-20
5.5.2	Estructura de la ventana	5-21
5.5.3	Elementos de los cuadros de diálogo	5-22
5.5.4	Crear y manejar objetos	5-23
5.5.5	Elegir objetos en cuadros de diálogo	5-28
5.5.6	Memoria de sesión	5-29
5.5.7	Cambiar la organización de las ventanas	5-29
5.5.8	Guardar y restablecer la organización de las ventanas	5-30
5.6	Manejo de las teclas	5-31
5.6.1	Combinaciones de teclas para comandos de menú	5-31
5.6.2	Combinaciones de teclas para mover el cursor	5-33
5.6.3	Combinaciones de teclas para seleccionar textos	5-35
5.6.4	Combinaciones de teclas para acceder a la Ayuda en pantalla	5-35
5.6.5	Combinaciones de teclas para cambiar de una ventana a otra	5-36
6	Elaboración de proyectos	6-1
6.1	Estructura de los proyectos	6-1
6.2	Información importante sobre la protección de acceso	6-3
6.3	Información importante sobre el informe de cambios	6-4
6.4	Utilizar juegos de caracteres de otros idiomas	6-5
6.5	Ajustar el idioma de Windows	6-8
6.6	Crear proyectos	6-9
6.6.1	Crear un proyecto	6-9
6.6.2	Insertar equipos	6-11
6.6.3	Insertar un programa S7 o M7	6-12
6.7	Elaborar un proyecto	6-14
6.7.1	Comprobar los paquetes de software utilizados en el proyecto	6-15
6.8	Gestionar textos en varios idiomas	6-15
6.8.1	Tipos de textos gestionados en varios idiomas	6-17
6.8.2	Estructura del archivo de exportación	6-18
6.8.3	Gestionar textos de usuario cuya fuente no está instalada	6-20
6.8.4	Información sobre el archivo de informe	6-21
6.8.5	Optimización de la plantilla para la traducción	6-22
6.8.6	Optimización del proceso de traducción	6-23

6.9	Micro Memory Card (MMC) como soporte de datos	6-24
6.9.1	Información importante sobre Micro Memory Cards (MMC)	6-24
6.9.2	Utilizar la Micro Memory Card como soporte de datos.....	6-25
6.9.3	Archivo Memory Card	6-25
6.9.4	Guardar datos de proyecto en la Micro Memory Card (MMC)	6-26
7	Editar proyectos con diferentes versiones de STEP 7	7-1
7.1	Editar proyectos y librerías de la versión 2.....	7-1
7.2	Ampliar esclavos DP creados con versiones anteriores de STEP 7	7-1
7.3	Editar configuraciones actuales con versiones anteriores de STEP 7	7-3
7.4	PC SIMATIC - Utilizar configuraciones de versiones anteriores	7-4
7.5	Representación de módulos configurados con nuevas versiones de STEP 7 o con paquetes opcionales.....	7-6
8	Definir símbolos	8-1
8.1	Direccionamiento absoluto y simbólico.....	8-1
8.2	Símbolos globales y locales	8-3
8.3	Representación de símbolos globales y locales.....	8-4
8.4	Ajustar la preferencia de operandos (simbólica/absoluta).....	8-5
8.5	Tabla de símbolos para los símbolos globales.....	8-8
8.5.1	Estructura y componentes de la tabla de símbolos.....	8-8
8.5.2	Direcciones y tipos de datos admisibles en la tabla de símbolos	8-10
8.5.3	Símbolos incompletos y ambiguos en la tabla de símbolos	8-11
8.6	Métodos para introducir símbolos globales	8-12
8.6.1	Observaciones generales para introducir símbolos	8-12
8.6.2	Introducir símbolos globales en un cuadro de diálogo	8-13
8.6.3	Introducir símbolos globales en la tabla de símbolos.....	8-14
8.6.4	Mayúsculas y minúsculas en los símbolos	8-15
8.6.5	Exportar e importar tablas de símbolos	8-17
8.6.6	Formatos de archivos para importar/exportar una tabla de símbolos	8-17
8.6.7	Editar áreas en tablas de símbolos	8-20
9	Crear bloques y librerías	9-1
9.1	Seleccionar el método de creación.....	9-1
9.2	Elegir el lenguaje de programación	9-2
9.2.1	Lenguaje de programación KOP (esquema de contactos)	9-4
9.2.2	Lenguaje de programación FUP (diagrama de funciones).....	9-5
9.2.3	Lenguaje de programación AWL (lista de instrucciones)	9-6
9.2.4	Lenguaje de programación S7-SCL	9-7
9.2.5	Lenguaje de programación S7-GRAPH (control secuencial)	9-8
9.2.6	Lenguaje de programación S7-HiGraph (grafo de estado)	9-9
9.2.7	Lenguaje de programación S7-CFC	9-11
9.3	Crear bloques	9-12
9.3.1	Carpeta de bloques.....	9-12
9.3.2	Tipos de datos de usuario (UDT).....	9-13
9.3.3	Propiedades del bloque	9-14
9.3.4	Visualizar longitudes de bloques	9-17
9.3.5	Comparar bloques	9-18
9.3.6	Recablear.....	9-21
9.3.7	Atributos de bloques y parámetros	9-21
9.4	Trabajar con librerías	9-22
9.4.1	Estructura jerárquica de las librerías	9-23
9.4.2	Librerías estándar disponibles.....	9-24

10	Crear bloques lógicos	10-1
10.1	Nociones básicas para crear bloques lógicos	10-1
10.1.1	Estructura de la ventana del editor de programas.....	10-1
10.1.2	Procedimiento básico para crear bloques lógicos	10-3
10.1.3	Preajustes para el editor de programas KOP/FUP/AWL.....	10-4
10.1.4	Derechos de acceso a bloques y fuentes.....	10-4
10.1.5	Instrucciones de la vista general de elementos de programa	10-5
10.2	Editar la declaración de variables.....	10-6
10.2.1	Utilización de la declaración de variables en los bloques lógicos	10-6
10.2.2	Interacción entre la declaración de variables y el área de instrucciones	10-7
10.2.3	Estructura de la ventana de declaración de variables.....	10-8
10.3	Multiinstancias en la tabla de declaración	10-9
10.3.1	Utilización de multiinstancias.....	10-9
10.3.2	Reglas para declarar multiinstancias.....	10-10
10.3.3	Introducir la multiinstancia en la ventana de declaración de variables	10-10
10.4	Indicaciones generales para introducir instrucciones y comentarios	10-11
10.4.1	Estructura del área de instrucciones	10-11
10.4.2	Procedimiento para introducir instrucciones.....	10-12
10.4.3	Introducir símbolos globales en un programa	10-13
10.4.4	Títulos y comentarios de bloques y segmentos	10-14
10.4.5	Introducir comentarios de bloques/segmentos.....	10-15
10.4.6	Trabajar con plantillas de segmentos.....	10-16
10.4.7	Función de búsqueda de errores en el área de instrucciones	10-17
10.5	Editar instrucciones KOP en el área de instrucciones.....	10-18
10.5.1	Ajustes para el lenguaje de programación KOP.....	10-18
10.5.2	Reglas para introducir operaciones KOP	10-18
10.5.3	Interconexiones no admisibles en KOP.....	10-21
10.6	Editar instrucciones FUP en el área de instrucciones.....	10-22
10.6.1	Ajustes para el lenguaje de programación FUP	10-22
10.6.2	Reglas para introducir operaciones FUP.....	10-23
10.7	Editar instrucciones AWL en el área de instrucciones	10-25
10.7.1	Ajustes para el lenguaje de programación AWL	10-25
10.7.2	Reglas para introducir instrucciones AWL.....	10-25
10.8	Actualizar llamadas a bloques	10-26
10.8.1	Cambiar interfaces.....	10-27
10.9	Guardar bloques lógicos.....	10-28
11	Crear bloques de datos	11-1
11.1	Nociones básicas para crear bloques de datos.....	11-1
11.2	Ver declaración de bloques de datos	11-2
11.3	Ver datos de los bloques de datos	11-3
11.4	Editar bloques de datos y guardar.....	11-4
11.4.1	Introducir la estructura de los bloques de datos globales	11-4
11.4.2	Introducir / visualizar la estructura de bloques de datos con FB asociado (DBs de instancia).....	11-5
11.4.3	Introducir la estructura de los tipos de datos de usuario (UDT).....	11-6
11.4.4	Introducir / visualizar la estructura de bloques de datos con UDT asociado.....	11-7
11.4.5	Cambiar valores de datos en la vista "Datos"	11-8
11.4.6	Inicializar los valores de datos.....	11-8
11.4.7	Guardar bloques de datos	11-9
12	Parametrizar bloques de datos	12-1
12.1	Parametrizar funciones tecnológicas.....	12-2

13	Crear fuentes AWL	13-1
13.1	Nociones básicas para programar en fuentes AWL	13-1
13.2	Reglas para programar en fuentes AWL	13-2
13.2.1	Reglas para introducir instrucciones en fuentes AWL.....	13-2
13.2.2	Reglas para declarar variables en fuentes AWL	13-3
13.2.3	Orden de los bloques en fuentes AWL.....	13-4
13.2.4	Reglas para definir los atributos de sistema en fuentes AWL.....	13-4
13.2.5	Reglas para definir las propiedades de bloques en fuentes AWL.....	13-5
13.2.6	Propiedades válidas para los diferentes tipos de bloques	13-7
13.3	Estructura de bloques en fuentes AWL	13-8
13.3.1	Estructura de los bloques lógicos en fuentes AWL	13-8
13.3.2	Estructura de bloques de datos en fuentes AWL	13-9
13.3.3	Estructura de tipos de datos de usuario en fuentes AWL	13-9
13.4	Sintaxis y formatos de bloques en fuentes AWL	13-10
13.4.1	Formato de los OBs.....	13-10
13.4.2	Formato de los FBs.....	13-11
13.4.3	Formato de las FCs	13-12
13.4.4	Formato de los DBs	13-13
13.5	Crear fuentes AWL	13-14
13.5.1	Crear fuentes AWL	13-14
13.5.2	Editar fuentes S7	13-14
13.5.3	Definir el diseño del texto fuente	13-15
13.5.4	Insertar plantillas de bloques en fuentes AWL	13-15
13.5.5	Insertar el contenido de otras fuentes AWL	13-15
13.5.6	Insertar en fuentes AWL el código de bloques existentes.....	13-15
13.5.7	Insertar fuentes externas.....	13-16
13.5.8	Generar fuentes AWL de bloques	13-16
13.5.9	Importar fuentes.....	13-17
13.5.10	Exportar fuentes.....	13-17
13.6	Guardar y compilar fuentes AWL y comprobar su coherencia.....	13-18
13.6.1	Guardar fuentes AWL	13-18
13.6.2	Comprobar la coherencia en fuentes AWL.....	13-18
13.6.3	Buscar errores en fuentes AWL.....	13-18
13.6.4	Compilar fuentes AWL	13-19
13.7	Ejemplos de fuentes AWL	13-20
13.7.1	Ejemplos de declaración de variables en fuentes AWL	13-20
13.7.2	Ejemplo de OBs en fuentes AWL	13-21
13.7.3	Ejemplo de FCs en fuentes AWL.....	13-22
13.7.4	Ejemplo de FBs en fuentes AWL.....	13-24
13.7.5	Ejemplos de DBs en fuentes AWL.....	13-26
13.7.6	Ejemplo de UDTs en fuentes AWL	13-27
14	Mostrar datos de referencia	14-1
14.1	Sinopsis de los datos de referencia.....	14-1
14.1.1	Lista de referencias cruzadas	14-2
14.1.2	Estructura del programa	14-3
14.1.3	Plano de ocupación	14-5
14.1.4	Símbolos no utilizados	14-7
14.1.5	Operandos sin símbolo	14-8
14.1.6	Visualizar la información del bloque en KOP, FUP, AWL	14-8
14.2	Trabajar con datos de referencia.....	14-9
14.2.1	Métodos de visualización de los datos de referencia.....	14-9
14.2.2	Visualizar listas en ventanas de trabajo adicionales	14-9
14.2.3	Crear y visualizar datos de referencia	14-10
14.2.4	Posicionamiento rápido en los puntos de aplicación del programa	14-11
14.2.5	Ejemplo para trabajar con puntos de aplicación.....	14-12

15	Comprobar coherencia del bloque y fecha y hora como propiedad del bloque	15-1
15.1	Comprobar coherencia de bloques.....	15-1
15.2	Fecha y hora como propiedad del bloque y conflictos de fecha y hora	15-3
15.3	Indicación de fecha y hora de bloques lógicos.....	15-4
15.4	Indicación de fecha y hora de bloques de datos globales.....	15-5
15.5	Indicación de fecha y hora de bloques de datos de instancia.....	15-5
15.6	Indicación de fecha y hora de UDTs y DBs derivados de UDTs.....	15-6
15.7	Corregir los interfaces en una FC, un FB o un UDT.....	15-6
15.8	Evitar errores al llamar bloques	15-7
16	Configurar avisos	16-1
16.1	El sistema de avisos	16-1
16.1.1	¿Qué procedimientos existen para crear avisos?	16-1
16.1.2	Selección del procedimiento de notificación.....	16-3
16.1.3	Componentes SIMATIC	16-5
16.1.4	Partes de un aviso	16-5
16.1.5	¿Qué bloques de notificación existen?.....	16-6
16.1.6	Parámetros formales, atributos del sistema y bloques de notificación.....	16-8
16.1.7	Plantilla y avisos	16-9
16.1.8	Crear una fuente AWL desde bloques notificadores	16-10
16.1.9	Asignar números a los avisos.....	16-10
16.1.10	Diferencias entre asignar números de aviso para todo el proyecto o para toda la CPU	16-11
16.1.11	Procedimientos para cambiar la asignación de números de aviso de un proyecto	16-11
16.2	Configurar avisos para todo el proyecto	16-12
16.2.1	Asignación de números de aviso para todo el proyecto.....	16-12
16.2.2	Asignar y editar avisos de bloque.....	16-12
16.2.2.1	Crear avisos de bloque (para todo el proyecto)	16-13
16.2.2.2	Editar avisos de bloque (para todo el proyecto).....	16-16
16.2.2.3	Configuración de avisos PCS7 (para todo el proyecto).....	16-17
16.2.3	Asignar y editar avisos de símbolos	16-18
16.2.3.1	Asignar y editar avisos de símbolos (para todo el proyecto).....	16-18
16.2.4	Crear y editar avisos de diagnóstico personalizados	16-19
16.3	Configurar avisos para toda la CPU	16-20
16.3.1	Asignación de números de aviso para toda la CPU	16-20
16.3.2	Asignar y editar avisos de bloque.....	16-21
16.3.2.1	Crear avisos de bloques (para toda la CPU).....	16-21
16.3.2.2	Editar avisos de bloques (para toda la CPU)	16-24
16.3.2.3	Configuración de avisos PCS 7 (para toda la CPU).....	16-24
16.3.3	Asignar y editar avisos de símbolos	16-26
16.3.3.1	Asignar y editar avisos de símbolos (para toda la CPU).....	16-26
16.3.4	Crear y editar avisos de diagnóstico personalizados	16-27
16.4	Trucos para editar avisos	16-28
16.4.1	Insertar valores asociados en avisos.....	16-28
16.4.2	Integrar textos de librerías en avisos.....	16-30
16.4.3	Borrar valores asociados	16-31
16.5	Traducir y editar textos para operarios	16-32
16.5.1	Traducir y editar textos de usuario	16-32
16.6	Editar y traducir librerías de textos	16-34
16.6.1	Librerías de textos de usuario	16-34
16.6.2	Crear librerías de textos de usuario.....	16-34
16.6.3	Editar librerías de textos de usuario	16-35
16.6.4	Librerías de textos de sistema.....	16-35
16.6.5	Traducir librerías de textos	16-36

16.7	Transferir los datos de configuración al sistema de destino	16-38
16.8	Visualizar avisos de CPU y avisos de diagnóstico personalizados.....	16-39
16.8.1	Configurar avisos de CPU	16-42
16.8.2	Visualizar los avisos de CPU guardados.....	16-42
16.9	Configurar 'Notificación de errores del sistema'	16-43
16.9.1	Componentes compatibles y su funcionalidad	16-45
16.9.2	Ajustes para notificar errores del sistema	16-48
16.9.3	Generar bloques para la notificación de errores del sistema	16-49
16.9.4	FB, DB generados	16-49
16.9.5	Crear textos de aviso en otros idiomas en 'Notificar errores de sistema'.....	16-51
17	Configurar variables para manejo y visualización	17-1
17.1	Configurar variables para manejo y visualización	17-1
17.2	Configurar atributos M+V con AWL, KOP y FUP	17-3
17.3	Configurar atributos M+V a través de la tabla de símbolos.....	17-4
17.3.1	Configurar atributos M+V desde de la tabla de símbolos.....	17-4
17.4	Modificar atributos para manejo y visualización con CFC.....	17-5
17.5	Transferir los datos de configuración al sistema de destino M+V	17-6
18	Establecer enlaces online y ajustar la CPU	18-1
18.1	Establecer enlaces online.....	18-1
18.1.1	Establecer un enlace online desde la ventana "Estaciones accesibles".....	18-2
18.1.2	Establecer un enlace online desde la ventana online del proyecto.....	18-3
18.1.3	Acceso online a sistemas de destino desde el multiproyecto	18-4
18.1.4	Protección por contraseña para acceder a sistemas de destino.....	18-6
18.1.5	Nota respecto a la actualización del contenido de ventanas	18-7
18.2	Visualizar y cambiar el estado operativo	18-8
18.3	Visualizar y ajustar la fecha y la hora	18-9
18.3.1	Relojes de la CPU con ajuste de zonas horarias y horario de verano/invierno	18-9
18.4	Actualizar el firmware.....	18-10
18.4.1	Actualizar el firmware de módulos y submódulos online.....	18-10
19	Cargar	19-1
19.1	Cargar en el sistema de destino desde la PG	19-1
19.1.1	Requisitos para cargar.....	19-1
19.1.2	Diferencia entre guardar y cargar bloques	19-2
19.1.3	Memorias de carga y de trabajo de la CPU.....	19-3
19.1.4	Procedimientos de carga en función de la memoria de carga	19-4
19.1.5	Cargar el programa en una CPU S7.....	19-5
19.1.5.1	Cargar con gestión de proyectos.....	19-5
19.1.5.2	Cargar sin gestión de proyectos	19-5
19.1.5.3	Recargar bloques en el sistema de destino	19-6
19.1.5.4	Guardar en la EPROM integrada los bloques cargados	19-6
19.1.5.5	Cargar mediante Memory Cards EPROM	19-7
19.2	Compilar varios objetos de la PG y cargar	19-8
19.2.1	Requisitos y observaciones para cargar.....	19-8
19.2.2	Compilar objetos y cargar.....	19-10
19.3	Cargar en la PG desde el sistema de destino	19-12
19.3.1	Cargar un equipo en la PG	19-13
19.3.2	Cargar bloques desde una CPU S7	19-14
19.3.3	Editar los bloques cargados en la PG/en el PC	19-15
19.3.3.1	Editar bloques cargados si la PGo el PC contiene el programa de usuario.....	19-16
19.3.3.2	Editar bloques cargados si la PGo el PC no contiene el programa de usuario	19-16
19.4	Borrar en el sistema de destino	19-17
19.4.1	Borrar las memorias de carga/de trabajo y borrado total de la CPU.....	19-17
19.4.2	Borrar bloques S7 en el sistema de destino.....	19-18
19.5	Comprimir la memoria de usuario (RAM)	19-19
19.5.1	Creación de huecos en la memoria de usuario (RAM).....	19-19
19.5.2	Comprimir el contenido de la memoria de una CPU S7.....	19-20

20	Test	20-1
20.1	Introducción al test con tablas de variables.....	20-1
20.2	Procedimiento básico para observar y forzar con tablas de variables.....	20-2
20.3	Editar y guardar tablas de variables.....	20-3
20.3.1	Crear y abrir una tabla de variables.....	20-3
20.3.2	Copiar/mover tablas de variables.....	20-3
20.3.3	Guardar una tabla de variables.....	20-4
20.4	Introducir variables en tablas de variables.....	20-4
20.4.1	Insertar operandos o símbolos en una tabla de variables.....	20-4
20.4.2	Insertar un área de operandos conexos en una tabla de variables.....	20-7
20.4.3	Insertar valores de forzado.....	20-7
20.4.4	Valores máximos para introducir temporizadores.....	20-8
20.4.5	Valores máximos para introducir contadores.....	20-9
20.4.6	Introducir líneas de comentario.....	20-9
20.4.7	Ejemplos.....	20-10
20.4.7.1	Ejemplo de introducción de operandos en tablas de variables.....	20-10
20.4.7.2	Ejemplo de introducción de un área de operandos conexos.....	20-11
20.4.7.3	Ejemplos de introducción de valores de forzado normal y de forzado permanente.....	20-12
20.5	Establecer un enlace con la CPU.....	20-14
20.6	Observar variables.....	20-15
20.6.1	Introducción a la observación de variables.....	20-15
20.6.2	Ajustar el disparo para observar variables.....	20-15
20.7	Forzar variables.....	20-17
20.7.1	Introducción al forzado de variables.....	20-17
20.7.2	Ajustar las condiciones de disparo para forzar variables.....	20-18
20.8	Forzado permanente de variables.....	20-20
20.8.1	Reglas de seguridad para el forzado permanente de variables.....	20-20
20.8.2	Introducción al forzado permanente de variables.....	20-21
20.8.3	Diferencias entre el forzado normal y el forzado permanente.....	20-23
21	Test con el estado del programa	21-1
21.1	Visualización del estado de programas.....	21-3
21.2	Información importante para comprobar en modo Etapa individual / Puntos de parada.....	21-5
21.3	Información importante sobre el estado operativo PARADA.....	21-7
21.4	Observar el estado de bloques de datos.....	21-8
21.5	Definir la visualización del estado del programa.....	21-9
21.6	Definir el modo de funcionamiento para el test.....	21-10
22	Test con el programa de simulación S7-PLCSIM (software opcional)	22-1
22.1	Test con el programa de simulación (software opcional).....	22-1
23	Diagnóstico	23-1
23.1	Diagnóstico del hardware y búsqueda de errores.....	23-1
23.2	Símbolos de diagnóstico en la vista online.....	23-3
23.3	Diagnosticar el hardware: Vista rápida.....	23-5
23.3.1	Acceder a la vista rápida.....	23-5
23.3.2	Funciones de información de la vista rápida.....	23-5
23.4	Diagnosticar el hardware: Vista del diagnóstico.....	23-6
23.4.1	Acceder a la vista de diagnóstico de HW Config.....	23-6
23.4.2	Funciones de información de la vista de diagnóstico.....	23-8
23.5	Información del módulo.....	23-9
23.5.1	Métodos para acceder a la información del módulo.....	23-9
23.5.2	Funciones de la información del módulo.....	23-10
23.5.3	Volumen de información del módulo en función del tipo de módulo.....	23-12
23.5.4	Visualizar la información de módulo de aparatos de campo PA y esclavos DP a través de un enlace Y.....	23-14

23.6	Diagnóstico en el estado operativo STOP.....	23-16
23.6.1	Procedimiento básico para averiguar la causa de un STOP.....	23-16
23.6.2	Contenido de las pilas en estado operativo STOP.....	23-16
23.7	Control de los tiempos de ciclo para evitar errores de tiempo.....	23-18
23.8	Transmisión de informaciones de diagnóstico.....	23-19
23.8.1	Lista de estado del sistema (SZL).....	23-20
23.8.2	Enviar avisos de diagnóstico personalizados.....	23-23
23.8.3	Funciones de diagnóstico.....	23-24
23.9	Medidas en el programa para tratar fallos.....	23-25
23.9.1	Evaluar el parámetro de salida RET_VAL.....	23-26
23.9.2	OBs de error para reaccionar a errores detectados.....	23-27
23.9.3	Insertar valores de sustitución al detectar errores.....	23-32
23.9.4	Error de redundancia en periferia (OB 70).....	23-34
23.9.5	Error de redundancia en CPU (OB 72).....	23-35
23.9.6	Error de tiempo (OB 80).....	23-36
23.9.7	Fallo de alimentación (OB 81).....	23-37
23.9.8	Alarma de diagnóstico (OB 82).....	23-38
23.9.9	Alarma de presencia de módulo (OB 83).....	23-39
23.9.10	Fallo de CPU (OB 84).....	23-40
23.9.11	Error de ejecución del programa (OB 85).....	23-41
23.9.12	Fallo del bastidor (OB 86).....	23-42
23.9.13	Error de comunicación (OB 87).....	23-43
23.9.14	Error de programación (OB 121).....	23-43
23.9.15	Error de acceso a la periferia (OB 122).....	23-44
24	Imprimir y archivar	24-1
24.1	Imprimir la documentación de un proyecto.....	24-1
24.1.1	Procedimiento básico para imprimir.....	24-2
24.1.2	Funciones para imprimir.....	24-2
24.1.3	Particularidades al imprimir el árbol de objetos.....	24-3
24.2	Archivar proyectos y librerías.....	24-4
24.2.1	Utilización de las funciones 'Guardar / Archivar'.....	24-5
24.2.2	Requisitos para poder archivar.....	24-6
24.2.3	Procedimiento para archivar y desarchivar.....	24-6
25	Trabajar con sistemas de automatización M7	25-1
25.1	Procedimiento para sistemas M7.....	25-1
25.2	Software opcional para la programación M7.....	25-3
25.3	Sistemas operativos para M7-300/400.....	25-5
26	Consejos y trucos	26-1
26.1	Intercambiar módulos en la tabla de configuración.....	26-1
26.2	Proyectos con un gran número de equipos en red.....	26-1
26.3	Reorganización.....	26-2
26.4	Editar símbolos en varios segmentos.....	26-2
26.5	Test con la tabla de variables.....	26-3
26.6	Forzar variables desde el editor de programas.....	26-4
26.7	Memoria de trabajo virtual.....	26-5

A	Anexos	A-1
A.1	Estados operativos	A-1
A.1.1	Estados operativos y cambios de estado	A-1
A.1.2	Estado operativo STOP	A-4
A.1.3	Estado operativo ARRANQUE	A-5
A.1.4	Estado operativo RUN	A-13
A.1.5	Estado operativo PARADA	A-14
A.2	Áreas de memoria de las CPUs S7	A-15
A.2.1	Subdivisión de la memoria en áreas	A-15
A.2.2	Memorias de carga y de trabajo	A-16
A.2.3	Memoria de sistema	A-18
A.2.3.1	Uso de las áreas de memoria del sistema	A-18
A.2.3.2	Imagen del proceso de las entradas y salidas (E/S)	A-20
A.2.3.3	Pila de datos locales	A-24
A.2.3.4	Pila de interrupción (USTACK)	A-25
A.2.3.5	Pila de bloques (BSTACK)	A-26
A.2.3.6	Búfer de diagnóstico	A-27
A.2.3.7	Evaluación del búfer de diagnóstico	A-27
A.2.3.8	Áreas remanentes de la memoria en CPUs S7-300	A-29
A.2.3.9	Áreas remanentes de la memoria de las CPUs S7-400	A-30
A.2.3.10	Objetos configurables en la memoria de trabajo	A-31
A.3	Tipos de datos y parámetros	A-32
A.3.1	Introducción a los tipos de datos y de parámetros	A-32
A.3.2	Tipos de datos simples	A-33
A.3.2.1	Formato del tipo de datos INT (enteros de 16 bits)	A-34
A.3.2.2	Formato del tipo de datos DINT (enteros de 32 bits)	A-34
A.3.2.3	Formato del tipo de datos REAL (números en coma flotante)	A-35
A.3.2.4	Formato de los tipos de datos WORD y DWORD en los números decimales codificados en binario	A-39
A.3.2.5	Formato del tipo de datos S5TIME (intervalo)	A-40
A.3.3	Tipos de datos compuestos	A-41
A.3.3.1	Formato del tipo de datos DATE_AND_TIME (fecha y hora)	A-42
A.3.3.2	Uso de tipos de datos compuestos	A-43
A.3.3.3	Uso de arrays (matrices) para acceder a los datos	A-44
A.3.3.4	Uso de estructuras para acceder a los datos	A-47
A.3.3.5	Uso de tipos de datos de usuario para acceder a los datos	A-49
A.3.4	Tipos de parámetros	A-51
A.3.4.1	Formato de los tipos de parámetros BLOCK, COUNTER, TIMER	A-52
A.3.4.2	Formato del tipo de parámetro POINTER	A-52
A.3.4.3	Uso del tipo de parámetro POINTER	A-53
A.3.4.4	Bloque para modificar el puntero	A-55
A.3.4.5	Formato del tipo de parámetro ANY	A-58
A.3.4.6	Uso del tipo de parámetro ANY	A-61
A.3.4.7	Asignar tipos de datos a los datos locales de bloques lógicos	A-64
A.3.4.8	Tipos de datos admisibles al transferir parámetros	A-66
A.3.4.9	Transferencia a parámetros IN_OUT de un FB	A-71
A.4	Trabajar con proyectos de versiones anteriores	A-72
A.4.1	Convertir proyectos de la versión 1	A-72
A.4.2	Convertir proyectos de la versión 2	A-73
A.4.3	Observaciones sobre los proyectos de STEP 7 V2.1 con comunicación de datos globales (GD)	A-74
A.4.4	Esclavos DP con archivos GSD defectuosos o inexistentes	A-74

A.5	Programas de ejemplo.....	A-75
A.5.1	Proyectos y programas de ejemplo.....	A-75
A.5.2	Programa de ejemplo para un proceso de mezcla industrial.....	A-77
A.5.2.1	Definir bloques lógicos.....	A-80
A.5.2.2	Asignar nombres simbólicos.....	A-81
A.5.2.3	Crear el FB para el motor.....	A-84
A.5.2.4	Crear la FC para las válvulas.....	A-89
A.5.2.5	Crear el OB 1.....	A-91
A.5.3	Ejemplo de aplicación de las alarmas horarias.....	A-97
A.5.3.1	Estructura del programa de usuario "Alarmas horarias".....	A-97
A.5.3.2	FC 12.....	A-99
A.5.3.3	OB 10.....	A-101
A.5.3.4	OB 1 y OB 80.....	A-103
A.5.4	Ejemplo de aplicación de las alarmas de retardo.....	A-105
A.5.4.1	Estructura del programa de usuario "Alarmas de retardo".....	A-105
A.5.4.2	OB 20.....	A-107
A.5.4.3	OB 1.....	A-109
A.5.4.4	Ejemplo de enmascarar y desenmascar eventos de errores síncronos.....	A-111
A.5.4.5	Ejemplo de bloquear y habilitar eventos de alarma y de error asíncrono (SFC 39 y 40).....	A-114
A.5.4.6	Ejemplo del tratamiento retardado de eventos de alarma y de error asíncrono (SFC 41 y 42).....	A-115
A.6	Acceso a áreas de datos del proceso y de la periferia.....	A-116
A.6.1	Acceso al área de datos de proceso.....	A-116
A.6.2	Acceso al área de datos de periferia.....	A-117
A.7	Ajustar el comportamiento del sistema.....	A-120
A.7.1	Modificar el comportamiento y las propiedades de los módulos.....	A-121
A.7.2	Actualización offline del firmware (del sistema operativo) de módulos y tarjetas ..	A-123
A.7.3	Aplicación de las funciones de reloj.....	A-124
A.7.4	Uso de marcas de ciclo y temporizadores.....	A-126

Índice alfabético

Índice alfabético-1

1 Presentación del producto e instalación

1.1 Guía de orientación de STEP 7

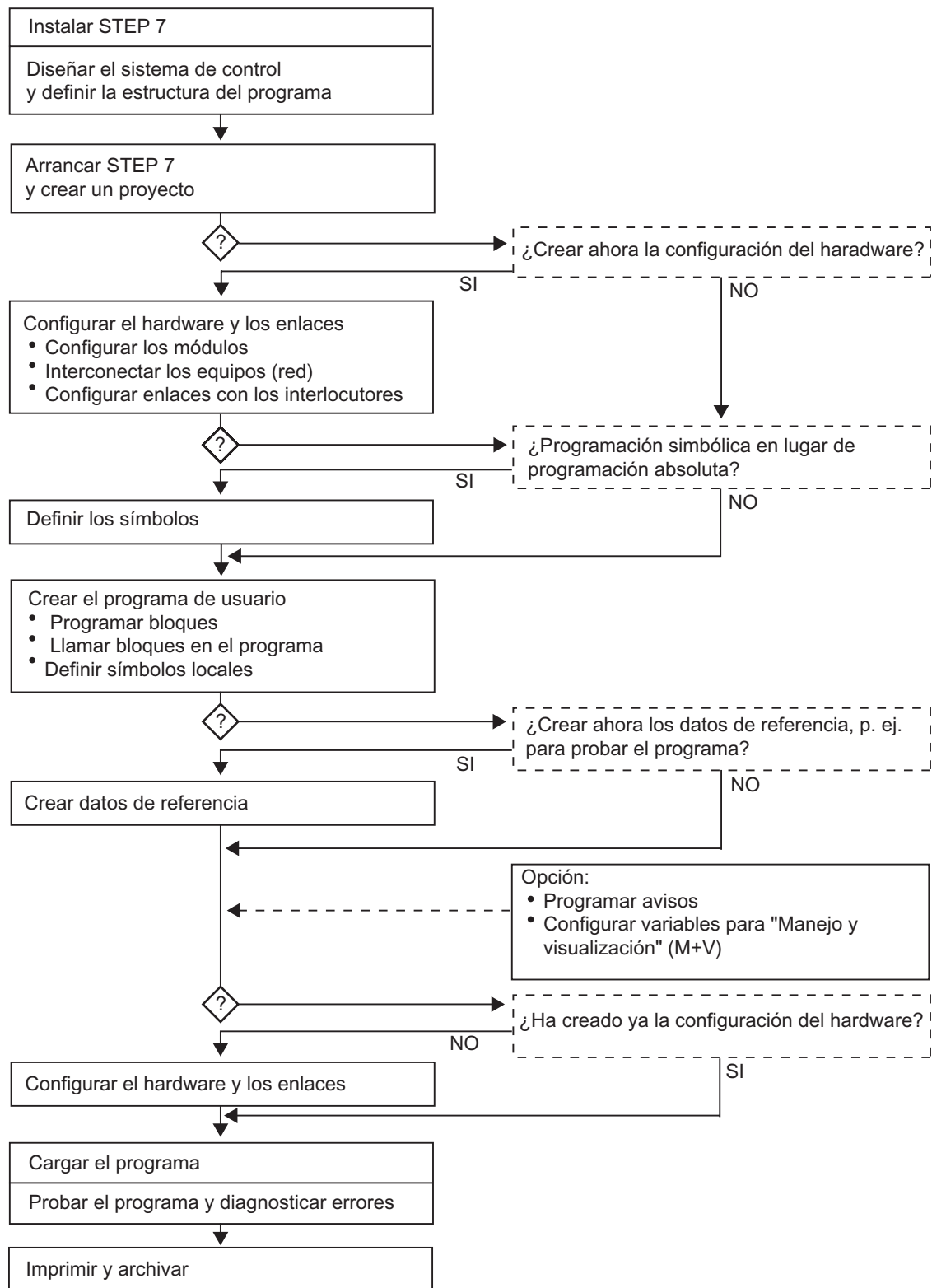
¿En qué consiste el software STEP 7?

STEP 7 es el software estándar para configurar y programar los sistemas de automatización SIMATIC. STEP 7 forma parte del software industrial SIMATIC. El software estándar STEP 7 presenta las siguientes variantes:

- STEP 7-Micro/DOS y STEP 7-Micro/WIN para aplicaciones stand-alone sencillas en sistemas de automatización SIMATIC S7-200.
- STEP 7 para aplicaciones en sistemas de automatización SIMATIC S7-300/400, SIMATIC M7-300/400 y SIMATIC C7 con funciones ampliadas::
 - ampliable con los productos de software opcionales integrados en el Software Industrial SIMATIC (consulte también Posibilidades de ampliar el software estándar STEP 7)
 - Posibilidad de parametrizar bloques de función y de comunicación
 - Forzado y modo multiprocesador
 - Comunicación de datos globales
 - Transferencia de datos controlada por eventos con bloques de comunicación y de función
 - Configuración de enlaces
 - La presente documentación contempla el software STEP 7, mientras que STEP 7-Micro se describe en la documentación "STEP 7-Micro/DOS".

Trabajos básicos

Al crear una solución de automatización con STEP 7 se deben realizar los trabajos que describiremos a continuación. La siguiente figura muestra las tareas básicas que se deben realizar en la mayoría de los proyectos, las cuales aparecen representadas en la figura en forma de organigrama, al cual nos referiremos en adelante con "Guía de orientación de STEP 7". Ésta señala los capítulos en cuestión, permitiéndole orientarse a través del manual en función de las tareas requeridas.



Procedimientos alternativos

Como muestra la figura anterior, hay dos procedimientos alternativos:

- Puede configurar primero el hardware y programar luego los bloques, o bien
- programar primero los bloques sin tener que configurar antes el hardware. Esto es especialmente recomendable cuando se deban realizar trabajos de mantenimiento, p.ej. al integrar bloques programados en un proyecto ya existente.

Resumen de los pasos a realizar:

- **Instalar STEP 7 y claves de licencia**
Al utilizar STEP 7 por primera vez, es preciso instalar el software y transferir las claves de licencia residente en el disquete al disco duro (consulte también Instalar STEP 7 y Autorización).
- **Diseñar el control**
Antes de trabajar con STEP 7, planifique su solución de automatización dividiendo primero el proceso en diversas tareas y creando luego un plano de configuración (consulte también Procedimiento básico para diseñar una solución de automatización).
- **Crear la estructura del programa**
Las tareas descritas en el diseño del control se tienen que plasmar en un programa estructurado en base a los bloques disponibles en STEP 7 (consulte también Bloques en el programa de usuario).
- **Iniciar STEP 7**
STEP 7 se arranca desde el interface de Windows (consulte también Arrancar STEP 7).
- **Crear la estructura del proyecto**
Un proyecto es una carpeta que contiene todos los datos estructurados jerárquicamente, estando disponibles en cualquier momento. Tras crear un proyecto, todos los demás trabajos se realizan en el mismo (consulte también Estructura de los proyectos).
- **Crear el equipo**
Al crear el equipo se define el sistema de automatización utilizado: p.ej. SIMATIC 300, SIMATIC 400, SIMATIC S5 (consulte también Insertar equipos).
- **Configurar el hardware**
Al configurar el hardware se define en una tabla de configuración qué módulos se utilizarán para la solución de automatización y a través de qué direcciones se accederá a los módulos desde el programa de usuario. Además, las propiedades de los módulos se pueden ajustar mediante parámetros (consulte también Pasos fundamentales para configurar el hardware).
- **Configurar redes y enlaces de comunicación**
Para poder establecer comunicaciones con otras estaciones primero hay que configurar una red. Para ello se deben crear las subredes necesarias para la red de autómatas, definir las propiedades de las subredes, parametrizar las propiedades de conexión de los equipos que la integran, así como determinar los enlaces de comunicación requeridos (consulte también Procedimiento para configurar una subred).
- **Definir los símbolos**
En lugar de utilizar direcciones absolutas es posible definir símbolos locales o globales en una tabla de símbolos, empleando nombres autoexplicativos que se utilizarán luego en el programa (consulte también Crear una tabla de símbolos)

- **Crear el programa**
El programa, que puede estar asignado o no a un módulo, se crea utilizando uno de los lenguajes de programación disponibles. Después se deposita en una carpeta en forma de bloque, fuente o esquema (consulte también Procedimiento básico para crear bloques lógicos y Nociones básicas para programar en fuentes AWL).
- **Sólo S7: Crear y evaluar los datos de referencia**
Los datos de referencia se pueden utilizar para poder comprobar y modificar más fácilmente el programa de usuario (consulte también Sinopsis de los posibles datos de referencia).
- **Configurar avisos**
Por ejemplo, se pueden crear avisos de bloques con sus textos y atributos. Utilizando el programa de transferencia, los datos de configuración de avisos que se hayan creado se transfieren al contingente de datos del sistema de manejo y visualización (p.ej. SIMATIC WinCC, SIMATIC ProTool), consulte también Configurar avisos.
- **Configurar variables M+V**
En STEP 7 se crea una sola vez una variable M+V, asignándole a la misma los atributos deseados. Utilizando el programa de transferencia, las variables M+V que se hayan creado se transfieren al contingente de datos del sistema de manejo y visualización WinCC (consulte también Configurar variables para manejo y visualización).
- **Cargar programas en el sistema de destino**
Sólo para S7: Tras concluir la configuración, la parametrización y la creación del programa, es posible cargar el programa de usuario entero o cualquiera de sus bloques en el sistema de destino (módulo programable perteneciente a la configuración de hardware; consulte también Requisitos para cargar). La CPU ya contiene el sistema operativo.

Sólo para M7: Elija el sistema operativo más adecuado para su solución de automatización y cárguelo desde el soporte de datos deseado en el sistema de destino M7, bien sea por separado o junto con el programa de usuario.
- **Comprobar los programas**
Sólo para S7: Para probar el programa puede visualizar los valores de las variables de su programa de usuario o de una CPU, asignarles valores a las mismas y crear una tabla de las variables que desea visualizar o forzar (consulte también Introducción al test con tablas de variables).
Sólo M7: Comprobar el programa de usuario utilizando un depurador con un lenguaje de alto nivel.
- **Vigilar el funcionamiento, diagnosticar el hardware**
La causa de un fallo en un módulo se determina visualizando informaciones online acerca del mismo. La causa de un fallo en la ejecución del programa de usuario se determina evaluando el búfer de diagnóstico y el contenido de las pilas. Asimismo es posible comprobar si un programa de usuario se puede ejecutar en una CPU determinada (consulte también Diagnosticar el hardware y buscar errores).
- **Documentar la instalación**
Tras crear un proyecto o una instalación es recomendable documentar claramente los datos del proyecto para facilitar las tareas de ampliación y modificación y los trabajos de mantenimiento (consulte también Imprimir la documentación de un proyecto). DOCPRO, la herramienta opcional para crear y gestionar la documentación de la instalación, permite estructurar los datos del proyecto, prepararlos en forma de manuales de circuitos normalizados e imprimirlos con un diseño homogéneo.

Incorporación de temas especiales en la guía de orientación

A la hora de crear una solución de automatización hay diversos temas especiales que pueden ser de interés:

- Modo multiprocesador - Funcionamiento síncrono de varias CPUs (consulte también Modo multiprocesador - Funcionamiento síncrono de varias CPUs)
- Creación de un proyecto por un grupo de usuarios (consulte también Elaboración de proyectos por un grupo de usuarios)
- Trabajo con sistemas M7 (consulte también Procedimiento para sistemas M7)

1.2 El software estándar STEP 7

Estándares utilizados

Los lenguajes de programación SIMATIC integrados en STEP 7 cumplen con la norma DIN EN 6.1131-3. El software estándar se ejecuta bajo los sistemas operativos MS Windows 2000 Professional (en adelante llamado Windows 2000) y MS Windows XP Professional (en adelante llamado Windows XP) y MS Windows Server 2003, estando adaptado a su funcionamiento gráfico y orientado a los objetos.

Funciones del software estándar

El software estándar le asiste en todas las fases de creación de soluciones de automatización, tales como

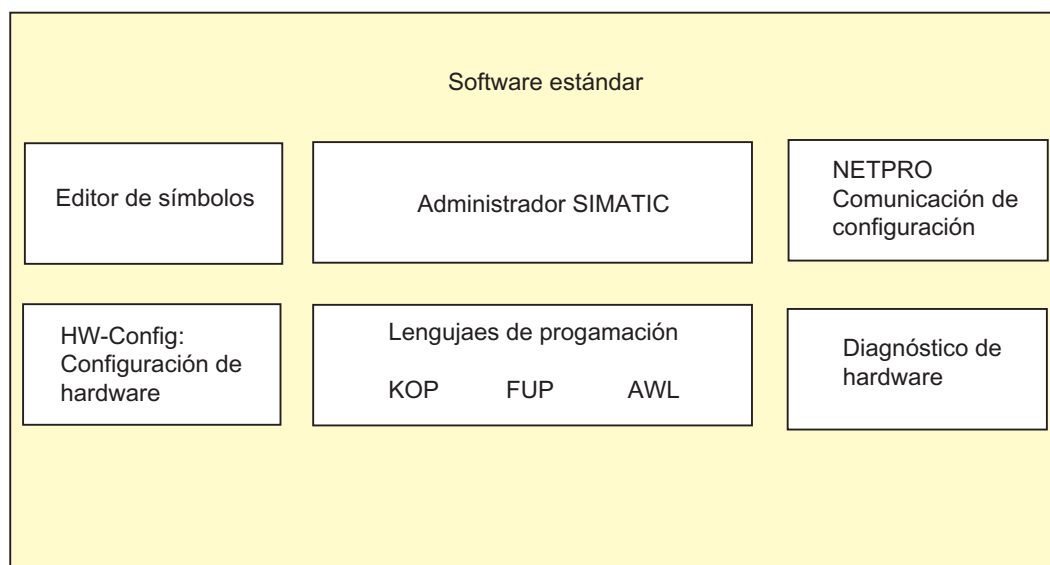
- crear y gestionar proyectos
- configurar y parametrizar el hardware y la comunicación
- gestionar símbolos
- crear programas, p.ej. para sistemas de destino S7
- cargar programas en sistemas de destino
- comprobar el sistema automatizado
- diagnosticar fallos de la instalación

El interface de usuario del software STEP 7 ha sido diseñado siguiendo los criterios ergonómicos más avanzados, lo que permite conocer rápidamente sus funciones.

La documentación del software STEP 7 contiene la información completa en la Ayuda en pantalla y en los manuales electrónicos en formato PDF.

Herramientas auxiliares

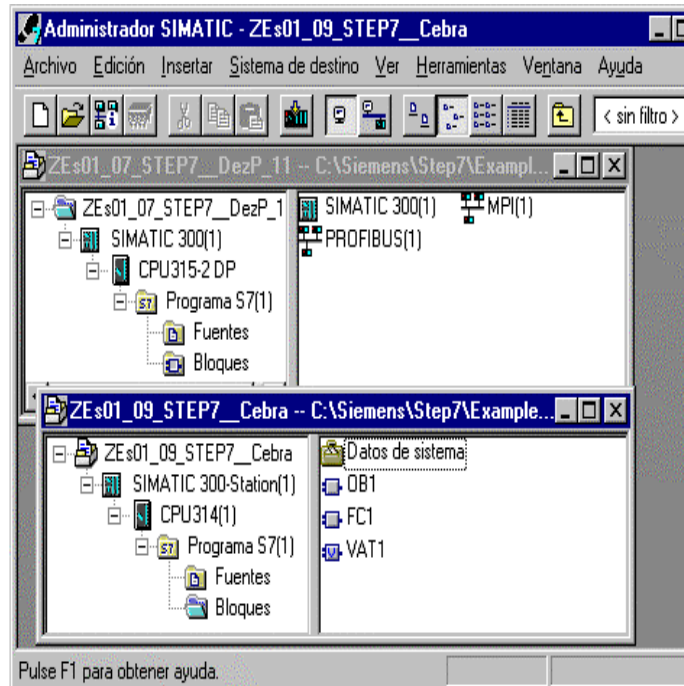
El software estándar STEP 7 ofrece toda una serie de herramientas:



Las herramientas no se deben llamar por separado, puesto que arrancan automáticamente al seleccionarse una determinada función o al abrirse un objeto.

Administrador SIMATIC

El Administrador SIMATIC gestiona todos los datos pertenecientes al proyecto de automatización, independientemente del sistema de destino (S7/M7/C7) donde se encuentren. El Administrador SIMATIC arranca automáticamente las herramientas necesarias para tratar los datos seleccionados.



Editor de símbolos

Con el editor de símbolos se gestionan todas las variables globales. Se dispone de las siguientes funciones:

- definir nombres simbólicos y comentarios para las señales del proceso (entradas y salidas), las marcas y los bloques,
- funciones de ordenación,
- importación/exportación de/hacia otros programas de Windows.

Todas las herramientas pueden acceder a la tabla de símbolos creada. Por consiguiente, detectan automáticamente si se ha modificado un parámetro de un símbolo.

Diagnóstico del hardware

El diagnóstico del hardware permite visualizar el estado del sistema de automatización, mostrando una vista general en la que aparece un símbolo cuando alguno de los módulos presenta un fallo o no. Con un doble clic en el módulo averiado se visualizan información detallada sobre el error. El volumen de información disponible depende del módulo en cuestión:

- visualización de informaciones generales sobre el módulo (p.ej. número de referencia, versión, denominación) y sobre su estado (p.ej. fallo),
- visualización de los fallos del módulo (p.ej. errores de canal) de la periferia centralizada y de los esclavos DP,
- visualización de los avisos del búfer de diagnóstico.

En el caso de las CPUs se visualizan además las siguientes informaciones:

- causas de una ejecución errónea del programa de usuario,
- duración del ciclo (máximo, mínimo y último),
- características y grado de utilización de la comunicación MPI,
- datos característicos (cantidad de entradas y salidas, marcas, contadores, temporizadores y bloques posibles).

Lenguajes de programación

Los lenguajes de programación KOP, AWL y FUP para S7-300/400 son parte integrante del software estándar.

- KOP (esquema de contactos) es un lenguaje de programación gráfico. La sintaxis de las instrucciones es similar a la de un esquema de circuitos. KOP permite observar la circulación de la corriente a través de contactos, elementos complejos y bobinas.
- AWL (lista de instrucciones) es un lenguaje de programación textual orientado a la máquina. En un programa creado en AWL, las instrucciones equivalen en gran medida a los pasos con los que la CPU ejecuta el programa. Para facilitar la programación, AWL se ha ampliado con estructuras de lenguajes de alto nivel (tales como accesos estructurados a datos y parámetros de bloques).
- FUP (diagrama de funciones) es un lenguaje de programación gráfico que utiliza los cuadros del álgebra booleana para representar la lógica. Asimismo, permite representar funciones complejas (p.ej. funciones matemáticas) mediante cuadros lógicos.

Además ofrecemos otros lenguajes de programación opcionales.

HW-Config: Configuración del hardware

Esta herramienta se utiliza para configurar y parametrizar el hardware de un proyecto de automatización. Se dispone de las siguientes funciones:

- Para configurar el sistema de automatización, se eligen primero los bastidores (racks) de un catálogo electrónico y luego se asignan los módulos seleccionados a los slots de los bastidores.
- La configuración de la periferia descentralizada se efectúa del mismo modo. También se asiste la periferia canal a canal (granular).
- Al parametrizar la CPU se pueden ajustar mediante menús propiedades tales como el comportamiento en el arranque y la vigilancia del tiempo de ciclo. Se asiste el modo multiprocesador. Los datos introducidos se depositan en bloques de datos del sistema.
- Al configurar los módulos, todos los datos se pueden ajustar en cuadros de diálogo. No es preciso efectuar ajustes mediante los interruptores DIP. La parametrización de los módulos se efectúa automáticamente durante el arranque de la CPU. Por consiguiente se puede p.ej. sustituir un módulo sin necesidad de repetir la parametrización.
- La parametrización de módulos de función (FMs) y de procesadores de comunicaciones (CPs) se efectúa con la misma herramienta de configuración del hardware de forma idéntica a como se parametrizan los demás módulos. Para los FM y CP se dispone de cuadros de diálogo específicos de los módulos (que forman parte del volumen de suministro del paquete de funciones FM/CP). El sistema impide que se efectúen entradas incorrectas, ya que los cuadros de diálogo sólo ofrecen las entradas admisibles.

NetPro

Con NetPro, los datos se pueden transferir de forma cíclica y temporizada a través de MPI, permitiendo

- seleccionar las estaciones que intervienen en la comunicación e
- introducir la fuente y el destino de los datos en una tabla. La creación de todos los bloques a cargar (SDB) y su transferencia completa a todas las CPUs se efectúa de forma automática.

Además, existe la posibilidad de transferir los datos de forma controlada por eventos, pudiéndose

- definir los enlaces de comunicación,
- seleccionar los bloques de comunicación o de función de la librería de bloques integrada,
- parametrizar en el lenguaje de programación habitual los bloques de comunicación o de función seleccionados.

1.3 Novedades de la versión 5.4 de STEP 7

Se han actualizado los siguientes temas:

- SIMATIC Manager
- Configurar y diagnosticar el hardware
- Configurar redes y enlaces
- Librerías estándar
- Notificar errores de sistema

SIMATIC Manager

- A partir de STEP 7 V5.4 dispone de dos formatos para visualizar la fecha y la hora. Se puede elegir entre visualizar el formato del idioma que esté seleccionado en STEP 7 o el del estándar ISO 8601. Este ajuste se lleva a cabo en el SIMATIC Manager, en la ficha "Fecha y hora" del cuadro de diálogo "Ajustes".
- A partir de STEP 7 V5.4 puede ver la hora de los módulos como hora local de su PG/PC. Este ajuste se lleva a cabo en el SIMATIC Manager, en la ficha "Fecha y hora" del cuadro de diálogo "Ajustes".
- A partir de STEP 7 V5.4 existe la posibilidad de configurar una protección de acceso a proyectos y librerías activando una contraseña de proyecto. Sin embargo, para ello hay que tener instalado SIMATIC Logon V1.3 SP1, en adelante llamado SIMATIC Logon (véase Información importante sobre la protección de acceso).
- A partir de STEP 7 V5.4, después de crear una protección de acceso para proyectos y librerías existe la posibilidad de crear un informe de cambios que registre acciones online como p. ej. "Cargar", "Cambios del estado operativo" o "Borrado total". Sin embargo, para ello hay que tener instalado SIMATIC Logon V1.3 SP1, en adelante llamado SIMATIC Logon (véase Información importante sobre el informe de cambios).

Configurar y diagnosticar el hardware

- Se soporta el procedimiento "Information and Maintenance" para leer y escribir los datos de identificación de los módulos. Esta función también está disponible en el SIMATIC Manager (véase Identification and Maintenance (I&M)).
- También es posible escribir datos de identificación en el caso de las interfases PROFIBUS DP en funcionamiento redundante (vía "Estaciones accesibles"). La interfase (IM) debe soportar esta función.
- Los datos CAx pueden ser importados y exportados. Así es posible intercambiar datos comunes entre STEP 7 y p. ej. los sistemas de ingeniería CAD o CAE (véase Exportar e importar datos CAx)
- El firmware de interfases PROFIBUS DP (IMs) también se puede actualizar en funcionamiento redundante, siempre y cuando los IMs sean apropiados para ello. Cada uno de los IM utilizados de forma redundante es capaz de transferir automáticamente el firmware a través del bus posterior activo al IM utilizado de forma redundante.
- Para la función "Redundancia de software" también es posible copiar e insertar de forma redundante PA-Links con sus esclavos PA (véase Configuración de la redundancia de SW)
- Las aplicaciones para procesar objetos en HW Config se pueden iniciar con el comando **Edición > Abrir objeto** (véase Abrir objetos en HW Config).

- En los dispositivos PROFINET IO se puede configurar un tiempo de supervisión de respuesta (véase Configurar el tiempo de supervisión de respuesta)
- A partir de STEP 7 V5.4 es posible visualizar la hora de los módulos como hora local de su PG/PC.

Configurar redes y enlaces

- Se soporta PROFINET IO con comunicación IRT (Isochronous Realtime). De este modo también es posible configurar para PROFINET IO tiempos de ciclo cortos y equidistantes (véase Introducción: Isochronous Realtime Ethernet).
- Manejo mejorado de la inserción de dispositivos IO copiados en otro equipo. Si las direcciones IP ya están ocupadas, es posible establecer cuál será su comportamiento al insertarlos (conservar las direcciones o asignarlas nuevamente).
- Al igual que en el caso de los esclavos PROFIBUS DP ahora también es posible ajustar la supervisión de respuesta en los esclavos PROFINET IO: como propiedad de objeto del dispositivo IO en la ficha "Ciclo IO".
- Si utiliza componentes ópticos para PROFIBUS DP: en configuraciones con anillo óptico es posible especificar los OLM utilizados. El cálculo de los parámetros de bus resulta más preciso y en caso de utilizar componentes más potentes se acorta el tiempo de ciclo del bus.

Librerías estándar

- La librería estándar "Communication Blocks" se amplía con los bloques FB 67 y FB 68 para la comunicación TCP/IP abierta.
- La librería estándar "Communication Blocks" se amplía con los bloques.
- FB 20, FB 21, FB 22 y FB 23 para el acceso cíclico a los datos útiles según PNO.
- A partir de STEP 7 V5.4 se ha añadido a la librería Redundant IO (V1) específica de los módulos la nueva librería de bloques Redundant IO CGP (channel granular peripheral devices). Esta soporta la redundancia en canales de módulos individuales. Encontrará indicaciones adicionales en la Ayuda contextual de los bloques o bien en el archivo Léame de STEP 7. Asimismo, encontrará una lista actual de los módulos soportados en las FAQs bajo <http://support.automation.siemens.com/>.

Notificar errores de sistema

- A partir de la V5.4 se soporta un bloque de datos (DB 125) para PROFIBUS con el que se pueden visualizar gráficamente eventos de diagnóstico en un dispositivo HMI.

1.4 Posibilidades de ampliar el software estándar STEP 7

El software estándar se puede ampliar con productos de software opcionales agrupados en las tres categorías siguientes:

- Herramientas de ingeniería:
abarcan lenguajes de programación de alto nivel y software orientado a la tecnología.
- Software Runtime:
contiene software ya listo para el proceso de producción.
- Interfaces hombre-máquina (HMI):
constituyen el software especial para manejo y visualización.

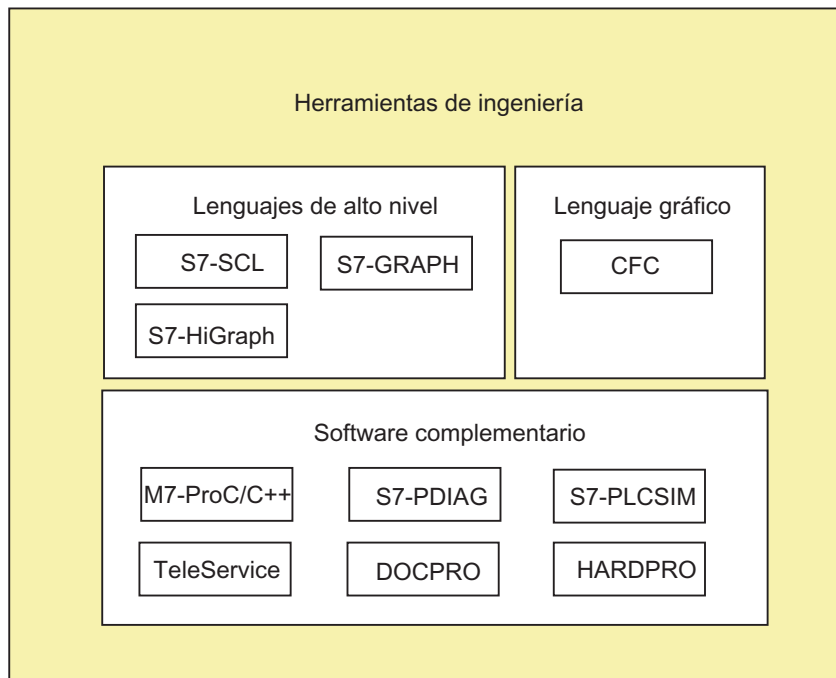
La siguiente tabla muestra el software opcional disponible en función del sistema de automatización utilizado:

	STEP 7		
	S7-300 S7-400	M7-300 M7-400	C7-620
Herramientas de ingeniería			
Borland C/C++		o	
CFC	+ ¹⁾	+	+ ²⁾
DOCPRO	+	+ ³⁾	+
HARDPRO	+		
• M7-ProC/C++		o	
• S7-GRAPH	+ ¹⁾		+ ²⁾
• S7-HiGraph	+		+
• S7-PDIAG	+		
• S7-PLCSIM	+		+
S7-SCL	+		+
• TeleService	+	+	+
Software Runtime			
• Fuzzy Control	+		+
• M7-DDE-Server		+	
• M7-SYS RT		o	
• Modular PID Control	+		+
• PC-DDE-Server	+		
• PRODAVE MPI	+		
• Standard PID Control	+		+
Interfaces hombre-máquina			
• ProAgent			
• SIMATIC ProTool			
• SIMATIC ProTool/Lite			o
• SIMATIC WinCC			
o = absolutamente necesario + = opcional ¹⁾ = recomendado a partir de S7-400 ²⁾ = no recomendado para C7-620 ³⁾ = no apto para programas C			

1.4.1 Herramientas de ingeniería

Las herramientas de ingeniería están orientadas a aquellas tareas que se pueden instalar para ampliar el software estándar. Las herramientas de ingeniería abarcan:

- Lenguajes de alto nivel para los programadores
- Lenguaje gráfico para los tecnólogos
- Software opcional para el diagnóstico, la simulación, la asistencia técnica a distancia, la documentación de la instalación, etc.



Lenguajes de alto nivel

Para programar los sistemas de automatización SIMATIC S7-300/400 se dispone de los siguientes lenguajes opcionales:

- S7-GRAPH es un lenguaje de programación que permite describir cómodamente controles secuenciales (programación de cadenas secuenciales) dividiendo el proceso en diferentes etapas. Estas últimas contienen sobre todo acciones para controlar las salidas. El paso de una etapa a otra se controla mediante condiciones de transición.
- S7-HiGraph es un lenguaje de programación que permite describir cómodamente los procesos asíncronos y no secuenciales en forma de grafos de estado. Para ello se divide la instalación en unidades funcionales que pueden adoptar diversos estados. Las unidades funcionales se pueden sincronizar mediante el intercambio de avisos.
- S7-SCL es un lenguaje textual de alto nivel según la norma DIN EN 61131-3. Contiene estructuras similares a las de los lenguajes de programación Pascal y C. Por consiguiente, S7-SCL es especialmente apropiado para los usuarios que ya estén acostumbrados a utilizar lenguajes de nivel superior. S7-SCL se puede utilizar p.ej. para programar funciones repetitivas o muy complejas.

Lenguaje gráfico

CFC para S7 y M7 es un lenguaje de programación para interconectar gráficamente las funciones existentes. Estas últimas abarcan una amplia gama que incluye desde combinaciones lógicas sencillas hasta regulaciones y controles complejos. Se dispone de un gran número de funciones en forma de bloques contenidos en una librería. La programación se lleva a cabo copiando los bloques en un esquema (plano) y conectándolos entre sí mediante líneas.

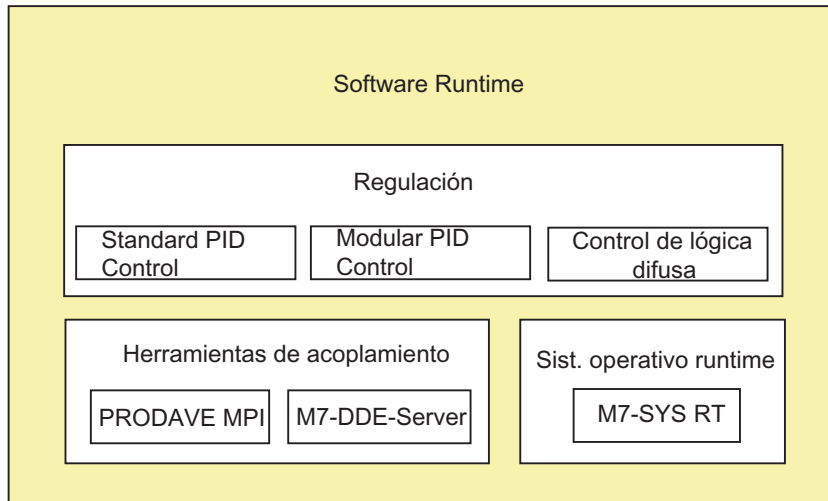
Software opcional

- Borland C++ (sólo M7) contiene el entorno de desarrollo Borland.
- Con DOCPRO se pueden organizar todos los datos de configuración que se hayan creado con STEP 7 en manuales de circuitos. Dichos manuales permiten gestionar cómodamente los datos de configuración y preparar una impresión normalizada.
- HARDPRO es el sistema de configuración de hardware para S7-300 que le facilita al usuario la configuración de tareas de automatización complejas.
- M7-ProC/C++ (sólo M7) permite incorporar el entorno de desarrollo Borland para los lenguajes de programación C y C++ en el entorno de desarrollo STEP 7.
- Con S7-PLCSIM (sólo S7) es posible simular sistemas de automatización S7 conectados al sistema de origen (PC/PG) para someterlos a un test.
- S7-PDIAG (sólo S7) permite configurar globalmente el diagnóstico de procesos para SIMATIC S7-300/400. Con dicho diagnóstico se pueden detectar estados erróneos fuera del sistema de automatización (p.ej. final de carrera no alcanzado).
- A través de la red telefónica, TeleService permite programar los sistemas de automatización S7 y M7, así como realizar trabajos de mantenimiento, utilizando para ello una PG/un PC.

1.4.2 Software Runtime

El Software Runtime abarca programas ya listos que se pueden llamar desde el programa de usuario. El Software Runtime se incorpora directamente en la solución de automatización e incluye:

- regulaciones para SIMATIC S7, p.ej. regulaciones estándar, modulares y de lógica difusa
- herramientas para acoplar los sistemas de automatización con aplicaciones de Windows
- un sistema operativo de tiempo real para SIMATIC M7



Regulaciones para SIMATIC S7

- Standard PID Control permite integrar en el programa de usuario los reguladores continuos, por impulsos y discontinuos. La herramienta de parametrización con ajuste de regulación incorporado permite parametrizar y ajustar rápidamente los reguladores.
- Modular PID Control se utiliza cuando un regulador PID sencillo no es suficiente para solucionar la tarea de automatización. Interconectando el bloque de función estándar incorporado se pueden realizar casi todas las estructuras de regulación.
- Con el Control de lógica difusa se pueden crear sistemas de lógica polivalente. Dichos sistemas son necesarios cuando resulta muy difícil o imposible describir los procesos matemáticamente, cuando éstos se ejecutan de forma impredecible o cuando se presentan no linealidades, y por otro lado se dispone de conocimientos acerca del comportamiento del proceso.

Herramientas de acoplamiento

- PRODAVE MPI es una caja de herramientas para la comunicación de datos del proceso entre SIMATIC S7, SIMATIC M7 y SIMATIC C7. PRODAVE MPI realiza independientemente la comunicación de datos a través del interface MPI.
- Con el M7-DDE-Server (**D**ynamic **D**ata **E**xchange), las aplicaciones de Windows se pueden conectar a las variables de proceso de SIMATIC M7 sin necesidad de programas adicionales.

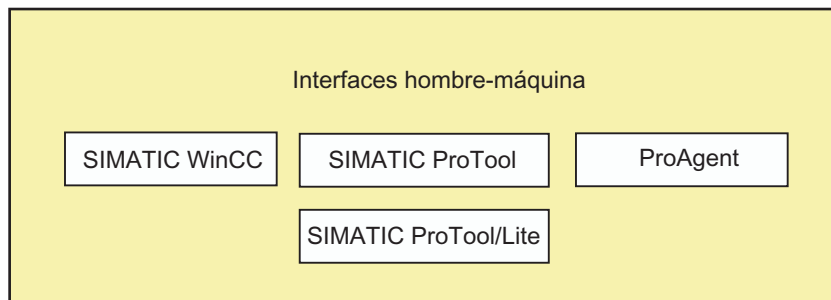
Sistema operativo runtime

- M7-SYS RT incorpora el sistema operativo M7 RMOS 32 y programas de sistema. Se requiere para poder utilizar los paquetes M7 ProC/C++ y CFC para SIMATIC M7.

1.4.3 Interfaces hombre-máquina

Human Machine Interface, el software especial de manejo y visualización para SIMATIC.

- El sistema abierto de visualización de procesos SIMATIC WinCC es un sistema básico independiente del ramo y de la tecnología, que incorpora todas las funciones importantes de control y supervisión.
- SIMATIC ProTool y SIMATIC ProTool/Lite son herramientas modernas para configurar los paneles de operador SIMATIC y los equipos compactos SIMATIC C7.
- ProAgent facilita un diagnóstico de procesos puntualizado y rápido en instalaciones y máquinas, averiguando informaciones relativas a la ubicación y a la causa del error.



2 Instalación

2.1 Automation License Manager

2.1.1 Autorización de utilización con el Automation License Manager

Automation Licence Manager

Para la utilización del software de programación STEP 7 se requiere una clave de licencia (autorización de utilización) específica para del producto, cuya instalación se ejecuta a partir de la versión 5.3 de STEP 7 con el Automation License Manager.

El Automation License Manager es un producto de software de Siemens AG. Se utiliza en todos los sistemas para el procesamiento de claves de licencia (representantes técnicos de licencias).

El Automation License Manager puede encontrarse:

- En el soporte de instalación del producto de software para el que se requiere una clave de licencia o
- En un soporte de instalación separado o
- En las páginas de Internet de A&D Customer Support de Siemens AG como WebDownload.

En el Automation License Manager se ha integrado una ayuda en pantalla, a la que puede acceder contextualmente tras la instalación pulsando la tecla F1 o a través del comando de menú **Ayuda > Ayuda sobre License Manager**. En esta ayuda encontrará la información detallada sobre la funcionalidad y la manipulación del Automation License Managers.

Licencias

Para la utilización de paquetes de software de STEP 7 protegidos con licencia se requieren licencias. Una licencia se otorga a modo de derecho a la utilización de productos. Los representantes de este derecho son:

- El CoL (**Certificate of License**) y
- La clave de licencia.

Certificate of License (CoL)

El "Certificate of License" (certificado de licencia) suministrado con el producto es la prueba jurídica el derecho de utilización. El producto correspondiente sólo puede utilizarlo el propietario del CoL o personas autorizadas.

Claves de licencia

La clave de licencia es el representante técnico de una licencia (sello de licencia electrónico).

Para cada software protegido por licencia SIEMENS AG otorga una clave de licencia. Sólo si al abrir el software se detecta una clave de licencia válida podrá utilizarse el software correspondiente de acuerdo con las condiciones de licencia y de utilización relacionadas con la licencia en cuestión.

Notas

- Puede utilizar el software básico de STEP 7 sin clave de licencia para familiarizarse con la interfaz de usuario y con sus funciones.
 - El uso ilimitado bajo la consideración de las determinaciones jurídicas de la licencia sólo es posible con una clave de licencia instalada.
 - Si **no** ha instalado la clave de licencia le aparecerá regularmente un aviso que le solicitará la instalación de la clave.
-

Las claves de licencia pueden guardarse y transferirse a soportes individuales de la siguiente manera:

- En disquetes de claves de licencia,
- En discos duros locales y
- En memorias de disco duro de ordenadores y de redes

Si se han instalado productos de software para los que no se dispone de licencia, puede conocer qué licencia necesita y después adquirirla.

Encontrará información detallada sobre la manipulación de claves de licencia en la ayuda en pantalla sobre el Automation License Manager.

Tipos de licencia

Para productos de software de Siemens AG se distingue entre los siguientes tipos de licencia orientadas a la aplicación. El comportamiento del software depende de las claves de licencia de los diferentes tipos de licencia. El tipo de utilización depende a su vez del correspondiente Certificate of License.

Tipo de licencia	Descripción
Single License	La utilización del software es ilimitada en tiempo y es posible en cualquier ordenador.
Floating License	Derecho de utilización temporalmente ilimitada sujeta a la utilización en red ("remote") de un software.
Trial License	El uso del software está limitado a: <ul style="list-style-type: none"> • Una validez máxima de 14 días, • Un número determinado de días a partir de la primera utilización, • El uso para pruebas y para validación (exclusión de responsabilidades).
Rental License	El uso del software está limitado a: <ul style="list-style-type: none"> • Una validez máxima de 50 días Un determinado número de horas en caso de utilización
Upgrade License	Para una actualización pueden ser necesarios determinados requisitos en cuanto al estado del sistema: <ul style="list-style-type: none"> • Con una licencia de actualización puede convertirse la licencia de una antigua versión x a una versión >x+.... • Una actualización puede ser necesaria p.ej. por la ampliación del equipamiento.

2.1.2 Instalar el Automation Licence Manager

El Automation License Manager se instala con un programa de instalación MSI. El software de instalación del Automation License Manager puede encontrarse en el CD de STEP 7.

Puede instalar el Automation License Manager junto con STEP 7 o hacerlo posteriormente.

Notas

- Consulte información detallada sobre el procedimiento de instalación del Automation License Manager en el archivo Léame.wri actual.
 - En la ayuda en pantalla sobre Automation License Manager encontrará toda la información necesaria sobre la funcionalidad y la manipulación de claves de licencia.
-

Instalar claves de licencia posteriormente

Si abre el software STEP 7 sin disponer de una clave de licencia, aparecerá el aviso correspondiente.

Notas

- Puede utilizar el software básico de STEP 7 sin clave de licencia para familiarizarse con la interfaz de usuario y con sus funciones.
 - El uso ilimitado bajo la consideración de las determinaciones jurídicas de la licencia sólo es posible con una clave de licencia instalada.
 - Si **no** ha instalado la clave de licencia le aparecerá regularmente un aviso que le solicitará la instalación de la clave.
-

Para instalar la clave de licencia posteriormente dispone de las siguientes posibilidades:

- Instalar las claves de licencia desde diquets
- Instalar las claves de licencia a través de WebDownload (es necesaria una solicitud previa)
- Utilizando las claves de Floating License disponibles en la red.

Encontrará información detallada sobre el procedimiento en la ayuda en pantalla sobre el Automation License Manager, a la que puede acceder tras la instalación pulsando la tecla F1 o con el comando de menú **Ayuda > Ayuda sobre el License Manager**.

Notas

- Las claves de licencia o License Keys sólo funcionarán en Windows 2000/XP/Server 2003 si se encuentran en un soporte de disco duro que permita accesos de escritura.
 - Las Floating Licenses también puede utilizarse a través de una red, es decir en modo "remote".
-

2.1.3 Reglas para la manipulación de claves de licencia



Cuidado

Tenga en cuenta las indicaciones sobre la manipulación de claves de licencia especificadas en la ayuda en pantalla referente al Automation License Manager así como en el archivo Léame.wri de STEP 7 que encontrará en el CD-ROM. Si no las observa es posible que pierda las claves de licencia de forma irrecuperable.

La ayuda en pantalla referente al Automation License Manager se puede abrir contextualmente pulsando la tecla F1 o con el comando de menú **Ayuda > Ayuda sobre el Automation License Manager**.

En esta ayuda encontrará toda la información necesaria sobre la funcionalidad y la manipulación de las claves de licencia.

2.2 Instalar STEP 7

STEP 7 contiene un programa que efectúa la instalación automáticamente. En la pantalla van apareciendo instrucciones que le conducirán paso a paso por todo el proceso de instalación. Dicho programa se inicia de la forma usual en Windows 2000/XP/Server 2003 para instalar software.

Las principales fases de la instalación son:

- copiar los datos en el sistema de origen,
- instalar los drivers para la EPROM y la comunicación,
- instalar las claves de licencia (opcional).

Nota

Las unidades de programación de SIEMENS se suministran con software STEP 7 a instalar en el disco duro.

Requisitos de instalación

- Sistema operativo
Microsoft Windows 2000, Windows XP. Windows Server 2003
- Hardware básico
PC o unidad de programación (PG) con
 - procesador Pentium (600 MHz),
 - memoria RAM: 32 MB como mínimo 256 MB
 - monitor color, teclado y ratón compatibles con Microsoft Windows.

Una unidad de programación (PG) es un ordenador personal compacto e idóneo para fines industriales. Posee el equipamiento completo para programar los sistemas de automatización SIMATIC.

- Capacidad de memoria
En el archivo "LEAME.WRI" se indica el espacio de memoria requerido en el disco duro.
- Interface MPI (opcional)
El interface MPI entre el sistema de origen (unidad de programación o PC) y el sistema de destino sólo es necesario para comunicarse en STEP 7 vía MPI con el sistema de destino.
Utilice para ello:
 - un PC-Adapter USB conectado al interface de comunicación de su unidad de programación o
 - una tarjeta MPI (p.ej. un CP 5611) incorporada en el ordenador.

Las unidades de programación incorporan el interface MPI de forma estándar.

- Prommer externo (opcional)
Si utiliza un PC, sólo necesita un prommer externo en caso de que desee grabar EPROMs.

Indicaciones

Lea las indicaciones de instalación de STEP 7 contenidas en el archivo LEAME.WRI y la "Lista de compatibilidad de los paquetes de software SIMATIC con las versiones del software estándar STEP 7".

El archivo Léame se encuentra en el menú Inicio bajo **Inicio > Simatic > Indicaciones sobre el producto**.

La lista de compatibilidad se encuentra en el menú Inicio bajo **Inicio > Simatic > Documentación**.

2.2.1 Procedimiento de instalación

Preparativos

Antes de poder comenzar con la instalación se deberá haber iniciado el sistema operativo Windows 2000, XP o Server 2003.

- No se requieren soportes de datos externos si el software STEP 7 a instalar ya se encuentra en el disco duro de la PG.
- Para instalar STEP 7 desde un CD-ROM, introdúzcalo en la unidad de CD-ROM de su PC.

Iniciar el programa de instalación

Para iniciar la instalación:

1. Introduzca el CD-ROM y arranque el programa de instalación haciendo doble clic en el archivo "setup.exe".
2. Siga paso a paso las instrucciones que el programa de instalación le muestra en la pantalla.

El programa le conduce paso a paso por el proceso de instalación, pudiéndose avanzar al paso siguiente o retroceder al paso anterior.

Durante el proceso de instalación aparecen cuadros de diálogo con consultas, o bien con opciones a elegir. Lea las observaciones que se indican a continuación para responder a los diálogos de forma más rápida y fácil.

Si ya está instalada una versión de STEP 7 ...

Si el programa de instalación detecta que ya se ha instalado una versión de STEP 7 en el sistema de origen, aparece un mensaje, pudiéndose optar por una de las siguientes alternativas:

- interrumpir la instalación (para desinstalar luego con Windows la antigua versión de STEP 7 y arrancar posteriormente la instalación) o
- proseguir con la instalación, sobrescribiendo así la antigua versión con la nueva.

Si ya existe una versión antigua, es recomendable desinstalarla antes de instalar la nueva. Si decide sobrescribir la versión antigua, puede suceder que al desinstalarla posteriormente no se borren las secciones que pudieran existir de una versión anterior.

Volumen de instalación

Para determinar el volumen de instalación, puede elegir una de las tres posibilidades siguientes:

- Normal: todos los idiomas del interface de usuario, todas las aplicaciones y todos los ejemplos. En la "Información sobre el producto" actual se indica el espacio de memoria requerido para ello.
- Mínima: sólo un idioma, sin ejemplos. En la "Información sobre el producto" actual se indica el espacio de memoria requerido.
- Personalizada: se puede elegir el volumen de instalación que necesite en lo que respecta a los programas, a la base de datos, a los ejemplos y a la comunicación.

Número de identificación

Durante la instalación se le solicita que introduzca un número de identificación. Este número se encuentra en el certificado de producto del software o en el correspondiente disquete de claves de licencia.

Autorización de claves de licencia

Durante la instalación se comprueba si en el disco duro existe la correspondiente clave de licencia. Si ésta no se detecta, aparece un mensaje indicando que el software sólo se puede utilizar con la debida clave de licencia. Si lo desea, puede transferir la clave de licencia inmediatamente o continuar con la instalación de STEP 7 y transferirla posteriormente. En el primer caso, introduzca el disquete con la clave de licencia suministrado en la correspondiente unidad cuando se le solicite.

Ajuste del interface PG/PC

Durante la instalación aparece un cuadro de diálogo que permite ajustar el interface de la PG/del PC. Consulte a este respecto el tema de la Ayuda "Ajustar interface PG/PC".

Parametrizar Memory Cards

Durante la instalación aparece un cuadro de diálogo para parametrizar Memory Cards.

- Si no va a utilizar Memory Cards, no necesita ningún driver EPROM. En este caso, elija la opción "sin driver EPROM".
- En caso contrario, elija el driver correspondiente a su PG.
- Si va a utilizar un PC, puede elegir un driver para un prommer externo. A tal efecto debe indicar adicionalmente el interface al que está conectado el prommer (p. ej., LPT1).

Los parámetros ajustados pueden modificarse después de la instalación, llamando el programa "Parametrizar Memory Card" en el grupo de programas de STEP 7 o en el Panel de control.

Flash File System

En el cuadro de diálogo para parametrizar Memory Cards puede indicar si desea instalar un Flash File System.

El Flash File System se utiliza, por ejemplo, para escribir o borrar en SIMATIC M7 algunos archivos que se encuentren en una Memory Card EPROM sin tener que modificar el resto de la misma.

Si utiliza una unidad de programación apropiada (PG 720/740/760, Field PG o Power PG) o un prommer externo adecuado y desea utilizar esta función, elija la instalación del Flash File System.

Errores durante la instalación

La instalación se interrumpe cuando se presenta alguno de los siguientes errores:

- Si inmediatamente después de arrancar el programa de instalación se presenta un error de inicialización, es muy probable que el *setup* no se haya iniciado en Windows.
- El espacio de memoria no es suficiente: dependiendo del volumen de la instalación, se requieren para el software estándar unos 100 MB de memoria en el disco duro.
- CD defectuoso: si hay un CD defectuoso, diríjase a su representante SIEMENS más próximo.
- Errores de usuario: comience de nuevo con la instalación y lea las instrucciones atentamente.

Al concluir la instalación ...

Una vez concluida la instalación, aparece el mensaje correspondiente en la pantalla.

Si durante la instalación se actualizaron archivos de sistema, será necesario reiniciar Windows. Entonces es posible arrancar (en caliente) el Administrador SIMATIC, que es el interface de STEP 7.

Una vez concluida la instalación, se creará un grupo de programas para STEP 7.

2.2.2 Ajustar el interface PG/PC

Los ajustes que se efectúen aquí determinan la comunicación entre la PG/el PC y el sistema de automatización. Durante la instalación aparece un cuadro de diálogo que permite ajustar el interface de la PG/del PC. El cuadro de diálogo también se puede visualizar después de la instalación, llamando el programa "Ajustar interface PG/PC". Así es posible cambiar los parámetros del interface a posteriori, independientemente de la instalación.

Procedimiento básico

Para poder utilizar un interface es preciso:

- realizar los ajustes necesarios en el sistema operativo y
- parametrizar el interface de forma adecuada

Si utiliza un PC con una tarjeta MPI o con procesadores de comunicaciones (CP), compruebe a través del "Panel de control" de Windows la asignación de interrupciones y de direcciones para asegurarse de que no se presenten conflictos de interrupciones y de que las áreas de direccionamiento no se solapen.

Windows 2000 y Windows XP y Windows Server 2003 ya no soportan el componente ISA MPI-ISA-Card, por lo que no se ofrecen para la instalación.

Para simplificar la parametrización del interface de la PG/del PC, es posible elegir en varios cuadros de diálogo entre diversos juegos predeterminados de parámetros básicos (parametrizaciones de interfaces).

Parametrizar el interface PG/PC

Proceda de la forma siguiente (en la Ayuda en pantalla se incluye una descripción detallada al respecto):

1. En el "Panel de control" de Windows, haga doble clic en el icono "Ajustar interface PG/PC".
2. Ajuste "S7ONLINE" en el cuadro "Punto de acceso de la aplicación".
3. En la lista "Parametrización utilizada", elija la parametrización deseada. Si no se visualiza la parametrización deseada, deberá instalar previamente un módulo o un protocolo mediante el botón "Seleccionar". La parametrización del interface se creará entonces automáticamente. En los sistemas Plug&Play, los CPs aptos para Plug&Play (CP 5611 y CP 5511) no se pueden instalar manualmente. Éstos se integran automáticamente en "Ajustar interface PG/PC" si se encuentran montados en el hardware de la PG/el PC.
 - Si elige un interface **con reconocimiento automático de los parámetros de bus** (p.ej.: CP 5611 (Auto)), puede conectar la PG o el PC a MPI o a PROFIBUS, sin necesidad de ajustar previamente los parámetros de bus. No obstante, si la velocidad de transferencia es inferior a 187,5 kbit/s, se pueden producir tiempos de espera de hasta un minuto.
Requisito para el reconocimiento automático: Hay maestros conectados al bus que distribuyen cíclicamente parámetros de bus. Este es el caso en todos los nuevos componentes MPI. En las subredes PROFIBUS, la distribución cíclica de los parámetros de bus no puede estar desconectada (ajuste estándar de red PROFIBUS).
 - Si se elige un interface **sin reconocimiento automático de los parámetros de bus**, habrá que visualizar las propiedades y adaptarlas a la subred.

Asimismo, es preciso efectuar cambios cuando se presenten conflictos con otros ajustes (p. ej., asignación de interrupciones o de direcciones). En estos casos los cambios necesarios se deberán realizar con la herramienta de detección de hardware y el "Panel de control" de Windows (v. más abajo).



Cuidado

¡Si se visualiza la parametrización "TCP/IP", **no la retire!**

Ello podría afectar a la ejecución de otras aplicaciones.

Comprobar la asignación de interrupciones y direcciones

Si utiliza un PC con tarjeta MPI, deberá comprobar si la interrupción y el área de direccionamiento predeterminadas están libres.

2.3 Desinstalar STEP 7

Para desinstalar el software utilice el procedimiento usual en Windows:

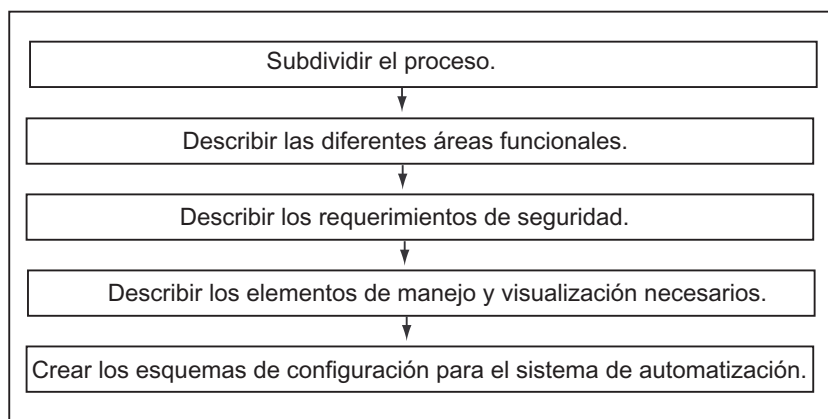
1. En el "Panel de control" de Windows, haga doble clic en el icono "Agregar o quitar programas" para abrir el cuadro de diálogo que permite instalar programas.
2. Seleccione la entrada STEP 7 en la lista de programas instalados. A continuación, haga clic en el icono de "Agregar o quitar programas".
3. Si aparecen cuadros de diálogo "Eliminar archivo compartido", haga clic en el botón "No" en caso de duda.

3 Cómo diseñar una solución de automatización

3.1 Procedimiento básico para planificar una solución de automatización

El presente capítulo contiene informaciones sobre las tareas básicas a tener en cuenta al estudiar una solución de automatización para un sistema de automatización (PLC). A la vista de un ejemplo se describe cómo proceder paso a paso para automatizar un proceso de mezcla industrial.

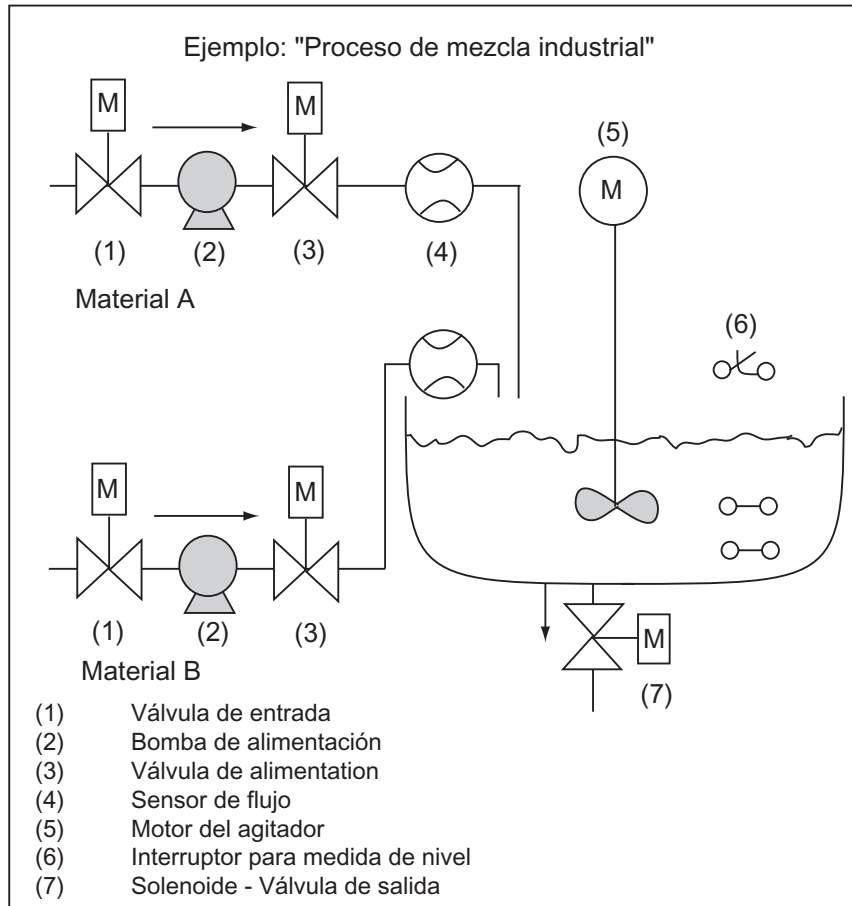
Existen muchos métodos para estudiar una solución de automatización. El procedimiento básico, que se puede utilizar para cualquier proyecto, se representa en la figura siguiente.



3.2 Subdividir el proceso en tareas y áreas

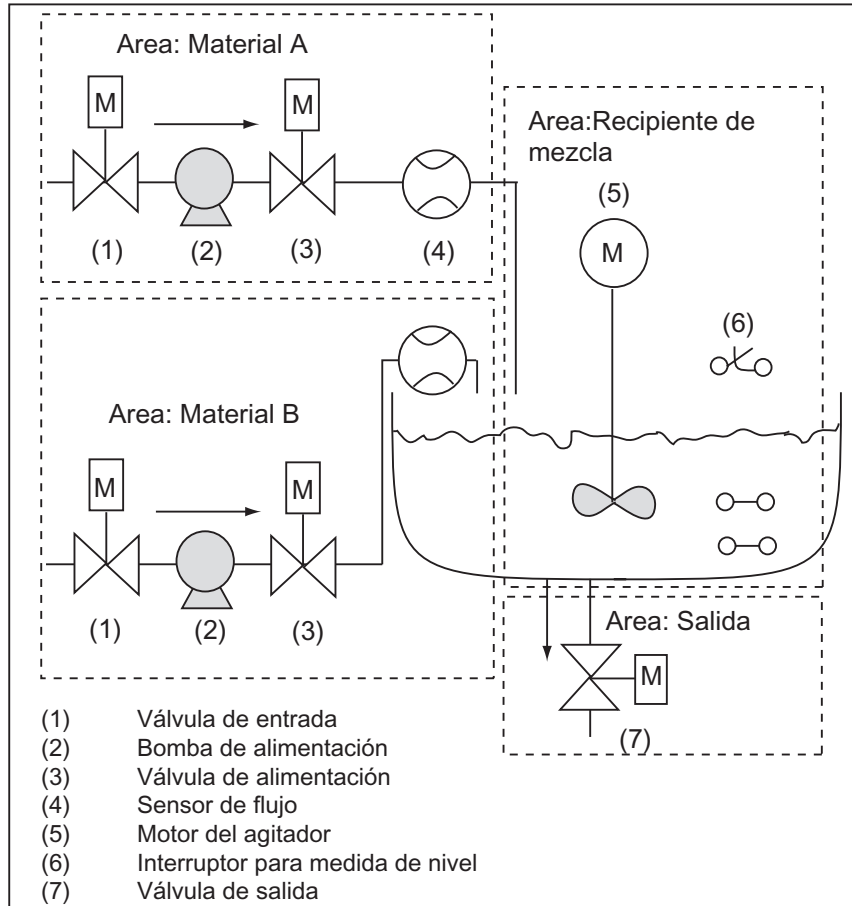
Un proceso de automatización siempre se divide en distintas tareas. Incluso el más complicado de los procesos puede ser definido, siempre y cuando se indique cómo están relacionadas las distintas tareas en las que se divide el proceso y se subdividan éstas en tareas más pequeñas.

El ejemplo siguiente ilustra, a la vista de un proceso de mezcla industrial, cómo estructurar un proceso en áreas funcionales y tareas.



Definir las áreas de un proceso

Después de haber definido el proceso a controlar, éste se ha de subdividir en grupos o áreas relacionadas entre sí:



Subdividiendo cada área en tareas más pequeñas se simplifican las tareas de control.

En el ejemplo del proceso de mezcla industrial se pueden definir cuatro áreas (v. siguiente tabla). El área para el material A contiene los mismos aparatos que el área para el material B.

Area funcional	Aparatos correspondientes
Material A	Bomba de alimentación para material A Válvula de entrada para material A Válvula de alimentación para material A Sensor de flujo para material A
Material B	Bomba de alimentación para material B Válvula de entrada para material B Válvula de alimentación para material B Sensor de flujo para material B
Recipiente de mezcla	Motor del agitador Interruptor para medida de nivel
Salida	Válvula de salida

3.3 Describir el funcionamiento en diversas áreas

Al describir cada área y tarea de un proceso, se define no solamente el funcionamiento de cada área, sino también los diferentes elementos que controlan dicha área. Estos comprenden:

- entradas y salidas eléctricas, mecánicas y lógicas de cada tarea
- enclavamientos y dependencias entre las diferentes tareas

En el ejemplo del proceso de mezcla industrial se utilizan bombas, motores y válvulas. Estos se deben definir exactamente para determinar las características operativas y el tipo de enclavamientos que se requieren durante el funcionamiento. Las tablas siguientes contienen ejemplos de descripción de los aparatos que se utilizan en un proceso de mezcla industrial. Esta descripción puede utilizarse también para adquirir los aparatos necesarios.

Material A/B: motores para bombas de alimentación
Las bombas de alimentación conducen los materiales A y B al recipiente de mezcla. <ul style="list-style-type: none"> • caudal: 400 l por minuto • potencia: 100 KW a 1200 rpm
Las bombas se controlan desde un panel de mando (Marcha/Paro), localizado en las cercanías del recipiente de mezcla. El número de puestas en marcha se cuenta para fines de mantenimiento. La puesta a 0 del contador y la cancelación del indicador de mantenimiento se realizan mediante un pulsador común.
Rigen las siguientes condiciones de habilitación: <ul style="list-style-type: none"> • El recipiente de mezcla no está lleno. • La válvula de salida del recipiente de mezcla está cerrada. • El PARO DE EMERGENCIA no está activado.
Rige la siguiente condición de desconexión: <ul style="list-style-type: none"> • El sensor de flujo señala que no hay caudal 7 segundos tras arrancar el motor de la bomba. • Mientras funciona el motor, el sensor de flujo no señala circulación de material.

Material A/B: válvulas de entrada y alimentación
Las válvulas de entrada y alimentación para los materiales A y B permiten/impiden la entrada de los materiales al recipiente de mezcla. Disponen de un solenoide con muelle antagonista. <ul style="list-style-type: none"> • Cuando el solenoide está activo, está abierta la válvula. • Cuando el solenoide está desactivado, está cerrada la válvula.
Las válvulas de entrada y alimentación son controladas por el programa de usuario.
Rige la siguiente condición de habilitación: <ul style="list-style-type: none"> • El motor de la bomba de alimentación marcha por lo menos durante 1 segundo.
Rige la siguiente condición de desconexión: <ul style="list-style-type: none"> • El sensor de flujo no señala ningún caudal.

Motor del agitador

Con el motor del agitador se mezclan los materiales A y B en el recipiente de mezcla.

- potencia: 100 KW a 1200 rpm

El motor del agitador se controla desde un panel de mando (Marcha/Paro) localizado en las cercanías del recipiente de mezcla. El número de puestas en marcha se cuenta para fines de mantenimiento. La puesta a 0 del contador y la cancelación del indicador de mantenimiento se realizan mediante un pulsador común.

Rigen las siguientes condiciones de habilitación:

- El interruptor de nivel no señala "recipiente debajo del mínimo".
- La válvula de salida del recipiente de mezcla está cerrada.
- El PARO DE EMERGENCIA no está activado.

Rige la siguiente condición de desconexión:

- El sensor de flujo no señala como muy tarde 10 s tras el arranque del motor que se ha alcanzado la velocidad nominal.

Válvula de salida

La mezcla pasa por gravedad a la próxima fase del proceso. La válvula dispone de un solenoide con muelle antagonista.

- Cuando el solenoide está activado, está abierta la válvula de salida.
- Cuando el solenoide está desactivado, está cerrada la válvula de salida.

La válvula de salida se controla desde un panel de mando (abrir/cerrar).

La válvula de salida se puede abrir bajo las siguientes condiciones:

- El motor del agitador está desconectado.
- El interruptor de nivel no señala "recipiente vacío".
- El PARO DE EMERGENCIA no está activado.

Rige la siguiente condición de desconexión:

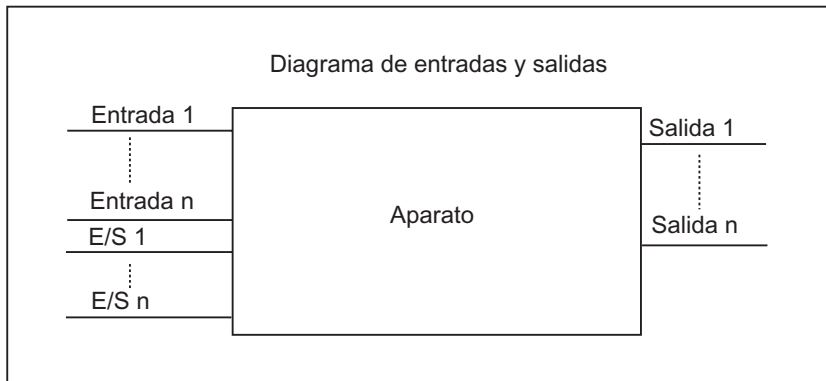
- El sistema de medida de nivel señala "Recipiente vacío".

Interruptor para medida de nivel

Los interruptores del recipiente de mezcla informan sobre el nivel de llenado del recipiente y se utilizan para el enclavamiento de las bombas de alimentación y del motor del agitador.

3.4 Listado de entradas y salidas

Después de haber definido físicamente cada uno de los aparatos a controlar, se han de dibujar diagramas de entradas y salidas para cada aparato o cada área de tareas.



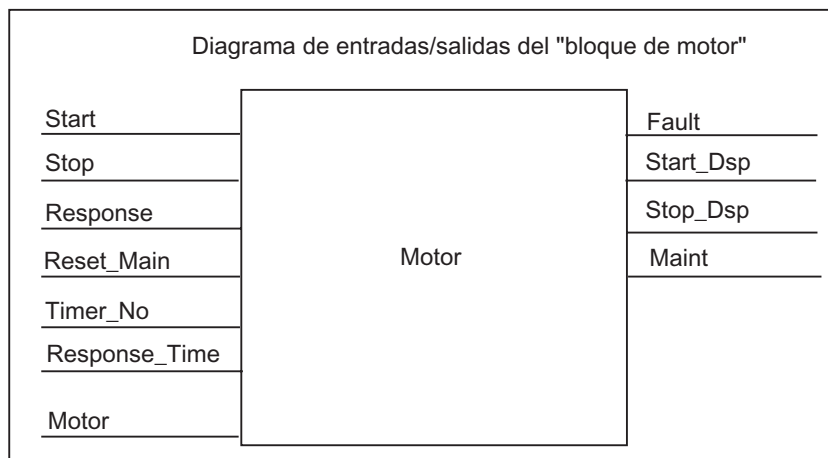
Estos diagramas equivalen a los bloques lógicos o de código que han de ser programados.

3.5 Crear un diagrama de E/S para los motores

En el ejemplo del proceso de mezcla industrial se utilizan dos bombas de alimentación y un agitador. El control de cada uno de los motores se efectúa mediante un "bloque de motor" idéntico para los tres aparatos. Este bloque necesita seis entradas: dos para marcha o paro, una para poner a 0 (Reset) el indicador de mantenimiento, una entrada para señalar el estado del motor (motor gira/detenido), una entrada para la temporización dentro de la cual debe llegar la señal de estado (respuesta) y una entrada para el número de temporizador que deberá utilizarse para medir el tiempo.

Además, el bloque lógico precisa cuatro salidas: dos para señalar el estado del motor, una para señalar errores y otra para señalar la necesidad de mantenimiento.

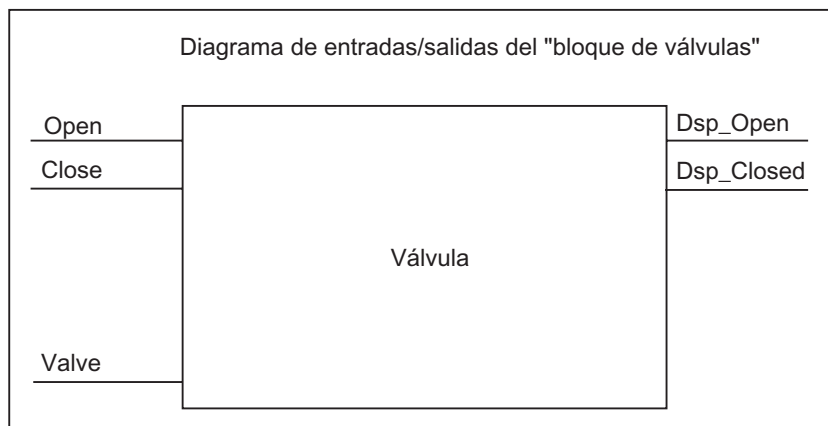
Además, se requiere una entrada/salida que sirve para controlar el motor, pero también simultáneamente para procesar o modificar el "bloque de motor" dentro del programa.



3.6 Crear un diagrama de E/S para las válvulas

Cada válvula se controla a través de un "bloque de válvulas" idéntico para todas las válvulas utilizadas. El bloque lógico dispone de dos entradas: una para abrir la válvula y otra para cerrarla. Además existen dos salidas: una para señalar que la válvula está abierta y otra para señalar que la válvula está cerrada.

El bloque dispone de una entrada/salida: sirve para controlar la válvula, pero también simultáneamente para procesar o modificar el "bloque de válvulas" dentro del programa.



3.7 Definir los requerimientos de seguridad

Determine - conforme a las prescripciones legales y a las directrices de procedimiento de la empresa - qué elementos se requieren para garantizar la seguridad del proceso. Describa también en qué medida influyen estos elementos de seguridad sobre las áreas del proceso.

Definir los requerimientos de seguridad

Definir los equipos que, por razones de seguridad, requieren circuitos fijamente cableados. Por definición, estos circuitos de seguridad trabajan independientemente del sistema de automatización (a pesar de que el circuito de seguridad ofrece normalmente un interface de entrada/salida para la coordinación con el programa de usuario). Usualmente se configura una matriz para conectar cada actuador con su propia área de PARO DE EMERGENCIA. Esta matriz constituye la base para los esquemas de los circuitos de seguridad.

Proceda de la siguiente manera al diseñar los dispositivos de protección:

- Definir los enclavamientos lógicos y mecánicos/eléctricos entre las diferentes tareas de automatización.
- Diseñar circuitos para poder manejar manualmente, en caso de emergencia, los aparatos integrantes del proceso.
- Definir otros requerimientos de seguridad para garantizar un desarrollo seguro del proceso.

Diseñar un circuito de seguridad

En el ejemplo del proceso de mezcla industrial se utiliza el siguiente circuito de seguridad:

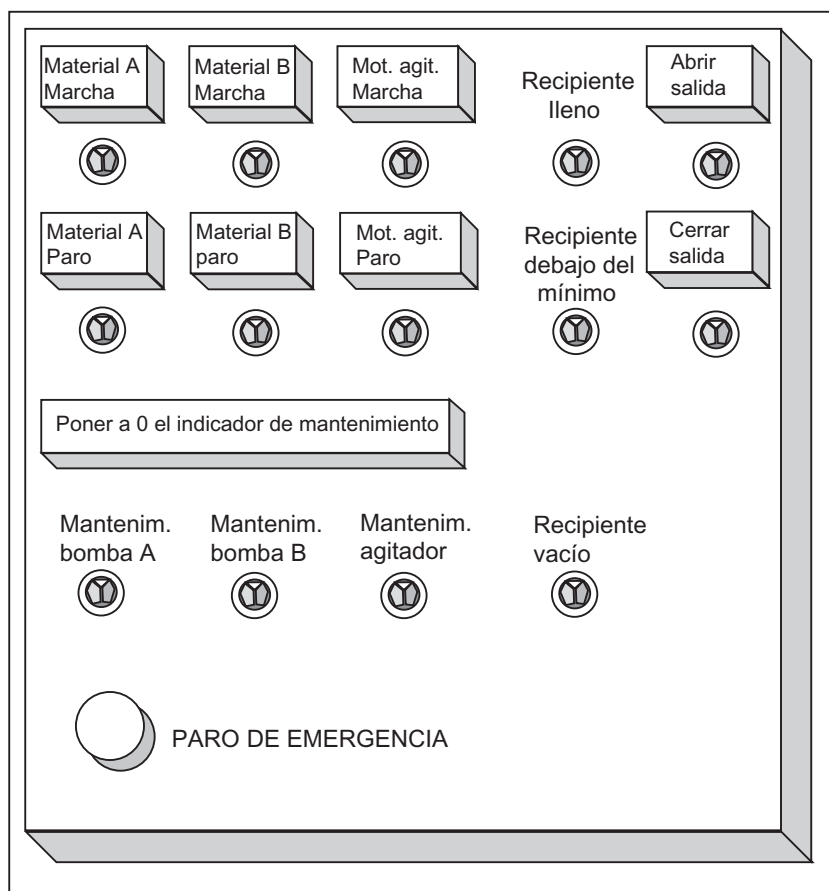
- Un pulsador de PARO DE EMERGENCIA que desconecta, independientemente del sistema de automatización (PLC), los aparatos siguientes:
 - Bomba de alimentación para material A
 - Bomba de alimentación para material B
 - Motor del agitador
 - Válvulas
- El pulsador de PARO DE EMERGENCIA está localizado en el panel de manejo.
- Una entrada del autómatas capta el estado del pulsador de PARO DE EMERGENCIA.

3.8 Describir los elementos necesarios para manejo y visualización

Cada proceso requiere un sistema de manejo y visualización que permita que las personas puedan intervenir en dicho proceso. Como parte de la descripción del proyecto se define también la estructura del panel de mando.

Definir un panel de mando

En el proceso de mezcla industrial descrito en nuestro ejemplo, cada aparato se pone en marcha o se para a través de un interruptor localizado en el panel de mando. Este panel de mando dispone de elementos de señalización que informan sobre el estado operativo (v. siguiente figura).



Incluye también las lámparas de señalización para aquellos equipos que requieren mantenimiento tras una determinada cantidad de arranques, así como el pulsador de PARO DE EMERGENCIA para parar de inmediato el proceso en caso necesario. El panel incluye también un pulsador de puesta a "0" o rearme de los indicadores de mantenimiento de los tres motores. Con él se apagan las lámparas de señalización de mantenimiento del motor afectado y se pone a 0 el contador que controla los intervalos de mantenimiento.

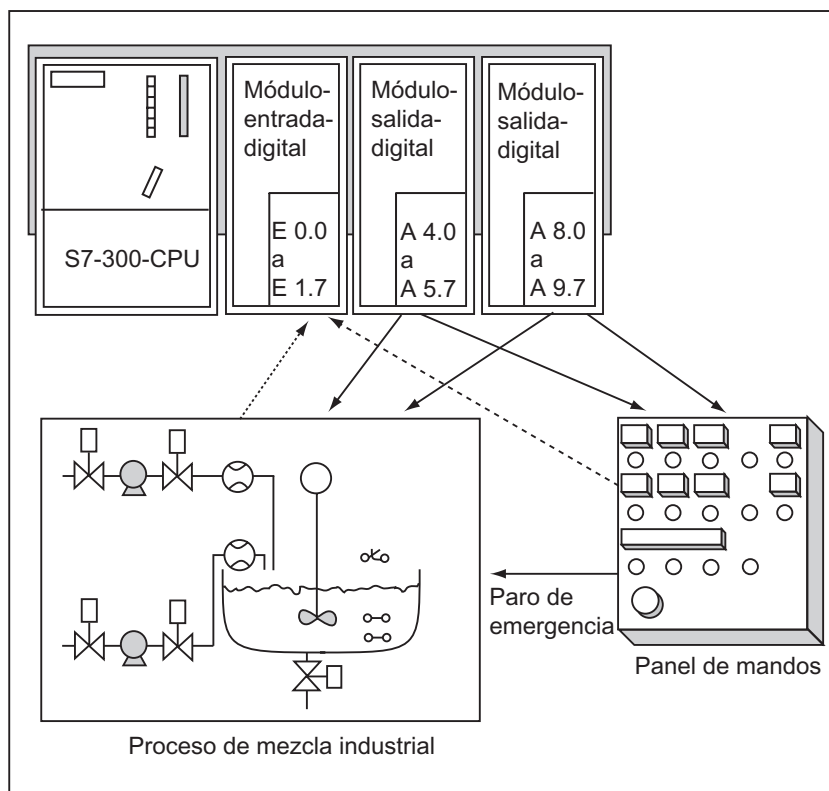
3.9 Crear un esquema de configuración

Después de haber documentado los requerimientos de diseño, se han de definir los equipos de control requeridos para este proyecto.

Al decidir qué módulos se han de utilizar, se define prácticamente la estructura del sistema de automatización. Crear un esquema de configuración considerando los puntos siguientes:

- tipo de la CPU
- cantidad y tipo de los módulos de señales
- configuración de las entradas y salidas físicas

La figura siguiente muestra un ejemplo de configuración S7 para el proceso de mezcla industrial.



4 Nociones básicas para diseñar la estructura del programa

4.1 Programas de una CPU

En una CPU se ejecutan principalmente dos programas diferentes:

- el sistema operativo y
- el programa de usuario.

Sistema operativo

El sistema operativo, que está integrado en las CPUs, organiza todas las funciones y procesos de la CPU que no están ligados a una tarea de control específica. Sus funciones son:

- gestionar el re arranque completo (en caliente) y el re arranque normal
- actualizar la imagen de proceso de las entradas y emitir la imagen de proceso de las salidas
- llamar el programa de usuario
- detectar las alarmas y llamar los OBs de tratamiento de alarmas
- detectar y tratar los errores
- administrar las áreas de memoria
- comunicar con unidades de programación y otras estaciones de comunicación

Modificando los parámetros del sistema operativo (preajustes) se puede controlar el comportamiento de la CPU en áreas determinadas.

Programa de usuario

El programa de usuario primero se ha de crear y luego se ha de cargar en la CPU. Contiene todas las funciones requeridas para procesar la tarea específica de automatización. Las tareas del programa de usuario son:

- definir las condiciones del re arranque completo (en caliente) y del re arranque normal de la CPU (p.ej. preestablecer un valor determinado para las señales)
- tratar datos del proceso (p.ej. efectuar combinaciones lógicas de señales binarias, leer y evaluar valores analógicos, definir señales binarias de salida, emitir valores analógicos)
- reaccionar a alarmas
- tratamiento de perturbaciones en el desarrollo normal del programa.

4.2 Bloques del programa de usuario

El software de programación STEP 7 permite estructurar el programa de usuario, es decir, subdividirlo en distintas partes. Esto aporta las siguientes ventajas:

- los programas de gran tamaño se pueden programar de forma clara
- se pueden estandarizar determinadas partes del programa
- se simplifica la organización del programa
- las modificaciones del programa pueden ejecutarse más fácilmente
- se simplifica el test del programa, ya que puede ejecutarse por partes
- se simplifica la puesta en marcha.

En el ejemplo del proceso de mezcla industrial ha podido apreciar cómo se puede subdividir un proceso de automatización en tareas individuales. Las partes de un programa de usuario estructurado equivalen a las distintas tareas y se definen como los bloques del programa.

Tipos de bloques

En un programa de usuario S7 se pueden utilizar diversos tipos de bloques:

Bloque	Descripción breve de la función	Consulte también
Bloques de organización (OB)	Los OBs definen la estructura del programa de usuario.	"Bloques de organización y estructura del programa"
Bloques de función del sistema (SFBs) y funciones de sistema (SFCs)	Los SFBs y SFCs están integrados en la CPU S7, permitiéndole acceder a importantes funciones del sistema.	"Bloques de función de sistema (SFB) y funciones de sistema (SFC)"
Bloques de función (FB)	Los FBs son bloques con "memoria" que puede programar el mismo usuario.	"Bloques de función (FB)"
Funciones (FC)	Las FCs contienen rutinas de programa para funciones frecuentes.	"Funciones (FC)"
Bloques de datos de instancia (DBs de instancia)	Al llamarse a un FB/SFB, los DBs de instancia se asocian al bloque. Los DBs de instancia se generan automáticamente al efectuarse la compilación.	"Bloques de datos de instancia"
Bloques de datos (DB)	Los DBs son áreas de datos para almacenar los datos de usuario. Adicionalmente a los datos asociados a un determinado bloque de función, se pueden definir también datos globales a los que pueden acceder todos los bloques.	"Bloques de datos globales (DB)"

Los OBs, FBs, SFBs, FCs y SFCs contienen partes del programa, por lo que se denominan también bloques lógicos. El número permitido de bloques de cada tipo y su longitud admisible dependen de la CPU.

4.2.1 Bloques de organización y estructura del programa

Los bloques de organización (OB) constituyen el interface entre el sistema operativo y el programa de usuario. Son llamados por el sistema operativo y controlan el procesamiento cíclico y controlado por alarmas del programa, el comportamiento de arranque del sistema de automatización y el tratamiento de los errores. Programando los bloques de organización se define el comportamiento de la CPU.

Prioridad de los bloques de organización

Los bloques de organización determinan la secuencia (eventos de arranque) en la que habrán de ejecutarse las diferentes partes del programa. La ejecución de un OB puede ser interrumpida por la llamada de otro OB. Qué OB puede interrumpir a otro OB depende de su prioridad. Los OBs de mayor prioridad pueden interrumpir a los de menor prioridad. La prioridad más baja la tiene el OB de tarea no prioritaria.

Tipos de alarma y prioridades

Los eventos de arranque que provocan la llamada de un determinado OB se denominan también alarmas. La tabla siguiente muestra los tipos de alarma en STEP 7 y la prioridad de los bloques de organización asociados. No todos los bloques de organización y sus prioridades existen en todas las CPUs S7 (consulte los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos").

Tipo de alarma	Bloque de organización	Prioridad (predeterminada)	Consulte también:
Ciclo libre	OB 1	1	"Bloque de organización para la ejecución cíclica del programa (OB 1)"
Alarmas horarias	OB 10 a OB 17	2	"Bloques de organización de alarma horaria (OB 10 a OB 17)"
Alarmas de retardo	OB 20	3	"Bloques de organización de alarma de retardo (OB 20 a OB 23)"
	OB 21	4	
	OB 22	5	
	OB 23	6	
Alarmas cíclicas	OB 30	7	"Bloques de organización de alarma cíclica (OB 30 a OB 38)"
	OB 31	8	
	OB 32	9	
	OB 33	10	
	OB 34	11	
	OB 35	12	
	OB 36	13	
	OB 37	14	
	OB 38	15	

Tipo de alarma	Bloque de organización	Prioridad (predeterminada)	Consulte también:
Alarmas de proceso	OB 40 OB 41 OB 42 OB 43 OB 44 OB 45 OB 46 OB 47	16 17 18 19 20 21 22 23	"Bloques de organización de alarma de proceso (OB 40 a OB 47)"
Alarmas DPV1	OB 55 OB 56 OB 57	2 2 2	Programación de equipos DPV1
Alarma de multiprocesamiento	OB 60 multiprocesamiento	25	Modo multiprocesador - Funcionamiento síncrono de varias CPUs
Alarma de sincronía al ciclo	OB 61 OB 62 OB 63 OB 64	25	Configurar tiempos de reacción de proceso cortos y equidistantes en PROFIBUS-DP
Error de redundancia	OB 70 Error de redundancia en periferia (sólo en sistemas H) OB 72 Error de redundancia en CPU (sólo en sistemas H)	25 28	"Bloques de organización para el tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)"
Errores asíncronos	OB 80 Error de tiempo OB 81 Fallo de alimentación OB 82 Alarma de diagnóstico OB 83 Alarma de presencia de módulo OB 84 Avería de CPU OB 85 Error de ejecución del programa OB 86 Fallo del bastidor OB 87 Error de comunicación	25 (o 28, si el OB de error asíncrono existe en el programa de arranque)	"Bloques de organización para el tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)"
Ciclo no prioritario	OB 90	29 ¹⁾	"Bloque de organización de tarea no prioritaria (OB 90)"
Arranque	OB 100 Rearranque completo (en caliente) OB 101 Rearranque normal OB 102 Arranque en frío	27 27 27	"Bloques de organización de arranque (OB 100/OB 101/OB 102)"
Errores síncronos	OB 121 Error de programación OB 122 Error de acceso a la periferia	Prioridad del OB que ha ocasionado el error	"Bloques de organización para el tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)"

1) La prioridad 29 equivale a la prioridad 0.29. Por tanto, el ciclo de tarea no prioritaria tiene menor prioridad que el ciclo libre.

Cambiar la prioridad

Las alarmas se pueden parametrizar con STEP 7. Así es posible deseleccionar, por ejemplo, los OBs de alarma o cambiar las prioridades en los bloques de parámetros: alarmas horarias, alarmas de retardo, alarmas cíclicas y alarmas de proceso.

En las CPUs S7-300, la prioridad de los bloques de organización está asignada fijamente.

En las CPUs S7-400 (y en la CPU 318) se puede modificar con STEP 7 la prioridad de los siguientes bloques de organización:

- OB 10 a OB 47
- OB 70 a OB 72 (sólo CPUs H) y OB 81 a OB 87 en estado operativo RUN.

Son admisibles:

- para OB 10 a OB 47 las prioridades 2 a 23,
- para OB 70 a OB 72 las prioridades 2 a 28, y
- para OB 81 a OB 87 las prioridades 24 a 26; en las CPUs desde aprox. mediados del 2001 (versión 3.0 del firmware) se amplió el número de prioridades posibles: para el OB 81 al OB 84 así como para el OB 86 y el OB 87 se pueden ajustar las prioridades 2 a 26.

Es posible asignar la misma prioridad a varios OBs. Los OBs de igual prioridad se ejecutan según el orden de aparición de sus eventos de arranque.

Los OBs de errores que arrancan cuando se producen errores síncronos se ejecutan con la misma prioridad que el bloque que se está ejecutando al detectarse el error.

Datos locales

Al crear los bloques lógicos (OBs, FCs, FBs) se pueden definir datos locales temporales. El área de datos locales disponible en la CPU se divide entre las diferentes prioridades.

Las CPUs S7-400 permiten modificar la cantidad de datos locales por prioridades a través de STEP 7 en el bloque de parámetros "Prioridad".

Información de arranque de un OB

Cada bloque de organización contiene una información de arranque de 20 bytes de datos locales, suministrada por el sistema operativo durante el arranque de un OB. La información de arranque informa sobre el evento de arranque del OB, la fecha y hora de arranque de OB, así como errores ocurridos y eventos de diagnóstico.

La información de arranque del OB 40 de alarma de proceso indica, por ejemplo, en la información de arranque la dirección del módulo que ha causado la alarma.

OBs de alarma desactivados

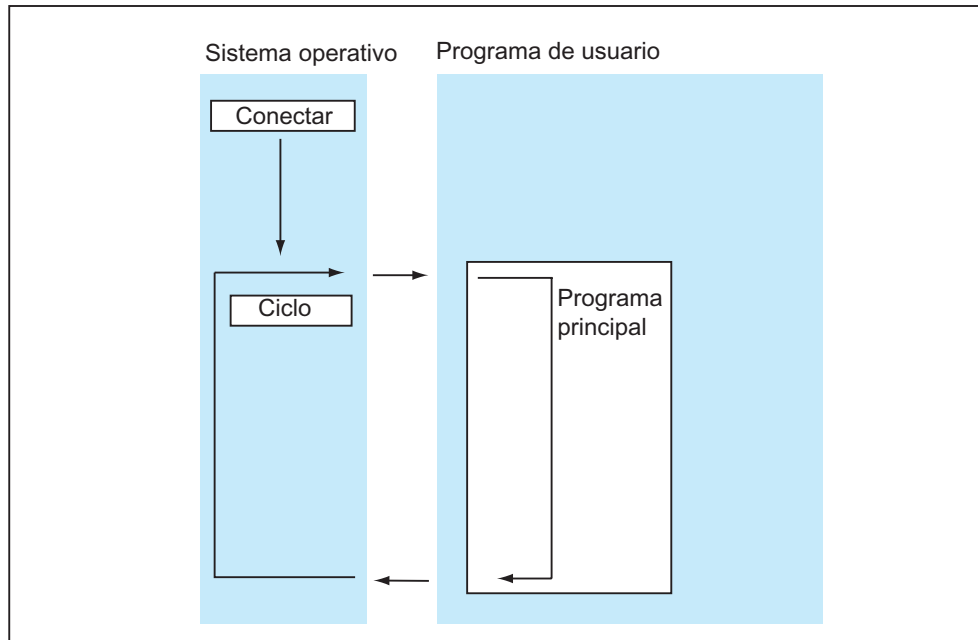
Eligiendo la prioridad 0 o asignando menos de 20 bytes de datos locales a una prioridad se desactiva el correspondiente OB de alarma. Los OBs de alarma desactivados:

- no se pueden copiar en el estado operativo RUN ni integrar en el programa de usuario.
- se pueden copiar en el estado operativo "STOP" e integrar en el programa de usuario, pero en caso de re arranque completo (en caliente de la CPU no provocan una interrupción del arranque y generan una entrada en el búfer de diagnóstico).

Desactivando los OBs de alarma no requeridos se incrementa el área disponible para datos locales que se puede utilizar para memorizar datos temporales de otras prioridades.

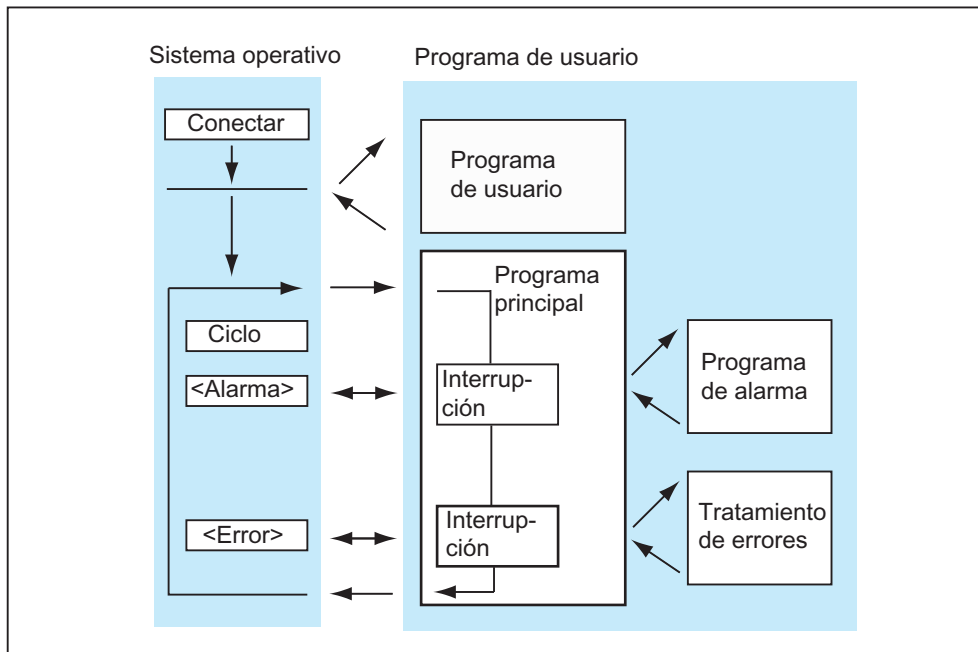
Ejecución cíclica de programas

La ejecución cíclica de programas es la ejecución "normal" en autómatas programables, es decir, el sistema operativo se ejecuta en un bucle llamado ciclo. Cada vez que se recorre un ciclo, el sistema operativo llama al bloque de organización OB 1 en el programa principal. Por consiguiente, el programa de usuario se trata cíclicamente en el OB 1.



Ejecución del programa controlada por alarmas

La ejecución cíclica del programa puede ser interrumpida por determinados eventos de arranque (alarmas). Si se presenta un evento de arranque tal, se interrumpe el bloque que está en tratamiento en el límite de una instrucción y se procesa el bloque de organización asignado al evento de arranque. Luego se continúa ejecutando el programa cíclico a partir del punto de interrupción.

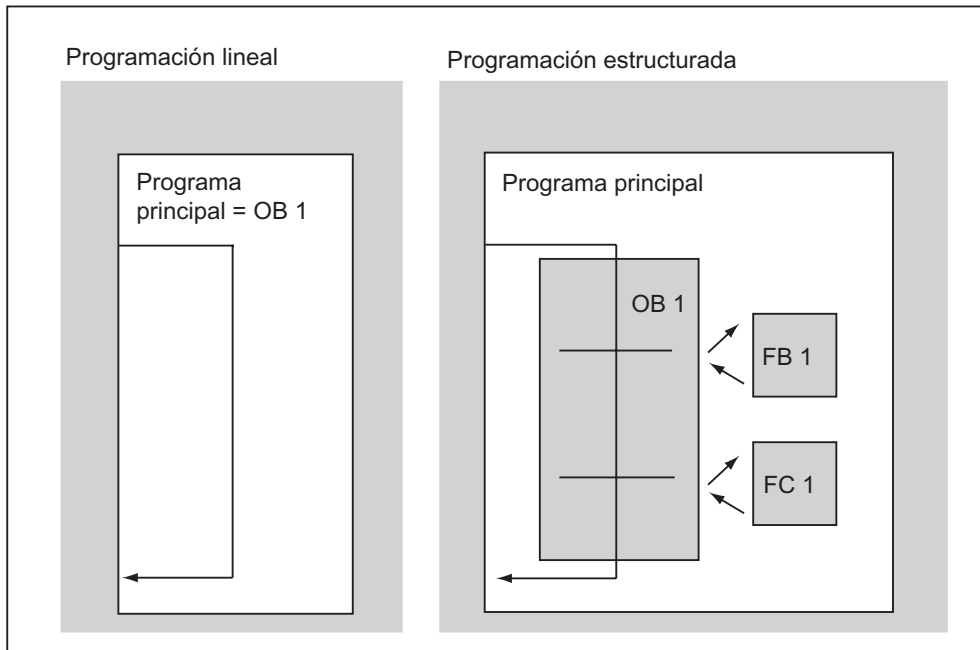


De este modo existe la posibilidad de ejecutar sólo en caso necesario aquellas partes del programa de usuario que no deben procesarse cíclicamente. El programa de usuario se puede dividir en subprogramas y repartir en diferentes bloques de organización. Si el programa de usuario debe reaccionar a una señal importante que se presente con poca frecuencia (p.ej., si el indicador de nivel de un depósito indica que se ha alcanzado el nivel de llenado), el subprograma que se deba ejecutar cuando se emita la señal se puede depositar en un OB que se ejecute de forma controlada por eventos.

Programación lineal o estructurada

El programa de usuario completo se puede escribir en el OB 1 (programación lineal). Esto se recomienda únicamente cuando los programas son simples y se ejecutan en las CPUs del S7-300 con poco volumen de memoria.

Las funciones complejas de automatización se pueden procesar mejor si se dividen en tareas más pequeñas que correspondan a las funciones tecnológicas del proceso de automatización o si se deben utilizar frecuentemente. Estas tareas parciales están representadas (programación estructurada) en el programa de usuario mediante bloques.



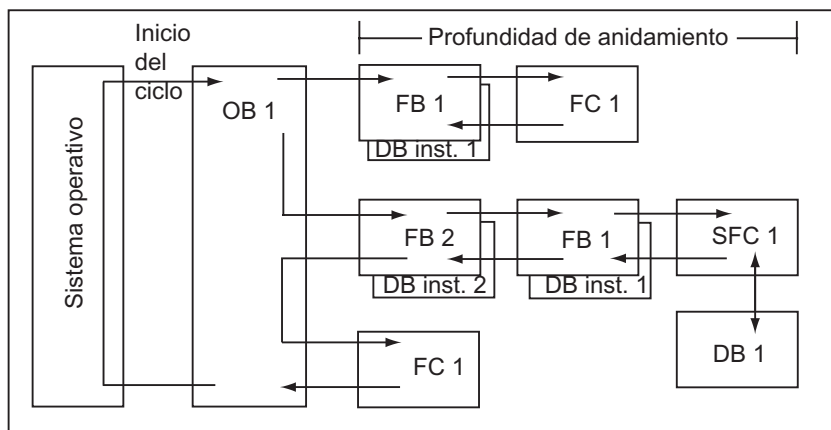
4.2.2 Jerarquía de llamada en el programa de usuario

Para que el programa de usuario pueda funcionar, los bloques que lo componen se deben poder llamar. Esto se efectúa mediante las llamadas de bloques, siendo éstas operaciones especiales de STEP 7 que sólo se pueden programar e iniciar en bloques lógicos.

Secuencia y profundidad de anidamiento

La secuencia y el anidamiento de las llamadas de bloques se denomina jerarquía de llamadas. La profundidad de anidamiento admisible depende del tipo de CPU.

La figura siguiente muestra, a la vista de un ejemplo, la secuencia y la profundidad de anidamiento de las llamadas de bloques dentro de un ciclo de ejecución.



Orden de creación de bloques:

- Los bloques se crean de arriba hacia abajo, es decir, que se comienza con la fila superior de bloques.
- Cada bloque que se llame ya deberá existir. Por tanto, en una fila de bloques, el orden de creación deberá ser de derecha a izquierda.
- El OB 1 es el último bloque que se crea.

Conforme a dichas reglas, el orden de creación de los bloques de la figura de ejemplo sería el siguiente:

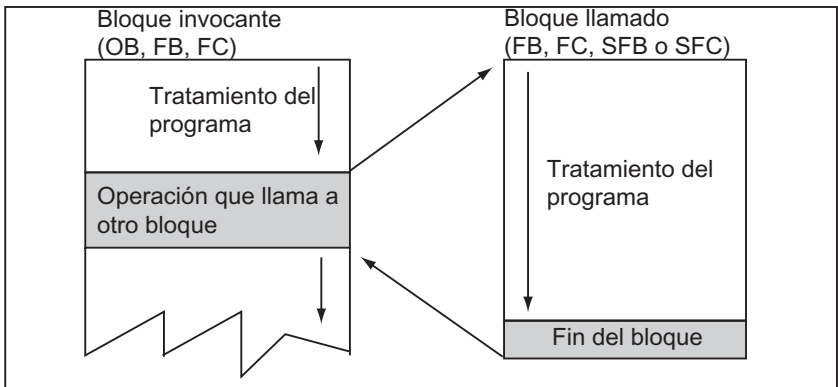
FC 1 > FB 1 + DB 1 de instancia > DB 1 > SFC 1 > FB 2 + DB 2 de instancia > OB 1

Nota

Si la profundidad de anidamiento es excesiva, la pila de datos locales puede desbordarse (consulte también Pila de datos locales).

Llamadas de bloques

La figura siguiente ilustra el desarrollo de la llamada de un bloque dentro del programa de usuario: el programa llama el segundo bloque, cuyas operaciones se ejecutan luego hasta el final. Finalizado el tratamiento del bloque llamado, se continúa el procesamiento del bloque invocante con la operación subsiguiente a la llamada del bloque.



Antes de programar un bloque, se debe definir previamente con qué datos ha de efectuarse la ejecución del programa: se han de declarar las variables del bloque.

Nota

Los parámetros OUT se deben describir en cada llamada de bloques.

Nota

El sistema operativo inicializa las instancias del SFB 3 "TP" al efectuarse un arranque en frío. Si las instancias de ese SFB se deben inicializar tras un re arranque (arranque en caliente), dichas instancias a inicializar se deberán llamar con PT = 0 ms a través del OB 100. Esto se puede llevar a cabo p.ej. mediante una rutina de inicialización en los bloques que contengan instancias de dicho SFB.

4.2.3 Tipos de bloques

4.2.3.1 Bloque de organización [0] para la ejecución cíclica del programa (OB 1)

La ejecución cíclica de programas es la ejecución "normal" en los sistemas de automatización. El sistema operativo llama cíclicamente al OB 1 y arranca la ejecución cíclica del programa de usuario.

Ejecución cíclica del programa

La tabla siguiente muestra las fases de la ejecución cíclica del programa:

Paso	Secuencia en las CPUs hasta 10/98	Secuencia en las CPUs a partir de 10/98
1	El sistema operativo inicia el tiempo de vigilancia del ciclo.	El sistema operativo inicia el tiempo de vigilancia del ciclo.
2	La CPU lee el estado de las entradas en los módulos de entradas y actualiza la imagen de proceso de las entradas.	La CPU escribe los valores de la imagen de proceso de las salidas en los módulos de salida.
3	La CPU ejecuta el programa de usuario y las operaciones indicadas en dicho programa.	La CPU lee el estado de las entradas en los módulos de entradas y actualiza la imagen de proceso de las entradas.
4	La CPU escribe los valores de la imagen de proceso de las salidas en los módulos de salida.	La CPU ejecuta el programa de usuario y las operaciones indicadas en dicho programa.
5	Al final del ciclo, el sistema operativo realiza las tareas pendientes, p.ej. cargar y borrar bloques, recibir y enviar datos globales.	Al final del ciclo, el sistema operativo realiza las tareas pendientes, p.ej. cargar y borrar bloques, recibir y enviar datos globales.
6	Finalmente, la CPU regresa al principio del ciclo y arranca nuevamente la vigilancia del tiempo de ciclo.	Finalmente, la CPU regresa al principio del ciclo y arranca nuevamente la vigilancia del tiempo de ciclo.

Imágenes del proceso

Para garantizar que la CPU disponga de una imagen coherente de las señales del proceso durante la ejecución cíclica del programa, al activarse las áreas de operandos entradas (E) y salidas (A), la CPU no accede directamente a los módulos de señales, sino a un área de memoria interna de la CPU que contiene una imagen de las entradas/salidas.

Programar ejecución cíclica del programa

La ejecución cíclica se programa escribiendo el programa de usuario con STEP 7 en el OB 1 y en los bloques allí llamados.

La ejecución cíclica del programa comienza tan pronto el programa de arranque sea finalizado sin errores.

Posibilidades de interrupción

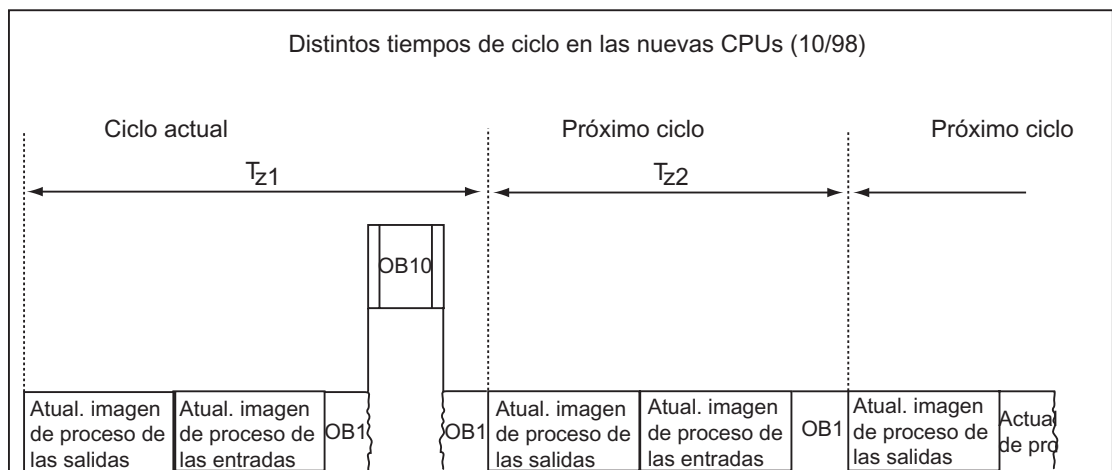
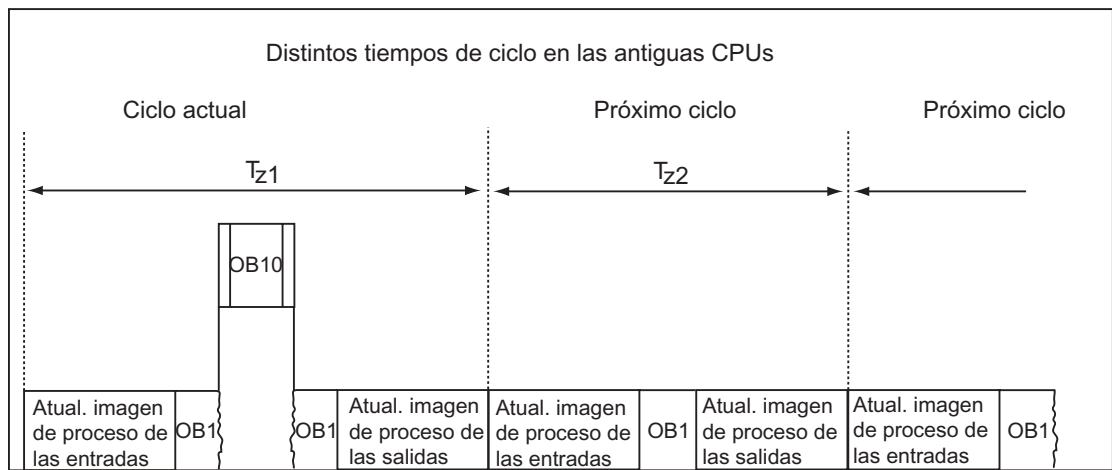
La ejecución cíclica del programa puede ser interrumpida por:

- una alarma
- una orden STOP (selector de modo de operación, comando de menú desde la PG, SFC 46 STP, SFB 20 STOP)
- un corte de tensión de red (alimentación)
- el fallo de un aparato o por un error del programa

Tiempo de ciclo

El tiempo de ciclo es el tiempo que el sistema operativo necesita para ejecutar el programa cíclico, así como todas las partes del programa que interrumpen dicho ciclo (p.ej. la ejecución de otros bloques de organización) y las actividades del sistema (p.ej. la actualización de las imágenes del proceso). Este tiempo es vigilado por el sistema.

El tiempo de ciclo (TZ) no es igual para cada ciclo. Las siguientes figuras muestran los diversos tiempos de ciclo ($TZ1 \neq TZ2$) para las CPUs hasta 10/98 y CPUs a partir de 10/98.



En el ciclo actual, el OB 1 es interrumpido por una alarma horaria.

Tiempo de vigilancia del ciclo

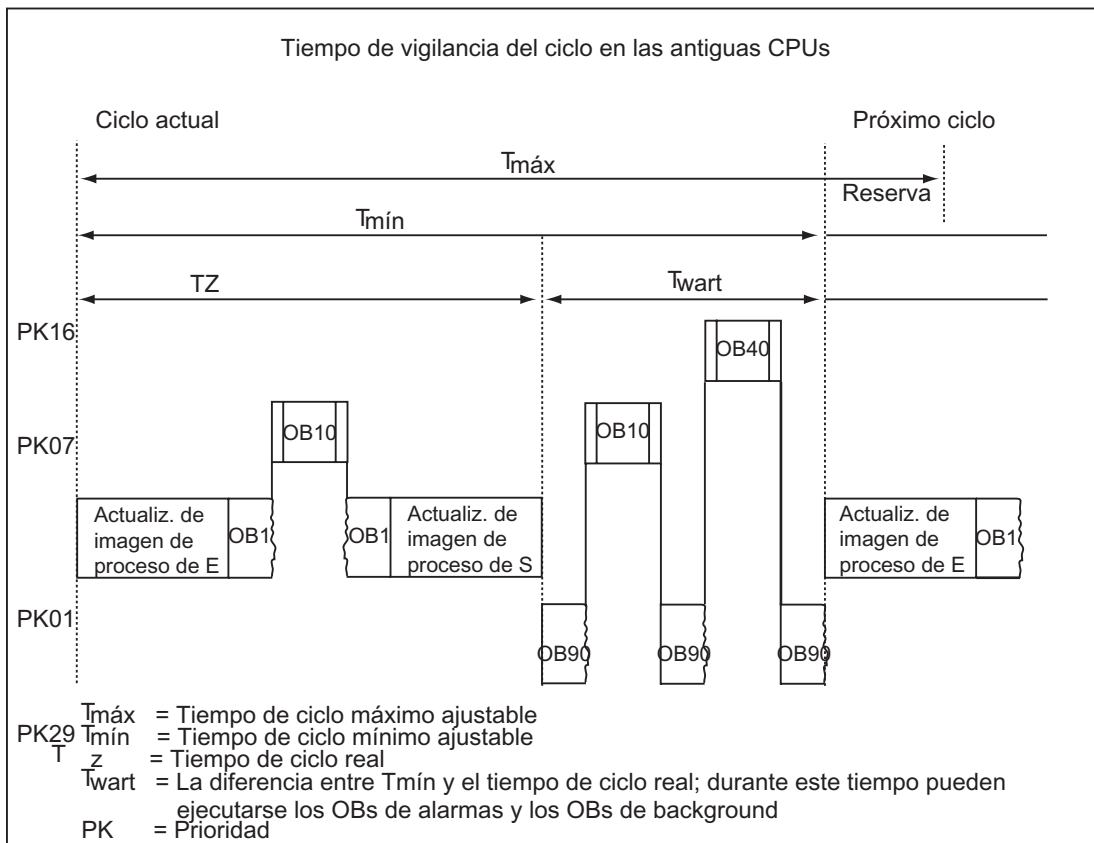
Con STEP 7 se puede modificar el tiempo de vigilancia del ciclo preajustado. Transcurrido este tiempo, la CPU pasa a STOP o se llama el OB 80, en el cual puede definirse cómo debe reaccionar la CPU al error de tiempo.

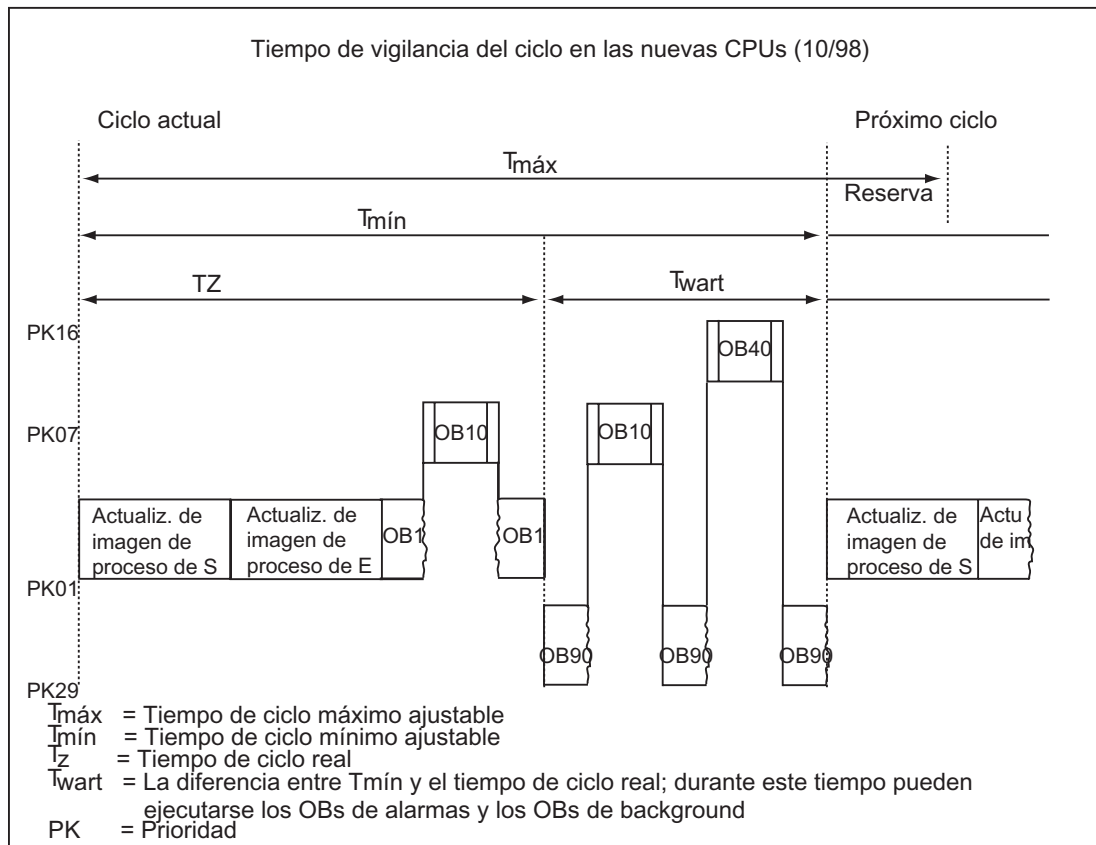
Tiempo de ciclo mínimo

Con STEP 7 se puede ajustar un tiempo de ciclo mínimo para las CPUs S7-400 y para la CPU 318. Esto se recomienda, cuando

- los intervalos entre los arranques de la ejecución del programa del OB 1 (ciclo libre) deban ser iguales o
- el tiempo de ciclo es muy corto, para evitar que las imágenes del proceso se actualicen innecesariamente con demasiada frecuencia.

Las siguientes figuras muestran la función del tiempo de vigilancia del ciclo en la ejecución del programa de las CPUs hasta 10/98 y CPUs a partir de 10/98.





Actualización de la imagen del proceso

En la ejecución cíclica del programa de la CPU la imagen del proceso se actualiza automáticamente. En las CPUs S7-400 y en la CPU 318 se puede deseleccionar la actualización de la imagen del proceso si se desea

- acceder directamente a la periferia o
- actualizar en otro momento una o varias imágenes de proceso de las entradas y las salidas a través de las funciones del sistema SFC 26 UPDAT_PI y SFC 27 UPDAT_PO.

Carga por comunicaciones

Con el parámetro de CPU "Carga del ciclo por comunicaciones" puede controlar dentro de cierto margen la duración de los procesos de comunicación, que a su vez siempre prologan el tiempo de ciclo. Pueden ser procesos de comunicación, p.ej: la transferencia de datos a otra CPU vía MPI o la carga de bloques activada mediante PG.

Este parámetro apenas influye en las funciones de test con la PG, pero pueden prolongar considerablemente el tiempo de ciclo. El tiempo disponible para funciones de test puede limitarse en el proceso (sólo S7-300).

Efecto del parámetro

El sistema operativo de la CPU pone permanentemente a disposición de la comunicación el porcentaje configurado de la capacidad total de procesamiento de la CPU (técnica de segmentación de tiempos). Si esta capacidad de procesamiento no se necesita para la comunicación, queda disponible para otras labores de procesamiento.

Efecto sobre el tiempo de ciclo real

Si no hay eventos asíncronos adicionales, el tiempo de ciclo del OB 1 se prolonga en un factor que puede calcularse con la siguiente fórmula:

$$\frac{100}{100 - \text{"Cargar del ciclo por comunicaciones (\%)\"}}$$

Ejemplo 1 (sin eventos asíncronos adicionales):

Si la carga del ciclo por comunicaciones se ajusta al 50%, el tiempo de ciclo del OB 1 puede duplicarse.

Simultáneamente, los eventos asíncronos (p.ej. alarmas de proceso o alarmas cíclicas) influyen en el tiempo de ciclo del OB1. Estadísticamente hablando, la prolongación del tiempo de ciclo debida a la parte de comunicaciones hace que se produzcan más eventos asíncronos dentro de un ciclo de OB1, lo que prolonga adicionalmente el tiempo de ciclo del OB1. Esta prolongación depende del número de eventos que se produzcan por cada ciclo del OB1 y de la duración del procesamiento del evento.

Ejemplo 2 (considerando eventos asíncronos adicionales):

Si el tiempo de ejecución del OB1 es 500 ms, una carga por comunicaciones del 50 % puede producir un tiempo de ciclo real hasta de 1000 ms (presuponiendo que la CPU tenga siempre suficientes peticiones de comunicación que procesar). Si paralelamente cada 100 ms se activa una alarma cíclica de 20 ms de tiempo de procesamiento, la repercusión sobre el ciclo sin carga por comunicaciones sería en total $5 \cdot 20 \text{ ms} = 100 \text{ ms}$; es decir, el tiempo real de ciclo sería 600 ms. Dado que una alarma cíclica también interrumpe la comunicación, con una carga por comunicaciones del 50 % la repercusión sobre el tiempo de ciclo es $10 \cdot 20 \text{ ms}$; es decir, en este caso el tiempo real de ciclo no es 1000 ms, sino 1200 ms.

Notas

- Compruebe las repercusiones de un cambio de valor del parámetro "Carga del ciclo por comunicaciones" con la instalación en funcionamiento.
 - Al ajustarse el tiempo de ciclo mínimo es preciso tener en cuenta la carga por comunicaciones, puesto que de lo contrario se producirán errores de tiempo.
-

Recomendaciones

- Si es posible, adopte el valor predeterminado
- Aumente el valor predeterminado sólo si la finalidad principal de la CPU es la comunicación y el programa de usuario no es crítico en el tiempo.
- En todos los demás casos, a lo sumo reduzca el valor predeterminado.
- Ajuste el proceso (sólo con S7-300) y restrinja el tiempo necesario en el mismo para las funciones de test.

4.2.3.2 Funciones (FC)

Las funciones son bloques programables. Una función es un bloque lógico "sin memoria". Las variables temporales de las FCs se memorizan en la pila de datos locales. Estos datos se pierden tras el tratamiento de las FCs. Para fines de memorización de datos, las funciones pueden utilizar bloques de datos globales.

Como una FC no tiene asignada ninguna memoria, se han de indicar siempre parámetros actuales. A los datos locales de una FC no se pueden asignar valores iniciales.

Campo de aplicación

La FC contiene un programa que se ejecuta cada vez que la FC es llamada por otro bloque lógico. Las funciones se pueden utilizar para

- devolver un valor de función al bloque invocante (ejemplo: funciones matemáticas)
- ejecutar una función tecnológica (ejemplo: control individual con combinación binaria).

Asignación de parámetros actuales a parámetros formales

El parámetro formal es un comodín para el parámetro real, es decir, el parámetro actual. Los parámetros actuales sustituyen a los parámetros formales al efectuar la llamada a una FC. A los parámetros formales de una FC se han de asignar siempre parámetros actuales (p.ej. al parámetro formal "Start" un parámetro actual "E3.6"). Los parámetros de entrada, de salida y de entrada/salida utilizados por la FC se depositan en forma de punteros en los parámetros actuales del bloque lógico que ha llamado a la FC.

Diferencia importante en los parámetros de salida de FC y FB

En los bloques de función (FB), al acceder a parámetros se utiliza la copia del parámetros actual del DB de instancia. Si en la llamada de un FB no se transfiere un parámetro de entrada o si en el bloque no se escribe un parámetro de salida, se continúan utilizando los valores antiguos todavía existentes en el DB de instancia (DB de instancia = memoria del FB).

Las funciones (FC) no tienen memoria. Por eso, al contrario que en el caso del FB, el hecho de proporcionar los parámetros formales no es opcional, sino necesario. El acceso a parámetros de FC se realiza a través de direcciones (indicadores de varias áreas). Si se utiliza como parámetro actual un operando del área Datos (bloque de datos) o una variable local del bloque que realiza la llamada, para la transmisión de parámetros se guarda temporalmente una copia del parámetro actual en los datos locales del bloque que realiza la llamada.

Atención

¡Si en ese caso no se escribe un parámetro OUTPUT en un FC, los valores resultantes pueden ser casuales!

El área de los datos locales del bloque que realiza la llamada dispuesta para la copia no se escribe debido a la falta de asignación al parámetro OUTPUT, por lo que permanece sin modificar. De ese modo se edita casualmente el valor existente en este área, ya que los datos locales no están ocupados automáticamente p.ej. con 0.

Por ello debe tener en cuenta los siguientes puntos:

- Si es posible inicialice parámetros OUTPUT.
- Los comandos de activar y desactivar dependen de RLO. Si con estos comandos se determina el valor de un parámetro OUTPUT, si RLO = 0 no se forma ningún valor.
- Tenga en cuenta que - independientemente de todas las rutas de programa posibles dentro del bloque - los parámetros OUTPUT se describen siempre. Debe prestar especial atención a comandos de salto y a la salida ENO en KOP y FUP. Tampoco debe olvidar BEB y el efecto de los comandos MCR.

Nota

En el caso de los parámetros OUTPUT de un FB o los parámetros INOUT de un FC y FB los valores no pueden ser casuales, ya que en este caso, sin descripción del parámetro se mantiene el valor de salida antiguo o bien el valor de entrada como valor de salida. No obstante, también en este caso debe tener en cuenta las instrucciones arriba indicadas para no procesar involuntariamente valores "antiguos".

4.2.3.3 Bloques de función (FB)

Los bloques de función son bloques programables. Un FB es un bloque "con memoria". Dispone de un bloque de datos asignado como memoria (bloque de datos de instancia). Los parámetros que se transfieren al FB, así como las variables estáticas, se memorizan en el DB de instancia. Las variables temporales se memorizan en la pila de datos locales.

Los datos memorizados en el DB de instancia no se pierden al concluir el tratamiento del FB. Los datos memorizados en la pila de datos locales se pierden al concluir el tratamiento del FB.

Nota

Con objeto de evitar errores al trabajar con FBs, lea el tema de la Ayuda Tipos de datos admisibles al transferir parámetros.

Campo de aplicación

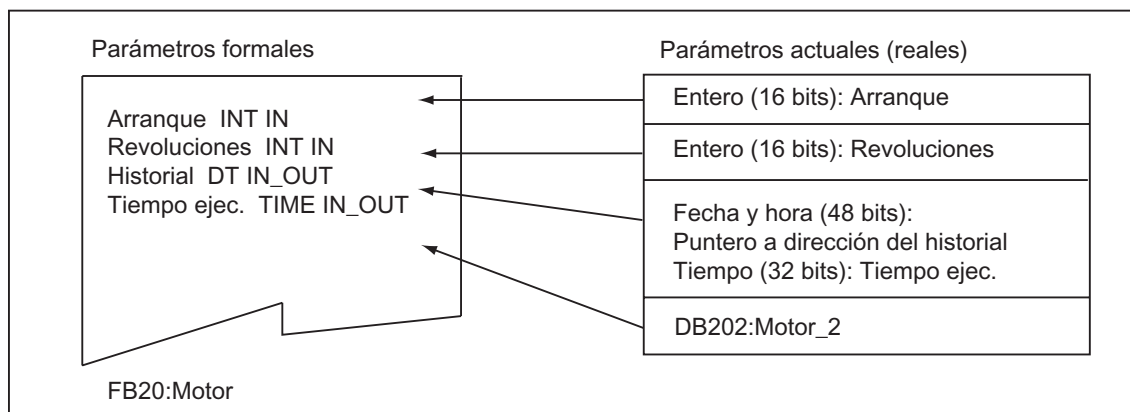
Un FB contiene un programa que se ejecuta siempre cuando el FB es llamado por otro bloque lógico. Los bloques de función simplifican la programación de funciones complejas de uso frecuente.

FBs y DBs de instancia

A cada llamada de un bloque de función que transfiere parámetros está asignado un bloque de datos de instancia.

Mediante la llamada de varias instancias de un FB es posible controlar varios equipos con un FB. Un FB para un tipo de motor puede controlar, por ejemplo, diferentes motores, utilizando datos de instancia diferentes para los diferentes motores. Los datos para cada motor (tales como número de revoluciones, rampas, tiempo de funcionamiento acumulado, etc.) se pueden memorizar en uno o varios DBs de instancia.

La figura siguiente muestra los parámetros formales de un FB que utiliza los parámetros actuales. Los parámetros formales están memorizados en el DB de instancia.



Variable del tipo de datos FB

Si el programa de usuario está estructurado de tal manera que en un FB se puedan llamar bloques de función ya existentes, los FBs a llamar se pueden incluir como variables estáticas del tipo de datos FB en la tabla de declaración de variables del FB invocante. Esto permite anidar las variables y concentrar los datos en un bloque de datos de instancia (multiinstancia).

Asignación de parámetros actuales a parámetros formales

En general, en STEP 7 no es necesario asignar parámetros actuales al parámetro formal de un FB. No obstante, hay ciertas excepciones. Los parámetros actuales se han de asignar:

- a los parámetros de entrada/salida de un tipo de datos compuestos (p.ej. STRING, ARRAY o DATE_AND_TIME)
- a todos los tipos de parámetros (p.ej. TIMER, COUNTER o POINTER)

STEP 7 asigna los parámetros actuales a los parámetros formales de un FB de la siguiente manera:

- *Si se indican parámetros actuales en la instrucción de llamada:* Las operaciones del FB utilizan los parámetros actuales suministrados.
- *Si en la instrucción de llamada no se indica ningún parámetro actual:* Las operaciones del FB utilizan los valores memorizados en el DB de instancia.

La tabla siguiente muestra qué variables del parámetro actual FB se deben asignar.

Variables		Tipo de datos	
	Tipo de datos simple	Tipo de datos compuesto	Tipo de parámetro
Entrada	Parámetro no necesario	Parámetro no necesario	Parámetro actual necesario
Salida	Parámetro no necesario	Parámetro no necesario	Parámetro actual necesario
Entrada/ salida	Parámetro no necesario	Parámetro actual necesario	–

Asignación de valores iniciales a parámetros formales

En el área de declaración del FB se pueden asignar valores iniciales a los parámetros formales. Estos datos se incluyen en el DB de instancia asignado al FB.

Si en la instrucción de llamada no se asignan parámetros actuales a los parámetros formales, entonces STEP 7 utiliza los valores memorizados en el DB de instancia. Estos datos pueden ser valores iniciales, que han sido indicados en la tabla de declaración de variables de un FB.

La tabla siguiente muestra qué variables se pueden asignar a un valor inicial. Como los datos temporales no se memorizan tras el tratamiento del bloque, entonces no se les puede asignar valores.

	Tipo de datos		
Variables	Tipo de datos simple	Tipo de datos compuesto	Tipo de parámetro
Entrada	Valor inicial admisible	Valor inicial admisible	–
Salida	Valor inicial admisible	Valor inicial admisible	–
Entrada/ salida	Valor inicial admisible	–	–
Estáticas	Valor inicial admisible	Valor inicial admisible	–
Temporales	–	–	–

4.2.3.4 Bloques de datos de instancia

A cada llamada de un bloque de función que transfiere parámetros está asignado un bloque de datos de instancia. En el DB de instancia están depositados los parámetros actuales y los datos estáticos del FB. Las variables declaradas en el FB definen la estructura del bloque de datos de instancia. La instancia define la llamada de un bloque de función. Si, por ejemplo, un bloque de función se llama cinco veces en el programa de usuario S7, existen cinco instancias de dicho bloque.

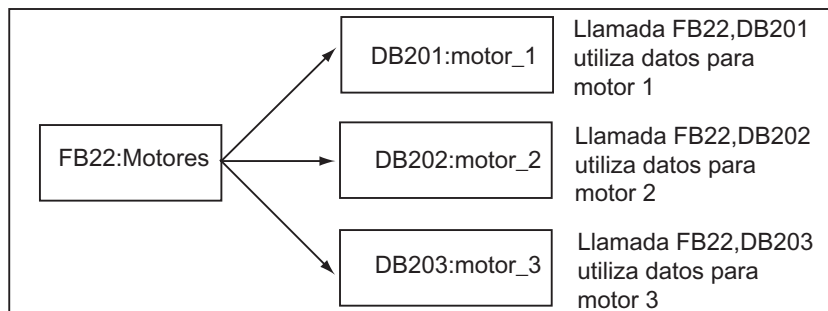
Crear un DB de instancia

Antes de crear un bloque de datos de instancia debe existir el FB asociado. El número de dicho FB se debe indicar al crear el bloque de datos de instancia.

Un DB de instancia para cada instancia

Si se asignan varios bloques de datos de instancia a un bloque de función (FB) que controla un motor, se puede utilizar este FB para controlar varios motores.

Los diversos datos de cada uno de los motores (p.ej. número de revoluciones, tiempo de aceleración, tiempo total de servicio) se memorizan en los diversos bloques de datos. Dependiendo de qué DB se asigne al FB al efectuar la llamada, se puede controlar un motor diferente. De esta manera se utiliza un solo bloque de función para varios motores (v. siguiente figura).

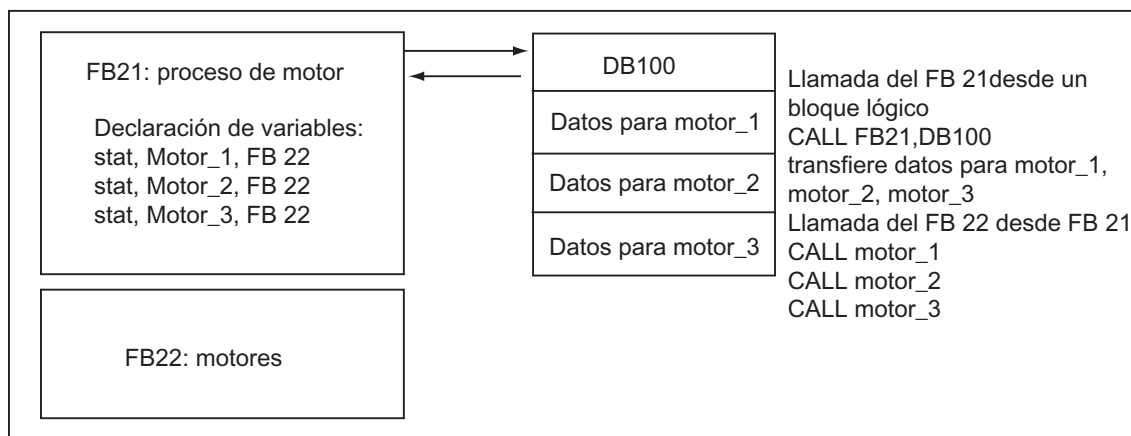


Un DB de instancia para varias instancias de un FB (multiinstancias)

A un FB se pueden transferir conjuntamente en un DB de instancia los datos de instancia para diferentes motores. A tal efecto, la llamada de los controles de motores se ha de efectuar en otro FB y en el área de declaración del FB invocante se deben declarar las variables estáticas con el tipo de datos de un FB para las diferentes instancias.

Utilizando un DB de instancia para varias instancias de un FB se ahorra capacidad de memoria y optimiza el uso de los bloques de datos.

En el caso ejemplificado por la figura siguiente, el bloque invocante es el FB 21 "Proceso de motor", las variables son del tipo de datos FB 22 y las instancias son designadas con motor_1, motor_2 y motor_3.



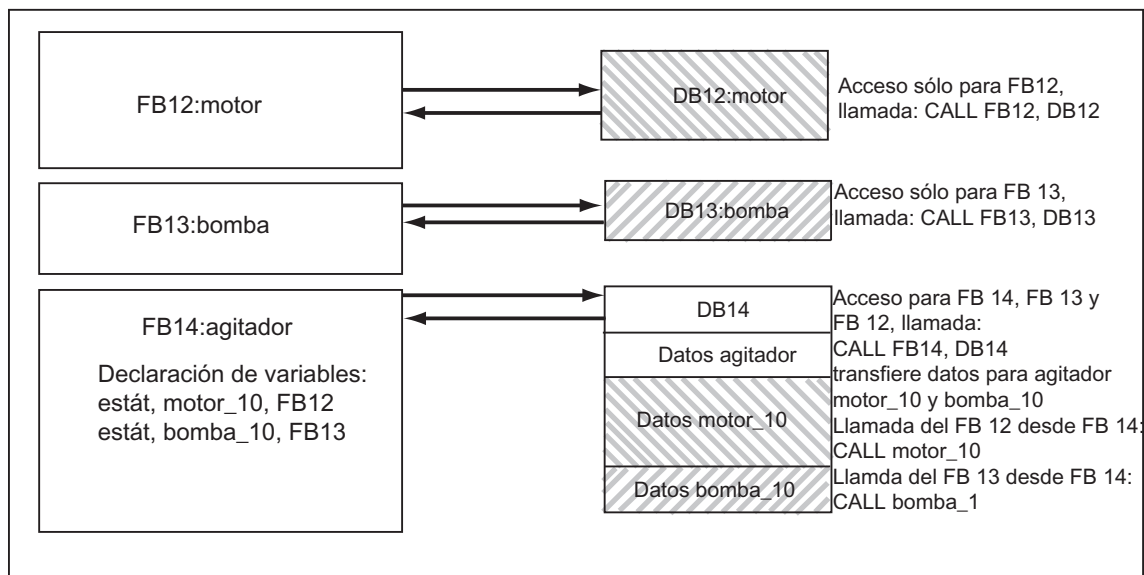
En este ejemplo, el FB 22 no necesita bloque de datos de instancia propio, ya que sus datos de instancia están memorizados en el bloque de datos de instancia del FB invocante.

Un DB de instancia para varias instancias de FB diferentes (multiinstancias)

En un bloque de función se pueden llamar a instancias de otros FBs ya existentes. Los datos de instancia necesarios al respecto se pueden asignar al bloque de datos de instancia del FB invocante, es decir que en este caso no se necesitan bloques de datos adicionales para los FBs que se han llamado.

Para dichas multiinstancias de un DB de instancia deberá declarar, en la tabla del FB invocante, variables estáticas del mismo tipo de datos del FB llamado, haciéndolo para cada una de las instancias. La llamada en el FB se efectúa entonces sólo con el nombre de la variable, es decir, sin indicar un DB de instancia.

En el ejemplo de la figura, los datos de instancia asignados se memorizan conjuntamente en un DB de instancia.



4.2.3.5 Bloques de datos globales (DB)

Al contrario de los bloques lógicos, los bloques de datos no contienen instrucciones STEP 7. En cambio, sirven para depositar datos de usuario, es decir que los bloques de datos contienen datos variables con los que trabaja el programa de usuario. Los bloques de datos globales contienen datos de usuario utilizables desde otros bloques.

El tamaño de los DBs puede variar. El tamaño máximo admisible se indica en las descripciones de la CPUs /70/ y /101/.

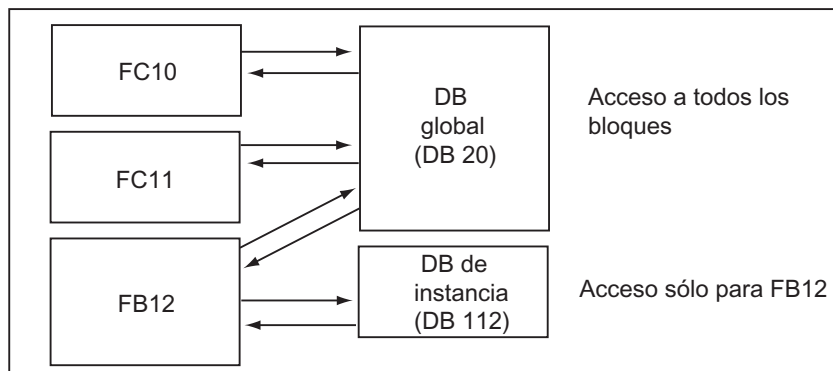
La estructura de bloques de datos globales se puede definir discrecionalmente.

Bloques de datos globales en el programa de usuario

Si se llama un bloque lógico (FC, FB o OB), éste puede ocupar determinada capacidad de memoria en el área de datos locales (pila L). Además de este área de datos locales, un bloque lógico puede abrir un área de memoria en forma de un DB. Al contrario de los datos en el área de datos locales, los datos contenidos en un DB no son borrados al cerrar el DB o al concluir el tratamiento del correspondiente bloque lógico.

Cada FB, FC o OB puede leer los datos de un DB global o escribir datos en un DB global. Estos datos se conservan en el DB incluso al abandonar dicho DB.

Un DB global y un DB de instancia pueden estar abiertos al mismo tiempo. La figura siguiente ilustra diferentes accesos a bloques de datos.



4.2.3.6 Bloques de función de sistema (SFB) y funciones del sistema (SFC)

Bloques preprogramados

No es necesario programar cada función. Las CPUs S7 ofrecen bloques preprogramados que se pueden llamar desde el programa de usuario.

Para obtener informaciones más detalladas, consulte los temas de Ayuda de los bloques del sistema y las funciones del sistema (saltos a descripciones de lenguajes, ayudas acerca de bloques, atributos del sistema).

Bloques de funciones del sistema

Un SFB es un bloque de funciones integrado en la CPU S7. Como los SFBs forman parte del sistema operativo, no se cargan como parte integrante del programa. Al igual que los FBs, los SFBs son bloques "con memoria". Para los SFBs se han de crear también bloques de datos de instancia y cargar en la CPU como parte integrante del programa.

Las CPUs S7 ofrecen SFBs

- para la comunicación vía enlaces configurados
- para las funciones especiales integradas (p.ej. SFB 29 "HS_COUNT" en la CPU 312 IFM y en la CPU 314 IFM)

Funciones del sistema

Una función del sistema es una función preprogramada integrada en la CPU S7. La SFC se puede llamar desde el programa. Como las SFCs forman parte del sistema operativo, no se cargan como parte integrante del programa. Al igual que las FCs, las SFCs son bloques "sin memoria".

Las CPUs S7 ofrecen SFCs para:

- funciones de copia y de bloque
- control del programa
- manipulación del reloj y del contador de horas de funcionamiento
- transferencia de registros
- transferencia de eventos en el modo Multiprocesamiento desde una CPU a todas las CPUs insertadas
- manipulación de alarmas horarias y de retardo
- manipulación de eventos de errores síncronos, eventos de errores de alarma y asíncronos
- información sobre datos de sistema estáticos y dinámicos, p. ej. diagnóstico
- actualización de imágenes del proceso y tratamiento de campos de bits
- direccionamiento de módulos
- periferia descentralizada
- comunicación por datos globales
- la comunicación vía enlaces no configurados
- generar avisos de bloque

Informaciones adicionales

Para obtener informaciones más detalladas sobre los SFBs y las SFCs, consulte el manual de referencia "Software de sistema para S7-300/400, funciones estándar y funciones de sistema". En los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" o "Sistemas de automatización S7-400, M7-400, Datos de los módulos" se indican los SFBs y las SFCs que se encuentran disponibles.

4.2.4 Bloques de organización para la ejecución controlada por alarmas

Gracias a los OBs de alarma las CPUs ofrecen las siguientes prestaciones:

- Posibilidad de ejecutar partes del programa por control de tiempo
- Posibilidad de reaccionar eficazmente a señales externas del proceso.

El programa de usuario cíclico no necesita consultar una y otra vez si han aparecido eventos de alarma, sino que cuando aparece una alarma el sistema operativo se encarga de que se ejecute aquella parte del programa que reside en el OB de alarma y que define cómo ha de reaccionar sistema de automatización a esta alarma.

Tipos de alarmas y su aplicación

La tabla siguiente muestra la aplicación que se les da a los distintos tipos de alarmas.

Tipo de alarma	OBs de alarma	Ejemplos de aplicación
Alarma horaria	OB 10 a OB 17	Calcular el flujo de un proceso de mezcla al final de un turno
Alarma de retardo	OB 20 a OB 23	Controlar un ventilador para que se pare 20 segundos después de pararse un motor.
Alarma cíclica	OB 30 a OB 38	Muestrear el nivel de una señal para una planta de regulación
Alarma de proceso	OB 40 a OB 47	Notificar que se ha alcanzado el nivel máximo de un depósito.

4.2.4.1 Bloques de organización de alarma horaria (OB 10 a OB 17)

Las CPUs S7 ofrecen OBs de alarmas horarias que pueden ejecutarse a una fecha determinada o en intervalos específicos.

Las alarmas horarias se pueden activar:

- una vez, en una fecha determinada (indicación de hora absoluta con fecha)
- periódicamente, indicando la fecha de arranque y la frecuencia de repetición (p. ej., cada minuto, cada hora, cada día).

Reglas para las alarmas horarias

Las alarmas horarias sólo se pueden tratar si se ha parametrizado la alarma horaria y el correspondiente bloque de organización está contenido en el programa de usuario. En caso contrario se escribe un aviso de error en el búfer de diagnóstico y se efectúa un tratamiento asíncrono del error (OB 80, véase "Bloques de organización de tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)").

Las alarmas horarias periódicas deben corresponder a una fecha real. No será posible la repetición mensual de un OB 10 cuyo tiempo de arranque sea 31.1. En este caso, el OB sólo sería arrancado en los meses que tengan realmente 31 días (por lo tanto no en febrero, ni en abril, ni en junio, etc.)

Una alarma horaria activada durante el arranque (rearranque completo (en caliente) o rearranque) se ejecuta sólo tras finalizar el arranque.

Los OBs de alarmas horarias desactivados por parametrización no se pueden arrancar. La CPU reconoce un error de programación y pasa a STOP.

Tras un rearranque completo (en caliente) será preciso reactivar todas las alarmas horarias ajustadas (p.ej. mediante la SFC 30 ACT_TINT en el programa de arranque).

Arrancar la alarma horaria

Para que la CPU pueda arrancar una alarma horaria, ésta ha de ser ajustada previamente y activada luego. Se dispone de tres posibilidades de arranque:

- arranque automático de la alarma horaria por parametrización con STEP 7 (bloque de parámetros "alarmas horarias")
- ajustar y activar desde el programa de usuario la alarma horaria a través de la SFC 28 SET_TINT y SFC 30 ACT_TINT
- ajustar la alarma horaria por parametrización con STEP 7 y activarla desde el programa de usuario a través de la SFC 30 ACT_TINT.

Consultar la alarma horaria

Para consultar si hay alarmas horarias ajustadas, y a qué hora, se puede

- llamar la SFC 31 QRY_TINT o
- solicitar la lista parcial "Estado de alarma" de la lista de estado del sistema.

Desactivar la alarma horaria

Las alarmas horarias aún no ejecutadas se pueden desactivar con la SFC 29 CAN_TINT. Las alarmas horarias desactivadas se pueden ajustar nuevamente con la SFC 28 SET_TINT y activar con la SFC 30 ACT_TINT.

Prioridad de los OBs de alarma horaria

Los ocho OBs de alarmas horarias están preajustados con la misma prioridad (2) y, por consiguiente, se ejecutan en la secuencia de sus eventos de arranque. La prioridad se puede modificar por parametrización.

Cambiar la hora ajustada

Para cambiar la hora ajustada se dispone de las siguientes posibilidades:

- un reloj maestro sincroniza la hora para maestros y esclavos
- la hora se ajusta nuevamente en el programa de usuario con la SFC 0 SET_CLK.

Comportamiento en caso de cambiar la hora

La tabla siguiente ilustra cómo se comportan las alarmas horarias tras cambiar la hora.

Si...	entonces...
por adelantar la hora se saltan una o varias alarmas horarias,	se arranca el OB 80 y se registra en la información de arranque del OB 80 qué alarmas horarias han sido saltadas.
no se han desactivado en el OB 80 las alarmas horarias saltadas,	no se reactivan dichas alarmas horarias.
no se han desactivado en el OB 80 las alarmas horarias saltadas,	se reactiva la primera alarma horaria saltada y se ignoran las demás alarmas horarias saltadas.
por retrasar la hora, quedan pendientes nuevamente alarmas horarias ya ejecutadas,	se repite la ejecución de estas alarmas horarias en las CPUs S7-300 no se repite en las CPUs S7-400 y CPU 318.

4.2.4.2 Bloques de organización de alarma de retardo (OB 20 a OB 23)

Las CPUs S7 ofrecen OBs de alarmas de retardo que permite programar la ejecución retardada de partes del programa de usuario.

Reglas para las alarmas de retardo

Las alarmas de retardo sólo se pueden ejecutar si el correspondiente bloque de organización está contenido en el programa de la CPU. En caso contrario se escribe un aviso de error en el búfer de diagnóstico y se efectúa un tratamiento asíncrono del error (OB 80, véase "Bloques de organización de tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)").

Los OBs de alarmas de retardo desactivados por parametrización no se pueden arrancar. La CPU reconoce un error de programación y pasa a STOP.

Las alarmas de retardo se activan al transcurrir el tiempo de retardo ajustado en la SFC 32 SRT_DINT.

Arrancar la alarma de retardo

Para arrancar una alarma de retardo se debe ajustar en la SFC 32 el tiempo de retardo, tras cuya expiración se ha de llamar al correspondiente OB de alarma de retardo. En los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos" se indica la duración máxima admisible del tiempo de retardo.

Prioridad de los OBs de alarma de retardo

Para los OBs de alarma de retardo se han preajustado las prioridades 3 a 6. Estas últimas se pueden modificar por parametrización.

4.2.4.3 Bloques de organización de alarma cíclica (OB 30 a OB 38)

Las CPUs S7 ofrecen OBs de alarmas cíclicas que interrumpen la ejecución cíclica del programa en intervalos determinados.

Las alarmas cíclicas se activan en intervalos de tiempo determinados. El tiempo de arranque del período corresponde al cambio de estado operativo de STOP a RUN.

Reglas para las alarmas cíclicas

Al ajustar los períodos se debe tener en cuenta que entre los eventos de arranque de las diferentes alarmas cíclicas haya tiempo suficiente para la ejecución de dichas alarmas.

Los OBs de alarma cíclica desactivados por parametrización no se pueden arrancar. La CPU reconoce un error de programación y pasa a STOP.

Arrancar alarmas cíclicas

Para arrancar una alarma cíclica es necesario indicar el período (base de tiempo) correspondiente con STEP 7 en el bloque de parámetros de alarmas cíclicas. El período es siempre un múltiplo entero del período básico de 1 ms.

Período = $n \cdot$ período básico 1 ms

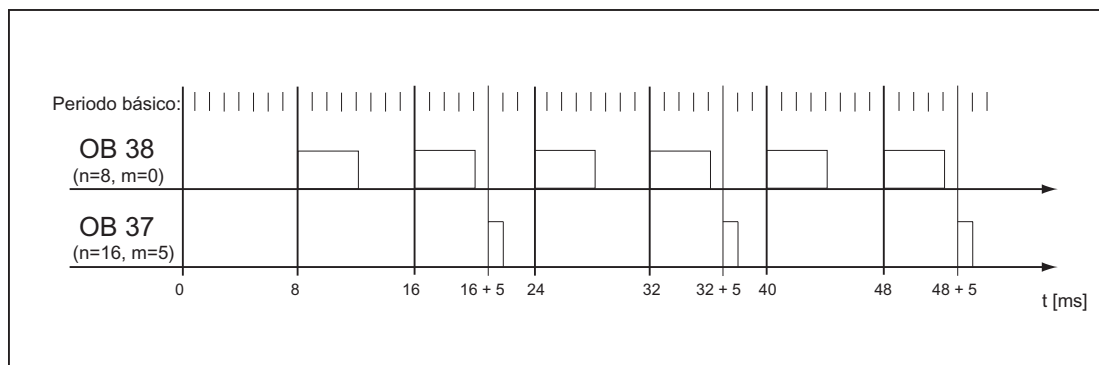
Los nueve OBs de alarmas cíclicas disponibles contienen períodos preajustados (v. tabla siguiente). El período predeterminado entra en vigor cuando está cargado el OB de alarmas cíclicas asignado. Sin embargo, los valores preajustados se pueden modificar por parametrización. En los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos" se indica el límite superior.

Alarmas cíclicas con desfase

Para impedir que las alarmas de diferentes OBs de alarmas cíclicas reciban al mismo tiempo una petición de arranque y se produzca eventualmente un error de tiempo (rebase del tiempo de ciclo), existe la posibilidad de prescribir un desfase. El desfase permite que, al transcurrir el período, la ejecución de una alarma cíclica se desplace en un intervalo de tiempo determinado.

Desfase = $m \cdot$ período básico (con $0 \leq m < n$)

La figura siguiente muestra la ejecución de un OB de alarma cíclica con desfase (OB 37) en contraposición a una alarma cíclica sin desfase (OB 38).



Prioridad del OB de alarma cíclica

La tabla siguiente muestra los períodos preajustados y las prioridades de los OBs de alarma cíclica. Tanto el período como la prioridad se pueden modificar por parametrización.

OBs de alarma cíclica	Período en ms	Prioridad
OB 30	5000	7
OB 31	2000	8
OB 32	1000	9
OB 33	500	10
OB 34	200	11
OB 35	100	12
OB 36	50	13
OB 37	20	14
OB 38	10	15

4.2.4.4 Bloques de organización de alarma de proceso (OB 40 a OB 47)

Las CPUs S7 ofrecen OBs de alarma de proceso que reaccionan a las señales de los módulos (p.ej. módulos de señales SMs, procesadores de comunicaciones CPs, módulos de función FMs). Para los módulos digitales y analógicos parametrizables se puede ajustar con STEP 7 qué señal debe arrancar al OB. Para los CPs y FMs se deben utilizar al respecto las pantallas de parametrización correspondientes.

Las alarmas de proceso son activadas cuando un módulo de señales con habilitación de alarma de proceso parametrizada transfiere a la CPU una señal de proceso recibida o cuando un módulo de función señala a la CPU una alarma.

Reglas para las alarmas de proceso

Las alarmas de proceso sólo se pueden ejecutar si el correspondiente bloque de organización está contenido en el programa de la CPU. En caso contrario se escribe un aviso de error en el búfer de diagnóstico y se efectúa un tratamiento asíncrono del error (véase "Bloques de organización de tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)").

Los OBs de alarmas de proceso desactivados por parametrización no se pueden arrancar. La CPU reconoce un error de programación y pasa a STOP.

Parametrizar módulos de señales aptos para alarmas de proceso

Cada canal de un módulo de señales apto para alarmas de proceso puede disparar una alarma de proceso. Por consiguiente, es necesario definir con STEP 7 en los registros de parámetros de tales módulos de señales:

- con qué se ha de disparar una alarma de proceso
- qué OB de alarma de proceso se debe ejecutar (el preajuste prevé el OB 40 para la ejecución de todas las alarmas de proceso).

Con STEP 7 se activa la generación de alarmas de proceso de los módulos de función. Otros parámetros se pueden asignar en las pantallas de parametrización de dichos módulos de función.

Prioridad de los OBs de alarma de proceso

Para los OBs de alarma de proceso se han preajustado las prioridades 16 a 23. Estas últimas se pueden modificar por parametrización.

4.2.4.5 Bloques de organización de arranque (OB 100 / OB 101 / OB 102)

Tipos de arranque

Se diferencia entre los siguientes tipos de arranque:

- Rearranque (no en S7-300 y S7-400H)
- Rearranque completo (en caliente)
- Arranque en frío

En la tabla siguiente se indica qué OB llama al sistema operativo durante el arranque.

Tipo de arranque	OB correspondiente
Rearranque	OB 101
Rearranque completo (en caliente)	OB 100
Arranque en frío	OB 102

Eventos de arranque de los OBs de arranque

La CPU realiza un arranque por

- tras CONEXIÓN
- al cambiar el selector de modos de operación de STOP a "RUN"/"RUN-P"
- tras petición por una función de comunicación
- tras sincronización en modo multiprocesador
- en un sistema H tras acoplar (sólo en CPU de reserva)

El OB de arranque (OB 100, OB 101 u OB 102) se invoca dependiendo del evento de arranque, de la CPU existente y de los parámetros ajustados en la misma.

Programa de arranque

Las condiciones para el comportamiento durante el arranque (valores de inicialización para RUN, valores de arranque para módulos de periferia) de la CPU se pueden definir depositando el programa para el arranque en el bloque de organización OB 100 para el rearranque completo (en caliente), el OB 101 para el rearranque u OB 102 para el arranque en frío.

El programa de arranque puede tener una longitud discrecional, su ejecución no tiene límite de tiempo y la vigilancia del tiempo de ciclo no está activa. La ejecución controlada por tiempo o por alarma no es posible en el programa de arranque. En el arranque, todas las salidas digitales tienen el estado de señal 0.

Modo de arranque tras arranque manual

En las CPUs S7-300 sólo se puede efectuar un re arranque completo (en caliente) manual o un arranque en frío (sólo CPU 318-2).

Las CPUs S7-400 permiten efectuar un re arranque manual a través del selector de modos de operación y del selector de modos de arranque (CRST/WRST), siempre y cuando se haya parametrizado correspondientemente con STEP 7. El re arranque completo (en caliente) manual también es posible sin parametrización previa.

Modo de arranque tras arranque automático

En las CPUs S7-300 sólo es posible efectuar un re arranque completo (en caliente) tras CONEXION.

En las CPUs S7-400 se puede definir si, tras CONEXION, el arranque automático debe provocar un re arranque completo (en caliente) o un re arranque normal.

Borrar la imagen de proceso

Durante el re arranque de una CPU S7-400 se borra, por defecto, la imagen de proceso de las salidas tras la ejecución del ciclo residual. El borrado de la imagen de proceso se puede inhibir si se desea que el programa de usuario continúe utilizando los valores que eran actuales antes de dicho re arranque.

Vigilancia de la configuración teórica/real de los módulos

A través de la parametrización se puede definir si, antes de efectuar el arranque, se debe comprobar si todos los módulos listados en la tabla de configuración están insertados realmente y si el tipo de módulo es correcto.

Si la vigilancia de módulos está activa, el arranque no se ejecuta en caso que se detecte una diferencia entre las configuraciones teórica y real.

Tiempos de vigilancia

Para garantizar un arranque sin errores del sistema de automatización, se han de parametrizar los siguientes tiempos de vigilancia:

- el tiempo máximo admisible para la transmisión de parámetros a los módulos
- el tiempo máximo admisible para el mensaje "ready" de los módulos tras CONEXION
- en las CPUs S7-400, el tiempo máximo de interrupción en el cual puede ejecutarse aún un re arranque.

Transcurridos los tiempos de vigilancia, la CPU pasa a STOP o sólo es posible efectuar un re arranque completo (en caliente).

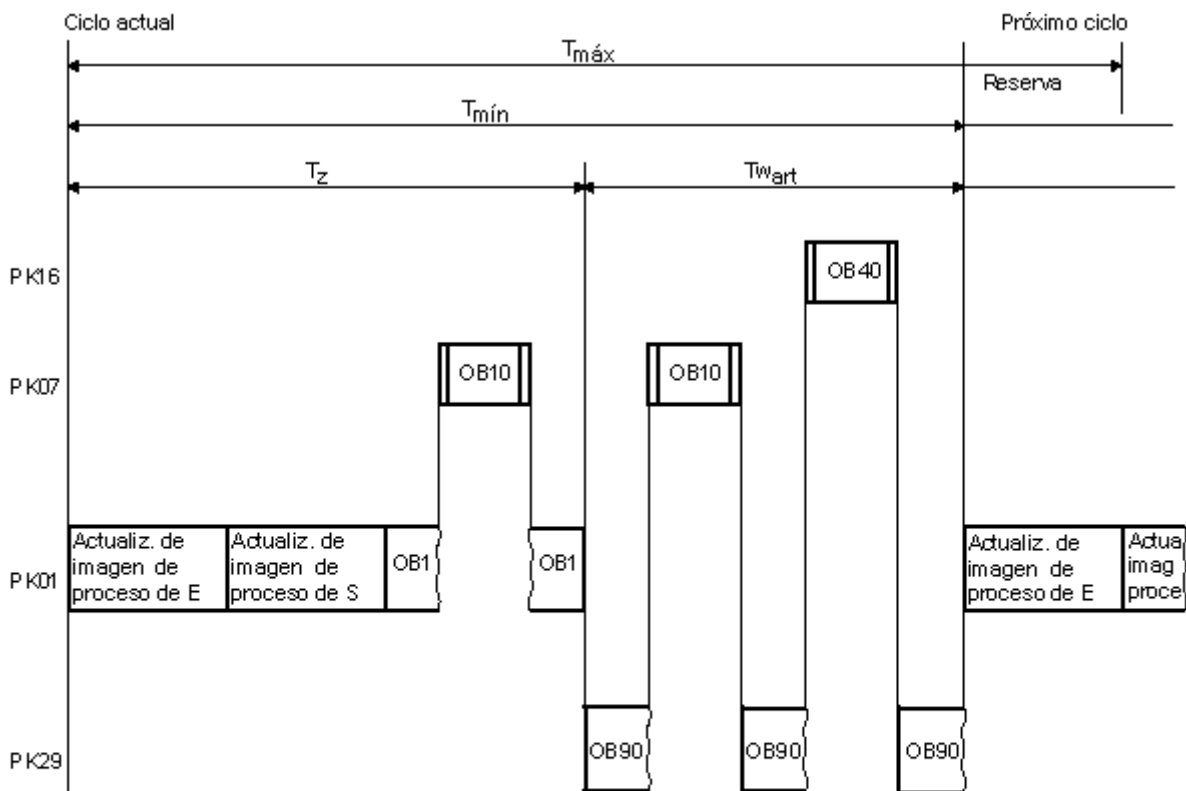
4.2.4.6 Bloque de organización de tarea no prioritaria (OB 90)

Si con STEP 7 se ha definido un tiempo de ciclo mínimo y éste es mayor que el tiempo de ciclo real, al finalizar el programa cíclico la CPU dispone aún de tiempo de ejecución. Dicho tiempo se utiliza para procesar el OB de tarea no prioritaria. Si la CPU utilizada carece de OB 90, entonces espera hasta que transcurre el tiempo de ciclo mínimo seleccionado. Por tanto, a través del OB 90 se pueden ejecutar procesos de tiempo no crítico, evitando así tiempos de espera.

Prioridad del OB de tarea no prioritaria

El OB de tarea no prioritaria tiene la prioridad 29, que equivale a la prioridad 0.29. Por tanto, se trata del OB con la prioridad más baja. La prioridad no puede modificarse por parametrización.

La figura siguiente muestra un ejemplo de ejecución del ciclo de tarea no prioritaria, del ciclo libre y del OB 10 (en las CPUs a partir de 10/98).



- $T_{máx}$ = Tiempo de ciclo máximo ajustable
- $T_{mín}$ = Tiempo de ciclo mínimo ajustable
- T_z = Tiempo de ciclo real
- T_{wart} = La diferencia entre $T_{mín}$ y el tiempo de ciclo real; durante este tiempo pueden ejecutarse los OBs de alarmas y los OBs de background
- PK = Prioridad

Programar el OB 90

El sistema operativo de la CPU no supervisa el tiempo de ejecución del OB 90, por lo que en éste es posible programar bucles de cualquier longitud. Asegúrese de que los datos utilizados en el programa de baja prioridad sean coherentes, observando lo siguiente al programar:

- los eventos de inicialización del OB 90 (consulte a este respecto el manual de referencia "Software de sistema para S7-300/400, funciones estándar y funciones de sistema"),
- la actualización de la imagen del proceso asíncrona al OB 90.

4.2.4.7 Bloques de organización de tratamiento de errores (OB 70 a OB 87 / OB 121 a OB 122)

Tipos de errores

Los errores reconocibles por las CPUs S7 y a los cuales se puede reaccionar con los bloques de organización se clasifican en dos categorías:

- Errores síncronos: estos errores se pueden asignar a una parte determinada del programa de usuario. El error es provocado por una determinada operación durante la ejecución. Si el OB de error síncrono correspondiente no está cargado, la CPU pasa a STOP al presentarse el error.
- Errores asíncronos: estos errores no se pueden asignar directamente al programa de usuario en ejecución. Se trata de errores de prioridad, errores en el sistema de automatización (p. ej., fallos de módulos) o de errores de redundancia. Si el OB de error asíncrono correspondiente no está cargado, la CPU pasa a STOP al presentarse el error. (Excepciones: OB 70, OB 72, OB 81, OB 87).

La tabla siguiente muestra los tipos de errores que se pueden presentar, subdivididos según la categoría de los OBs de error.

Errores asíncronos / errores de redundancia	Errores síncronos
OB 70 Errores de redundancia en la periferia (sólo en CPUs H)	OB 121 Error de programación (p.ej. DB sin cargar)
OB 72 Errores de redundancia en la CPU (sólo en CPUs H, p.ej. fallo de una CPU)	OB 122 Error de acceso a periferia (p.ej. acceso a un módulo de señales que no existe)
OB 80 Error de tiempo (p.ej. tiempo de ciclo sobrepasado)	
OB 81 Error de alimentación (p.ej. fallo de pila)	
OB 82 Alarma de diagnóstico (p.ej. cortocircuito en un módulo de entradas)	
OB 83 Alarma de presencia de módulo (p.ej. extraer un módulo de entradas)	
OB 84 Avería de hardware CPU (error en interface a red MPI)	
OB 85 Error de ejecución del programa (p.ej. OB sin cargar)	
OB 86 Error en el bastidor	
OB 87 Error de comunicación (p.ej. identificación de telegrama errónea en comunicación por datos globales)	

Uso de los OBs para errores síncronos

Los errores síncronos son causados durante la ejecución de una operación determinada. Cuando ocurren estos errores, el sistema operativo registra una entrada en la pila U (USTACK) y arranca el OB para errores síncronos.

Los OBs de errores que son llamados por errores síncronos se ejecutan como parte del programa con la misma prioridad que el bloque en tratamiento al detectarse el error. Si desea obtener más información sobre el error que ha provocado la llamada del OB, consulte la información de arranque del OB. Esta información se puede utilizar para reaccionar a la condición de error y regresar luego a la ejecución del programa (p.ej. en caso de errores de acceso a módulo de entradas analógicas, predeterminar un valor de sustitución en el OB 122 con la SFC 44 RPL_VAL). No obstante, los datos locales de los OBs de error representan una carga adicional para la LSTACK (pila de datos locales) de esta prioridad.

En las CPUs S7-400 es posible arrancar desde un OB de error síncrono a otro OB de error síncrono. Esto no es posible en las CPUs S7-300.

Uso de los OBs para errores asíncronos

Si el sistema operativo de la CPU detecta un error asíncrono, arranca el correspondiente OB de error (OB 70 a OB 73 y OB 80 a OB 87). Los OBs para errores asíncronos tienen la máxima prioridad por defecto: No pueden ser interrumpidos por otros OBs, si todos los OBs para errores asíncronos tienen la misma prioridad. Al presentarse simultáneamente varios OBs para errores asíncronos de igual prioridad, serán ejecutados en el orden de aparición.

Enmascarar eventos de arranque

Las funciones del sistema (SFC) permiten enmascarar, retardar o inhibir eventos de arranque para algunos OBs de tratamiento de errores. Para obtener informaciones más detalladas a este respecto, así como en relación con los diversos bloques de organización, consulte el manual de referencia "Software de sistema para S7-300/400, funciones estándar y funciones de sistema".

Tipo de OB de error	SFC	Función de la SFC
OB de errores síncronos	SFC 36 MSK_FLT	Enmascarar eventos de error síncronos individuales. Los eventos de error enmascarados no arrancan ningún OB de error y no conducen a la reacción de sustitución programada.
	SFC 37 DMSK_FLT	Desenmascarar eventos de error síncronos.
OB de errores asíncronos	SFC 39 DIS_IRT	Inhibir globalmente los eventos de alarma y los de errores asíncronos. Los eventos de error inhibidos no arrancan OBs de error en ninguno de los ciclos posteriores de la CPU y no conducen a la reacción de sustitución programada.
	SFC 40 EN_IRT	Habilitar eventos de alarma y de errores asíncronos
	SFC 41 DIS_AIRT	Retardar los eventos de alarma y los de errores asíncronos de mayor prioridad hasta el final del OB.
	SFC 42 EN_AIRT	Habilitar los eventos de alarma y los de errores asíncronos de mayor prioridad

Nota

Para ignorar alarmas resulta más efectivo inhibirlas mediante una SFC durante el arranque, en vez de cargar un OB vacío (con resultado binario RB).

5 **Cómo arrancar y utilizar STEP 7**

5.1 **Cómo iniciar STEP 7**



Una vez arrancado Windows, en la pantalla aparece el icono del Administrador SIMATIC que permite acceder al software STEP 7.

La manera más rápida de arrancar STEP 7 consiste en hacer un doble clic en el icono "Administrador SIMATIC", con lo que se abrirá la ventana del Administrador SIMATIC. Desde allí es posible acceder a todas las funciones instaladas, tanto del paquete estándar como de los paquetes opcionales.

Asimismo, puede iniciar el Administrador SIMATIC a través del botón "Inicio" de la barra de tareas del sistema operativo (entrada en "Simatic").

Nota

Para obtener más información sobre las funciones y opciones estándar de Windows, consulte el manual del usuario o la Ayuda en pantalla de su sistema operativo Windows.

Administrador SIMATIC

El Administrador SIMATIC es el interface de acceso a la configuración y programación. Éste permite:

- crear proyectos,
- configurar y parametrizar el hardware,
- configurar redes de hardware,
- programar bloques,
- probar y hacer funcionar los programas.

El acceso a las funciones es orientado a objetos, con lo cual resulta fácil de aprender.

Puede trabajar con el Administrador SIMATIC

- offline, es decir, sin conectar el sistema de automatización, o bien
- online, es decir, estando conectado el sistema de automatización.

(Tenga en cuenta las correspondientes observaciones de seguridad.)

Procedimiento ulterior

Las soluciones de automatización se crean en forma de "proyectos". Si se familiariza previamente con los siguientes temas generales, el trabajo le resultará más sencillo:

- el interface de usuario,
- algunos manejos básicos,
- la Ayuda en pantalla.

5.2 Arrancar STEP 7 con los parámetros de arranque estándar

A partir de la versión 5 de STEP 7 se pueden crear varios iconos del Administrador SIMATIC e indicar parámetros de arranque en la línea de llamada con objeto de que el Administrador SIMATIC se posicione en el objeto descrito por esos parámetros. Con un doble clic puede acceder directamente a los puntos correspondientes en un determinado proyecto.

Llamando a **s7tgotpx.exe** puede indicar los siguientes parámetros de arranque:

/e <ruta física completa del proyecto>

/o <ruta lógica del objeto donde se debe posicionar>

/h <ID del objeto>

A continuación se indica el método más sencillo para averiguar los parámetros apropiados.

/onl

El parámetro de arranque **/onl** hace que el proyecto se abra online y que se llame la ruta indicada.

/off

El parámetro de arranque **/off** hace que el proyecto se abra offline y que se llame la ruta indicada.

/keep

El parámetro de arranque **/keep** hace que estando abierto el Administrador SIMATIC se abra el proyecto que se va a abrir desde la línea de comandos además de los proyectos ya visualizados. Si el Administrador SIMATIC todavía no está abierto, entonces se abre el proyecto que se va a abrir además de los proyectos guardados en la memoria de sesión del Administrador SIMATIC.

Si no se indica este parámetro de arranque, primero se cerrarán los proyectos que estén abiertos, se ignora la memoria de sesión y sólo se abre el proyecto transferido.

Para determinar qué parámetros son los apropiados, proceda de la manera siguiente.

Determinar los parámetros utilizando las funciones "Copiar" y "Pegar"

Proceda como se describe a continuación:

1. Cree en su escritorio un nuevo acceso directo al archivo s7tgotpx.exe. Encontrará este archivo en el directorio de instalación bajo S7bin.
2. Abra el cuadro de diálogo de propiedades.
3. Elija la ficha "Acceso directo". La entrada que aparece en "Destino" se complementa de la siguiente forma:
4. Seleccione el objeto deseado en el Administrador SIMATIC.
5. Copie el objeto al portapapeles, utilizando la combinación de teclas CTRL+ALT+C.
6. Sitúe el cursor al final de la entrada "Destino" en la ficha "Acceso directo".
7. Pegue el contenido del portapapeles utilizando la combinación de teclas CTRL+V.
8. Cierre el cuadro de diálogo haciendo clic en "Aceptar".

Ejemplo de indicación de parámetros:

```
/e F:\SIEMENS\STEP7\S7proj\MyConfig\MyConfig.s7p /keep  
/o "1,8:MyConfig\SIMATIC 400(1)\CPU416-1\ProgramaS7(1)\Bloques\FB1"  
/h T00112001;129;T00116001;1;T00116101;16e /keep
```

Nota sobre la estructura de la ruta del proyecto

La ruta del proyecto es la ruta física del sistema de archivos.

La ruta lógica completa está estructurada de la siguiente forma:

[Identificador de visualización,identificador online]:nombre del proyecto\{nombre del objeto}*\ nombre del objeto

Ejemplo: /o 1,8:MyConfig\SIMATIC 400(1)\CPU416-1\ProgramaS7(1)\Bloques\FB1

Nota respecto a la estructura de la ruta lógica

La ruta lógica completa y el ID del objeto sólo se pueden crear utilizando las funciones Copiar y Pegar. No obstante, también existe la posibilidad de indicar la ruta que el usuario puede leer, es decir, conforme al ejemplo indicado arriba:

```
/o "MyConfig\SIMATIC 400(1)\CPU416-1\ProgramaS7(1)\Bloques\FB1".
```

Con /onl u /off, el usuario puede indicar si la ruta es válida en la ventana online u offline. Esta indicación no es necesaria al utilizarse las funciones Copiar y Pegar.

Nota importante: Si la ruta contiene espacios en blanco, se deberá colocar entre comillas.

5.3 Cómo acceder a las funciones de ayuda

Ayuda en pantalla

La Ayuda en pantalla ofrece informaciones directas. Así puede consultar de forma puntualizada la información que necesite, sin tener que buscarla en un manual. En la Ayuda en pantalla encontrará:

- **Temas de Ayuda:** ofrece diversos accesos para visualizar los temas de Ayuda.
- **Ayuda contextual** (tecla F1): muestra informaciones sobre el objeto seleccionado, o bien, acerca del cuadro de diálogo o de la ventana actual.
- **Introducción:** presenta una panorámica de la aplicación, de sus principales características y del volumen de sus funciones.
- **Primeros pasos:** resume los primeros pasos que se deben dar para poder comenzar a trabajar con éxito.
- **Uso de la Ayuda:** describe las posibilidades disponibles para encontrar determinadas informaciones en la Ayuda en pantalla.
- **Acerca de:** proporciona información sobre la versión actual de la aplicación.

A través del menú "Ayuda" es posible acceder también desde cualquier ventana a los temas que guarden relación con el cuadro de diálogo actual.

Llamar la Ayuda en pantalla

Para llamar la Ayuda en pantalla puede utilizar alguna de las siguientes alternativas:

- Elegir un comando del menú "Ayuda" en la barra de menús.
- Hacer clic en el botón "Ayuda" de un cuadro de diálogo. A continuación, obtendrá ayuda sobre dicho cuadro.
- Situar el puntero del ratón en una ventana o en un cuadro de diálogo, en el tema sobre el que requiera ayuda y pulsando la tecla F1 o elegir el comando **Ayuda > Ayuda contextual**.
- Hacer clic con el cursor de interrogación de Windows (¿Qué es esto?).

Las tres últimas alternativas constituyen la Ayuda contextual.

Llamar la Ayuda rápida

Si sitúa el puntero del ratón en un botón de la barra de herramientas y lo deja allí por corto tiempo, obtendrá ayuda rápida sobre ese botón.

5.4 Objetos y su jerarquía

La jerarquía de objetos para proyectos y librerías del Administrador SIMATIC está estructurada de la misma forma que los directorios con carpetas y archivos del Explorador de Windows.

La figura siguiente muestra un ejemplo de jerarquía de objetos.

<pre> graph TD Proyecto --> Equipo Equipo --> Modulo[Módulo programable] Modulo --> Programa[Programa S7] Programa --> Fuentes Programa --> Bloques </pre>	<ul style="list-style-type: none"> • Objeto "Proyecto" • Objeto "Equipo" • Objeto "Módulo programable" • Objeto "Programa S7/M7" • Objeto "Carpeta de fuentes" • Objeto "Carpeta de bloques"
---	--

Los objetos sirven de:

- portadores de propiedades,
- carpetas,
- portadores de funciones (p.ej., para arrancar una aplicación determinada).

Objetos portadores de propiedades

Los objetos pueden ser tanto portadores de funciones como de propiedades (p. ej., de ajustes). Una vez seleccionado un objeto es posible

- editar el objeto eligiendo el comando **Edición > Abrir objeto**.
- abrir un cuadro de diálogo con el comando **Edición > Propiedades del objeto** y determinar allí los ajustes específicos del objeto.

Una carpeta puede ser también portador de propiedades.

Objetos del tipo "Carpeta"

Una carpeta puede contener otras carpetas u objetos. Estos se visualizan al abrir la carpeta.

Objetos portadores de funciones

Al abrir un objeto aparece una ventana para poder editarlo.

Un objeto puede ser una carpeta o un portador de funciones. Los equipos constituyen la excepción a la regla, puesto que son tanto carpetas (de módulos programables) como portadores de funciones (para configurar el hardware).

- Si hace doble clic en un equipo, se mostrarán los objetos que contiene: los módulos programables y la configuración del equipo (equipo como carpeta).
- Si abre un equipo con el comando **Edición > Abrir objeto**, puede configurar y parametrizar dicho equipo (equipo portador de una función). Este comando de menú tiene el mismo efecto que un doble clic en el objeto "Hardware".





5.4.1 Objeto "Proyecto"

Un proyecto representa la totalidad de los datos y programas de una tarea de automatización, constituyendo el nivel superior de la jerarquía de objetos.

Posicionamiento en la ventana del proyecto


	<ul style="list-style-type: none"> • Objeto "Proyecto" • Objeto "Equipo" • Objeto "Módulo programable" • Objeto "Programa S7/M7" • Objeto "Carpeta de fuentes" • Objeto "Carpeta de bloques"
--	--


Símbolo	Carpeta de objetos	Selección de funciones importantes
	Proyecto	<ul style="list-style-type: none"> • Crear un proyecto • Archivar proyectos y librerías • Gestionar textos en varios idiomas • Comprobar los paquetes opcionales utilizados en el proyecto • Imprimir la documentación de un proyecto • Reorganizar • Traducir y editar textos de usuario • Insertar estaciones de operador (OS) • Compartir la elaboración de proyectos • Convertir proyectos de la versión 1 • Convertir proyectos de la versión 2 • Ajustar interface PG/PC



Símbolo	Objetos en el nivel de proyecto	Selección de funciones importantes
	Equipo: Equipo SIMATIC 300 Equipo SIMATIC 400	Insertar equipos <ul style="list-style-type: none"> Los equipos son al mismo tiempo objetos (del nivel de proyectos) y carpetas de objetos (del nivel de equipo). En el tema Objeto "Equipo" se describen las demás funciones.
 	Programa S7 Programa M7	<ul style="list-style-type: none"> Programa S7/M7 sin equipo ni CPU Los programas S7/M7 son al mismo tiempo objetos (del nivel de proyectos) y carpetas de objetos (del nivel de programas). En el tema Objeto "Programa S7/M7" se describen las demás funciones.
	Red para arrancar la aplicación para configurar la red y ajustar sus propiedades	<ul style="list-style-type: none"> Propiedades de subredes y estaciones de la red Comunicación de datos globales. Sinopsis Procedimiento para configurar la comunicación de datos globales

5.4.2 Objeto "Librería"

Una librería puede contener programas S7/M7 y sirve para depositar bloques. Se encuentra en el nivel superior de una jerarquía de objetos.

	<ul style="list-style-type: none"> • Objeto "Librería" • Objeto "Programa S7/M7" • Objeto "Carpeta de fuentes" • Objeto "Carpeta de bloques"
---	--

Símbolo	Carpeta de objetos	Selección de funciones importantes
	Librería	<ul style="list-style-type: none"> • Panorámica de las librerías estándar • Trabajar con librerías • Archivar proyectos y librerías

Símbolo	Objetos en el nivel de librerías	Selección de funciones importantes
	Programa S7	<ul style="list-style-type: none"> • Insertar un programa S7 o M7 • Los programas S7/M7 son tanto objetos (en el nivel de proyectos) como carpetas de objetos (en el nivel de programas). En el tema Objeto "Programa S7/M7" se describen las demás funciones.
	Programa M7	

5.4.3 Objeto "Equipo"

Un equipo SIMATIC 300/400 representa una configuración de hardware S7 compuesta por uno o varios módulos programables.

Posicionamiento en la ventana del proyecto

	<ul style="list-style-type: none"> • Objeto "Proyecto" • Objeto "Equipo" • Objeto "Módulo programable" • Objeto "Programa S7/M7" • Objeto "Carpeta de fuentes" • Objeto "Carpeta de bloques"
--	--

Símbolo	Carpeta de objetos	Selección de funciones importantes
	Equipo	<ul style="list-style-type: none"> • Insertar equipos • Cargar equipo en PG • Cargar una configuración en un sistema de destino • Cargar la configuración desde otro equipo • Visualizar los avisos de la CPU y los avisos de diagnóstico personalizados • Configurar la función 'Notificar errores del sistema' • Diagnóstico del hardware y visualizar el estado del módulo • Visualizar y cambiar el estado operativo de la CPU • Visualizar y ajustar la hora y la fecha de la CPU • Borrar las memorias de carga/de trabajo y borrado total de la CPU
	Equipo PC Simatic (no asignado)	<ul style="list-style-type: none"> • Crear y parametrizar equipos PC SIMATIC • Configurar enlaces para un equipo PC SIMATIC • Carga de un equipo PC SIMATIC
	Equipo PC SIMATIC (asignado)	<ul style="list-style-type: none"> • Resaltar un equipo PC SIMATIC que se va a configurar en la vista de la red

Símbolo	Objetos en el nivel de equipos	Selección de funciones importantes
	Hardware	<ul style="list-style-type: none"> • Pasos fundamentales para configurar el hardware • Pasos fundamentales para configurar un equipo • Visión de conjunto: Procedimiento para configurar y parametrizar un sistema centralizado • Procedimiento básico para configurar un sistema maestro DP • Configurar el modo multiprocesador
	Módulo programable	<ul style="list-style-type: none"> • Los módulos programables son al mismo tiempo objetos (del nivel de equipo) y carpetas de objetos (del nivel módulos programables). En el tema Objeto "Módulos programables" se describen las demás funciones.



5.4.4 Objeto "Módulo programable"

Un módulo programable representa los datos de parametrización de un módulo programable (CPUxxx, FMxxx, CPxxx). Los datos de sistema de los módulos que no disponen de memoria remanente (p.ej. CP441), se cargan a través de la CPU del equipo. Por tanto, a dichos módulos no se ha asociado ningún objeto "Datos de sistema", por lo que no se visualizan en la jerarquía de proyectos.

Posicionamiento en la ventana del proyecto

	<ul style="list-style-type: none"> • Objeto "Proyecto" • Objeto "Equipo" • Objeto "Módulo programable" • Objeto "Programa S7/M7" • Objeto "Carpeta de fuentes" • Objeto "Carpeta de bloques"
--	--

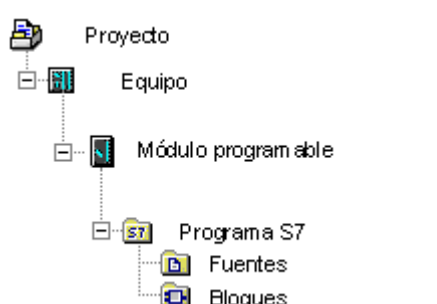
Símbolo	Carpeta de objetos	Selección de funciones importantes
	Módulo programable	<ul style="list-style-type: none"> • Visión de conjunto: Procedimiento para configurar y parametrizar un sistema centralizado • Visualizar los avisos de la CPU y los avisos de diagnóstico personalizados • Configurar 'Notificar errores del sistema' • Diagnóstico del hardware y visualizar el estado del módulo • Cargar mediante Memory Cards EPROM • Protección con contraseña para acceder a sistemas de destino • Visualizar ventanas de forzado permanente • Visualizar y cambiar el estado operativo • Visualizar y ajustar la fecha y hora • Ajustar el comportamiento del sistema • Borrar las memorias de carga/de trabajo y borrado total de la CPU • Símbolos de diagnóstico en la ventana online • Subdivisión de la memoria en áreas • Guardar en la EPROM integrada los bloques cargados • Actualizar el sistema operativo en el sistema de destino
	Objeto sustituto para un módulo programable	<ul style="list-style-type: none"> • Representación de módulos configurados con nuevas versiones de STEP 7




Símbolo	Objetos en el nivel de "Módulos programables"	Selección de funciones importantes
	<p>Programas:</p> <p>Programa S7</p> <p>Programa M7</p> <p>Programa</p>	<ul style="list-style-type: none"> • Insertar un programa S7 o M7 • Los programas S7/M7 son tanto objetos (en el nivel de proyectos) como carpetas de objetos (en el nivel de programas). En el tema Objeto "Programa S7/M7" se describen las demás funciones.
	<p>Enlaces para definir enlaces en la red</p>	<ul style="list-style-type: none"> • Conectar equipos de un mismo proyecto a la red • Tipos de enlaces e interlocutores • Informaciones importantes sobre los distintos tipos de enlaces • Introducir un nuevo enlace • Configurar enlaces para módulos en un equipo SIMATIC





5.4.5 Objeto "Programa S7/M7"

Un programa (S7/M7) es una carpeta de software que contiene módulos CPU S7/M7 o módulos que no sean CPUs (p.ej. módulos CP o FM programables).

Emplazamiento en la vista del proyecto

	<ul style="list-style-type: none"> • Objeto "Proyecto" • Objeto "Equipo" • Objeto "Módulo programable" • Objeto "Programa S7/M7" • Objeto "Carpeta de fuentes" • Objeto "Carpeta de bloques"
---	---

Símbolo	Carpeta de objetos	Selección de funciones importantes
	Programa S7	<ul style="list-style-type: none"> • Insertar un programa S7 o M7 • Ajustar la preferencia de operandos (simbólica/absoluta) • Procedimiento básico para crear bloques lógicos • Asignar números de aviso • Crear y editar avisos de diagnóstico personalizados (para todo el proyecto) • Crear avisos de diagnóstico personalizados (para toda la CPU) • Traducir y editar textos de usuario • Gestionar textos en varios idiomas • Visualizar avisos de CPU y avisos de diagnóstico personalizados • Medidas en el programa para tratar fallos
	Programa M7	Procedimiento para sistemas M7
	Programa	<ul style="list-style-type: none"> • Crear el software en el proyecto

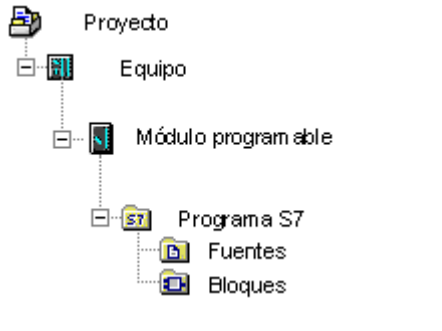
Símbolo	Objetos en el nivel de programas	Selección de funciones importantes
	Carpeta de fuentes	En el tema Objeto "Carpeta de fuentes" se describen las demás funciones.
	Carpeta de bloques	<ul style="list-style-type: none"> • En el tema Objeto "Carpeta de bloques" se describen las demás funciones.
	Carpeta de librerías de texto	<ul style="list-style-type: none"> • Librerías de texto de usuario
	Tabla de símbolos para asignar símbolos a señales y otras variables	<ul style="list-style-type: none"> • Direccionamiento absoluto y simbólico • Estructura y componentes de la tabla de símbolos • Distintas posibilidades para introducir símbolos globales • Observaciones generales para introducir símbolos • Asignar y editar avisos de símbolos (para todo el proyecto) • Asignar y editar avisos de símbolos (para toda la CPU) • Traducir y editar textos de usuario • Configurar atributos M+V mediante la tabla de símbolos • Editar el atributo de comunicación • Exportar e importar tablas de símbolos


5.4.6 Objeto "Carpeta de bloques"





Una carpeta de bloques de una vista offline puede contener bloques lógicos (OB, FB, FC, SFB, SFC), bloques de datos (DB), tipos de datos de usuario (UDT) y tablas de variables. El objeto "Datos de sistema" representa bloques de datos de sistema.








La carpeta de bloques de una vista online contiene las partes ejecutables del programa residentes en el sistema de destino.


Posicionamiento en la ventana del proyecto

	<ul style="list-style-type: none"> • Objeto "Proyecto" • Objeto "Equipo" • Objeto "Módulo programable" • Objeto "Programa S7/M7" • Objeto "Carpeta de fuentes" • Objeto "Carpeta de bloques"
---	--

Símbolo	Carpeta de objetos	Selección de funciones importantes
	Bloques	<ul style="list-style-type: none"> • Cargar con gestión de proyectos • Cargar sin gestión de proyectos • Visión de conjunto de los datos de referencia posibles • Recablear • Comparar bloques • Traducir y editar textos de usuario • Saltos a descripciones de lenguajes, ayudas sobre bloques y atributos de sistema

Símbolo	Objetos en la carpeta de bloques	Selección de funciones importantes
	Bloques en general	<ul style="list-style-type: none"> • Procedimiento básico para crear bloques lógicos • Crear bloques • Reglas para programar en fuentes AWL • Comparar bloques
	OB (Bloques de organización)	<p>Funciones adicionales:</p> <ul style="list-style-type: none"> • Introducción a los tipos de datos y a los tipos de parámetros • Requisitos para cargar • Test con el estado del programa • Información importante para el test en modo Etapa individual / Puntos de parada • Recablear • Ayuda de bloques
	FC (Funciones)	<p>Funciones adicionales:</p> <ul style="list-style-type: none"> • Introducción a los tipos de datos y a los tipos de parámetros • Requisitos para cargar • Test con el estado del programa • Información importante para el test en modo Etapa individual / Puntos de parada • Recablear • Atributos para bloques y parámetros
	FB (Bloques de función)	<p>Funciones adicionales:</p> <ul style="list-style-type: none"> • Introducción a los tipos de datos y a los tipos de parámetros • Uso de multiinstancias • Requisitos para cargar • Test con el estado del programa • Información importante para el test en modo Etapa individual / Puntos de parada • Recablear • Atributos para bloques y parámetros • Asignar y editar avisos de bloques (para todo el proyecto) • Crear avisos de bloques (para toda la CPU) • Configuración de avisos PCS7 (para todo el proyecto) • Configuración de avisos PCS 7 (para toda la CPU) • Traducir y editar textos de usuario • Asignar atributos de M+V a parámetros FB
	UDT (Tipos de datos de usuario)	<ul style="list-style-type: none"> • Crear bloques • Reglas para programar en fuentes AWL • Introducción a los tipos de datos y a los tipos de parámetros • Uso de tipos de datos de usuario para acceder a los datos • Atributos para bloques y parámetros

Símbolo	Objetos en la carpeta de bloques	Selección de funciones importantes
	DB (Bloques de datos globales)	<ul style="list-style-type: none"> • Ver datos de los bloques de datos • Ver declaración de bloques de datos • Requisitos para cargar • Observar el estado de bloques de datos • Introducción a los tipos de datos y a los tipos de parámetros • Utilización de multiinstancias • Atributos para bloques y parámetros • Asignar y editar avisos de bloques (para todo el proyecto - sólo DBs de instancia) • Editar avisos de bloques (para toda la CPU - sólo DBs de instancia) • Configuración de avisos PCS7 (para todo el proyecto - sólo DBs de instancia) • Configuración de avisos PCS 7 (para toda la CPU - sólo DBs de instancia) • Traducir y editar textos de usuario(sólo DBs de instancia)
	SFC (Funciones de sistema)	<ul style="list-style-type: none"> • Requisitos para cargar • Atributos para bloques y parámetros • Ayuda de bloques
	SFB (Bloques de función de sistema)	<ul style="list-style-type: none"> • Requisitos para cargar • Atributos para bloques y parámetros • Asignar y editar avisos de bloques (para todo el proyecto) • Crear avisos de bloques (para toda la CPU) • Configuración de avisos PCS7 (para todo el proyecto) • Configuración de avisos PCS7 (para toda la CPU) • Traducir y editar textos de usuario • Ayuda sobre bloques
	Módulo con protección KNOW HOW	<ul style="list-style-type: none"> • Reglas para programar en fuentes AWL • Propiedades del bloque
	Bloque apto para el diagnóstico	Encontrará más información en la documentación del paquete opcional S7-PDIAG.
	El bloque se ha creado con los lenguajes de creación F-FUP/-KOP/-AWL/-DB	Encontrará más información en la documentación del paquete opcional S7 Distributed Safety.
	VAT (tabla de variables).	<ul style="list-style-type: none"> • Procedimiento básico para observar y forzar con tablas de variables • Introducción a la comprobación de programas utilizando una tabla de variables • Introducción a la observación de variables • Introducción al forzado normal de variables • Introducción al forzado permanente de variables

Símbolo	Objetos en la carpeta de bloques	Selección de funciones importantes
	Datos de sistema (SDB)	Los SDBs sólo se procesan indirectamente a través de funciones de: <ul style="list-style-type: none"> • Introducción a la configuración del hardware • Propiedades de subredes y estaciones de la red • Comunicación de datos globales. Sinopsis • Asignar y editar avisos de símbolos • Requisitos para cargar

5.4.7 Objeto "Carpeta de fuentes"

Una carpeta de fuentes contiene programas fuente en forma de texto.

Posicionamiento en la ventana del proyecto

	<ul style="list-style-type: none"> • Objeto "Proyecto" • Objeto "Equipo" • Objeto "Módulo programable" • Objeto "Programa S7/M7" • Objeto "Carpeta de fuentes" • Objeto "Carpeta de bloques"
--	---

Símbolo	Carpeta de objetos	Selección de funciones importantes
	Carpeta de fuentes	<ul style="list-style-type: none"> • Reglas para programar en fuentes AWL • Exportar fuentes • Importar fuentes

Símbolo	Objetos en la carpeta de fuentes	Selección de funciones importantes
	Fuente (p. ej., fuente AWL)	<ul style="list-style-type: none"> • Reglas para programar en fuentes AWL • Crear fuentes AWL • Insertar plantillas de bloques en fuentes AWL • Insertar en fuentes AWL el código de bloques existentes • Comprobar la coherencia en fuentes AWL • Compilar fuentes AWL • Generar fuentes AWL de bloques • Exportar fuentes • Importar fuentes
	Plantilla de segmento	<ul style="list-style-type: none"> • Trabajar con plantillas de segmentos

5.4.8 Programa S7/M7 sin equipo ni CPU

Es posible crear programas sin haber configurado previamente un equipo SIMATIC. Con ello, el usuario puede trabajar sin considerar el módulo a programar ni los ajustes del mismo.

Crear el programa S7/M7

1. Abra el proyecto que corresponda con el comando de menú **Archivo > Abrir** o active la ventana del proyecto.
2. Seleccione el proyecto en la ventana offline del proyecto.
3. Dependiendo del sistema de destino para el que esté creando el programa, elija el comando de menú:
Insertar > Programa > Programa S7, si desea ejecutar su programa en un SIMATIC S7 o
Insertar > Programa > Programa M7, si desea ejecutar su programa en un SIMATIC M7.

El programa S7/M7 se inserta directamente bajo el proyecto. Contiene una carpeta para los bloques y una tabla de símbolos vacía. Ahora puede proceder a crear bloques y programarlos.

Asignación a un módulo programable

Los programas insertados que no estén asignados a ningún módulo pueden ser posteriormente asignados a un módulo en la ventana del proyecto mediante Drag & Drop (Arrastrar y soltar), al copiarlos o arrastrarlos al símbolo del módulo.

Integrar en librerías

Si el programa está destinado al sistema de destino SIMATIC S7 y debe ser reutilizado igual que un "Software-Pool", entonces se puede insertar bajo una librería. Para efectuar un test, los programas se tienen que encontrar bajo un proyecto, ya que sólo entonces es posible establecer un enlace con el sistema de destino.

Acceso a un sistema de destino

Elija la vista online del proyecto. En el diálogo de propiedades del programa puede ajustar las direcciones.

Nota

Al borrar equipos o módulos programables se le preguntará si también desea borrar el programa que contienen. Si responde "No", el programa se insertará directamente bajo el proyecto en calidad de "programa sin equipo".

5.5 Interface de usuario

5.5.1 Filosofía de manejo

Objetivo: manejo sencillo

La interfaz gráfica pretende hacer el manejo lo más intuitivo posible. Por esa razón puede encontrar objetos que ya conoce de mundo laboral cotidiano, como por ejemplo estaciones, módulos, programas o bloques.

Las acciones que tiene que realizar al trabajar con STEP 7 incluyen la creación, la selección y la manipulación de tales objetos.

Diferencias con el manejo por herramientas

Con el manejo convencional mediante herramientas era necesario pensar primero qué herramienta es la más adecuada para resolver una cuestión determinada, y a continuación abrir esa herramienta.

El procedimiento básico en el caso del manejo orientado a los objetos consiste en saber primero qué objeto debe procesarse y a continuación abrir ese objeto y procesarlo.

En el manejo orientado a los objetos ya no se requieren conocimientos sobre una sintaxis de comandos determinada. Los objetos se representan en la interfaz mediante símbolos gráficos que pueden abrirse a través de comandos de menú o haciendo clic con el ratón.

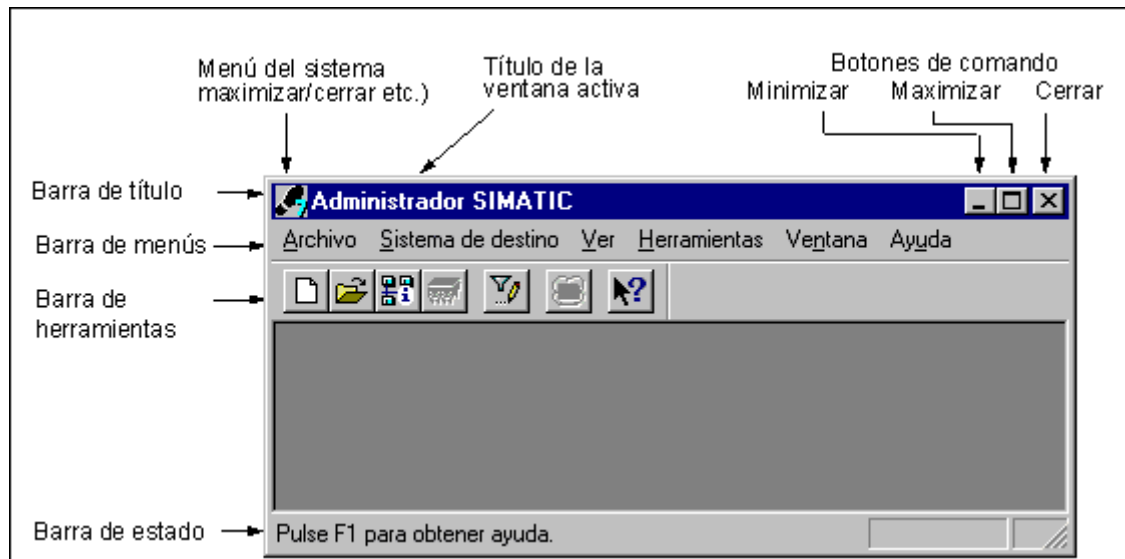
Al abrir un objeto, se abre automáticamente el componente de software necesario para mostrar y procesar el contenido del objeto.

Continúe leyendo ...

A continuación se describen las acciones básicas para el procesamiento de objetos. Familiarícese con ellas ya que en muchas ocasiones se hará referencia a estas instrucciones básicas.

5.5.2 Estructura de la ventana

La figura siguiente muestra los componentes estándar de una ventana:



Barra de título y barra de menús

La barra de título y la barra de menús se encuentran siempre en el borde superior de la ventana. La barra de título contiene el título de la ventana y los botones para modificar el tamaño de la misma y para cerrarla. La barra de menús contiene todos los menús disponibles en la ventana.

Barra de herramientas

La barra de herramientas contiene botones con los que es posible ejecutar rápidamente con un clic del ratón los comandos de menú de uso frecuente que estén disponibles en ese momento. Situando el puntero del ratón unos instantes en un botón, se obtiene breve información sobre su función. Además, en la barra de estado se visualiza una explicación adicional.

Si no es posible acceder a la configuración actual, los botones aparecen atenuados.

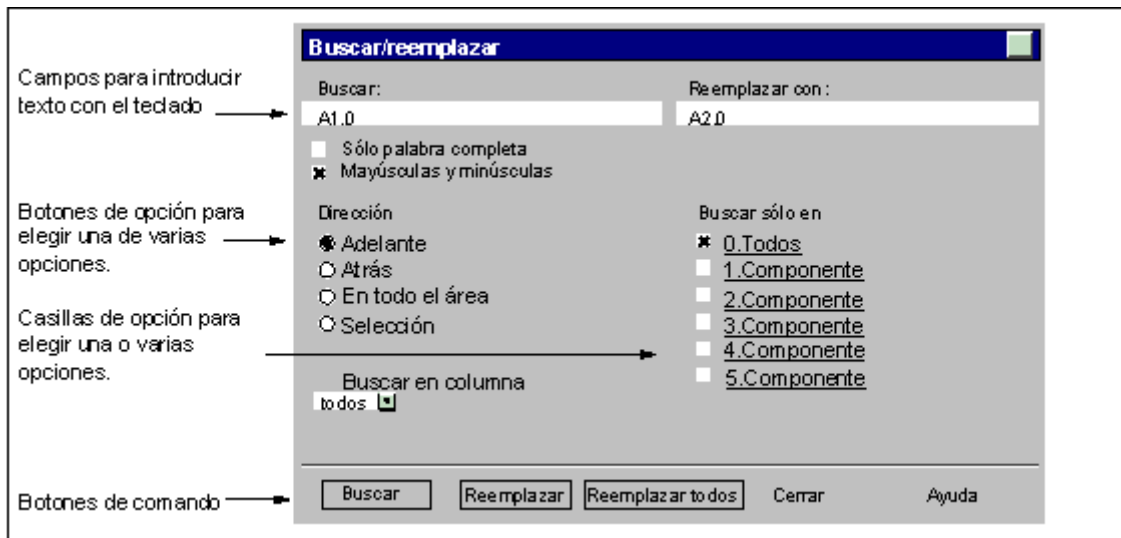
Barra de estado

En la barra de estado se muestran informaciones contextuales.

5.5.3 Elementos de los cuadros de diálogo

Uso de los cuadros de diálogo

En los cuadros de diálogo es posible introducir informaciones necesarias para ejecutar una tarea determinada. Los componentes más frecuentes de los cuadros de diálogo se muestran en forma de ejemplo en la figura siguiente:



Cuadros de lista / cuadros combinados

Algunos cuadros de texto tienen una flecha que señala hacia abajo. Dicha flecha indica que para ese cuadro se dispone de más posibilidades de selección. Haga clic en la flecha para abrir un cuadro de lista o un cuadro combinado. Si hace clic en una de las entradas, ésta será adoptada automáticamente en el cuadro de texto.

Cuadros de diálogo con fichas

Algunos cuadros de diálogo comprenden diversas fichas para facilitar la orientación (v. la figura siguiente).



El título de cada una de las fichas se indica en la pestaña correspondiente. Para que la ficha deseada se visualice en primer plano, haga clic en su título.

5.5.4 Crear y manejar objetos

Algunas funciones básicas son aplicables a todos los objetos. A continuación se ofrece una sinopsis de las mismas. Al describirse procedimientos en los siguientes apartados del manual se da por supuesto que el usuario conoce dichas funciones básicas.

El modo habitual de proceder al manipular objetos es el siguiente:

- crear objetos,
- seleccionar objetos,
- realizar acciones con el objeto (p. ej., copiar, borrar, etc.).

Ajustar la ruta para guardar nuevos proyectos o librerías

Los nuevos proyectos de usuario, librerías o multiproyectos se guardan por defecto en el directorio "\\Siemens\Step7\S7proj". Si desea definir un directorio diferente, debería ajustar la ruta donde se guardarán los nuevos proyectos, librerías o multiproyectos antes de crearlos por primera vez. Elija para ello el comando **Herramientas > Preferencias**. En la ficha "General" del cuadro de diálogo que aparece entonces es posible determinar la ruta donde se han de guardar los nuevos proyectos o librerías.

Crear objetos

El Asistente STEP 7 "Nuevo proyecto" ayuda a crear nuevos proyectos y a insertar objetos. Para llamarlo, elija el comando **Archivo > Asistente 'Nuevo proyecto'**. En los cuadros de diálogo que aparecen entonces puede determinar la estructura de su proyecto y hacerlo crear por el Asistente.

Si no desea utilizar el Asistente, puede crear proyectos y librerías con el comando **Archivo > Nuevo**. Dichos objetos constituyen el nivel superior de una jerarquía. Todos los demás objetos de la jerarquía - a no ser que se creen automáticamente - pueden ser creados utilizando los comandos del menú "Insertar". No obstante, los módulos de un equipo SIMATIC constituyen una excepción, ya que sólo pueden ser generados por el Asistente 'Nuevo Proyecto' al configurarse el hardware.

Abrir objetos

Existen varias alternativas para abrir un objeto en modo de visualización "Detalles":

- hacer doble clic en el icono del objeto o
- seleccionar el objeto y elegir el comando **Edición > Abrir objeto**. Esta alternativa sólo puede ser aplicada a objetos que no sean carpetas.

Una vez abierto el objeto, su contenido se puede editar o modificar.

Una vez abierto un objeto de este último tipo, su contenido se muestra en una nueva ventana para que pueda ser editado. No es posible modificar objetos cuyo contenido esté siendo utilizado.

Nota

Excepción: Los equipos aparecen en forma de carpetas para módulos programables (al hacer doble clic) y para la configuración de equipos. La aplicación para configurar el hardware se arranca bien haciendo doble clic en el objeto "Hardware", o bien seleccionando el equipo y eligiendo el comando de menú **Edición > Abrir objeto**.

Estructurar la jerarquía de objetos

Utilice el Asistente 'Nuevo proyecto' para estructurar el árbol de objetos. Tras abrir una carpeta, se representan en la pantalla los objetos que contiene. Con el menú "Insertar" es posible crear entonces otros subobjetos, como p.ej. otros equipos de un proyecto. En dicho menú se pueden activar sólo aquellos comandos que permiten insertar aquellos objetos que son admisibles en la carpeta actual.

Ajustar las propiedades de objetos

Las propiedades de objetos son atributos que determinan el comportamiento de los mismos. El cuadro de diálogo para ajustar dichas propiedades aparece automáticamente al crear un objeto. No obstante, éstas se pueden también modificar posteriormente.

Con el comando **Edición > Propiedades del objeto** se llama un cuadro de diálogo donde se pueden visualizar o ajustar las propiedades del objeto seleccionado.

Con el comando **Edición > Propiedades especiales del objeto** es posible llamar diversos cuadros de diálogo e introducir los datos necesarios para manejo y visualización (M+V) y para configurar avisos.

Por ejemplo, para llamar las propiedades especiales de un bloque para manejo y visualización, dicho bloque se deberá haber preparado para M+V. Es decir, en la ficha "Atributos" de las propiedades del bloque es preciso ajustar el atributo "S7_m_c" con el valor "true".

Nota

- Las propiedades de la carpeta "Datos de sistema" y del objeto "Hardware" no se pueden visualizar ni modificar.
 - Los cuadros de diálogo de propiedades de los objetos de un proyecto protegido contra escritura no se pueden editar. En este caso, los campos de entrada aparecen atenuados.
 - Si desea visualizar las propiedades de los módulos programables, los parámetros visualizados no se podrán editar por motivos de coherencia. Para poder editar los parámetros deberá llamar la aplicación "Configurar hardware".
 - Si se modifican en el sistema de origen los ajustes realizados en objetos (p.ej. los datos de parametrización de un módulo), estos cambios no tendrán efecto en el sistema de destino. Para que tengan efecto, los bloques de datos de sistema en los que se guardan estos ajustes tienen que encontrarse en el sistema de destino.
 - Si se carga un programa de usuario entero, los bloques de datos de sistema se transfieren automáticamente. Si, tras haber cargado el programa, se modifican los ajustes, puede volver a cargar el objeto "Datos de sistema" para transferir los ajustes al sistema de destino.
 - Se recomienda editar la carpeta únicamente con STEP 7, ya que ésta puede tener una estructura física diferente de la que se ve en el Administrador SIMATIC.
-

Cortar, pegar y copiar

La mayoría de los objetos se pueden cortar, pegar o copiar siguiendo el procedimiento habitual en Windows. Los comandos correspondientes se encuentran en el menú "Edición".

Asimismo, con "arrastrar y soltar" (drag & drop) es posible copiar objetos. Si se señala un destino no admisible, aparecerá un cursor de prohibición.

Al copiar un objeto se copian también todos los niveles subordinados que contiene, lo que permite reutilizar componentes que se hayan creado en otra solución de automatización.

Nota

La tabla de enlaces en la carpeta "Enlaces" no se puede copiar. Tenga en cuenta que al copiar listas de textos importantes para el usuario sólo se adoptarán los idiomas instalados en el objeto de destino.

El tema Copiar objetos contiene instrucciones paso a paso al respecto.

Cambiar el nombre de objetos

El Administrador SIMATIC asigna nombres estandarizados a algunos objetos nuevos que se hayan insertado. Esos nombres resultan por lo general del tipo de objeto y, si se pueden crear varios objetos del mismo tipo en una misma carpeta, de un número correlativo.

Por ejemplo, el primer programa S7 se llamará "Programa S7(1)" y el segundo, "Programa S7(2)". En cambio, la tabla de símbolos se llama sólo "Símbolos", puesto que existe sólo una vez en cada carpeta de orden superior.

Si lo desea, puede cambiar los nombres de la mayoría de los objetos por denominaciones que se autoexplicativas.

En los proyectos, los nombres de los directorios indicados en la ruta pueden comprender 8 caracteres como máximo. De lo contrario pueden surgir problemas al archivar y al emplear "C para M7" (compilador Borland).

Para cambiar el nombre de un objeto (en caso de que se pueda modificar) puede elegir una de las siguientes posibilidades:

- Edición directa:
Haga clic dos veces lentamente en el nombre de un objeto seleccionado en la ventana del proyecto. Aparecerá un marco alrededor del texto. Entonces podrá introducir el nombre utilizando el teclado.
- Con el comando "Propiedades del objeto":
En la ventana del proyecto, seleccione el objeto deseado y ejecute comando **Edición > Cambiar nombre**. Aparecerá un marco al rededor del texto. Ahora puede editar el nombre mediante el teclado.

Si no es posible cambiar el nombre de un objeto, aparecerá atenuado el campo de entrada en el cuadro de diálogo. Entonces se visualizará el nombre actual, siendo imposible introducir texto.

Nota

Si, durante la edición, el puntero del ratón se desplaza fuera del campo del nombre para realizar otra acción (p.ej. para elegir un comando de menú), el proceso de edición finalizará. El nuevo nombre será adoptado en caso de que sea admisible.

En el tema Cambiar el nombre de objetos encontrará instrucciones detalladas al respecto.

Mover objetos

Con el Administrador SIMATIC se puede desplazar los objetos de una carpeta a otra, aunque la carpeta de destino forme parte de un proyecto diferente. Al desplazar una carpeta se traslada también todo su contenido.

Nota

Los siguientes objetos no se pueden desplazar:

- Enlaces
 - Bloques de datos de sistema (SDB) en la vista online
 - Funciones de sistema (SFC) y bloques de función de sistema (SFB) en la vista online
-

El tema Desplazar objetos contiene instrucciones detalladas al respecto.

Ordenar objetos

En el modo de visualización "Detalles" (comando de menú **Ver > Detalles**), los objetos se pueden ordenar según sus atributos. A tal efecto, haga clic en el título del atributo deseado. Haciendo nuevamente clic se invierte la secuencia de ordenamiento. Los bloques se ordenan conforme a su complemento numérico, p.ej. FB 1, FB 2, FB 11, FB 12, FB 21, FC 1.

Secuencia de ordenamiento preajustada:

Cuando se abre un proyecto, los objetos se abren en el modo de visualización "Detalles" conforme a la secuencia de ordenamiento preajustada. Ejemplo:

- Los bloques se representan en el siguiente orden: "Datos de sistema, OB, FB, FC, DB, UDT, VAT, SFB, SFC".
- En el proyecto se visualizan primero todos los equipos y luego los programas S7.

Por tanto, el ajuste estándar del modo de visualización "Detalles" no es una secuencia alfanumérica en orden ascendente o descendente.

Restablecer la secuencia de ordenamiento preajustada:

Tras haberse cambiado el orden, p.ej. haciendo clic en el título de la columna "Nombre del objeto", se puede restablecer la secuencia de ordenamiento preajustada. Proceda a tal efecto de la siguiente forma:

- En el modo de visualización "Detalles", haga clic en el título de la columna "Tipo".
- Cierre el proyecto y ábralo de nuevo.

Borrar objetos

Tanto las carpetas como los objetos se pueden borrar. Si borra una carpeta se borrarán también todos los objetos contenidos en ella.

El proceso de borrado es irreversible. Si no está completamente seguro de que ya no necesita un objeto, es preferible que archive previamente el proyecto entero.

Nota

Los siguientes objetos no se pueden borrar:

- Enlaces
 - Bloques de datos de sistema (SDB) en la vista online
 - Funciones de sistema (SFC) y bloques de función de sistema (SFB) en la vista online
-

El tema Borrar objetos contiene instrucciones paso a paso al respecto.

5.5.5 Elegir objetos en cuadros de diálogo

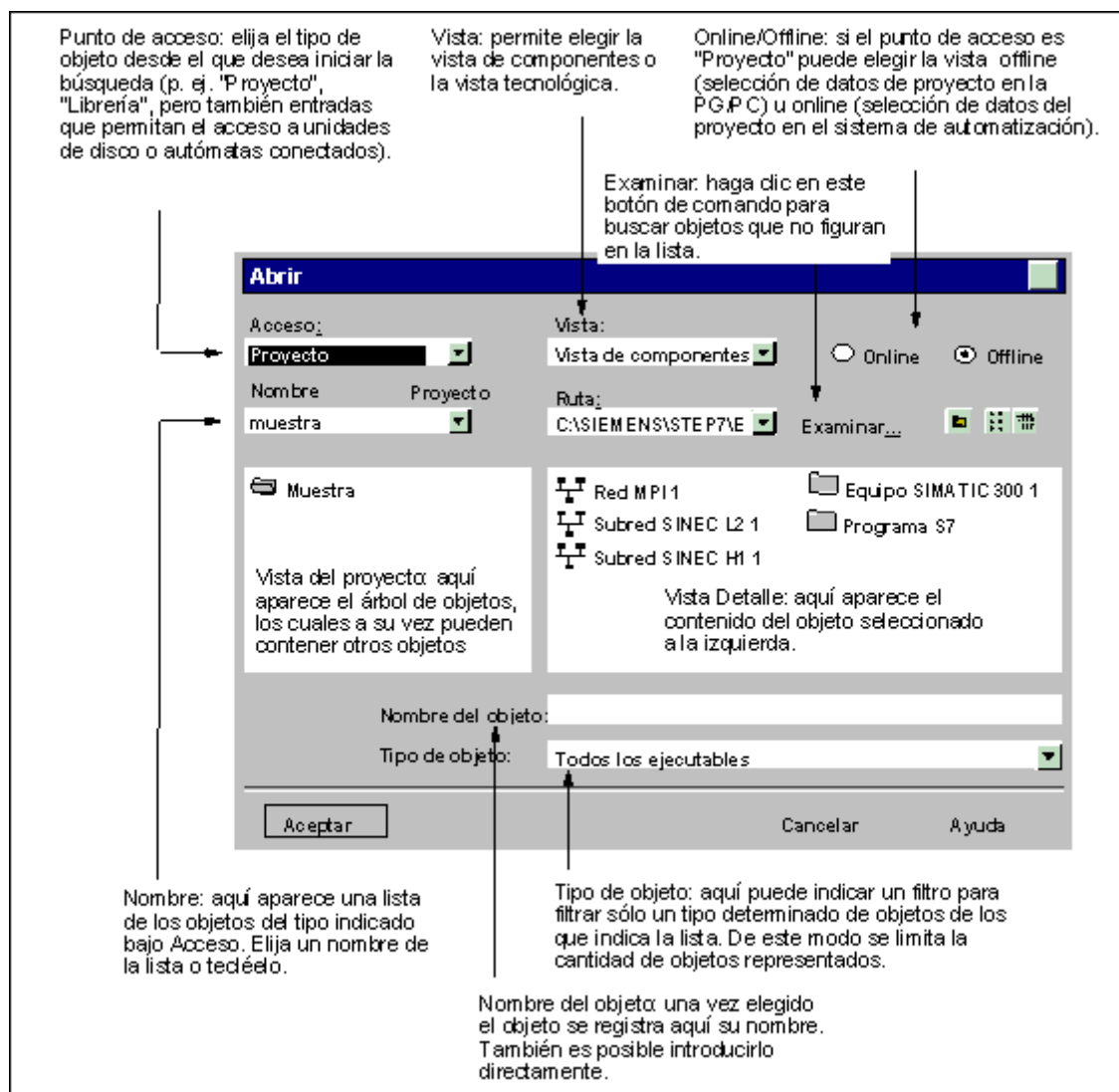
En diversas situaciones es necesario elegir objetos en un cuadro de diálogo.

Llamar el cuadro de diálogo

El cuadro de diálogo se llama, como p.ej. en el caso de la configuración del hardware, mediante comandos de menú tales como **Equipo > Nuevo.../Abrir...** (a excepción de la ventana de acceso "Administrador SIMATIC").

Estructura del cuadro de diálogo

La figura siguiente muestra las opciones del cuadro de diálogo.



5.5.6 Memoria de sesión

El Administrador SIMATIC memoriza el contenido de la ventana, es decir, los proyectos y librerías que están abiertos así como la disposición de las ventanas.

- Con el comando de menú **Herramientas > Preferencias** puede hacer que se memorice el contenido y la disposición de las ventanas al finalizar la sesión. Cuando empiece la siguiente sesión, la ventana se abrirá con el mismo contenido y la misma disposición. En los proyectos abiertos, el cursor se posiciona en la última carpeta que se abrió.
- Con el comando de menú **Ventana > Guardar organización** se guarda el contenido actual de la ventana así como su disposición.
- Con el comando de menú **Ventana > Restablecer organización** se restablece el contenido y la disposición de la ventana que se guardó con el comando **Ventana > Guardar organización** . En los proyectos abiertos el cursor se posiciona en la última carpeta que se seleccionó.

Nota

El contenido de las ventanas de proyectos online, así como de la ventana "Estaciones accesibles" y el de la ventana "Memory Card S7" no se memoriza.

Las posibles contraseñas que haya introducido para limitar el acceso a los sistemas de destino (S7-300/S7-400) no se memorizan para otras sesiones .

5.5.7 Cambiar la organización de las ventanas

Para disponer todas las ventanas una tras otra con los títulos visibles, dispone de las siguientes posibilidades:

- Elija el comando de menú **Ventana > Organizar > Cascada**.
- Teclee MAYUS + F5.

Para disponer todas las ventanas una bajo la otra elija el comando de menú **Ventana > Organizar > Mosaico horizontal**.

Para disponer todas las ventanas una junto a la otra elija el comando de menú **Ventana > Organizar > Mosaico vertical**.

5.5.8 Guardar y restablecer la organización de las ventanas

Las aplicaciones de STEP 7 permiten guardar y restablecer posteriormente la organización actual de las ventanas. El ajuste se puede efectuar con los comandos de menú **Herramientas > Preferencias > ficha "General"**.

¿Qué se guarda?

Al guardar la organización de las ventanas se almacenan las siguientes informaciones:

- Posición de la ventana principal
- Proyectos y librerías abiertos y las correspondientes posiciones de las ventanas
- Orden de las ventanas que pudieran estar superpuestas

Nota

No se guarda el contenido de las ventanas de proyectos online, ni tampoco el de las ventanas "Estaciones accesibles" y "Memory Card S7".

Guardar la organización

Para guardar la organización actual de las ventanas, elija el comando **Ventana > Guardar organización**.

Restablecer la organización

Para restablecer las ventanas como estaban dispuestas previamente, elija el comando **Ventana > Restablecer organización**.

Nota

Al restablecerse una ventana, se representa detalladamente sólo la parte de la jerarquía donde se encuentra el objeto que estaba seleccionado al guardar la organización.

5.6 Manejo de las teclas

Denominación internacional de las teclas	Denominación española de las teclas
Tecla HOME	Tecla INICIO
Tecla END	Tecla FIN
Tecla PAGE-UP	Tecla AvPág
Tecla PAGE-DOWN	Tecla RePág
Tecla CTRL	Tecla CTRL
Tecla ENTER	Tecla INTRO
Tecla DEL	Tecla SUPR
Tecla INSERT	Tecla INSERT

5.6.1 Combinaciones de teclas para comandos de menú

Se puede activar cualquier comando de menú pulsando la siguiente combinación: ALT y la(s) tecla(s) correspondiente(s).

Pulse las teclas indicadas en el orden siguiente:

- Tecla ALT
- La letra que aparece subrayada en el menú deseado (p.ej. ALT+A para el menú "Archivo" - en el caso de que el menú "Archivo" aparezca en el menú). El menú se desplegará.
- La letra que aparece subrayada en el comando de menú deseado (p.ej. N para el comando "Nuevo"). Si se trata de un comando que comprenda submenús, éstos se desplegarán entonces. Proceda de igual forma hasta que haya seleccionado el comando de menú completo introduciendo las letras correspondientes.

Una vez introducida la última letra de la combinación de teclas se ejecutará el comando de menú.

Ejemplo:

Comandos de menú Teclas

Archivo > Archivar ALT, A, A

Archivo > Abrir ALT, A, A

Teclas de método abreviado para comandos de menú

Función	Teclas de método abreviado
Nuevo (menú "Archivo")	CTRL+N
Abrir (menú "Archivo")	CTRL+O
Guardar como (menú "Archivo")	CTRL+S
Imprimir > Lista de objetos (menú "Archivo ")	CTRL+P
Imprimir > Objeto (menú "Archivo ")	CTRL+ALT+P
Salir (menú "Archivo")	ALT+F4
Cortar (menú "Edición")	CTRL+X
Copiar (menú "Edición")	CTRL+C
Pegar (menú "Edición")	CTRL+V
Borrar (menú "Edición")	SUPR
Seleccionar todo (menú "Edición")	CTRL+A
Cambiar nombre (menú "Edición ")	F2
Propiedades del objeto (menú "Edición")	ALT+INTRO
Abrir objeto (menú "Edición")	CTRL+ALT+O
Compilar (menú "Edición")	CTRL+B
Cargar (menú "Sistema de destino")	CTRL+L
Diagnóstico/Preferencias > Información del módulo (menú "Sistema de destino")	CTRL+D
Diagnóstico/Preferencias Estado operativo (menú "Sistema de destino")	CTRL+I
Actualizar (menú "Ver")	F5
actualiza la visualización del estado de las CPUs visibles en la vista online	CTRL+F5
Preferencias (menú "Herramientas")	CTRL+ALT+E
Datos de referencia > Mostrar (menú "Herramientas")	CTRL+ALT+R
Organizar > Cascada (menú "Ventana")	MAYÚS + F5
Organizar > Mosaico horizontal (menú "Ventana")	MAYÚS + F2
Organizar > Mosaico vertical (menú "Ventana")	MAYÚS + F3
Ayuda contextual (menú "Ayuda")	F1 (Si existe un contexto actual, p.ej., un comando de menú seleccionado, se llama al tema de Ayuda correspondiente. En caso contrario, al índice de la Ayuda.)

5.6.2 Combinaciones de teclas para mover el cursor

Desplazar el cursor en la barra de menús / en el menú contextual

Función	Teclas
A la barra de menús.	F10
Al menú contextual.	MAYUS + F10
Al menú que contiene la letra X subrayada.	ALT+X
Comando de menú asociado	Letra subrayada en el comando de menú
Un comando de menú a la izquierda.	FLECHA IZQUIERDA
Un comando de menú a la derecha.	FLECHA DERECHA
Un comando de menú hacia arriba.	FLECHA ARRIBA
Un comando de menú hacia abajo.	FLECHA ABAJO
Activar el comando de menú seleccionado.	TECLA INTRO
Abandonar el menú y volver al texto.	ESC

Desplazar el cursor al editar textos

Función	Teclas
Una línea hacia arriba o un carácter a la izquierda en un texto que se componga de una sola línea.	FLECHA ARRIBA
Una línea hacia abajo o un carácter a la derecha en un texto que se componga de una sola línea.	FLECHA ABAJO
Un carácter a la derecha.	FLECHA DERECHA
Un carácter a la izquierda.	FLECHA IZQUIERDA
Una palabra a la derecha.	CTRL+FLECHA DERECHA
Una palabra a la izquierda.	CTRL+FLECHA IZQUIERDA
Al comienzo de la línea.	INICIO
Al final de la línea.	FIN
Una pantalla hacia arriba.	RE PAG
Una pantalla hacia abajo.	AV PAG
Al comienzo del texto.	CTRL+INICIO
Al final del texto.	CTRL+FIN

Desplazar el cursor al editar tablas

Función	Teclas
Una línea hacia arriba	FLECHA ARRIBA
Una línea hacia abajo	FLECHA ABAJO
Un carácter o una línea hacia la izquierda	FLECHA DERECHA
Un carácter o una línea hacia la derecha	FLECHA IZQUIERDA
Al inicio de la línea	CTRL+FLECHA DERECHA
Al final de la línea	CTRL+FLECHA IZQUIERDA
Al inicio de la línea	INICIO
Al final de la línea	FIN
Una pantalla hacia arriba	RE PÁG
Una pantalla hacia abajo	AV PÁG
Al inicio de la tabla	CTRL+INICIO
Al final de la tabla	CTRL+FIN
Sólo en la tabla de símbolos: A la columna Símbolo	MAYÚSCULAS+INICIO
Sólo en la tabla de símbolos: A la columna Comentario	MAYÚSCULAS+FIN

Desplazar el cursor en cuadros de diálogo

Función	Teclas
Al siguiente cuadro de entrada (de izquierda a derecha y de arriba hacia abajo).	TAB
Retrocede un cuadro de entrada.	Mayusc+TAB
Al cuadro de entrada que contiene la letra X subrayada.	ALT+X
Selecciona en la lista de selección	TECLAS DE CURSOR
Abre una lista de selección	ALT+FLECHA ABAJO
Activa y desactiva la selección de un objeto	BARRA ESPACIADORA
Confirma las entradas efectuadas y cierra el cuadro de diálogo (botón "Aceptar").	TECLA INTRO
Cierra el cuadro de diálogo sin guardar la selección (botón "Cancelar").	ESC

5.6.3 Combinaciones de teclas para seleccionar textos

Función	Teclas
Un carácter a la derecha.	MAYÚS+FLECHA DERECHA
Un carácter a la izquierda.	MAYÚS+FLECHA IZQUIERDA
Al comienzo de la línea de comentario	MAYUS+INICIO
Al final de la línea de comentario	MAYUS+FIN
Una línea en una tabla	MAYUS+ESPACIO
Una línea arriba.	MAYÚS+FLECHA ARRIBA
Una línea abajo.	MAYÚS+FLECHA ABAJO
Una pantalla hacia arriba.	MAYUS+REPAG
Una pantalla hacia abajo.	MAYUS+AVPAG
Principio de archivo.	CTRL + MAYUS + INICIO
Fin de archivo.	CTRL+MAYUS+FIN

5.6.4 Combinaciones de teclas para acceder a la Ayuda en pantalla

Función	Teclas
Abre la Ayuda.	F1 (Si existe un contexto actual, p.ej., un comando de menú seleccionado, se llama al tema de Ayuda correspondiente. En caso contrario, al índice de la Ayuda.)
Activa el icono de interrogación para la ayuda contextual	MAYUS+F1
Cierra la ventana de ayuda, vuelve a la ventana de trabajo	ALT + F4

5.6.5 Combinaciones de teclas para cambiar de una ventana a otra

Función	Teclas
Cambiar de una sección de ventana a otra	F6
Retornar a la sección de ventana anterior si no existe ninguna ventana acoplable	MAYÚS + F6
Cambiar de la ventana de documento a la ventana acoplable del documento (p.ej. a la ventana de declaración de variables) Si no existe una ventana acoplable se retorna a la sección de ventana anterior.	MAYÚS + F6
Cambiar de una ventana de documento a otra	CTRL + F6
Retornar a la ventana de documento anterior	MAYÚS + CTRL + F6
Cambiar entre ventanas que no sean de documento (marco de la aplicación y ventanas acoplables del mismo; al retornar al marco se cambia a la última ventana activa del documento)	ALT+F6
Retornar a la ventana precedente que no sea de documento	MAYÚS + ALT + F6
Cerrar la ventana activa	CTRL + F4

6 Elaboración de proyectos

6.1 Estructura de los proyectos

Los proyectos sirven para almacenar de forma ordenada los datos y programas necesarios para crear una tarea de automatización. Los datos que conforman un proyecto comprenden principalmente:

- los datos para configurar el hardware y parametrizar los módulos,
- los datos para configurar la comunicación por redes y
- los programas para los módulos programables.

Al crear un proyecto el objetivo principal consiste en programar y en poner a disposición dichos datos.

En un proyecto, los datos se depositan en forma de objetos. Dentro de un proyecto, los objetos están estructurados en forma de árbol (jerarquía del proyecto). La representación de la jerarquía en la ventana del proyecto es similar a la del Explorador de Windows. Sólo los iconos de los objetos tienen un aspecto diferente.

El nivel superior de la jerarquía de los proyectos está estructurado de la siguiente forma:

1. Nivel : Proyecto
2. Nivel : Subredes, equipos o programas S7/M7
3. Nivel: Dependiendo del objeto que se encuentre en el segundo nivel.

Ventana del proyecto

La ventana del proyecto se divide en dos partes. En la mitad izquierda se representa la estructura en árbol del proyecto. En la mitad derecha aparece el contenido del objeto seleccionado a la izquierda, conforme a la visualización elegida (iconos grandes, iconos pequeños, lista o detalles) .

Para visualizar la estructura completa del proyecto en la mitad izquierda de la ventana, haga clic en la casilla "+". Aparecerá entonces una representación similar a la que muestra la figura siguiente.



A la cabeza de la jerarquía de objetos se encuentra el objeto "S7_Pro1", que es el icono del proyecto entero. Se puede utilizar para visualizar las propiedades del objeto y sirve de carpeta de redes (para configurar redes), de equipos (para configurar el hardware) y de programas S7 o M7 (para crear el software). Si se selecciona el icono del proyecto, los objetos contenidos allí se visualizan en la mitad derecha de la ventana. Los objetos que encabezan la jerarquía (que pueden ser tanto proyectos como librerías) permiten acceder a los cuadros de diálogo para seleccionar objetos.

Vista del proyecto

Puede visualizar en ventanas la estructura de proyectos para el contingente de datos del sistema de origen con el modo "offline", y la del correspondiente contingente de datos del sistema de destino con el modo "online".

Si está cargado el correspondiente paquete opcional, podrá disponer también de la vista del Administrador de instalaciones.

Nota

La configuración del hardware y de las redes sólo se puede llevar a cabo en el modo de visualización "offline".

6.2 Información importante sobre la protección de acceso

A partir de STEP 7 V5.4 existe la posibilidad de crear una protección de acceso para proyectos y librerías asignando una contraseña de proyecto. Sin embargo, para poder utilizar esta función es necesario tener instalado SIMATIC Logon.

Además, también es posible activar, desactivar y visualizar un informe de cambios.

Si su ordenador tiene instalado SIMATIC Logon, en el SIMATIC Manager aparecerán de forma dinámica los comandos siguientes, los cuales le permitirán gestionar la protección de acceso a un proyecto o una librería:

- Protección de acceso, Activar
- Protección de acceso, Desactivar
- Protección de acceso, Administrar usuarios
- Protección de acceso, Sincronizar en el multiproyecto
- Eliminar la protección de acceso y el informe de cambios

La protección de acceso se activa en el SIMATIC Manager con el comando de menú **Herramientas > Protección de acceso, Activar**. Si activa la protección de acceso con el comando de menú arriba indicado, se abre un cuadro de diálogo en el que podrá asignar la contraseña del proyecto. En dicho caso, el proyecto o la librería correspondientes podrán editarse únicamente introduciendo dicha contraseña.

Con el comando **Eliminar protección de acceso e informe de cambios** se elimina tanto la protección de acceso como el informe de cambios de un proyecto o librería que estén protegidos por contraseña. Después de eliminar la protección de acceso pueden reeditarse proyectos con una versión de STEP 7 anterior a la V5.4.

Notas

- Para poder activar o desactivar la protección de acceso, es preciso estar autenticado en SIMATIC Logon como administrador del proyecto.
 - Al activar la protección de acceso por primera vez se modifica el formato del proyecto. Aparece un aviso indicando que el proyecto modificado ya no puede ser editado con versiones anteriores de STEP 7.
 - La función **Herramientas > Protección de acceso > Eliminar protección de acceso e informe de cambios** permite editar nuevamente el proyecto o la librería con una versión de STEP 7 anterior a la V5.4. Sin embargo, perderá información sobre los usuarios que tienen acceso a este proyecto o librería así como todos los informes de cambios.
 - El usuario que está dado de alta en esos momentos se visualiza en la barra de estado del SIMATIC Manager.
 - El usuario del proyecto que está dado de alta en esos momentos y que activa la protección de acceso se registra como administrador del proyecto y se requiere asignar la contraseña del proyecto.
 - Para poder abrir un proyecto protegido es preciso estar autenticado en el SIMATIC Logon como administrador del proyecto o como editor del proyecto y la contraseña del proyecto debe estar activada o conoce la contraseña del proyecto.
-

6.3 Información importante sobre el informe de cambios

A partir de STEP 7 V5.4, después de crear una protección de acceso para proyectos y librerías existe la posibilidad de crear un informe de cambios que registre acciones online.

Ejemplos:

- Activación / desactivación / configuración de la protección de acceso y del informe de cambios
- Abrir / Cerrar proyectos y librerías
- Cargar en el sistema de destino (datos de sistema)
- Operaciones seleccionadas para cargar y copiar bloques
- Acciones para cambiar el estado operativo
- Borrado total

El informe de cambios se puede visualizar y en él también se pueden introducir los motivos del cambio en cuestión así como comentarios. Esta función requiere la instalación de SIMATIC Logon.

El informe de cambios se activa en el SIMATIC Manager con el comando **Herramientas > Informe de cambios, Activar**. Una vez activado el informe de cambios puede visualizarlo con el y volverlo a desactivar con el respectivo comando.

Dependiendo del objeto que haya seleccionado en la estructura del proyecto (p. ej. la carpeta del proyecto o un equipo incluido en la misma), se mostrará el informe de cambios correspondiente.

Notas

- La función **Herramientas > Protección de acceso > Eliminar protección de acceso e informe de cambios** permite editar nuevamente el proyecto o la librería con una versión de STEP 7 anterior a la V5.4. Sin embargo, perderá información sobre los usuarios que tienen acceso a este proyecto o librería así como todos los informes de cambios.
 - Para poder ejecutar esta función es preciso estar autenticado como administrador del proyecto en el SIMATIC Logon Admin Tool y la protección de acceso de dicho proyecto debe estar activada.
-

6.4 Utilizar juegos de caracteres de otros idiomas

A partir de STEP 7 V5.3 SP2 es posible introducir en proyectos y librerías textos en otros idiomas que no se correspondan con el idioma ajustado para STEP 7. Para ello es necesario ajustar el idioma de Windows correspondiente en el panel de control del sistema operativo. Así por ejemplo, en un Windows chino es posible utilizar STEP 7 en el idioma inglés de STEP 7, y no obstante introducir textos chinos.

Cabe distinguir las siguientes posibilidades de ajuste del idioma:

Ajuste del idioma de Windows

Este ajuste del idioma se efectúa en el panel de control de Windows. En el idioma seleccionado se visualizan textos del sistema operativo y además se pueden introducir textos en juegos de caracteres de otros idiomas.

Idioma del proyecto

El idioma del proyecto es el idioma que está ajustado en el panel de control de Windows al crear un proyecto. No es posible cambiar el idioma del proyecto por otro idioma. No obstante, con el ajuste "independiente del idioma" es posible abrir un proyecto en ordenadores que tengan ajustado otro idioma de Windows. Antes de marcar el idioma del proyecto como "independiente del proyecto", asegúrese de que al introducir textos en el proyecto se hayan utilizado exclusivamente caracteres del juego de caracteres inglés (caracteres ASCII 0x2a - 0x7f).

Para saber en qué idioma ha sido creado un proyecto o una librería, ejecute el comando de menú **Edición > Propiedades del objeto**. En el mismo cuadro de diálogo también se puede seleccionar la opción "Se puede abrir con cualquier idioma ajustado en Windows (independiente del idioma)".

Si copia un proyecto con el comando de menú **Guardar como** y el idioma del proyecto no coincide con el idioma actualmente ajustado en Windows, puede cambiar el idioma del proyecto en el proyecto copiado por el idioma actualmente ajustado en Windows. Esto es especialmente útil para crear variantes del proyecto específicas del idioma. En este caso, el proyecto maestro deberá contener exclusivamente caracteres del juego de caracteres inglés (caracteres ASCII 0x2a - 0x7f), de manera que no se puedan corromper datos al editar el proyecto en el idioma en cuestión.

Idioma de STEP 7

El idioma de STEP 7 se ajusta en el Administrador SIMATIC con el comando de menú **Herramientas > Preferencias**. En este idioma se visualizan los elementos de la interfaz de usuario, los comandos de menú, cuadros de diálogo y mensajes de error de STEP 7.

Si está utilizando otro idioma de Windows que no sea alemán, inglés, francés, italiano o español y desea asegurarse de que el interface de usuario se representará correctamente, elija inglés como idioma de STEP 7.

Reglas

Si pretende editar sus proyectos o librerías en ordenadores que tengan ajustados idiomas diferentes, tenga en cuenta las reglas que se indican a continuación para evitar incompatibilidades o que se corrompan datos al utilizar juegos de caracteres de otros idiomas:

- Instale STEP 7 únicamente en directorios que contengan caracteres del juego de caracteres inglés (caracteres ASCII 0x2a - 0x7f).
- Utilice solamente nombres y rutas de proyectos que contengan caracteres del juego de caracteres inglés (caracteres ASCII 0x2a - 0x7f). Si utiliza p. ej. diéresis alemanas o caracteres cirílicos o chinos, sólo podrá abrir el proyecto en ordenadores en los que se haya ajustado el idioma correspondiente en Windows.
- Utilice en los multiproyectos únicamente proyectos y librerías que tengan el mismo idioma de proyecto o únicamente sólo los que estén marcados como independientes del idioma.
- Cree solamente librerías que sean siempre independientes del idioma, para asegurarse de que pueden ser utilizadas en ordenadores con diferentes idiomas de Windows. Al asignar el nombre de los objetos de librería, en comentarios, nombres simbólicos etc. utilice solamente caracteres ASCII (0x2a - 0x7f), para que las librerías puedan ser utilizadas sin problemas.
- Al importar o exportar configuraciones de hardware o tablas de símbolos, vigile que los archivos que importe o exporte sean compatibles en lo que respecta a su idioma.
- En los nombres de atributos definidos por el usuario utilice solamente los caracteres del juego de caracteres inglés (caracteres ASCII 0x2a - 0x7f).
- Si en una fuente AWL utiliza para las propiedades de bloques TITLE, AUTHOR, FAMILY o NAME caracteres que no están incluidos en el juego de caracteres inglés (caracteres ASCII 0x2a - 0x7f), ponga dichas entradas entre comillas simples.

Notas

- Si modifica o copia proyectos o librerías que hayan sido creados en un ordenador, que si bien es independiente del idioma en lo que respecta al ajuste del idioma de Windows, no es compatible con el ordenador actual, pueden corromperse datos, si en el proyecto o en la librería se han utilizado caracteres que no están contenidos en el juego de caracteres inglés (caracteres ASCII 0x2a – 0x7f).
Por consiguiente, antes de editar proyectos o librerías "ajenos" asegúrese de que el idioma ajustado para Windows en su ordenador se corresponda con el idioma del proyecto.
 - Si va a exportar configuraciones de hardware o tablas de símbolos que deban ser importadas con otro idioma de Windows, asegúrese de que solamente se han utilizado caracteres ASCII (0x2a - 0x7f) y ningún carácter específico del idioma, como p. ej. diéresis alemanas o caracteres japoneses o cirílicos.
 - Las configuraciones de hardware exportadas o las tablas de símbolos que contienen caracteres específicos del idioma como p. ej. diéresis alemanas o caracteres japoneses o cirílicos, solo pueden ser importados en el idioma de Windows, en el que fueron exportados. Si p. ej. importa tablas de símbolos antiguas que contengan caracteres específicos del idioma, compruebe detenidamente el resultado: los símbolos deben ser inequívocos, no deb en contener signos de interrogación o marcas y deben ser pausibles.
 - En las tablas de símbolos con caracteres especiales que no están definidos en el idioma de Windows ajustado, pueden modificarse los nombres simbólicos y pueden aparecer errores al ordenar por nombres y comentarios en caso de haber utilizado signos de interrogación o marcas.
 - Tenga en cuenta que en el direccionamiento simbólico deben escribirse los nombres simbólicos entre comillas ("`<nombre simbólico>`").
-

Procedimiento básico

Para introducir textos en juegos de caracteres de otros idiomas en proyectos y librerías, proceda del siguiente modo:

1. Vaya al panel de control y ajuste el idioma de Windows deseado.
2. Cree un proyecto.
3. Introduzca los textos deseados en otro idioma.

En los proyectos y librerías que hayan sido creados con versiones anteriores a STEP 7 V5.3 SP2, el idioma del proyecto está "todavía sin establecer". En estos proyectos se puede cambiar el idioma del proyecto con el comando de menú **Edición > Propiedades del objeto** por el idioma de Windows actualmente visualizado y ajustado. Asegúrese previamente de que no se hayan utilizado caracteres que no estén definidos en el idioma actualmente ajustado en Windows.

6.5 Ajustar el idioma de Windows

Para ajustar el idioma de Windows, proceda del siguiente modo:

Ajuste del idioma bajo Windows XP y Windows Server 2003:

1. Vaya a Panel de control > Configuración regional y de idioma > Opciones avanzadas > Idioma para programas no Unicode y ajuste el idioma de visualización deseado.
2. Vaya al **Panel de control > Opciones regionales, de idioma y de fecha y hora > Agregar otros idiomas** y en el cuadro de diálogo **Configuración regional y de idioma elija Idiomas > Detalles** y seleccione el idioma de entrada (configuración regional predeterminada).
3. Vaya al **Panel de control > Opciones regionales, de idioma y de fecha y hora > Opciones regionales (Estándares y formatos)** y seleccione el idioma.

Tan solo cuando haya realizado todos los ajustes, podrá introducir y visualizar correctamente los textos en el idioma deseado.

Ajuste del idioma bajo Windows 2000:

1. Vaya a **Panel de control > Configuración regional > General > Su idioma (ubicación)** y ajuste el idioma deseado.
2. Vaya a **Panel de control > Configuración regional > General > Predeterminada** y ajuste el idioma de visualización deseado.
3. Vaya a **Panel de control > Configuración regional > Idiomas** y ajuste el idioma de entrada (configuración regional del sistema).

Sólo cuando haya realizado todos los ajustes, podrá introducir y visualizar correctamente los textos en el idioma deseado.

6.6 Crear proyectos

6.6.1 Crear un proyecto

Para poder solucionar su tarea de automatización a nivel de gestión de un proyecto deberá crear primero un nuevo proyecto. El nuevo proyecto se crea en el directorio que haya indicado en la ficha "General", una vez elegido el comando **Herramientas > Preferencias**.

Nota

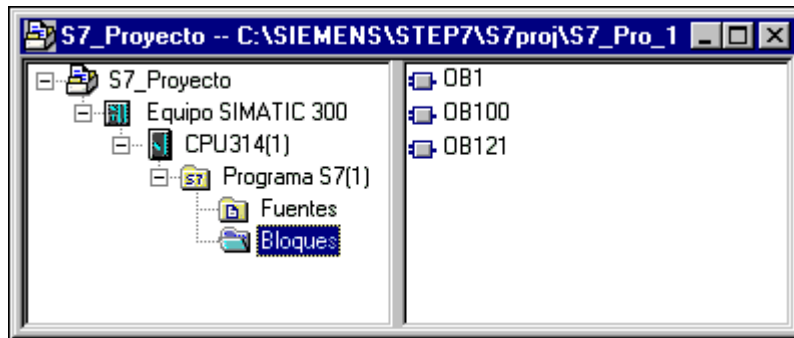
El Administrador SIMATIC permite utilizar nombres de más de 8 letras. Sin embargo, el nombre del directorio del proyecto queda limitado a 8 caracteres, por lo que resulta imprescindible que los nombres de los proyectos se distingan ya en los primeros 8 caracteres. Entre mayúsculas y minúsculas no se hace distinción alguna.

Los temas Crear un proyecto manualmente o Crear un proyecto utilizando el Asistente contienen instrucciones detalladas para crear un proyecto.

Crear un proyecto utilizando el Asistente

La manera más fácil de crear un nuevo proyecto es utilizando el Asistente 'Nuevo proyecto'. Para llamarlo, elija el comando **Archivo > Asistente 'Nuevo proyecto'**. El Asistente consulta los datos necesarios en diversos cuadros de diálogo y crea luego el proyecto. Además del equipo, la CPU, las carpetas de programas, fuentes y bloques, así como el OB1, se pueden seleccionar allí también los OBs para el tratamiento de errores y de alarmas.

La figura siguiente muestra un proyecto creado utilizando el Asistente.



Crear un proyecto manualmente

No obstante, un proyecto también se puede crear utilizando los comandos de menú **Archivo > Nuevo** en el Administrador SIMATIC. El proyecto contendrá ya el objeto "Redes MPI".

Alternativas para el trabajo ulterior

Para seguir elaborando un proyecto dispone de máxima flexibilidad. Una vez creado un proyecto puede, p.ej.,

- configurar primero el hardware y crear luego el software, o bien
- crear primero el software sin haber configurado antes el hardware.

1a. alternativa: configurar primero el hardware

Si desea configurar primero el hardware, proceda de la forma descrita en el segundo tomo del manual "Configuración del hardware con STEP 7". Al configurar el hardware, se insertan automáticamente las carpetas necesarias para crear el software, es decir, la de "Programa S7" o la de "Programa M7". Proceda entonces a insertar los objetos necesarios para crear el programa. Por último, cree el software para los módulos programables.

2a. alternativa: crear primero el software

También es posible crear el software sin haber configurado antes el hardware. La configuración se puede realizar posteriormente. Para crear programas no es necesario haber configurado antes el hardware de un equipo.

Si desea crear primero el software:

1. Inserte en su proyecto las carpetas de software necesarias (Programa S7/M7 sin equipo ni CPU).
Para ello basta con que decida si la carpeta "Programas" debe comprender programas para equipos S7 o M7.
2. Por último, cree el software para los módulos programables.
3. Configure el hardware.
4. Una vez configurado el hardware, asigne el programa S7 o M7 a una CPU.

6.6.2 Insertar equipos

En un proyecto, un equipo representa la configuración física del sistema de automatización, conteniendo los datos para configurar y parametrizar los diversos módulos.

Los nuevos proyectos que se hayan creado con el Asistente 'Nuevo proyecto' ya contienen un equipo. En caso contrario, el equipo se podrá crear mediante el comando de menú **Insertar > Equipo**.

Se puede elegir entre los siguientes equipos:

- Equipo SIMATIC 300
- Equipo SIMATIC 400
- Equipo H SIMATIC
- Equipo PC SIMATIC
- PC/PG
- SIMATIC S5
- Otros equipos, es decir, que no sean de SIMATIC S7/M7 o de SIMATIC S5

El equipo se inserta con un nombre estándar (p.ej. Equipo SIMATIC 300 (1), Equipo SIMATIC 300 (2) etc.). Los nombres de los equipos se pueden cambiar por denominaciones más explícitas.

El tema Insertar equipos contiene instrucciones paso a paso al respecto.

Configurar el hardware

En la configuración del hardware se determinan la CPU y todos los módulos contenidos en su sistema de automatización, utilizando para ello un catálogo de hardware. La configuración del hardware se inicia haciendo doble clic en el equipo.

Una vez guardada y concluida la configuración del hardware, para cada módulo programable que se haya configurado se crean automáticamente un programa S7 o M7 como carpeta de software y una tabla de enlaces (objeto "Enlaces"). Los nuevos proyectos que se hayan creado con el Asistente 'Nuevo proyecto' ya contienen estos objetos desde el principio.

El tema Configurar el hardware contiene instrucciones detalladas al respecto. En el tema Procedimiento básico para configurar un equipo encontrará los pasos fundamentales.

Crear una tabla de enlaces

Para cada módulo programable se crea automáticamente una tabla (vacía) de enlaces (objeto "Enlaces"). Dicha tabla se utiliza para definir los enlaces entre los módulos programables de una red. Tras abrirla se visualiza una ventana con una tabla para definir los enlaces entre los módulos programables.

Para obtener información detallada, consulte el tema Conectar equipos de un mismo proyecto a la red.

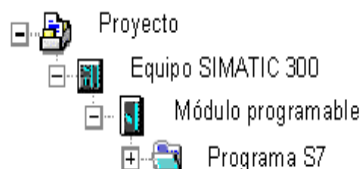
Pasos siguientes

Tras elaborar la configuración del hardware se puede crear el software para los módulos programables (consulte también Insertar un programa S7 o M7).

6.6.3 Insertar un programa S7 o M7

El software para los módulos programables se deposita en carpetas de objetos. En el caso de los módulos SIMATIC S7, dicha carpeta se denomina "Programa S7" y, en el caso de los módulos SIMATIC M7, "Programa M7".

A modo de ejemplo, la figura siguiente muestra un programa S7 contenido en un módulo programable de un equipo SIMATIC 300.



Componentes ya existentes

Para cada módulo programable se crea automáticamente un programa S7 o M7 como carpeta de software.

El programa S7 contiene ya los siguientes componentes:

- una tabla de símbolos (objeto "Símbolos"),
- una carpeta "Bloques" con el primer bloque y
- una carpeta "Fuentes" para programas fuente.

El programa M7 contiene ya los siguientes componentes:

- una tabla de símbolos (objeto "Símbolos"),
- una carpeta "Bloques".

Crear bloques S7

Si desea crear programas AWL, KOP o FUP, seleccione el objeto "Bloques" ya creado y elija el comando de menú **Insertar > Bloque S7**. En el menú siguiente podrá elegir el tipo de bloque (p.ej. bloque de datos, tipo de datos de usuario (UDT), función, bloque de función, bloque de organización, tabla de variables (VAT)).

Una vez abierto el bloque (vacío), podrá introducir el programa en AWL, KOP o FUP, respectivamente. Para obtener más informaciones al respecto, consulte el tema Procedimiento básico para crear bloques lógicos y los manuales de AWL, KOP y FUP, respectivamente.

Nota

El objeto "Datos del sistema" (SDB), que puede encontrarse en el programa de usuario, lo crea el sistema. Aunque se puede abrir, no es posible modificar su contenido por motivos de coherencia. Sirve para modificar la configuración a posteriori, es decir, una vez cargado un programa, y para cargar los cambios en el sistema de destino.

Utilizar bloques de librerías estándar

Para crear programas de usuario puede utilizar también bloques de las librerías estándar incluidas en el volumen de suministro. Para acceder a las librerías, elija el comando **Archivo > Abrir**. Para obtener más información sobre cómo utilizar las librerías estándar y para crear librerías propias, consulte el tema de la Ayuda en pantalla Trabajar con librerías.

Crear fuentes/esquemas CFC

Si desea crear una fuente en un lenguaje de programación determinado, o bien crear un plano CFC, seleccione en el programa S7 el objeto "Fuentes" o "Planos" (esquemas), respectivamente, y elija luego el comando de menú **Insertar > Software S7**. En el menú siguiente, elija la fuente correspondiente al lenguaje de programación. Una vez abierta la fuente vacía, podrá introducir el programa. Para obtener más información al respecto, consulte el tema Reglas para programar en fuentes AWL.

Crear programas para M7

Si desea crear programas para el sistema operativo RMOS de un módulo programable de la gama M7, seleccione en el programa M7 y elija luego el comando de menú **Insertar > Software M7**. En el menú siguiente, elija el objeto correspondiente al lenguaje de programación o al sistema operativo, respectivamente. Una vez abierto el objeto, podrá acceder al entorno de creación de programas.

Crear tablas de símbolos

Al crear un programa S7 o M7, se genera automáticamente una tabla de (vacía) símbolos (objeto "Símbolos"). Una vez abierta, aparece la ventana "Editor de símbolos", visualizándose allí la tabla de símbolos. Para obtener más información, consulte el tema Introducir varios símbolos globales en la tabla de símbolos.

Insertar fuentes externas

Es posible crear y editar archivos fuente con un editor ASCII cualquiera. Dichos archivos se pueden importar luego en un proyecto y compilar en bloques.

Los bloques creados al compilar una fuente importada se depositan en la carpeta "Bloques".

Para obtener más información, consulte el tema Insertar fuentes externas.

6.7 Elaborar un proyecto

Abrir un proyecto

Para abrir un proyecto, elija primero el comando **Archivo > Abrir**. Seleccione luego un proyecto en los cuadros de diálogo siguientes. Entonces se abrirá la ventana del proyecto.

Nota

Si el proyecto no se visualiza en la lista de proyectos, haga clic en el botón "Examinar". Así aparecerá el cuadro de diálogo "Examinar", con el cual podrá buscar otros proyectos e integrar los que se encuentren en la lista de proyectos. Las entradas en la lista de proyectos se pueden modificar seleccionando el comando de menú **Archivo > Gestionar**.

Copiar un proyecto

Para copiar un proyecto, elija el comando de menú **Archivo > Guardar como** y guarde el proyecto con otro nombre.

Los componentes del proyecto, tales como equipos, programas, bloques etc. se copian utilizando el comando de menú **Edición > Copiar**.

Los temas Copiar un proyecto y Copiar un componente de un proyecto contienen instrucciones detalladas al respecto.

Borrar un proyecto

Un proyecto se borra utilizando el comando de menú **Archivo > Borrar**.

Los componentes del proyecto, tales como los equipos, los programas, los bloques etc. se borran con el comando de menú **Edición > Borrar**.

Los temas Borrar un proyecto y Borrar un componente de un proyecto contienen instrucciones detalladas al respecto.

6.7.1 Comprobar los paquetes de software utilizados en el proyecto

Si se está editando un proyecto que contiene objetos que han sido elaborados con un paquete de software, éste será necesario para la edición.

STEP 7 le ayuda independientemente del sistema de origen que utilice para trabajar con los multiproyectos, proyectos o librerías, mostrándole los paquetes de software necesarios en la versión que se necesita.

La información sobre los paquetes de software necesarios es completa cuando:

- el proyecto (o todos los proyectos de un multiproyecto) o la librería se han creado con STEP 7 a partir de V5.2.
- ha comprobado los paquetes de software utilizados en el proyecto. Para ello debe seleccionar el proyecto en el administrador SIMATIC y seleccionar el comando de menú **Edición > Propiedades del objeto**. En el cuadro de diálogo que aparece, seleccione la ficha "Paquetes de software necesarios". A partir de la información que aparece en este cuadro de diálogo puede determinar si tiene que comprobar los paquetes de software utilizados en el proyecto.

6.8 Gestionar textos en varios idiomas

STEP 7 ofrece la posibilidad de exportar, traducir, volver a importar y ver en el idioma traducido los textos monolingües guardados en un proyecto.

Los siguientes tipos de textos se pueden gestionar en varios idiomas.

- Título y comentarios
 - Títulos y comentarios de bloques
 - Títulos y comentarios de segmentos
 - Comentarios de líneas y programas AWL
 - Comentarios de tablas de símbolos, tablas de declaración de variables, tipos de datos de usuario y bloques de datos
 - Comentarios, nombres de estado y de transición en programas HiGraph
 - Ampliaciones a nombres y comentarios de etapas en programas S7-GRAPH
- Textos visualizados
 - Textos de avisos generados por STEP 7, S7-GRAPH, S7-HiGraph, S7-PDIAG o ProTool
 - Librerías de textos del sistema
 - Librerías de textos de usuario
 - Textos de usuario

Exportar

La exportación se realiza para todos los bloques y tablas de símbolos que hay por debajo del objeto seleccionado. Para cada tipo de texto se genera un archivo de exportación. Éste contiene una columna para el idioma de origen y otra para el de destino. Los textos del idioma de origen no deben modificarse.

Importar

Al importar se traspasa al objeto seleccionado el contenido de las columnas para el idioma de destino (columna derecha). Al hacerlo sólo se adoptan aquellas traducciones cuyos textos de origen (textos exportados) coinciden con una entrada existente en la columna "Idioma fuente".

Cambiar idioma

Al cambiar de idioma se pueden seleccionar todos los idiomas que se indicaron al importar en el proyecto seleccionado. El cambio de idioma para "Títulos y comentarios" sólo se realiza en el objeto seleccionado. El cambio de idioma para "Textos para visualizadores" se realiza para todo el proyecto.

Borrar idioma

Al borrar un idioma se borran todos los textos de este idioma de la gestión de datos interna.

En su proyecto siempre debería existir un idioma de referencia. Este podría ser, por ejemplo, el idioma de su país. Este idioma no se debería borrar. Al exportar e importar deberá ajustar siempre este idioma de referencia como idioma de origen. El idioma de destino podrá ajustarlo a su elección.

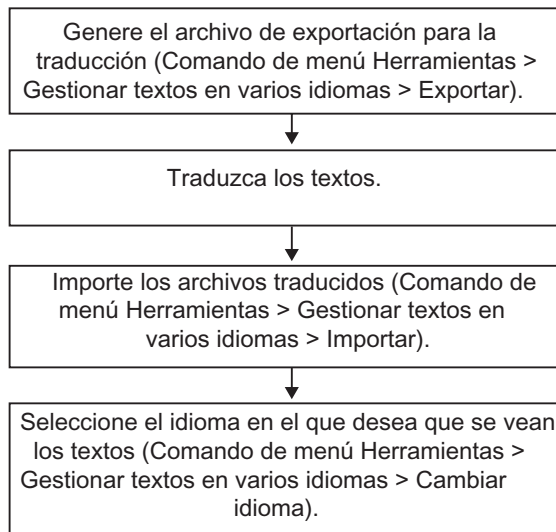
Reorganizar

Al reorganizar se ejecuta un cambio de idioma al idioma ajustado en ese momento. El idioma ajustado en ese momento es el idioma que ha seleccionado en los ajustes para el tratamiento de comentarios como "Idioma para bloques futuros". La reorganización sólo afecta a los títulos y a los comentarios.

Tratamiento de comentarios

Puede ajustar el modo en que deben tratarse comentarios de bloques en proyectos con textos gestionados en varios idiomas.

Procedimiento básico



6.8.1 Tipos de textos gestionados en varios idiomas

Al exportar se crea un archivo propio para cada tipo de texto. Este archivo tendrá como nombre el tipo de texto y como extensión, el formato de exportación (Tipo_de_texto.Formato: p. ej.: SymbolComment.CSV o SymbolComment.XLS). Los archivos que no cumplan esta convención de denominación no se pueden indicar como origen o destino.

Los textos traducibles dentro de un proyecto se dividen en los siguientes tipos de texto:

Tipo de texto	Significado
BlockTitle	Título de bloque
BlockComment	Comentario de bloque
NetworkTitle	Título de segmento
NetworkComment	Comentario de segmento
LineComment	Comentario de línea en AWL
InterfaceComment	Comentario de Var_Section (tabla de declaración en bloques lógicos) Comentario UDT (tipos de datos de usuario) Comentario de bloque de datos
SymbolComment	Comentario de símbolo
S7UserTexts	Textos introducidos por el usuario, que pueden mostrarse en visualizadores
S7SystemTextLibrary	Textos de librerías de textos de sistema que pueden integrarse en avisos, actualizarse dinámicamente durante el tiempo de ejecución y visualizarse en la PG o en otros visualizadores.
S7UserTextLibrary	Textos de librerías de textos de usuario que pueden integrarse en avisos, actualizarse dinámicamente durante el tiempo de ejecución y visualizarse en la PG o en otros visualizadores.

Tipo de texto	Significado
HiGraphStateName HiGraphStateComment	S7-HiGraph Nombre de estado Comentario de estado
HiGraphTransitionName HiGraphTransitionComment	Nombre de transición Comentario de transición
S7GraphStateName S7GraphStateComment	S7-GRAPH Extensión del nombre de la etapa Comentario de etapa

Puede haber otros tipos de texto específicos de aplicaciones de otros editores de otros paquetes opcionales (p.ej. Protool, WinCC,.....), que aquí no se describen.

6.8.2 Estructura del archivo de exportación

La estructura del archivo de exportación es la siguiente:

Ejemplo:

\$_Languages		
7(1) Español (España)	9(1) Inglés (USA)	
\$_Typ(NetworkTitle)		
//Longitud máxima del texto: 64 caracteres		
//\$_Exportado el 06.02.2006 11.44.19		
Primera secuencia de caracteres a traducir	Traducción	testPrograma S7(1)\BloquesOB1
Segunda secuencia de caracteres a traducir	Traducción	testPrograma S7(1)\BloquesOB1
Secuencia de caracteres que no debe ser mostrada en la traducción	\$_hide	testPrograma S7(1)\BloquesOB1

Idioma de origen

Idioma de destino

Punto de aplicación

Como norma general:

- No se deberá modificar, sobrescribir ni borrar:
 - los campos precedidos de "\$_" (se trata de palabras clave)
 - los números para el idioma (en el ejemplo superior 7(1) indica el idioma de origen alemán (Alemania) y 9(1) el idioma de destino inglés)
- En el archivo los textos son siempre de un solo tipo. En el ejemplo el tipo de texto es título de segmento (\$_Typ(NetworkTitle)). Las reglas para el traductor que trabaje con este archivo están en el texto introductorio del archivo de exportación.
- La información adicional sobre los textos o comentarios siempre deben estar delante de la definición del tipo (\$_Typ...) o detrás de la última columna.

Nota

Si la columna del idioma de destino aparece sobrescrita con "512(32) \$_Undefined", es porque al exportar no se indicó ningún idioma de destino. Para tener una mejor visión general, puede sustituir este texto con el idioma de destino, p. ej., "9(1) Inglés (USA)". Al importar deberá tener en cuenta el idioma de destino propuesto y, en caso necesario, seleccionarlo de nuevo.

Los textos que no deben visualizarse en el idioma de destino pueden ocultarse con la clave \$_hide. Eso no es aplicable a comentarios sobre variables (InterfaceComment) ni a símbolos (SymbolComment).

Formato del archivo de exportación

Puede escoger el formato en que desea guardar los archivos de exportación.

Si se decide por el formato CSV, debe tener en cuenta que en la edición con EXCEL éste sólo abre los archivos CSV correctamente desde el cuadro de diálogo Abrir. **Si se abre un archivo CSV haciendo doble clic en él desde el Explorador, lo más normal es que el archivo quede inutilizable.** El siguiente procedimiento facilitará la edición de archivos CVS con EXCEL:

1. Abrir los archivos de exportación con EXCEL
2. Guardar los archivos como archivos XLS
3. Traducir los textos en los archivos XLS
4. Guardar los archivos XLS con EXCEL en formato CSV.

Nota

¡No se debe cambiar el nombre de los archivos de exportación!

6.8.3 Gestionar textos de usuario cuya fuente no está instalada

También tiene la posibilidad de exportar, traducir, importar y finalmente guardar en su proyecto textos de usuario a un idioma cuya fuente no está instalada en el sistema operativo.

No obstante, esos textos sólo se podrán visualizar en un ordenador en el que esté instalada la fuente en cuestión.

Proceda de la siguiente manera p.ej. para traducir al ruso textos de usuario si no tiene instalada la escritura cirílica:

1. Exporte el textos de usuario a traducir con el idioma de origen español y el idioma de destino ruso.
2. Envíe los archivos de exportación al traductor, que, naturalmente, tiene instalada la escritura cirílica en su ordenador.
3. Importe los archivos de exportación traducidos.
Resultado: ahora el proyecto está guardado en su ordenador en varios idiomas (español y ruso).
4. Guarde todo el proyecto y envíelo al cliente final, que podrá ver los textos en ruso en su ordenador, en el que tendrá instalada la escritura cirílica.

6.8.4 Información sobre el archivo de informe

Los mensajes de error y advertencias que aparecen al trabajar con textos gestionados en varios idiomas, se emiten en forma de archivo de informe en formato txt. Este archivo se deposita en el mismo directorio que los archivos de exportación.

Por lo general, los mensajes suelen ser autoexplicativos. A continuación encontrará información más detallada al respecto:



Advertencia

El texto 'xyz' que figura en el archivo 'xyz' ya existe. Se han ignorado otras apariciones del texto.

Explicación

Un texto es utilizado independientemente de su idioma como clave para la traducción. En caso de que un texto idéntico sea utilizado en más de un idioma o varias veces en un mismo idioma para designar distintos términos, ya no se podrá asignar de forma unívoca y, por consiguiente, ya no se traducirá.

Ejemplo:

\$ _Languages		
7(1) Español (España)	7 (1) Inglés (USA)	
\$ _Typ(SymbolComment)		
Habilator Auto		
Habilitación automtica		
Habil. autom.		

Idioma de origen

Idioma de destino

Solo se aplica para títulos y comentarios.

Solución

Cambie el nombre de los textos en cuestión en el archivo de texto (en el caso del ejemplo habría que utilizar un solo término en lugar de los tres utilizados) y vuelva a importar los textos

6.8.5 Optimización de la plantilla para la traducción

El material lingüístico original se puede depurar para la traducción unificando términos o expresiones diferentes.

Ejemplo

Antes de la depuración (archivo de exportación):

\$_Languages		
7(1) Español (España)	7 (1) Inglés (USA)	
\$_Typ(SymbolComment)		
Habilator Auto		
Habilitación automtica		
Habil. autom.		

Idioma de origen

Idioma de destino

Unificación como una sola expresión:

\$_Languages		
7(1) Español	7 (1) Español	
\$_Typ(SymbolComment)		
Habilitar Auto	Habilitación automática	
Habilitación automática	Habilitación automática	
Habil. autom.	Habilitación automática	

Idioma de origen

Idioma de destino

Después de la depuración (es decir, después de importar y exportar):

\$_Languages		
7(1) Español	7 (1) Español	
\$_Typ(SymbolComment)		
Habilitación automática	Habilitación automática	

Idioma de origen

Idioma de destino

6.8.6 Optimización del proceso de traducción

En los proyectos en los que la estructura y los textos sean parecidos a un proyecto anterior, se puede optimizar el proceso de traducción.

Sobre todo en los proyectos que se han creado copiándolos y adaptándolos es recomendable seguir el proceso descrito a continuación.

Requisitos

Que exista un destino para la exportación ya traducido.

Procedimiento

1. Copie los archivos de exportación en el directorio del proyecto nuevo que se va a traducir.
2. Abra el nuevo proyecto y exporte los textos (comando de menú **Herramientas > Gestionar textos en varios idiomas > Exportar**)
Puesto que ya existe el destino de la exportación, se le preguntará si desea ampliarlo o sobrescribirlo.
3. Haga clic en el botón "Ampliar".
4. Mande a traducir los archivos de exportación (sólo habrá que traducir los textos nuevos).
5. A continuación, importe los archivos traducidos.

6.9 Micro Memory Card (MMC) como soporte de datos

6.9.1 Información importante sobre Micro Memory Cards (MMC)

Las Micro Memory Cards (MMC) son tarjetas enchufables para p.ej. una CPU 31xC o una IM 151/CPU (ET 200S) que se caracterizan por un diseño extremadamente compacto.

Para las MMCs existe un nuevo concepto de memoria que describiremos a continuación.

Contenido de las MMCs

La MMC realiza tanto la función de la memoria de carga como la función de un soporte de datos.

MMC como memoria de carga

Las MMCs contienen toda la **memoria de carga** de una CPU adecuada para MMCs. Incluye el programa con los bloques (OBs, DBs, FCs, ...) así como la configuración del hardware. Este contenido influye en el funcionamiento de la CPU. En calidad de memoria de carga, los bloques y la configuración del hardware pueden transferirse con funciones de carga (p. ej. **Cargar** en CPU). Los bloques cargados en la CPU son inmediatamente operativos, mientras que la configuración del hardware lo es tras arrancar la CPU.

Reacción tras un borrado total

Los bloques de la MMC no se pierden tras un borrado total.

Cargar y borrar

Los bloques de la MMC se pueden sobrescribir.

Los bloques de la MMC se pueden borrar.

Los bloques sobrescritos y borrados no se pueden recuperar.

Acceso a los bloques de datos de la MMC

Para grandes cantidades de datos o para datos que el programa de usuario requiere rara vez, es posible manejar bloques de datos y sus contenidos en la MMC. Para ello se dispone de nuevas funciones de sistema:

- SFC 82: Generar bloque de datos en la memoria de carga
- SFC 83: Leer del bloque de datos residente en la memoria de carga
- SFC 84: Escribir en un bloque de datos residente en la memoria de carga

MMC y protección por contraseña

Si una CPU equipada con una MMC (p. ej. una CPU de la gama 300-C) está protegida por contraseña, esta contraseña también deberá indicarse al abrir esta MMC en el Administrador SIMATIC (en la PG o el PC).

Visualizar la ocupación de memoria en STEP 7

En el diálogo de información del módulo (ficha "Memoria") se indica conjuntamente la ocupación de la memoria de carga de la EPROM y de la RAM.

Los bloques que residen en la MMC tienen el mismo comportamiento que la EPROM.

6.9.2 Utilizar la Micro Memory Card como soporte de datos

La SIMATIC Micro Memory Card (MMC) puede emplearse como un soporte de datos externo normal con STEP 7.

Con la condición de que la MMC sea lo suficientemente grande, pueden transferirse a la MMC todos los archivos visibles en el explorador de archivos del sistema operativo.

De este modo, puede poner a disposición de otros colaboradores más dibujos, instrucciones de servicio, descripciones de funcionamiento, etc. de su instalación.

6.9.3 Archivo Memory Card

Los archivos Memory Card (*.wld) se generan para:

- Software PLC **WinLC** (WinAC Basis y WinAC RTX) y
- SlotPLCs **CPU 41x-2 PCI** (WinAC Slot 412 y WinAC Slot 416).

En un archivo Memory Card se pueden depositar los bloques y los datos del sistema para una WinLC o una CPU 41x-2 PCI de la misma manera que en una Memory Card S7. El contenido de estos archivos equivale entonces al contenido de una Memory Card correspondiente para una CPU S7.

Este archivo se puede cargar por medio de un comando de menú del panel de mandos de WinLC o de la CPU 41x-2 PCI en su memoria de carga, lo cual equivale a cargar el programa de usuario con STEP 7.

En las CPUs 41x-2 PCI se puede cargar este archivo automáticamente al arrancar el sistema operativo del PC, si la CPU 41x-2 PCI no está respaldada y sólo funciona con una tarjeta RAM (función "Autoload").

Los archivos Memory Card son archivos convencionales en lo que a Windows se refiere. Es decir, se pueden mover o borrar con el Explorador o se pueden transportar por medio de un soporte de datos.

Para más información al respecto, consulte la documentación correspondiente de los productos WinAC.

6.9.4 Guardar datos de proyecto en la Micro Memory Card (MMC)

Con STEP 7 puede guardar los datos de su proyecto STEP 7 así como otros datos cualesquiera (p. ej. archivos WORD o EXCEL) en la SIMATIC Micro Memory Card (MMC) de una CPU adecuada o de una PG/un PC. De este modo tiene la ventaja de poder acceder a los datos de proyecto incluso con unidades de programación que no han grabado el proyecto.

Requisitos

Sólo es posible guardar datos de proyecto en una MMC cuando la MMC se encuentra en la ranura de la CPU adecuada o de una PG/un PC y existe un enlace online.

El tamaño de la MMC debe calcularse de modo que tenga espacio suficiente para los datos guardados.

Los datos siguientes pueden guardarse en una MMC

Con la condición de que la MMC sea lo suficientemente grande, pueden transferirse a la MMC todos los archivos visibles en el explorador de archivos del sistema operativo, por ejemplo:

- datos de proyecto completos de STEP 7
- configuraciones del equipo
- tablas de símbolos
- bloques y fuentes
- textos en varios idiomas
- otros datos cualesquiera, p. ej. archivos WORD o Excel

7 Editar proyectos con diferentes versiones de STEP 7

7.1 Editar proyectos y librerías de la versión 2

La versión V5.2 de STEP 7 ya **no** permite **modificar proyectos** y librerías de la versión **V2**. Si edita proyectos o librerías V2, es posible que se produzcan incoherencias que hagan que los proyectos o librerías V2 ya no se puedan seguir editando con versiones anteriores de STEP 7.

Para poder seguir editando proyectos o librerías V2 hay que utilizar una versión de STEP 7 anterior a V5.1.

7.2 Ampliar esclavos DP creados con versiones anteriores de STEP 7

Configuraciones que pueden crearse instalando nuevos archivos GSD

Instalando nuevos archivos GSD en el catálogo de hardware de HW-Config. pueden incorporarse nuevos esclavos DP. Después de la instalación están disponibles en la carpeta "Otros aparatos de campo".

Un esclavo DP modular no puede ampliarse (ni modificarse su configuración) de la forma habitual en los casos siguientes:

- si ha sido configurado con una versión anterior de STEP 7 o
- si en el catálogo de hardware no estaba representado por un archivo GSD, sino por un archivo de tipo, o bien
- si se ha instalado un archivo GSD nuevo

Remedio

Si desea utilizar el esclavo DP con **nuevos módulos** descritos en el archivo GSD:

- borre el esclavo DP y reconfigúrelo; el esclavo DP no se describirá mediante el archivo de tipo, sino que quedará completamente definido por el archivo GSD.

Si no desea utilizar **módulos nuevos**, descritos únicamente en el archivo GSD:

- seleccione la carpeta "Otros aparatos de campo/Esclavos compatibles Profibus-DP", dentro de PROFIBUS-DP en la ventana "Catálogo de hardware". STEP 7 desplaza a dicha carpeta los archivos de tipo "antiguos" cuando son reemplazados por archivos GSD nuevos. En dicha carpeta encontrará los módulos con los que podrá ampliar el esclavo DP configurado.

Constelación tras sustituir archivos de tipo por archivos GSD en STEP 7 V5.1 Servicepack 4

A partir de STEP 7 V5.1, Servicepack 4, los archivos de tipo se han sustituido o actualizado en su mayor parte por archivos GSD. Esta sustitución sólo afecta a los perfiles de catálogo suministrados con STEP 7, no a los creados por el usuario.

Los esclavos DP, cuyas propiedades estaban determinadas antes por archivos de tipo y ahora por archivos GSD, se han mantenido en la misma posición en el catálogo de hardware.

Los archivos de tipo "antiguos" no se han borrado, sino que se han movido a otro lugar en el catálogo de hardware. Ahora se encuentran en la carpeta "**Otros aparatos de campo esclavos compatibles Profibus DP\...**".

Ampliar una configuración DP ya existente con STEP 7 a partir de V5.1 Servicepack 4

Si edita un proyecto creado con una versión anterior de STEP 7 (anterior a V5.1, SP4) y desea ampliar un esclavo DP modular, no podrá utilizar los módulos ni bloques del lugar habitual del catálogo de hardware. En tal caso, utilice el esclavo DP almacenado en "**Otros aparatos de campo esclavos compatibles Profibus DP\...**".

Editar una configuración DP con una versión de STEP 7 inferior a V5.1, SP4)

Si configura un esclavo DP "actualizado" con STEP 7 a partir de V5.1, Servicepack 4, y después quiere editar el proyecto con una versión anterior (versión de STEP 7 anterior a V5.1, SP4), no podrá editar dicho esclavo DP, ya que no podrá reconocer el archivo GSD de la versión de STEP 7.

Remedio: Puede volver a instalar el archivo GSD en la versión anterior de STEP 7. En tal caso, el archivo GSD quedará guardado en el proyecto. Si después edita el proyecto con la versión actual de STEP 7 utilizará para la configuración el archivo GSD instalado posteriormente.

7.3 Editar configuraciones actuales con versiones anteriores de STEP 7

Configurar la comunicación directa

Configurar la comunicación directa con un maestro DP **sin** sistema maestro DP:

- no es posible con STEP 7 V5.0, Service Pack 2 (o anterior)
- sí es posible con STEP 7 V5.0, a partir de Service Pack 3 y STEP 7 V5.1

Si guarda un maestro DP sin sistema maestro DP propio con asignaciones configuradas para la comunicación directa, y continúa editando el proyecto con una versión anterior a STEP 7 V5 (STEP 7 V5.0, SP2 (o anterior)), pueden producirse los siguientes efectos:

- Un sistema maestro DP se visualiza con esclavos DP que se utilizan para el almacenamiento de datos interno a STEP 7 de las asignaciones de la comunicación directa. Estos esclavos DP no pertenecen al sistema maestro DP visualizado.
- A dicho maestro DP no se puede conectar ni un sistema maestro DP nuevo ni uno huérfano.

Enlace online con la CPU a través de interface DP PROFIBUS

Configuración del interface DP PROFIBUS **sin** sistema maestro DP:

- STEP 7 V5.0, Service Pack 2 (o anterior): no es posible establecer un enlace con la CPU a través de este interface.
- a partir deSTEP 7 V5.0, Service Pack 3: al compilar se crean datos del sistema para el interface DP PROFIBUS; es posible establecer un enlace con la CPU a través de este interface después de la carga.

7.4 PC SIMATIC - Utilizar configuraciones de versiones anteriores

Configuraciones de PC de proyectos de STEP 7 V5.1 (hasta SP 1)

A partir de STEP 7 V5.1, Servicepack 2 se pueden cargar enlaces en el equipo PC igual que en un equipo S7-300 o S7-400 (sin necesidad de un archivo de configuración). No obstante, al guardar y compilar siempre se genera un archivo de configuración con el que también se puede transferir la configuración al equipo PC de destino.

La consecuencia es que estos archivos de configuración contendrán información que los "antiguos" equipos PC no pueden interpretar. STEP 7 se adapta automáticamente a esta situación:

- Si reconfigura un equipo PC SIMATIC con STEP 7 a partir de V5.1, Servicepack 2, STEP 7 supondrá que el equipo PC de destino se configuró con el CD SIMATIC NET a partir de 7/2001, es decir, que S7RTM (Runtime Manager) está instalado. Los datos de configuración se generarán de tal forma que puedan ser interpretados por un equipo PC "nuevo".
- Si utiliza un equipo PC SIMATIC ya configurado con una versión anterior (p. ej., un equipo PC configurado con STEP 7 V5.1, Servicepack 1), STEP 7 **no** supondrá que el equipo PC de destino se configuró con el CD SIMATIC NET a partir de 7/2001. Los datos de configuración se generarán de tal forma que puedan ser interpretados por un equipo PC "antiguo".

Si este modo de actuación predefinido no es adecuado, puede cambiarlo según se describe a continuación:

Ajustes en el contexto "Configurar hardware":

1. Abra la configuración del hardware del equipo PC
2. Haga clic con el botón derecho del ratón en la ventana del equipo (en la superficie blanca)
3. Seleccione el menú contextual "Propiedades del equipo"
4. Active o desactive la casilla de verificación bajo "Compatibilidad".

Ajustes en el contexto "Configurar redes":

1. Abra la configuración de red
2. Marque el equipo PC
3. Seleccione el comando de menú Edición > Propiedades del objeto
4. Seleccione en el cuadro de diálogo la ficha "Configuración"
5. Active o desactive la casilla de verificación bajo "Compatibilidad".

Configuraciones de PC de proyectos de STEP 7 V5.0

Si desea continuar editando con STEP 7 a partir de V5.0, Servicepack 3, un equipo PC SIMATIC para configurar nuevos componentes que sólo son compatibles con versiones a partir del Servicepack 3, tendrá que convertir el equipo:

1. En el Administrador SIMATIC marque el equipo PC SIMATIC y seleccione el comando de menú **Edición > Propiedades del objeto**.
2. En la ficha "Funcionalidad" del cuadro de diálogo de propiedades, haga clic en el botón "Ampliar".

A continuación se convertirá el equipo PC SIMATIC y ya sólo se podrá editar con STEP 7 V5.0, Servicepack 3, o con versiones posteriores.

7.5 Representación de módulos configurados con nuevas versiones de STEP 7 o con paquetes opcionales


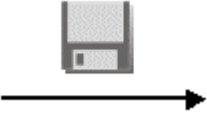

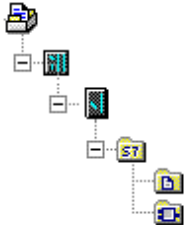

A partir de STEP 7 V5.1, Servicepack 3, se representan todos los módulos, incluso los configurados con una versión más nueva de STEP 7 y que podrían ser desconocidos por versiones "antiguas" de STEP 7. También se representan los módulos configurados con un paquete opcional, aun cuando el paquete necesario no esté instalado en la PG donde se abren los proyectos.

En versiones anteriores de STEP 7 no se representaban los módulos y sus objetos subordinados. En la versión actual, se pueden ver estos objetos y se puede editar en parte. Así, por ejemplo, puede utilizar esta función para modificar programas de usuario si el proyecto se ha creado en otro microcomputador con una versión posterior de STEP 7 y el módulo (p.ej. una CPU) no se puede configurar con la versión disponible de STEP 7 ya que, por ejemplo, tiene nuevas propiedades y parámetros.

El módulo "desconocido" para STEP 7 aparece como módulo sustituto con la siguiente representación:



Si abre el proyecto con la versión adecuada de STEP 7 o con el paquete opcional apropiado, aparecerán todos los módulos con la representación acostumbrada y se podrán volver a editar de forma ilimitada.

PG con la última versión de STEP 7 o con paquete opcional		PG con versiones antiguas de STEP 7 o sin paquete opcional
		
	>>>---Datos de proyecto--->>>	
Representación con el último módulo "conocido"		Representación del último módulo como módulo "desconocido"
		

Trabajar con el módulo sustituto en el Administrador SIMATIC

El módulo sustituto se encuentra por debajo del equipo. Todos los objetos subordinados, como el programa de usuario, los datos de sistema y la tabla de enlaces se pueden ver y se pueden cargar desde el Administrador SIMATIC.

Podrá abrir el programa de usuario (p.ej. bloques), editarlo, compilarlo y cargarlo. Sin embargo, los proyectos con módulos sustitutos presentan las siguientes limitaciones:

- No es posible copiar un equipo con módulos sustitutos.
- En "Guardar proyecto como..." no es posible ejecutar completamente la opción "Reorganizando". El módulo sustituto así como todas las referencias y objetos subordinados de este módulo faltarán a continuación en el proyecto que se ha copiado y reorganizado (p.ej., el programa de usuario).

Trabajar con el módulo sustituto en la configuración de hardware

El módulo sustituto aparece representado en el slot para el que se haya configurado.

Puede abrir el módulo, pero no cambiar la parametrización ni cargarlo. Las propiedades de módulo se limitan a la ficha "Sustituto". La configuración del equipo no se puede modificar (p.ej. no se pueden añadir nuevos módulos).

También es posible realizar el diagnóstico de hardware (p.ej. abrir un equipo online) aunque estará limitado y no se tendrán en cuenta las nuevas posibilidades y textos de diagnóstico.

Trabajar con el módulo sustituto en la configuración de red

El módulo sustituto también aparece representado en NetPro. El nombre del módulo aparece en el equipo correspondiente con un signo de interrogación.

Un proyecto con un módulo sustituto sólo se podrá abrir en NetPro si está protegido contra escritura.

Si abre el proyecto con la protección contra escritura activada, podrá ver e imprimir la configuración de red. También es posible ver el estado del enlace; como mínimo se muestra la información compatible con la versión de STEP 7 disponible.

Sin embargo, en general no podrá hacer cambios ni guardarlos, compilarlos o cargarlos.

Instalar módulos con posterioridad

Si el módulo procede de una versión posterior de STEP 7 y requiere una actualización de hardware, puede sustituir el módulo sustituto por el "real". Nada más abrir el equipo recibirá información sobre las actualizaciones de hardware o los paquetes opcionales y podrá instalar las actualizaciones de hardware desde ese cuadro de diálogo. También puede instalar el módulo posteriormente con el comando de menú **Herramientas > Instalar actualizaciones HW**.

8 Definir símbolos

8.1 Direccionamiento absoluto y simbólico

En un programa de STEP 7 se utilizan operandos tales como señales de E/S, marcas, contadores, temporizadores, bloques de datos y bloques de función. Si lo desea, puede direccionar dichos operandos en su programa de forma absoluta. No obstante, la legibilidad del programa aumentará considerablemente si utiliza nombres simbólicos (p.ej. Motor_A_On, o bien, denominaciones usuales en su ramo). En el programa de usuario será posible entonces direccionar un operando mediante dicho símbolo.

Dirección absoluta

Una dirección absoluta se compone de un identificador de operando y de una dirección (p. ej., A 4.0, E 1.1, M 2.0, FB 21).

Direccionamiento simbólico

Asignándoles nombres simbólicos a las direcciones absolutas, será posible diseñar el programa de forma más clara y facilitar la corrección de errores.

STEP 7 puede traducir automáticamente los nombres simbólicos a las direcciones absolutas necesarias. Si prefiere acceder a los ARRAYS, STRUCTs, bloques de datos, datos locales, bloques lógicos y tipos de datos de usuario a través de los nombres simbólicos, deberá asignarles previamente éstos últimos a las direcciones absolutas antes de poder direccionar los datos de forma simbólica.

Por ejemplo, puede asignar al operando A 4.0 el nombre simbólico MOTOR_ON y utilizar luego MOTOR_ON como dirección en una instrucción del programa. A través de las direcciones simbólicas podrá reconocer fácilmente si los elementos del programa concuerdan con los componentes de su solución de automatización.

Nota

En un nombre simbólico (denominación de variable) no se pueden utilizar dos caracteres de subrayado seguidos(p.ej.: MOTOR__ON).

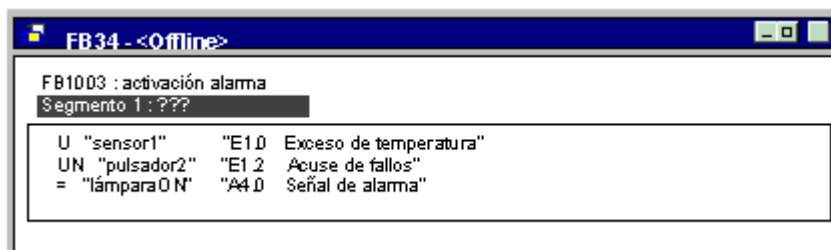
Ayudas al introducir el programa

Las direcciones, los parámetros y los nombres de los bloques se pueden indicar de forma absoluta o bien en forma de símbolos en los lenguajes de programación KOP, FUP y AWL.

Con el comando de menú **Ver > Representación simbólica** puede conmutar entre la representación absoluta y la representación simbólica de las direcciones.

Para facilitar la programación con direcciones simbólicas, es recomendable visualizar las direcciones absolutas y los comentarios de los símbolos utilizados. Utilice para ello el comando de menú **Ver > Información del símbolo**. Procediendo así, tras cada instrucción AWL la línea de comentario se sustituye por dicha información. El texto visualizado no se puede editar. Los cambios se deben efectuar en la tabla de símbolos o en la tabla de declaración de variables.

La figura siguiente muestra una información del símbolo en AWL.



Al imprimir el bloque, se reproduce la representación actual en pantalla con el comentario de la instrucción o con el del símbolo.

8.2 Símbolos globales y locales

Los símbolos permiten utilizar denominaciones claras en vez de direcciones absolutas. Mediante la combinación de símbolos breves con comentarios más detallados puede crear programas eficientemente y buenas documentaciones.

Se distingue entre símbolos locales y símbolos globales.

	Símbolos globales	Símbolos locales
Ámbito de validez	<ul style="list-style-type: none"> Se pueden aplicar en todo el programa de usuario Pueden ser utilizados por todos los bloques, Tienen la misma importancia en todos los bloques, Su denominación debe ser unívoca en todo el programa de usuario. 	<ul style="list-style-type: none"> Sólo se conocen en el bloque donde fueron definidos. Es posible utilizar la misma denominación en diversos bloques para fines diferentes.
Caracteres admisibles	<ul style="list-style-type: none"> Letras, cifras, caracteres especiales Diéresis, excepto 0x00, 0xFF y comillas; Al utilizarse caracteres especiales, el símbolo se deberá colocar entre comillas. 	<ul style="list-style-type: none"> Letras, cifras, carácter de subrayado (_),
Aplicación	<p>Los símbolos globales se pueden definir para:</p> <ul style="list-style-type: none"> señales de E/S (E, EB, EW, ED, A, AB, AW, AD) entradas y salidas periféricas (PE, PA) marcas(M, MB, MW, MD) temporizadores (T)/ contadores (Z) bloques lógicos (OB, FB, FC, SFB, SFC) bloques de datos (DB) tipos de datos de usuario tabla de variables (VAT) 	<p>Los símbolos locales se pueden definir para:</p> <ul style="list-style-type: none"> parámetros de bloques (parámetros de entrada, de salida y de entrada/salida), datos estáticos de un bloque, datos temporales de un bloque.
Lugar de definición	Tabla de símbolos	Tabla de declaración de variables del bloque

8.3 Representación de símbolos globales y locales

En el área de instrucciones de un programa, los símbolos globales se pueden diferenciar de los locales de la siguiente forma:

- Los símbolos (globales) de la tabla de símbolos se representan entre comillas "..".
- Los símbolos (locales) de la tabla de declaración de variables se representan precedidos de un "#".

No es necesario introducir las comillas o el signo "#". Tras introducir el programa en KOP, FUP o AWL, la identificación se agrega una vez concluida la comprobación de sintaxis.

No obstante, si hubiera confusión, debido a que un mismo símbolo se utiliza tanto en la tabla de símbolos como en la declaración de variables, es preciso identificar explícitamente el símbolo global. De lo contrario, los símbolos no identificados se interpretarán como variables locales de un bloque.

Además, será preciso identificar los símbolos globales que contengan espacios en blanco.

En caso de programar en una fuente AWL se deberán respetar los mismos signos y reglas de utilización. En la entrada orientada a la fuente, las identificaciones no se complementan automáticamente. No obstante, sólo serán necesarias si pudiesen presentarse confusiones.

Nota

Con el comando de menú **Ver > Representación simbólica** puede optar entre visualizar los símbolos globales declarados o la correspondiente dirección absoluta.

8.4 Ajustar la preferencia de operandos (simbólica/absoluta)

Cuando realiza cambios en la tabla de símbolos, la preferencia de operandos le ayuda a perfilar los cambios de los nombres de parámetros de bloques de datos o bloques de función o, si realiza cambios en los nombres de componentes de UDTs referenciados o de multiinstancias, le ayuda a perfilar el código de programa como desee.

Ajuste la preferencia de operandos para las siguientes situaciones de cambio de forma selectiva. Para poder aprovechar las ventajas de la preferencia de operandos, el proceso de cambio debe haber finalizado antes de iniciar un cambio de otro tipo.

En el administrador SIMATIC, seleccione la carpeta del bloque y seleccione el comando de menú **Editar > Propiedades del objeto** para ajustar la preferencia de operandos. En la ficha "Operandos preferentes" puede seleccionar los ajustes deseados.

A fin de poder ajustar una preferencia de operandos óptima, se distingue entre las siguientes situaciones de cambio:

- Corregir nombres individualmente
- Sustituir nombres o asignaciones
- Nuevos símbolos, variables, parámetros o componentes

Nota

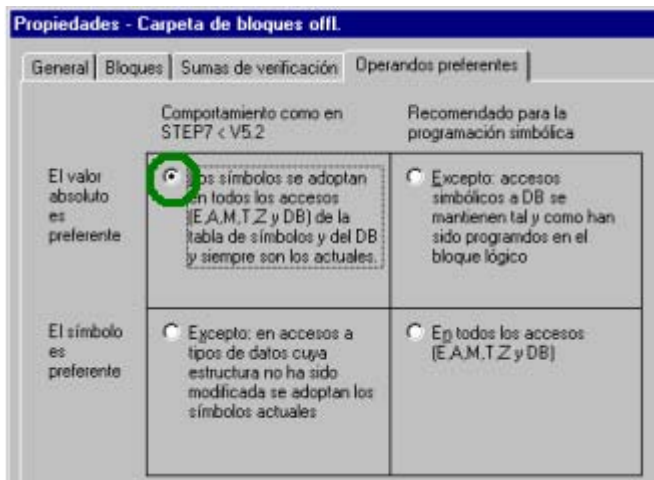
Recuerde que en las llamadas de bloque ("Call FC" o "Call FB, DB") para el bloque lógico siempre es determinante únicamente el número de bloque absoluto – ¡también aunque la preferencia de operandos se determine como simbólica!

Corregir nombres individualmente

Ejemplos:

En la tabla de símbolos o en el editor de programas/bloques debe corregirse un nombre debido a un fallo de ortografía. Esto es válido tanto para todos los nombres de la tabla de símbolos como para todos los nombres de parámetros, variables o componentes que pueden cambiarse con el editor de programas/bloques.

Ajuste de la preferencia de operandos:



Seguimiento de cambios:

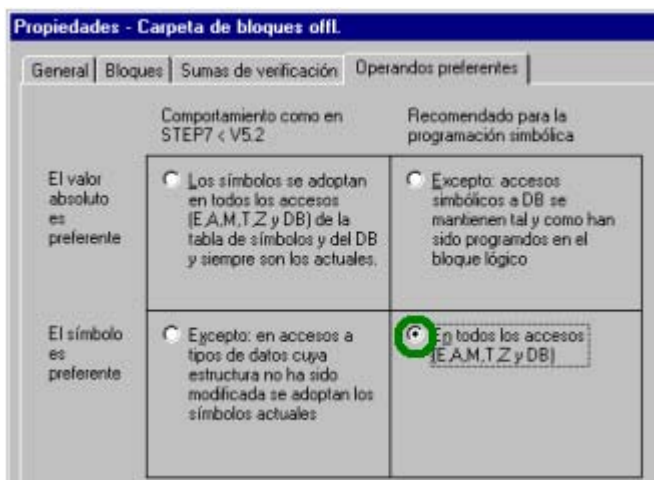
En el administrador SIMATIC, seleccione la carpeta de bloques y el comando de menú **Edición > Comprobar coherencia de bloques**. La función "Comprobar coherencia de bloques" realiza los cambios necesarios en cada uno de los diferentes bloques.

Sustituir nombres o asignaciones

Ejemplos:

- Se cambian los nombres de asignaciones existentes en la tabla de símbolos.
- A las asignaciones existentes en la tabla de símbolos se les asigna nuevas direcciones.
- En el editor de programas/bloques se cambiar nombres de variables, parámetros o componentes.

Ajuste de la preferencia de operandos:



Seguimiento de cambios:

En el administrador SIMATIC, seleccione la carpeta del bloque y seleccione el comando de menú **Edición > Comprobar coherencia de bloques**. La función "Comprobar coherencia de bloques" ejecuta los cambios necesarios en cada uno de los diferentes bloques.

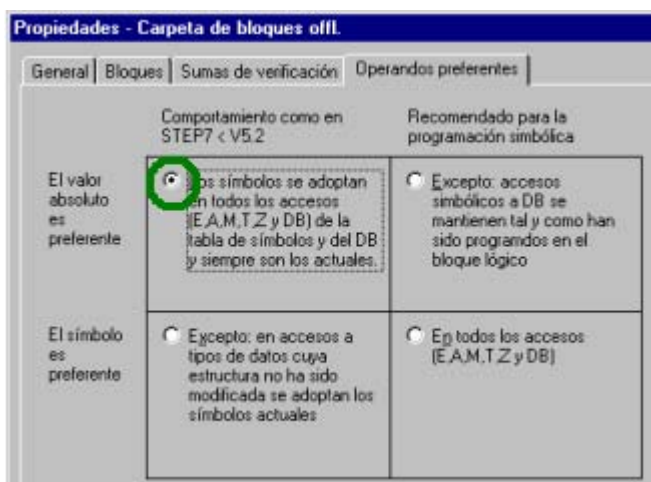
Nuevos símbolos, variables, parámetros o componentes

Ejemplos:

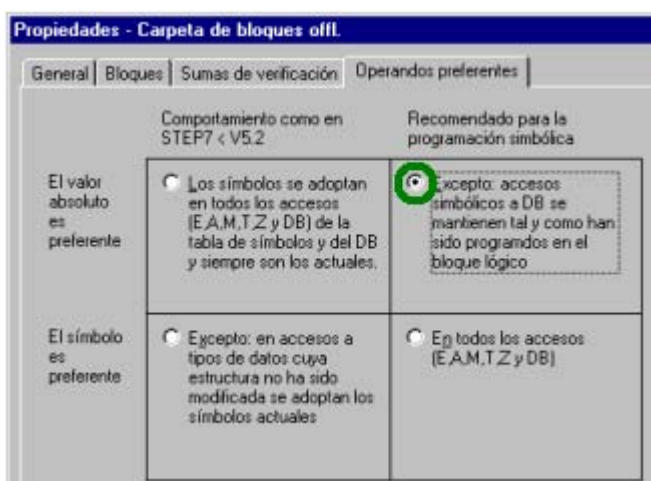
- Crea nuevos símbolos para direcciones utilizadas en el programa.
- Amplía bloques de datos, UDTs o bloques de funciones para nuevas variables o parámetros.

Ajuste de la preferencia de operandos:

- En caso de cambios en la tabla de símbolos.



- En caso de cambios en el editor de programas/bloques.



Seguimiento de cambios:

En el administrador SIMATIC, seleccione la carpeta del bloque y el comando de menú **Edición > Comprobar coherencia de bloques**. La función "Comprobar coherencia de bloques" ejecuta los cambios necesarios en cada uno de los diferentes bloques.

8.5 Tabla de símbolos para los símbolos globales

Los símbolos globales se definen en la tabla de símbolos.

Al crear un programa S7 o M7 se genera automáticamente una tabla de símbolos vacía (objeto "Símbolos").

Ámbito de validez

La tabla de símbolos será válida sólo para el módulo al que se asigne el programa. Si desea utilizar los mismos símbolos en diversas CPUs, deberá asegurarse de que coincidan las entradas correspondientes en las tablas de símbolos en cuestión (p. ej., copiándolas).

8.5.1 Estructura y componentes de la tabla de símbolos

Estructura de la tabla de símbolos

	Estado	S	V	M	C	AC	Símbolo	Dirección	Tipo de dato	Comentario
1		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Arrancar_MD	E 1.4	BOOL	Arrancar el motor
2		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Arrancar_MG	E 1.0	BOOL	Arrancar motor de
3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Automático ON	E 0.5	BOOL	para la función de
4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Conectar_ventila...	A 5.6	BOOL	Comando para cor
5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Conectar_ventila...	A 5.2	BOOL	Comando para cor
6		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Datos_globales	DB 3	DB 3	Bloque de datos gl

Columna de líneas

	Si se han ocultado las columnas para las "Propiedades especiales del objeto" (comando de menú Ver > Columnas S, MV, A, C, AC desactivado), en la columna de líneas aparece este símbolo si en la línea correspondiente se ha utilizado como mínimo una "Propiedad especial del objeto".
--	---

Columna "Estado"

	El nombre simbólico o el operando es idéntico a otra entrada de la tabla de símbolos..
	El símbolo todavía no está completo (falta el nombre simbólico o la dirección).

Columnas S/MV/A/C/AC

En las columnas S/MV/A/C/AC se indica si se han asignado propiedades especiales (atributos) a un símbolo:

- S significa que para este símbolo se han creado definiciones de error para el diagnóstico de procesos con el paquete opcional S7-PDIAG (V5).
- MV significa que el símbolo se puede manejar y visualizar con WinCC.
- A significa que al símbolo se le asignó un aviso (SCAN).
- C significa que al símbolo se le han asignado propiedades de comunicación.
- AC significa que el símbolo puede manejar y visualizar rápida y directamente en el editor de programas (Accionar contacto).

Haga clic sobre la casilla de verificación para activar o desactivar las "Propiedades especiales del objeto". También puede modificar las "Propiedades especiales del objeto" con el comando de menú **Edición > Propiedades especiales del objeto**.

Columna "Símbolo"

El nombre de un símbolo puede comprender 24 caracteres como máximo.

En la tabla de símbolos no se pueden asignar nombres a los operandos de bloques de datos (DBD, DBW, DBB, DBX). Sus nombres están predefinidos por haberlos declarado en los bloques de datos.

Para bloques de organización (OBs), así como para algunos bloques de funciones de sistema (SFBs) y funciones de sistema (SFCs), la tabla de símbolos dispone de registros estándar que pueden ser importados a la tabla de símbolos de su programa S7 cuando ésta es editada. El archivo importado se encuentra en el directorio de STEP 7 bajo ...\\S7data\\Symbol\\Symbol.sdf.

Columna "Dirección"

Una dirección sirve para identificar un operando determinado.

Ejemplo: entrada E 12.1

Al introducir una dirección, se comprueba automáticamente si su sintaxis es correcta.

Columna "Tipo de datos"

Puede elegir entre los tipos de datos disponibles en STEP 7. En el campo aparece un tipo de datos predeterminado que se puede cambiar si es necesario. Si el cambio no concuerda con la dirección o si la sintaxis es errónea, aparecerá un mensaje de error al salir del campo.

Columna "Comentario"

Es posible asignar comentarios a cada uno de los símbolos. Combinando símbolos breves con comentarios detallados se crea tanto un programa efectivo como una buena documentación del mismo. Los comentarios comprenden un máximo de 80 caracteres.

Conversión a variables C

Puede elegir símbolos de la tabla de símbolos de un programa M7 y, mediante el software opcional ProC/C++, convertirlos a las variables C correspondientes.


8.5.2 Direcciones y tipos de datos admisibles en la tabla de símbolos

Sólo hay una notación posible para toda la tabla de símbolos. La conmutación entre alemán (anteriormente SIMATIC) e inglés (anteriormente IEC) debe efectuarse en el Administrador SIMATIC eligiendo el comando de menú **Herramientas > Preferencias** y la ficha "Idioma".

Inglés	Alemán	Explicación:	Tipo de datos:	Direcciones:
I	E	Bit de entrada	BOOL	0.0..65535.7
IB	EB	Byte de entrada	BYTE, CHAR	0..65535
IW	EW	Palabra de entrada	WORD, INT, S5TIME, DATE	0..65534
ID	ED	Palabra doble de entrada	DWORD, DINT, REAL, TOD, TIME	0..65532
Q	A	Bit de salida	BOOL	0.0..65535.7
QB	AB	Byte de salida	BYTE, CHAR	0..65535
QW	AW	Palabra de salida	WORD, INT, S5TIME, DATE	0..65534
QD	AD	Palabra doble de salida	DWORD, DINT, REAL, TOD, TIME	0..65532
M	M	Bit de marcas	BOOL	0.0..65535.7
MB	MB	Byte de marcas	BYTE, CHAR	0..65535
MW	MW	Palabra de marcas	WORD, INT, S5TIME, DATE	0..65534
MD	MD	Palabra doble de marcas	DWORD, DINT, REAL, TOD, TIME	0..65532
PIB	PEB	Byte de entrada de periferia	BYTE, CHAR	0..65535
PQB	PAB	Byte de salida de periferia	BYTE, CHAR	0..65535
PIW	PEW	Palabra de entrada de periferia	WORD, INT, S5TIME, DATE	0..65534
PQW	PAW	Palabra de salida de periferia	WORD, INT, S5TIME, DATE	0..65534
PID	PED	Palabra doble de entrada de periferia	DWORD, DINT, REAL, TOD, TIME	0..65532
PQD	PAD	Palabra doble de salida de periferia	DWORD, DINT, REAL, TOD, TIME	0..65532
T	T	Temporizador	TIMER	0..65535
C	Z	Contadores	COUNTER	0..65535
FB	FB	Bloque de función	FB	0..65535
OB	OB	Bloque de organización	OB	0..65535
DB	DB	Bloque de datos	DB, FB, SFB, UDT	0..65535
FC	FC	Función	FC	0..65535
SFB	SFB	Bloque de función de sistema	SFB	0..65535
SFC	SFC	Función de sistema	SFC	0..65535
VAT	VAT	Tabla de variables		0..65535
UDT	UDT	Tipo de datos de usuario	UDT	0..65535


8.5.3 Símbolos incompletos y ambiguos en la tabla de símbolos

Símbolos incompletos

Existe la posibilidad de guardar símbolos incompletos. Así puede indicar en primer lugar, por ejemplo el nombre y fijar más tarde la dirección. La ventaja es que puede interrumpir en cualquier momento el trabajo en la tabla de símbolos y guardar el estado intermedio. Los símbolos incompletos se identifican en la columna "Estado" con el símbolo . No obstante, para poder utilizar el símbolo mientras se va creando el programa (sin que aparezca un mensaje de error), deberá haber introducido el nombre del mismo, la dirección y el tipo de datos.

Crear símbolos ambiguos

Un símbolo será ambiguo si se introduce en la tabla con un nombre (símbolo) o dirección que ya existan en otro símbolo. En tal caso, tanto el símbolo que existía ya como el que se ha insertado se declararán

ambiguos. Este estado se identifica en la columna "Estado" con el símbolo .

Dicho caso se presenta, por ejemplo, cuando se copia y se inserta un símbolo, modificando luego ligeramente la entrada en la copia.

Identificar símbolos ambiguos

Los símbolos ambiguos se destacan gráficamente (color, tipo de letra) en la tabla de símbolos. Ello indica que es necesario corregirlos. Es posible hacerse mostrar todos los símbolos o filtrar la visualización de manera que se vean sólo los símbolos unívocos o sólo los ambiguos.

Eliminar la ambigüedad

Un símbolo ambiguo se convertirá en un símbolo unívoco cuando se modifique el nombre o la dirección que haya causado la ambigüedad. Si hay dos símbolos ambiguos y uno de ellos se corrige, el otro volverá a ser unívoco.

8.6 Métodos para introducir símbolos globales

Existen tres métodos distintos para introducir los símbolos que se utilizarán en la programación:

- **Entrada en un cuadro de diálogo**
Al editar el programa, puede abrir un cuadro de diálogo en una ventana y definir allí un nuevo símbolo. Este procedimiento es especialmente apropiado para definir símbolos de forma individual, como p.ej. si, mientras está programando, se da cuenta que hace falta un símbolo o que es necesario corregir un símbolo ya existente. De esta forma no es necesario visualizar la tabla de símbolos.
- **Entrada en la tabla de símbolos**
En la "tabla de símbolos" se pueden introducir directamente los símbolos y las direcciones absolutas correspondientes. Este procedimiento es recomendable para introducir varios símbolos a la vez y crear por primera vez la tabla de símbolos. Los símbolos ya definidos se visualizan entonces en la pantalla, ofreciendo así una visión de conjunto.
- **Importar tablas de símbolos de otros editores de tablas**
Los datos de la tabla de símbolos también se pueden crear con un editor de tablas diferente (p.ej. Microsoft Excel), importando después a la tabla de símbolos el archivo creado con dicho editor.

8.6.1 Observaciones generales para introducir símbolos

Para poder introducir símbolos en la tabla, vaya a la primera línea vacía de la misma y rellene los campos. Para insertar nuevas líneas delante de la línea actual en la tabla de símbolos, ejecute el comando **Insertar > Símbolo**. Si la línea que se encuentra encima de la posición del cursor ya contiene un operando, al insertar nuevos símbolos se rellenarán automáticamente las columnas "Dirección" y "Tipos de datos": se adoptará la dirección que contenga la línea anterior y se introducirá el tipo de datos predeterminado.

Si desea modificar entradas ya existentes, utilice los comandos del menú "Edición". A continuación, guarde y cierre la tabla de símbolos. También es posible guardar símbolos que no se hayan terminado de definir.

Al introducir símbolos hay que prestar atención a las siguientes particularidades:

Columna	Nota
Símbolo	El nombre debe ser unívoco en toda la tabla de símbolos. Al aceptar el nombre en este campo y al salir del mismo, se le indicará si el símbolo es ambiguo. El símbolo puede tener como máximo 24 caracteres. No se admiten las comillas ".
Dirección	Si acepta el contenido de este campo o sale del mismo, se comprobará si la dirección dada es válida.
Tipo de datos	Después de introducir la dirección, a este campo se le asigna un valor predeterminado. Si lo modifica, entonces se comprueba si el nuevo tipo de datos corresponde a la dirección.
Comentario	Aquí puede introducir notas para describir brevemente (máx. 80 caracteres) las funciones del símbolo. Este campo de comentario es opcional.

8.6.2 Introducir símbolos globales en un cuadro de diálogo

Aquí se describe cómo modificar símbolos existentes o definir nuevos símbolos en un cuadro de diálogo al programar bloques y sin tener que abrir la tabla de símbolos.

Este procedimiento resulta muy útil si desea editar un solo símbolo. Para modificar varios símbolos es recomendable abrir la tabla de símbolos y trabajar allí directamente.

Activar la visualización de símbolos en el bloque

Estando abierto un bloque, visualice los símbolos eligiendo el comando **Ver > Representación simbólica**. Una marca de verificación delante del comando de menú indica que está activada la representación simbólica.

Definir símbolos al introducir el programa

1. Asegúrese de que en la ventana del bloque esté activada la representación simbólica (comando **Ver > Representación simbólica**).
2. En el área de instrucciones de su programa, seleccione la dirección absoluta que desee asignar a un símbolo.
3. Seleccione el comando de menú **Edición > Símbolo**.
4. Rellene el cuadro de diálogo que se visualiza entonces. Introduzca un símbolo y cierre el cuadro.

El símbolo definido se introduce en la tabla de símbolos. Los datos que pudieran causar símbolos ambiguos se rechazarán con un mensaje de error.

Editar en la tabla de símbolos

Con el comando de menú **Herramientas > Tabla de símbolos** puede abrir la tabla de símbolos para su edición.

8.6.3 Introducir símbolos globales en la tabla de símbolos

Abrir una tabla de símbolos

Existen dos posibilidades para abrir una tabla de símbolos:

- Haciendo doble clic en la tabla de símbolos en la ventana del proyecto,
- Seleccionando la tabla de símbolos en la ventana del proyecto y activando luego el comando **Edición > Abrir objeto**.

La tabla de símbolos del programa actual se muestra en una ventana por separado. Ahora se puede proceder a crear o modificar símbolos. La tabla de símbolos está vacía cuando se abre por primera vez.

Introducir símbolos

Para poder introducir nuevos símbolos en la tabla, vaya a la primera línea vacía de la misma y rellene los campos. Para insertar nuevas líneas vacías delante de la línea actual en la tabla de símbolos, active el comando **Insertar > Símbolo**. Las entradas ya existentes se pueden copiar con los comandos del menú "Edición" y modificar luego. A continuación, guarde y cierre la tabla de símbolos. También es posible guardar símbolos que no se hayan terminado de definir.

Ordenar símbolos

Los registros de la tabla de símbolos se pueden ordenar alfabéticamente por símbolos, por direcciones, por tipos de datos o por comentarios.

El criterio de ordenación, se puede modificar p.ej. en el cuadro de diálogo que se abre al elegir el comando de menú **Ver > Ordenar....**

Filtrar símbolos

Activando filtros puede seleccionar aquella parte de los registros de la tabla de símbolos que desee visualizar.

Con el comando **Ver > Filtrar** se abre el cuadro de diálogo "Filtrar".

Allí puede definir los criterios que deben cumplir los registros para que se puedan filtrar. Se puede filtrar por:

- nombres, direcciones, tipos de datos, comentarios
- símbolos con atributo de M+V, símbolos con propiedades de comunicación, símbolos para variables binarias en avisos (marca o entrada del proceso)
- símbolos con estado "válido", "no válido (ambiguo, incompleto)"

Los criterios están combinados entre sí mediante Y (AND). Los registros filtrados comienzan con las secuencias de caracteres indicadas.

Si desea obtener más información acerca del cuadro de diálogo "Filtrar", abra la Ayuda en pantalla pulsando la tecla de función F1.

8.6.4 Mayúsculas y minúsculas en los símbolos

Entre mayúsculas y minúsculas ya no se hace distinción.

En las versiones anteriores de STEP 7 se podían definir símbolos que se diferenciaban sólo por la escritura en mayúsculas o minúsculas de algunos caracteres. Esta característica ha sido modificada a partir de la versión 4.02 de STEP 7. A partir de ésta última, ya no es posible diferenciar los símbolos por su escritura en mayúsculas o minúsculas.

Esta modificación responde al deseo de nuestros clientes, puesto que así se reducen considerablemente las posibles causas de errores en los programas. Limitando así la definición de símbolos se apoyan también los objetivos de PLCopen relativos a la definición de un estándar para los programas transferibles.

De ahora en adelante no se asistirá la diferenciación de los símbolos basada sólo en el uso de mayúsculas o minúsculas. Hasta ahora se permitía, por ejemplo, la siguiente diferenciación en la tabla de símbolos:

Motor1 = E 0.0

motor1 = E 1.0

Dichos símbolos se diferenciaban por la escritura (mayúsculas/minúsculas) de la primera letra. Este tipo de diferenciación da lugar a numerosas confusiones. Gracias a la definición actual se ha eliminado esta posible causa de errores.

Posibles repercusiones en programas existentes

Si ha utilizado hasta ahora este criterio de diferenciación al definir los símbolos, es posible que se presenten conflictos debido a la nueva definición en caso de que:

- los símbolos **sólo** se diferencien entre sí por su escritura en mayúsculas/minúsculas
- los parámetros **sólo** se diferencien entre sí por su escritura en mayúsculas/minúsculas
- los símbolos **sólo** se diferencien de los parámetros por su escritura en mayúsculas/minúsculas

No obstante, dichos tres casos se pueden analizar y solucionar como se describe a continuación.

Símbolos que sólo se diferencien por su escritura en mayúsculas/minúsculas

Conflicto:

Si la tabla de símbolos no se ha editado todavía con la versión actual del software, al compilarse los archivos fuente se utilizará el símbolo ambiguo que se encuentre más arriba en la tabla de símbolos.

Si la tabla de símbolos ya se ha editado, dichos símbolos no serán válidos, es decir, que al abrir los bloques no se visualizarán los símbolos, en tanto que los archivos fuente que los utilicen ya no se podrán compilar sin errores.

Remedio:

Compruebe si su tabla de símbolos contiene errores. Para ello, ábrala y guárdela nuevamente. Ello permite reconocer los símbolos ambiguos. Estos sólo se podrán visualizar y corregir a través del filtro "Símbolos ambiguos". Corrija luego los archivos fuente que presenten conflictos. En los bloques no será necesario efectuar cambios, puesto que al abrirlos utilizan automáticamente la tabla de símbolos actual (que ya no contiene conflictos).

Parámetros que sólo se diferencien por su escritura en mayúsculas/minúsculas

Conflicto:

Los archivos fuente que contengan dichos interfaces ya no se podrán compilar sin errores. Aunque los bloques que contengan dichos interfaces todavía se pueden abrir, no es posible acceder al segundo de dichos parámetros. Al guardar, los accesos al segundo parámetro se aplican automáticamente al primer parámetro.

Remedio:

Para comprobar qué bloques contienen dichos conflictos, es recomendable generar un archivo fuente para todos los bloques de un programa utilizando la función "Generar fuente". Si se presentan errores al intentar compilar de nuevo el archivo fuente creado, significa que existe un conflicto.

Corrija los archivos fuente haciendo unívocos los parámetros, por ejemplo, mediante la función "Buscar/reemplazar" y compile dichos archivos nuevamente.

Símbolos que sólo se diferencien de los parámetros por su escritura en mayúsculas/minúsculas

Conflicto:

Si los símbolos globales y locales de un archivo fuente sólo se diferencian por su escritura en mayúsculas/minúsculas y si no se han utilizado caracteres de identificación para los símbolos globales ("nombre simbólico") o locales (#nombre simbólico), al compilar se utilizará siempre el símbolo local. Ello cambiará el código máquina.

Remedio:

En este caso es recomendable generar nuevamente una fuente para cada uno de los bloques. Así se les adjudican automáticamente los correspondientes caracteres de identificación a los accesos locales y globales, procesándose correctamente en los siguientes procesos de compilación.

8.6.5 Exportar e importar tablas de símbolos

La tabla de símbolos visualizada se puede exportar a un archivo de texto, p.ej., para modificarla con un editor cualquiera.

Asimismo, es posible importar a la tabla de símbolos otras tablas que se hayan creado con una herramienta diferente para seguir editándolas en la primera. La función "Importar" permite, p.ej., incorporar en la tabla de símbolos las listas de asignación que se hayan creado con STEP5/ST (después de su conversión).

Es posible elegir entre los formatos de archivos *.SDF, *.ASC, *.DIF y *.SEQ.

Reglas para exportar

De la tabla de símbolos se puede exportar bien su totalidad, bien una parte, o bien sólo las líneas que se hayan seleccionado.

No es posible exportar las propiedades de los símbolos que se ajusten con el comando Edición Propiedades especiales del objeto.

Reglas para importar

- Para bloques de función de sistema (SFB), funciones de sistema (SFC) y bloques de organización (OB) de uso frecuente, encontrará símbolos predefinidos en el archivo ...\\S7DATA\SYMBOL\SYMBOL.SDF que podrá utilizar en caso necesario.
- Las propiedades de los símbolos que se ajusten con el comando **Edición > Propiedades especiales del objeto** no se tienen en cuenta ni al exportar ni al importar la tabla.

8.6.6 Formatos de archivos para importar/exportar una tabla de símbolos

Se pueden importar/exportar los siguientes formatos de archivo desde/hacia la tabla de símbolos:

- Formato ASCII (ASC)
- Data Interchange Format (DIF)
Los archivos DIF se pueden abrir, editar y guardar de nuevo en Microsoft Excel.
- System Data Format (SDF)
Los archivos SDF se pueden abrir, editar y guardar de nuevo en Microsoft Excel.
 - Para la importación y exportación de datos de la aplicación Access de Microsoft utilice el formato SDF.
 - En ACCESS, elija el formato de datos "Texto (con carácter separador)".
 - Utilice las comillas (") como carácter separador de textos.
 - Utilice la coma (,) como carácter separador de campos.
- Lista de asignación (SEQ)
Cuidado: ¡Al exportar la tabla de símbolos a un archivo del tipo SEQ, los comentarios que excedan los 40 caracteres quedarán cortados tras el 40º carácter!

Formato ASCII (ASC)

Tipo de archivo:	*.ASC			
Estructura:	Longitud del registro, Coma separadora, Registro			
Ejemplo:	126,verde_fase_peat	T	2	TIMER Duración de la fase verde para peatones
	126,rojo_peat	A 0.0	BOOL	Rojo para peatones

Data Interchange Format (DIF)

Tipo de archivo:	*.DIF
Estructura:	Un archivo DIF se compone del encabezado (header) y los datos:

Encabezado	TABLE	Inicio de un archivo DIF
	0,1	
	"<Título>"	Cadena de caracteres de comentario
	VECTORS	Número de registros en el archivo
	0,<Número de registro>	
	""	
	TUPLES	Número de campos de datos de un registro
	0,<Cantidad de columnas>	
	""	
	DATA	Identificador de final del encabezado e inicio de datos
	0,0	
	""	
Datos (para cada registro)	<Tipo>,<valor numérico>	Identificador del tipo de datos, valor numérico.
	<Zeichenkette>	Sección alfanumérica, o
	V	Si no se utiliza la sección alfanumérica

Encabezado: El encabezado del archivo debe contener los tipos de registro TABLE, VECTORS, TUPLES y DATA en el orden indicado. En los archivos DIF, antes de DATA se pueden incluir opcionalmente otros tipos de registro. No obstante, éstos serán ignorados por el editor de símbolos.

Datos: En la sección de datos, cada registro consta de 3 partes: el identificador de tipo, un valor numérico y una sección alfanumérica.

Los archivos DIF se pueden abrir, editar y volver a guardar con Microsoft Excel. No obstante, hay que renunciar a la utilización de la diéresis y de otros caracteres especiales de diversos idiomas.

System Data Format (SDF)

Tipo de archivo:	*.SDF
Estructura:	Cadena de caracteres entre comillas, secciones separadas por comas
Ejemplo:	"verde_fase_peat","T 2","TIMER","Duración de la fase verde para peatones" "rojo_peat","A 0.0","BOOL","Rojo para peatones"

Para abrir un archivo SDF desde Microsoft Access, seleccione el formato de archivo "Texto (con caracteres de separación)". Introduzca las comillas como carácter separador de textos ("), y la coma (,) como carácter separador de campos.

Lista de asignación (SEQ)

Tipo de archivo:	*.SEQ
Estructura:	TAB Dirección TAB Símbolo TAB Comentario CR
Ejemplo:	T 2 verde_fase_peat Duración de la fase verde para peatones A 0.0 rojo_peat Rojo para peatones

TAB representa el carácter de tabulación (09H),
CR representa un salto de línea con la tecla INTRO (0DH).

8.6.7 Editar áreas en tablas de símbolos

A partir de la versión V5.3 de STEP 7 es posible marcar y editar áreas relacionadas de una tabla de símbolos. Puede copiar esas áreas en otra tabla de símbolos, cortarlas, insertarlas o borrarlas.

Eso le permite transferir datos cómodamente de una tabla de símbolos a otra y actualizar las tablas con mayor facilidad.

Áreas seleccionables:

- Puede marcar celdas completas haciendo clic en la primera columna. Si marca todas las celdas desde la columna "Estado" hasta la columna "Comentario", también se considera una selección de líneas.
- Puede seleccionar celdas de forma individual o conjunta si se trata de celdas relacionadas en forma de área. Para poder editar un área marcada, todas las celdas deben permanecer a las columnas "Símbolo", "Dirección", "Tipo de datos" y "Comentario". Si la selección no es válida, los comandos de menú para la edición no están activos.
- Las columnas S, MV, A, C, AC contiene las propiedades especiales del objeto correspondientes a los símbolos y sólo se copian si en el cuadro de diálogo "Preferencias", están ajustadas como llamables con el comando de menú **Herramientas > Preferencias**, activando la casilla de verificación "Copiar propiedades especiales del objeto" y se ha seleccionado líneas completa como un área.
- La copia de los contenidos de las columnas S, MV, A, C, AC depende de si estas columnas se visualizan o están ocultas. La vista puede cambiarse con el comando de menú **Ver > Columnas S, MV, A, C, AV**.

Proceda de la siguiente manera:

1. Seleccione el área deseada en la tabla de símbolos que desea editar:
 - Para ello haga clic con el **ratón** en la línea de inicio que desee y mueva el ratón con el botón izquierdo presionado por encima de toda el área que desea seleccionar o bien
 - Seleccione el área deseada con el **teclado** pulsando las teclas Mayúsculas + teclas de cursor.
2. El área marcada se representa en color de selección. La primera celda seleccionada de la tabla de símbolos se marca con un borde.
3. Edite el área seleccionada de la forma que desee.

9 Crear bloques y librerías

9.1 Seleccionar el método de creación

Dependiendo del lenguaje de programación utilizado para representar el programa, se puede utilizar la edición incremental y/o la edición orientada a la fuente.

Editores incrementales para los lenguajes de programación KOP, FUP, AWL o S7-GRAPH

Con los editores incrementales para KOP, FUP, AWL o S7-GRAPH es posible crear bloques que se depositan en el programa de usuario. Es recomendable que elija la edición incremental si desea que sus entradas se comprueben inmediatamente. Este tipo de edición es adecuado también para los programadores principiantes. En la edición incremental se comprueba inmediatamente la sintaxis de cada línea o elemento. En caso de detectarse errores, éstos se visualizan inmediatamente, debiendo ser corregidos antes de finalizar la edición. Las entradas con sintaxis correcta se compilan automáticamente y se depositan en el programa de usuario.

Los símbolos utilizados se deben definir antes de editar las instrucciones. En caso de que no existan determinados símbolos, el bloque no se puede compilar por completo. No obstante, este "estado intermedio incoherente" se podrá guardar.

Editores (de texto) orientados a la fuente para los lenguajes de programación AWL, S7-SCL o S7-HiGraph

Con estos editores se crean **fuentes** que se compilan luego en bloques.

Elija la edición orientada a la fuente si desea introducir un programa rápidamente.

En este tipo de edición, el programa o un bloque se editan en un archivo de texto que se compila luego.

Los archivos de texto (fuentes) se depositan en la carpeta de fuentes de su programa S7, por ejemplo en forma de **fuelle AWL** o de **fuelle SCL**. Un archivo fuente puede contener la lógica de uno o varios bloques. Con los editores de texto de AWL o de SCL se puede crear el código para **OBs, FBs, FCs, DBs y UDTs** (tipos de datos de usuario), es decir, también para un programa de usuario entero. El programa entero de una CPU, es decir, todos los bloques, puede estar depositado en un solo archivo de texto.

Al compilar el archivo fuente se crearán los bloques que luego se depositarán en el programa de usuario. Antes de la compilación es preciso definir los símbolos utilizados. El compilador anunciará los posibles errores una vez que se empieza a compilar.

Por lo que respecta a la compilación, es importante tener en cuenta la sintaxis prescrita para el respectivo lenguaje de programación. La sintaxis sólo se verifica al comprobar la coherencia o al compilar la fuente en bloques.

9.2 Elegir el lenguaje de programación

Definir el lenguaje de programación del editor

Al generar un bloque o un archivo fuente hay que indicar en las propiedades del objeto con qué lenguaje de programación y con qué editor se desea crear el bloque o la fuente. Conforme a su elección, se arrancará el editor correspondiente al abrir el bloque o el archivo fuente.

Abrir el editor de lenguaje

El editor de lenguaje se abre en el Administrador SIMATIC haciendo doble clic en el objeto correspondiente (bloque, archivo fuente, etc.), o bien activando el comando **Edición " Abrir objeto** o el botón correspondiente de la barra de herramientas.

Para crear el programa S7 se dispone de los lenguajes de programación indicados en la tabla. El software básico STEP 7 contiene los lenguajes de programación KOP, FUP y AWL. Los demás lenguajes de programación pueden adquirirse opcionalmente.

Gracias a ello tiene la posibilidad de elegir entre diversos lenguajes de programación (esquema de contactos, diagrama de funciones, lista de instrucciones, lenguaje de alto nivel, control secuencial o grafo de estado), pudiendo elegir también si desea programar de forma textual o gráfica.

Mediante la elección del lenguaje de programación se definen también los métodos de introducción admisibles (●).

Lenguaje de programación	Destinatarios	Caso de aplicación	Entrada incremental	Entrada orientada a fuente	Bloque autodocumentable de la CPU
Lista de instrucciones AWL	Usuarios que desean programar a nivel de máquina	Programas optimizados en tiempo de ejecución y uso de memoria	●	●	●
Esquema de contactos KOP	Usuarios familiarizados con esquemas eléctricos	Programación de controles combinacionales	●	–	●
Diagrama de funciones FUP	Usuarios familiarizados con los cuadros lógicos del álgebra de Bool	Programación de controles combinacionales	●	–	●
F-KOP, F-FUP paquete opcional	Usuarios familiarizados con los lenguajes de programación KOP y FUP.	Programación de programas de seguridad para sistemas F	●	–	●
SCL (Structured Control Language) Paquete opcional	Usuarios que han programado con lenguajes de alto nivel tales como PASCAL o C	Programar tareas de procesamiento de datos	–	●	–

Lenguaje de programación	Destinatarios	Caso de aplicación	Entrada incremental	Entrada orientada a fuente	Bloque autodocumentable de la CPU
GRAPH Paquete opcional	Usuarios que desean programar conforme a la tecnología sin tener conocimientos profundos de programación/ autómatas programables	Descripción confortable de procesos secuenciales	•	–	•
HiGraph Paquete opcional	Usuarios que desean programar conforme a la tecnología sin tener conocimientos profundos de programación/ autómatas programables	Descripción confortable de procesos asíncronos, no secuenciales	–	•	–
CFC Paquete opcional	Usuarios que desean programar conforme a la tecnología sin tener conocimientos profundos de programación/ autómatas programables	Descripción de procesos continuos	–	•	–

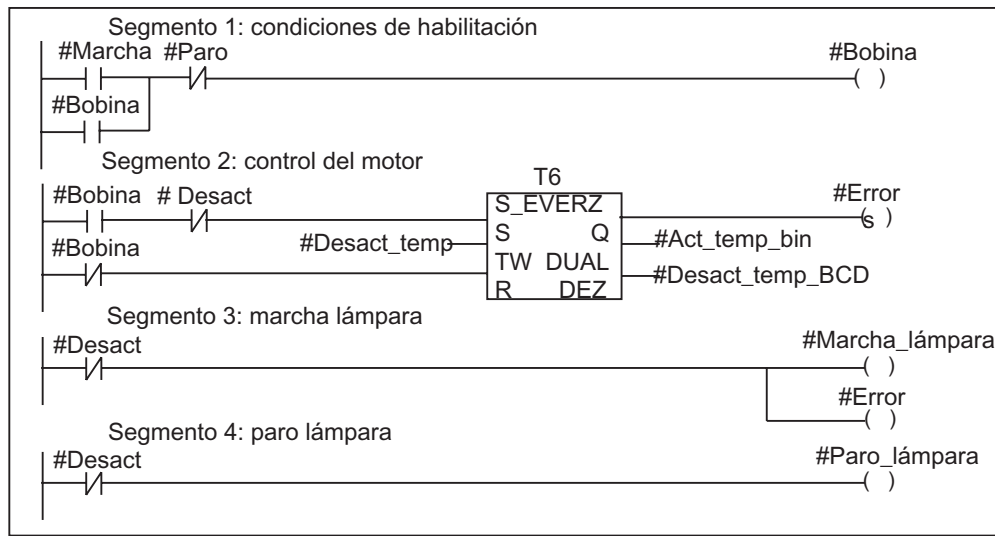
Es posible conmutar entre KOP, FUP y AWL si el bloque en cuestión no contiene errores. Las partes del programa no representables en el lenguaje de destino se representan en AWL.

Los bloques se pueden crear en AWL partiendo de archivos fuente y también se pueden volver a compilar en archivos fuente.

9.2.1 Lenguaje de programación KOP (esquema de contactos)

La representación del lenguaje de programación gráfico KOP (esquema de contactos) es similar a la de los esquemas de circuitos. Los elementos de un esquema de circuitos, tales como los contactos normalmente cerrados y normalmente abiertos, se agrupan en segmentos. Uno o varios segmentos constituyen el área de instrucciones de un bloque lógico.

Ejemplo de segmentos en KOP



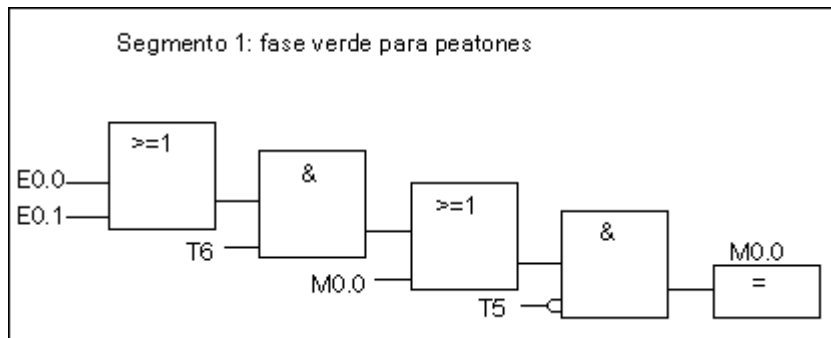
El lenguaje de programación KOP está incluido en el software básico STEP 7. Los programas se crean en KOP con un editor incremental.

9.2.2 Lenguaje de programación FUP (diagrama de funciones)

El lenguaje de programación FUP (diagrama de funciones) utiliza los símbolos gráficos del álgebra booleana para representar la lógica. También es posible representar en conexión directa con los cuadros lógicos funciones complejas, como por ejemplo funciones matemáticas.

El lenguaje de programación FUP está incluido en el software básico STEP 7.

Ejemplo de un segmento en FUP:



Los programas se crean en FUP con un editor incremental.

9.2.3 Lenguaje de programación AWL (lista de instrucciones)

El lenguaje de programación AWL (lista de instrucciones) es un lenguaje textual orientado a la máquina. Las diversas instrucciones equivalen a los pasos de trabajo con los que la CPU ejecuta el programa. Las instrucciones pueden ser agrupadas en segmentos.

Ejemplo de segmentos en AWL

```
U(   Segmento 1: control de la válvula de descarga
O
O   #Bobina
)
UN #Cerrar
= #Bobina
   Segmento 2: indicación "válvula abierta"
U   #Bobina
= #Indic_abierta
   Segmento 3: indicación "válvula cerrada"
UN  #Bobina
= #Indic_cerrada
```

El lenguaje de programación AWL está incluido en el software básico STEP 7. Con él puede editar bloques S7 de forma incremental o crear su programa en una fuente AWL con un editor orientado a la fuente para ser luego compilado en bloques.

9.2.4 Lenguaje de programación S7-SCL

El lenguaje de programación SCL (Structured Control Language), suministrado como software opcional, es un lenguaje textual de alto nivel cuya definición equivale básicamente a la norma IEC 1131-3. Este lenguaje, similar al PASCAL, ayuda a simplificar - en contraposición a AWL - gracias a sus instrucciones de alto nivel, como p.ej. la programación de bucles y de ramificaciones condicionadas. Por esta razón, SCL es especialmente apropiado para el cálculo de fórmulas, para complejos algoritmos de optimización o para gestionar grandes cantidades de datos.

Los programas se crean en S7-SCL con un editor orientado a la fuente (en una fuente SCL).

Ejemplo:

```

FUNCTION_BLOCK FB 20
VAR_INPUT
ENDWERT:          INT;
END_VAR
VAR_IN_OUT
IQ1   :          REAL;
END_VAR
VAR
INDEX:          INT;
END_VAR

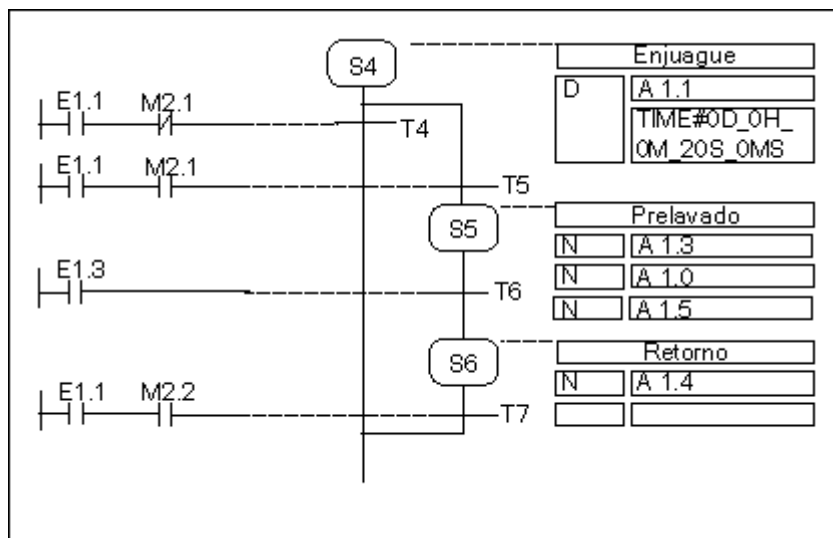
BEGIN
CONTROL:=FALSE;
FOR INDEX:= 1 TO ENDWERT DO
    IQ1:= IQ1 * 2;
    IF IQ1 >10000 THEN
        CONTROL = TRUE
    END_IF;
END_FOR;
END_FUNCTION_BLOCK
    
```

9.2.5 Lenguaje de programación S7-GRAPH (control secuencial)

El software opcional S7-GRAPH es un lenguaje gráfico de programación que permite programar controles secuenciales. Comprende la creación de una cadena de etapas, la definición de los contenidos de las mismas y las condiciones de transición. El contenido de las etapas se define con un lenguaje de programación especial (similar a AWL), en tanto que las condiciones de transición se introducen en una representación del esquema de contactos (parte del lenguaje de programación KOP).

S7-GRAPH permite representar también procesos complejos de forma muy clara, permitiendo así una programación y una búsqueda de errores efectivas.

Ejemplo de un control secuencial en S7 GRAPH



Crear bloques

Con el editor de S7-GRAPH se programa el bloque de función que contiene la cadena de etapas. Los datos de la cadena de etapas, tales como los parámetros del FB, así como las condiciones de etapas y de transición, están contenidos en un bloque de datos de instancia asignado. Con el editor de S7-GRAPH es posible crear automáticamente el DB de instancia mencionado.

Archivo fuente

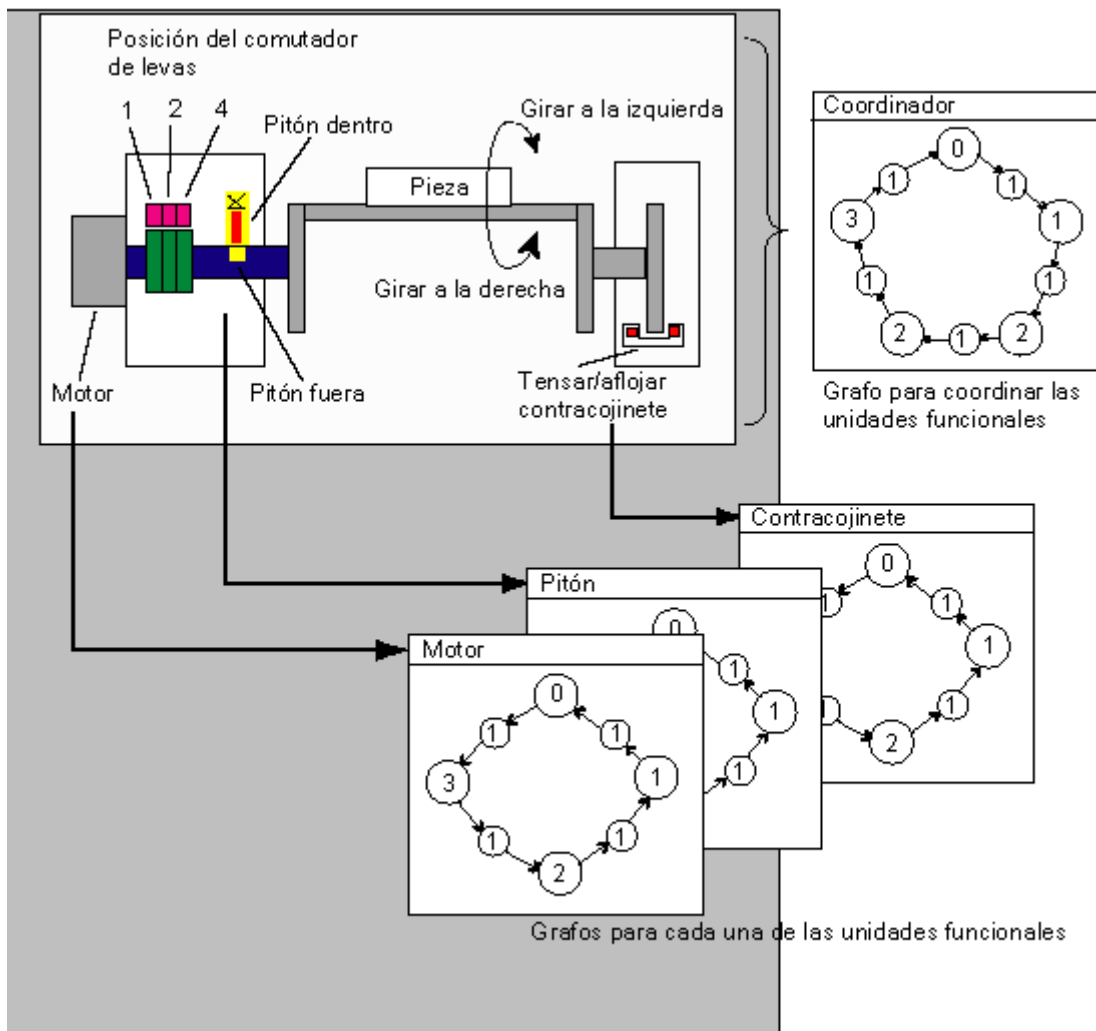
A partir de un bloque de función creado en S7-GRAPH es posible generar un archivo fuente textual (fuente GRAPH) que puede ser interpretado con paneles de operador o visualizadores de textos para visualizar el control secuencial.

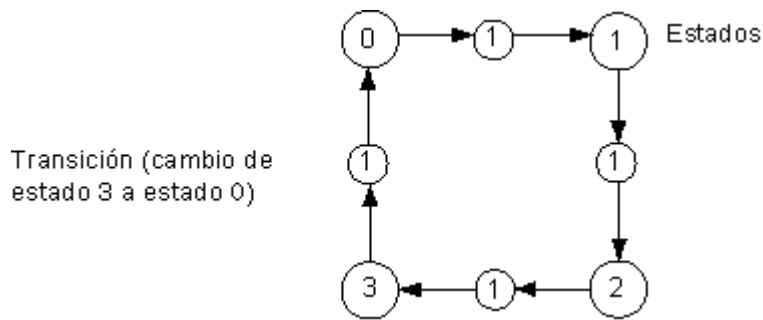
9.2.6 Lenguaje de programación S7-HiGraph (grafo de estado)

El lenguaje gráfico de programación S7-HiGraph (opcional) permite programar algunos bloques de su programa en forma de grafo de estado. Para ello debe subdividir su instalación en unidades funcionales independientes que puedan adoptar diversos estados. Para cambiar de un estado a otro debe definir transiciones. Las acciones asociadas a los estados y las condiciones de transición entre los mismos se describen en un lenguaje detallado similar a AWL.

Para cada unidad funcional se deberá crear un grafo que describa el comportamiento de la misma. Los grafos de una instalación se reúnen en grupos de grafos. Entre los diversos grafos es posible intercambiar avisos para la sincronización de las unidades funcionales.

La representación esquemática de las transiciones de estado de una unidad funcional permite programar sistemáticamente y facilita la búsqueda de errores. Al contrario de S7-GRAPH, en S7-HiGraph sólo hay un estado ("etapa" en S7-GRAPH) activado en un determinado momento. La figura siguiente muestra la creación de grafos para unidades funcionales (ejemplo).





Un grupo de grafos se guarda en una fuente HiGraph en la carpeta "Fuentes" bajo el programa S7. Dicha fuente se compila luego en los bloques S7 del programa de usuario.

Una vez efectuada la última entrada de un grafo (al cerrar la ventana de trabajo) se comprueban la sintaxis y los parámetros formales. Las direcciones y los símbolos se comprueban al compilarse la fuente.

9.2.7 Lenguaje de programación S7-CFC

El software opcional CFC (*C*ontinuous *F*unction *C*hart) es un lenguaje de programación que permite interconectar gráficamente las funciones complejas.

En el lenguaje de programación S7-CFC se programa interconectando gráficamente las funciones existentes. No es necesario programar numerosas funciones estándar, puesto que se dispone de librerías que contienen bloques predefinidos (p.ej. para funciones lógicas, aritméticas, de regulación y de procesamiento de datos). Para poder utilizar el lenguaje CFC no se requieren conocimientos especiales de programación o sobre sistemas de automatización, lo que permite concentrarse en la técnica.

El programa creado se guarda en forma de esquemas CFC. Estos se depositan en la carpeta "Planos" bajo el programa S7. De dichos planos o esquemas se compilan luego los bloques S7 para el programa de usuario.

Los bloques a interconectar se programan para el SIMATIC S7 con uno de los lenguajes de S7, y para el SIMATIC M7, con C/C++.

9.3 Crear bloques

9.3.1 Carpeta de bloques

Puede crear programas para una CPU S7 en forma de

- Bloques
- Fuentes

Para depositar los bloques se dispone de la carpeta "Bloques" bajo Programa S7.

La carpeta "Bloques" contiene los bloques que debe cargar en la CPU S7 para llevar a cabo la solución de automatización. Los bloques lógicos (OBs, FBs, FCs) y los bloques de datos (DB) forman parte de dichos bloques cargables. El bloque lógico OB1 se crea automáticamente con la carpeta "Bloques", puesto que se necesita en todo caso para editar el programa en la CPU S7.

Además, la carpeta "Bloques" comprende los siguientes objetos:

- Los tipos de datos de usuario que usted haya definido. Estos simplifican la programación, pero no se cargan en la CPU.
- Las tablas de variables (VATs) que puede crear para observar y forzar variables, a fin de comprobar su programa. Las tablas de variables no se cargan en la CPU.
- El objeto datos de sistema" (bloques de datos de sistema), que contiene información sobre el sistema (configuración y parámetros del sistema). Estos bloques se crean al configurarse el hardware, abasteciéndose con datos.
- Las funciones de sistema (SFCs) y los bloques de función del sistema (SFBs) que desee llamar en su programa de usuario. Las SFCs y los SFBs no se pueden editar.

Con excepción de los bloques de datos del sistema (que se crean y se editan sólo a través de la configuración del sistema de automatización), los bloques del programa de usuario se editan siempre con el editor asignado. Este se arranca automáticamente haciendo doble clic en el bloque en cuestión.

Nota

En la carpeta "Bloques" también se guardan los bloques que se hayan programado en forma de fuentes y compilado posteriormente.

9.3.2 Tipos de datos de usuario (UDT)

Los tipos de datos de usuario son estructuras de datos creadas por usted mismo y que, una vez definidas, pueden utilizarse en todo el programa de usuario S7:

- como tipos de datos simples o como tipos de datos compuestos en la declaración de variables de bloques lógicos (FC, FB, OB) o como tipo de datos para variables en un bloque de datos (DB). La ventaja es que basta con definir una sola vez una estructura especial de datos que vaya a utilizar varias veces, pudiéndola asignar luego a cualquier cantidad de variables.
- como plantilla para crear bloques de datos de estructura idéntica, es decir que basta con elaborar una sola vez la estructura, pudiendo crear luego los bloques de datos necesarios mediante la simple asignación del UDT (por ejemplo, en el caso de las recetas: la estructura del DB es siempre igual, variando sólo las cantidades).

Al igual que los demás bloques los tipos de datos de usuario (UDTs) se crean en el Administrador SIMATIC o con el editor incremental.

Estructura de un UDT

Después de abrir un UDT se visualiza en una nueva ventana de trabajo la tabla de este tipo de datos de usuario en modo "Declaración".

- La primera y la última línea contienen ya las declaraciones STRUCT y END_STRUCT, respectivamente, que indican el comienzo y el fin del tipo de datos de usuario. Estas dos líneas no se pueden editar.
- Puede editar el tipo de datos de usuario comenzando a partir de la segunda línea de la tabla de declaración en las columnas correspondientes.
- Los tipos de datos de usuario pueden estar formados por:
 - tipos de datos simples,
 - tipos de datos compuestos o
 - tipos de datos de usuario ya existentes.

Los tipos de datos de usuario del programa de usuario de S7 no pueden cargarse en la CPU S7. Se crean y se editan bien sea directamente con editores incrementales, o bien resultan de la compilación de fuentes.

9.3.3 Propiedades del bloque

Mediante las propiedades de bloques podrá identificar mejor los bloques que ha creado (p.ej., en las diferentes versiones) o proteger los bloques contra cambios no autorizados.

Para poder editar las propiedades debe estar abierto el bloque. Además de las propiedades editables, en el cuadro de diálogo correspondiente se visualizan también datos a título de información. Dichos datos no se pueden editar.

Aunque las propiedades de los bloques y los atributos de sistema se indican también en el Administrador SIMATIC, en las propiedades de objeto de un bloque, allí sólo se pueden editar las propiedades NAME, FAMILY, AUTHOR y VERSION.

Una vez insertado el bloque, las propiedades del objeto se editan desde el Administrador SIMATIC. En caso de que un bloque no haya sido creado con el Administrador SIMATIC, sino con uno de los editores, dichos ajustes (p.ej., el lenguaje de programación) se guardan automáticamente en las propiedades del objeto.

Nota

La nemotécnica en la que desee programar sus bloques S7 se elige en el Administrador SIMATIC con el comando **Herramientas > Preferencias** en la ficha "Idioma".

Tabla de las propiedades de bloques

Si desea introducir propiedades de bloques, deberá respetar el orden indicado en la tabla que aparece a continuación.

Palabra clave / propiedad	Significado	Ejemplo
[KNOW_HOW_PROTECT]	Protección de bloques: los bloques compilados con esta opción no permiten ver el área de instrucciones. La interfaz del bloque puede visualizarse, pero no puede modificarse.	KNOW_HOW_PROTECT
[AUTHOR:]	Nombre del autor, nombre de la empresa, del departamento u otros nombres (máx. 8 caracteres, sin blancos)	AUTHOR : Siemens, pero ninguna palabra clave
[FAMILY:]	Nombre de la familia de bloques: p.ej. regulador. (máx. 8 caracteres, sin espacios en blanco).	FAMILY : Reguladores, pero ninguna palabra clave
[NAME:]	Nombre del bloque (máx. 8 caracteres)	NAME : PID, pero ninguna palabra clave
[VERSION: int1 . int2]	Número de versión del bloque (ambos números entre 0 y 15, es decir, 0.0 - 15.15)	VERSION : 3.10
[CODE_VERSION1]	Indica si un FB es multiinstancia. Si desea declarar multiinstancias, el FB no puede llevar dicha propiedad.	CODE_VERSION1

Palabra clave / propiedad	Significado	Ejemplo
[UNLINKED] sólo para DBs	Los bloques de datos con la propiedad UNLINKED sólo se guardan en la memoria de carga, pero no ocupan espacio en la memoria de trabajo y no se vinculan en el programa. No puede accederse a ellos con comandos MC7. Los contenidos de tales DBs se pueden transferir a la memoria de trabajo solamente con la SFC 20 BLKMOV o la SFC 83 READ_DBL, depende de la CPU.	
[Non-Retain]	Los valores de carga de los bloques de datos con este atributo se restauran después de cada desconexión y conexión de red así como después de cada transición STOP-RUN.	
[READ_ONLY] sólo para DBs	Protección contra escritura para bloques de datos; sus datos sólo pueden leerse, pero no modificarse.	READ_ONLY
Bloque protegido contra escritura	Copia de un bloque que ha sido almacenado como referencia con protección contra escritura. Esta propiedad sólo puede ser asignada en el editor de programas con el comando de menú Archivo > Guardar con protección contra escritura.	

La protección de bloques KNOW_HOW_PROTECT tiene las siguientes consecuencias:

- Si desea visualizar posteriormente el bloque compilado en el editor AWL, FUP o KOP incremental, no podrá ver el área de instrucciones del bloque.
- En la tabla de declaración de variables del bloque se visualizan sólo las variables de tipo var_in, var_out y var_in_out. Las variables de tipo var_stat y var_temp permanecerán ocultas.

Asignar propiedades a tipos de bloques

La siguiente tabla muestra qué propiedades pueden asignarse a qué tipos de bloques.

Propiedad	OB	FB	FC	DB	UDT
KNOW_HOW_PROTECT	•	•	•	•	–
AUTHOR	•	•	•	•	–
FAMILY	•	•	•	•	–
NAME	•	•	•	•	–
VERSION	•	•	•	•	–
UNLINKED	–	–	–	•	–
READ_ONLY	–	–	–	•	–
Non-Retain	–	–	–	•	–
Bloque protegido contra escritura	•	•	•	•	•

La propiedad KNOW_HOW_PROTECT se puede ajustar en una fuente al programar el bloque. Dicha propiedad se visualiza en el cuadro de diálogo de propiedades del bloque, pero no se puede modificar.

9.3.4 Visualizar longitudes de bloques

Las longitudes de bloques se visualizan en la unidad "Byte".

Visualizar características de una carpeta de bloques

Las siguientes longitudes se muestran dentro de las características de una carpeta de bloques (vista offline):

- Tamaño (suma de todos los bloques sin datos del sistema) en la memoria de carga del sistema de destino
- Tamaño (suma de todos los bloques sin datos del sistema) en la memoria de trabajo del sistema de destino

No se muestran las longitudes de bloques en el sistema de origen (PG/PC) dentro de las propiedades de la carpeta de bloques.

Visualizar propiedades de un bloque

En las propiedades de un bloque se visualizan:

- Cantidad necesaria de datos locales: Tamaño de los datos locales (en bytes)
- MC7: tamaño del código MC7 (en bytes) o tamaño de los datos útiles DB
- Tamaño en la memoria de carga del sistema de destino
- Tamaño en la memoria de trabajo del sistema de destino: sólo se visualiza al conocerse la asignación del hardware.

Los datos se visualizarán independientemente de si el bloque se encuentra en una ventana de vista online u offline.

Visualizar en el Administrador SIMATIC (Ver > Detalles)

Si la carpeta de bloques está abierta y "**Ver > Detalles**" activado, en la ventana del proyecto se visualizará la memoria de trabajo requerida, independientemente de si la carpeta de bloques se encuentra en una ventana de vista online u offline.

Puede sumar las longitudes de todos los bloques seleccionando los relevantes. En este caso, la suma se mostrará en la barra de estado del Administrador SIMATIC.

En el caso de los bloques no cargables en el sistema de destino (p.ej. VATs) no se muestra la longitud.

En **Ver > Detalles** no se muestran las longitudes de los bloques del sistema de origen (PG/PC).

9.3.5 Comparar bloques

Introducción

Si desea comparar bloques que se encuentran en diferentes lugares, dispone de las posibilidades siguientes para llamar la comparación de bloques:

- Llamar la comparación de bloques en el Administrador SIMATIC con el comando de menú **Herramientas > Comparar bloques**. Haciendo clic en el botón "Ir a" del cuadro de diálogo siguiente "Comparar bloques - Resultados", la ficha "Comparación" del editor de programas (KOP/FUP/AWL) se llenará de contenido.
- Llamar la comparación de bloques en el editor de programas con el **comando de menú Herramientas > Comparar equivalente online/offline**.

A continuación se explica el funcionamiento y se distingue entre bloques lógicos (OBs, FBs, FCs) y bloques de datos (DBs).

Funcionamiento de la comparación de bloques: bloques lógicos

En un primer paso, STEP 7 compara la fecha y la hora de las interfaces de los bloques lógicos que deben compararse. Si son idénticas, STEP 7 parte de la base de que las interfaces son iguales.

Si la fecha y la hora de las interfaces son diferentes, STEP 7 compara los tipos de datos de las interfaces por secciones. Si encuentra diferencias, STEP 7 determina la primera diferencia de cada sección, es decir, la primera diferencia de las correspondientes áreas de declaración. Las multiinstancias y UDTs también se incluyen en la comparación. Si los tipos de datos de las secciones son iguales, STEP 7 compara a continuación los valores iniciales de las variables. Todas las diferencias se visualizan.

En un segundo paso, STEP 7 comprueba el código por segmentos (¡si la opción "Comparar código" no estaba activada, se realiza una comparación de códigos aunque se desactive el botón "Ir a" en el editor de programas!).

En primer lugar se determinan los segmentos insertados y borrados. En el resultado de la comparación, los segmentos que sólo existen en un bloque reciben el comentario "existe sólo en".

Seguidamente se comparan los demás segmentos, hasta que se encuentra la **primera** instrucción divergente. Las instrucciones se comparan del modo siguiente:

- En la parametrización "El operando absoluto es preferente", de acuerdo con los operandos absolutos
- En la parametrización "El símbolo es preferente", de acuerdo con el símbolo

Las instrucciones son iguales cuando el operador y el operando son iguales.

Si los bloques que deben compararse se programaron en diferentes lenguajes de creación, STEP 7 compara sobre la base de AWL.

Particularidad en la comparación offline-offline:

A diferencia de una comparación offline-online, en una comparación offline-offline STEP 7 determina además nombres de variable diferentes. En una comparación offline-online, esto no es posible, ya que online sólo hay disponible unos simbólicos de sustitución.

En la comparación no se incluyen los comentarios de bloques, segmentos y líneas, además de otros atributos de bloques (p. ej. informaciones PDIAG de S7 y avisos).

Funcionamiento de la comparación de bloques: bloques de datos

En un primer paso, igual que en los bloques lógicos, STEP 7 compara la fecha y la hora de las interfaces de los bloques de datos que deben compararse. Si son idénticas, STEP 7 parte de la base de que las estructuras de datos son iguales.

Si la fecha y la hora de las interfaces son diferentes, STEP 7 compara las estructuras hasta la **primera** diferencia en la estructura de datos. Si la estructura de datos de las secciones es igual, STEP 7 compara a continuación los valores iniciales y actuales. Todas las diferencias se visualizan.

Particularidad en la comparación offline-offline:

A diferencia de una comparación offline-online, en una comparación offline-offline STEP 7 determina además nombres de variable diferentes. En una comparación offline-online, esto no es posible, ya que online sólo hay disponible unos simbólicos de sustitución.

En la comparación no se incluyen los comentarios ni las estructuras de UDTs que se utilizan en el bloque de datos.

Funcionamiento de la comparación de bloques: tipos de datos (UDT)

En un primer paso, igual que en los bloques de datos, STEP 7 compara la fecha y la hora de las interfaces de los tipos de datos que deben compararse. Si son idénticas, STEP 7 parte de la base de que las estructuras de datos son iguales.

Si la fecha y la hora de las interfaces son diferentes, STEP 7 compara las estructuras hasta la **primera** diferencia en la estructura de datos. Si la estructura de datos de las secciones es igual, STEP 7 compara a continuación los valores iniciales. Todas las diferencias se visualizan.

Procedimiento: comparación de bloques en el editor de programas

1. Abra un bloque que desee comparar con la versión cargada.
2. Seleccione el comando de menú Herramientas > Comparar equivalente online/offline.

Si puede accederse al interlocutor online, en la ficha "7:Comparación" aparecerán los resultados de la comparación en la parte inferior de la ventana del editor de programas.

Consejo: si dos segmentos son "diferentes", haciendo doble clic en esta línea puede abrir el segmento en cuestión.

Procedimiento: comparación de bloques en el Administrador SIMATIC

1. En el Administrador SIMATIC, marque la carpeta de bloques o sólo los bloques que desea comparar.
2. Seleccione el comando de menú **Herramientas > Comparar bloques**.
3. En el cuadro de diálogo "Comparar bloques", seleccione el tipo de comparación (ONLINE/offline o ruta1/ruta2).
4. En la comparación ruta1/ruta2: en el Administrador SIMATIC, marque la carpeta de bloques o los bloques con los que debe realizarse la comparación. Los bloques se insertarán automáticamente en el cuadro de diálogo.
5. Active la casilla de verificación "incluir los SDBs" si también deben compararse SDBs.
6. Active la casilla de verificación "Comparar código" si desea ejecutar además una comparación de códigos. Si esta casilla de verificación está activada, es posible determinar posteriormente si se ejecuta la función "Comparar detalles". En la comparación de detalles, además de las partes de los bloques relevantes para el proceso (interfaces, y código) se muestran los cambios de nombres de variables locales y nombres de parámetros. Además puede determinar si la comparación se realiza con la opción "incluidos los bloques creados en distintos lenguajes" (p. ej. AWL, FUP). En este caso los bloques se comparan sobre la base de AWL.
7. Pulse el botón "Aceptar" del cuadro de diálogo.

Los resultados de la comparación aparecerán en el cuadro de diálogo siguiente "Comparar bloques - Resultados".
8. Si pulsa el botón "Detalles" en este cuadro de diálogo, aparecerán las propiedades de los bloques comparados (p. ej. momentos de las últimas modificaciones, sumas de verificación, etc.).

Si pulsa el botón "Ir a", se abre el editor de programas y los resultados de la comparación se muestran en la parte inferior de la ventana.

Nota

Al comparar una carpeta de bloques offline con una carpeta de bloques online, sólo se comparan los tipos de bloques cargables (OB, FB, ...).

En la comparación offline/online o ruta1/ruta2 de una multiselección se comparan todos los bloques seleccionados, incluso si se trata de bloques que no pueden cargarse (p. ej. tablas de variables o UDTs).

9.3.6 Recablear

Se pueden recablear los siguientes bloques y operandos :

- Entradas, salidas
- Marcas, temporizadores, contadores
- Funciones, bloques de función

Proceda como sigue:

1. En el Administrador SIMATIC, seleccione la carpeta "Bloques" en la que se encuentran los bloques que desea recablear.
2. Elija el comando de menú **Herramientas > Recablear**.
3. En el cuadro de diálogo "Recablear" elija los cambios deseados (Antiguo operando / Nuevo operando) en la tabla.
4. Elija la opción "Considerar todos los bits a los que accede el operando" si desea recablear todas las áreas de operandos (BYTE, WORD, DWORD).
Ejemplo: Si se indican las áreas de operandos EW0 y EW4, los operandos E0.0 - E1.7 se recablearán en los operandos E4.0 - E5.7. Los operandos del área recableada (p.ej. E0.1) no se podrán introducir luego de forma individual en la tabla.
5. Haga clic en el botón "Aceptar".

Así se inicia el proceso de recableado. Tras el recableado se puede decidir en un cuadro de diálogo si se desea visualizar el archivo de información al respecto. El archivo de información contiene la lista de operandos, "Antiguo operando" y "Nuevo operando". Además, se indican los diversos bloques con la cantidad de recableados que se hayan efectuado en ellos.

Al recablear debe tenerse en cuenta que:

- al recablear un bloque (es decir, al cambiar el nombre), el nuevo bloque no debe existir previamente. Si el bloque ya existe, el proceso se interrumpirá.
- Al recablear un bloque de función (FB), su DB de instancia se asignará automáticamente al FB recableado. El DB de instancia no cambiará (es decir, se mantendrá el número de DB).

9.3.7 Atributos de bloques y parámetros

En los temas de Ayuda de los atributos de sistema se incluye una descripción de los atributos:

Salto a descripciones de lenguajes, ayudas sobre bloques y atributos de sistema

9.4 Trabajar con librerías

Las librerías sirven para depositar componentes reutilizables de programas para SIMATIC S7/M7. Los componentes de programas pueden copiarse de proyectos existentes a una librería, o bien, crearse directamente en la librería, independientemente de los proyectos.

La programación se puede simplificar en gran medida depositando en un programa S7 o en una librería los bloques que se deseen utilizar una y otra vez. De allí se podrán copiar siempre al programa de usuario en cuestión.

Para crear programas S7/M7 en una librería se dispone - a excepción de las funciones de test - de las mismas funciones que en los proyectos.

Crear librerías

Al igual que los proyectos, las librerías se pueden crear utilizando el comando de menú **Archivo > Nuevo**. La nueva librería se crea en el directorio que haya indicado en la ficha "General", una vez elegido el comando de menú **Herramientas > Preferencias**.

Nota

El Administrador SIMATIC permite utilizar nombres de más de 8 letras. Sin embargo, el nombre del directorio de la librería queda cortado a 8 caracteres. Por consiguiente, es imprescindible que los nombres de las librerías se distingan en los primeros 8 caracteres. Entre mayúsculas y minúsculas no se hace distinción. Al abrir se volverá a visualizar el nombre completo en el examinador; al examinar aparecerá el corto.

Tenga en cuenta que en los proyectos de una versión anterior de STEP 7 no se pueden utilizar bloques de librerías de una versión posterior de STEP 7.

Abrir librerías

Para abrir una librería, elija primero el comando **Archivo > Abrir**. Seleccione luego una librería en los cuadros de diálogo siguientes. Entonces se abrirá la ventana de la librería.

Nota

Si no aparece en la lista de librerías la librería deseada, haga clic en el botón "Examinar" del cuadro de diálogo "Abrir". Entonces se visualizará el cuadro de diálogo estándar de Windows con la estructura de directorios en el que podrá buscar la librería.

Tenga en cuenta que el nombre del archivo corresponde siempre al nombre original de la librería que se haya creado, es decir que los cambios de nombre efectuados en el Administrador SIMATIC no tienen efecto en el nivel de archivos.

Una vez seleccionada la librería, ésta se adoptará en la lista de librerías. Las entradas en la lista de librerías se pueden modificar seleccionando el comando de menú **Archivo > Gestionar**.

Copiar librerías

Para copiar una librería, elija el comando de menú **Archivo > Guardar como** y guarde la librería con otro nombre.

Los componentes de la librería, tales como programas, bloques, fuentes, etc. se copian utilizando el comando de menú **Edición > Copiar**.

Borrar librerías

Las librerías se borran utilizando el comando de menú **Archivo > Borrar**.

Los componentes de la librería, tales como programas, bloques, fuentes, etc. se borran utilizando el comando de menú **Edición > Borrar**.

9.4.1 Estructura jerárquica de las librerías

Al igual que los proyectos, las librerías están estructuradas jerárquicamente:

- Las librerías pueden contener programas S7/M7.
- Un programa S7 puede comprender un sola carpeta "Bloques" (programa de usuario), una carpeta "Fuentes" y una carpeta "Esquemas", así como un objeto "Símbolos" (tabla de símbolos).
- Un programa M7 puede contener planos y programas C para módulos M7 programables, así como un objeto "Símbolos" (tabla de símbolos) y una carpeta de bloques para DBs y VATs (tablas de variables).
- La carpeta "Bloques" contiene los bloques que se cargan en la CPU S7. Las tablas de variables (VAT) y los tipos de datos de usuario no se cargan en la CPU.
- La carpeta "Fuentes" comprende las fuentes de los programas creados en los diversos lenguajes de programación.
- La carpeta "Esquemas" (planos) contiene los esquemas CFC (sólo en el caso del software opcional S7-CFC).

Al insertar un nuevo programa S7/M7, se crean automáticamente la carpeta "Bloques", la carpeta "Fuentes" (sólo para S7) y el objeto "Símbolos".

9.4.2 Librerías estándar disponibles

El software estándar STEP 7 contiene las siguientes librerías estándar:

- **Sytem Function Blocks:** funciones de sistema (SFC) y bloques de función de sistema (SFB)
- **S5-S7 Converting Blocks:** bloques para la conversión de programas STEP 5
- **IEC Function Blocks:** bloques para funciones IEC, tales como para editar indicaciones de fecha y hora, para operaciones de comparación, para el tratamiento de cadenas y para seleccionar el máximo y el mínimo
- **Organization Blocks:** bloques de organización estándar (OB)

- **PID Control Blocks:** bloques de función (FBs) para el PID Control
- **Communication Blocks:** funciones (FCs) y bloques de función (FBs) para CPs SIMATIC NET.
- **TI-S7 Converting Blocks:** funciones estándar utilizables en general
- **Miscellaneous Blocks:** bloques para la asignación de sello horario y la sincronización horaria

Si se instala software opcional se pueden añadir más librerías.

Borrar e instalar las librerías adjuntas

Las librerías adjuntas se pueden borrar e instalar luego nuevamente en el Administrador SIMATIC. Para la instalación es preciso volver a ejecutar el programa de instalación Setup de STEP 7.

Nota

Al instalarse STEP 7 las librerías adjuntas se copian automáticamente. Si ya las ha modificado, se sobrescribirán con las originales al instalarse STEP 7 de nuevo.

Por tanto, antes de efectuar cambios es recomendable copiar las librerías adjuntas y editar sólo las copias.

10 Crear bloques lógicos

10.1 Nociones básicas para crear bloques lógicos

10.1.1 Estructura de la ventana del editor de programas

La ventana del editor de programas está dividida en las siguientes áreas:

Vista general

La ficha "Elementos de programa" muestra una perspectiva completa de los elementos de programa que puede insertar en su programa KOP, FUP o AWL. La ficha "Estructura de llamada" muestra la jerarquía de llamada de los bloques en el programa S7 actual.

Declaración de variables

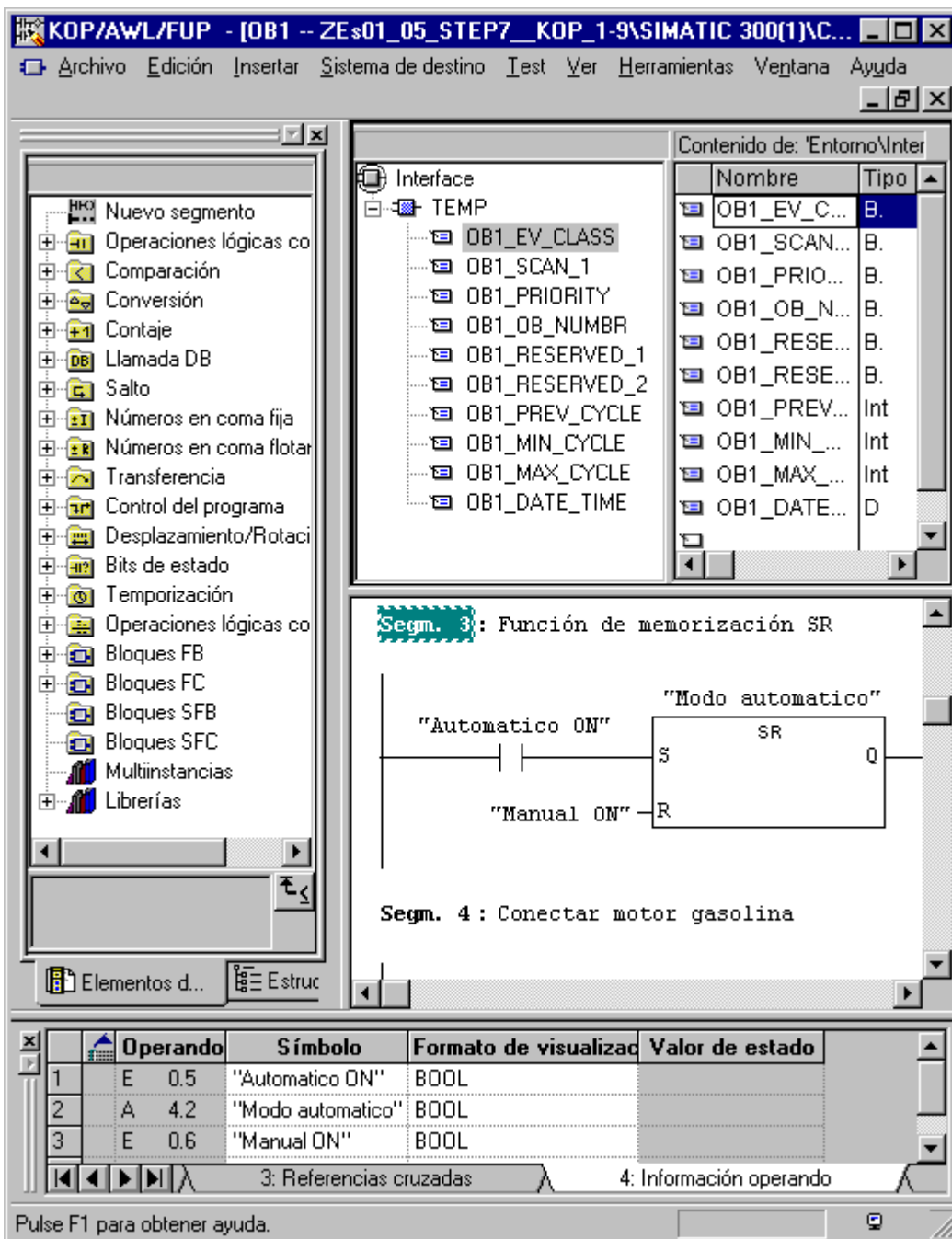
La declaración de variables está dividida en la vista general de variables y en la vista detallada de variables.

Instrucciones

En el área de instrucciones se muestra el código del bloque que debe procesar el PLC. Se compone de una varias redes.

Detalles

Las diferentes fichas de la ventana "Detalles" permiten p.ej. mostrar mensajes de error o información sobre operandos, editar símbolos, forzar operandos, comparar bloques o editar definiciones de error para el diagnóstico de procesos.

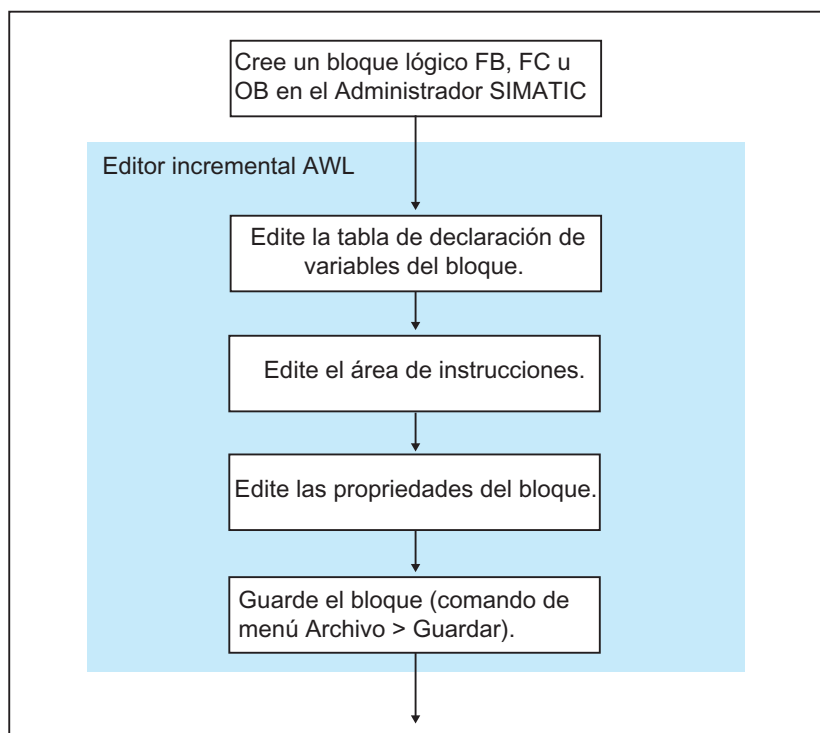


10.1.2 Procedimiento básico para crear bloques lógicos

Los bloques lógicos (OBs, FBs, FCs) se componen de una área de declaración de variables y una área de instrucciones. También tienen propiedades. En otras palabras, al programar se tienen que editar las tres partes siguientes:

- **Declaración de variables.** En la declaración de variables se definen los parámetros, los atributos de sistema de los parámetros y las variables locales del bloque.
- **Área de instrucciones:** En el área de instrucciones se programa el código del bloque que tiene que ser editado por el sistema de automatización. Este código se compone de uno o varios segmentos. Para crear los segmentos puede utilizar, p.ej., los lenguajes de programación lista de instrucciones (AWL), esquema de contactos (KOP) y diagrama de funciones (FUP).
- **Propiedades de bloques:** Las propiedades de los bloques contienen información adicional, como por ejemplo la indicación de fecha y hora o la ruta, que el mismo sistema registra. También puede indicar el nombre, la familia, la versión y el autor del bloque y asignarle atributos de sistema.

En principio es indistinto en qué orden se editan las distintas partes de un bloque lógico. También se pueden corregir/completar.



Nota

Si desea recurrir a símbolos de la tabla de símbolos, primero debería verificarlos y, si es necesario, completarlos.

10.1.3 Preajustes para el editor de programas KOP/FUP/AWL

Antes de comenzar a programar es recomendable que conozca las posibilidades de ajuste para poder trabajar de la forma más cómoda posible y conforme a sus preferencias.

Elija el comando de menú **Herramientas > Preferencias** y se abrirá un cuadro de diálogo compuesto por varias fichas. En las fichas puede efectuar preajustes para programar los bloques, p.ej. en la ficha "General"

- la fuente y el tamaño de letra de los textos y tablas;
- si al crear un nuevo bloque desea visualizar primero los símbolos y comentarios.

Los ajustes relativos al lenguaje, a los comentarios y a los símbolos se pueden modificar durante la edición mediante los comandos del menú **Ver > ...**

El color en el que se representan las selecciones (p.ej. de segmentos o de líneas de instrucciones) se puede cambiar en la ficha "KOP/FUP".

10.1.4 Derechos de acceso a bloques y fuentes

En la ejecución de un proyecto se utiliza a menudo una gestión de datos común, con lo cual puede suceder que varios usuarios quieran acceder al mismo bloque o a la misma fuente de datos.

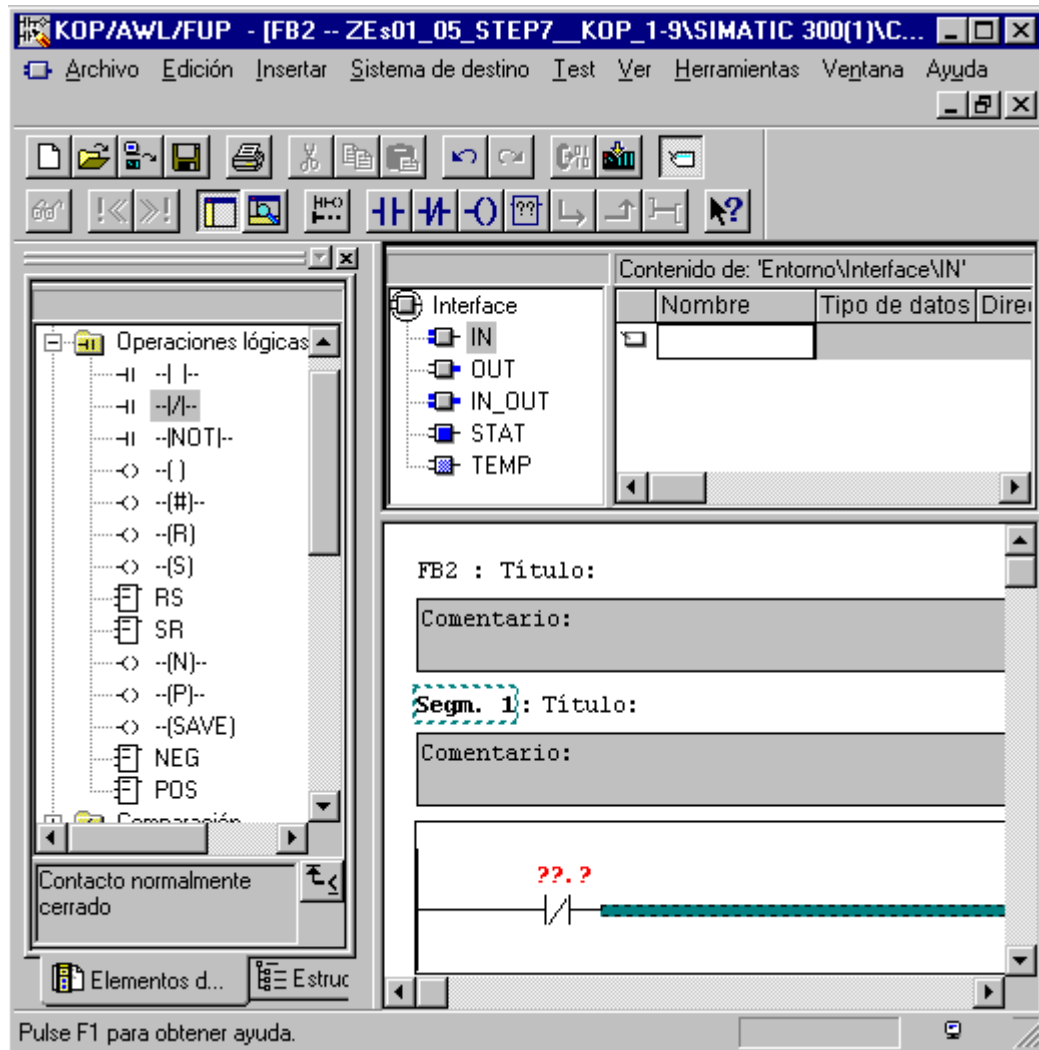
Los derechos de lectura y escritura se adjudican como sigue:

- Ejecución offline:
Al abrir un bloque o una fuente se comprueba si éstos pueden ser abiertos con el derecho 'de escritura'. Si el bloque o fuente ya están abiertos, sólo se puede trabajar con una copia. En el caso de que quiera guardar la copia, se le preguntará si desea sobrescribir el original o guardar la copia con otro nombre.
- Ejecución online:
Si el usuario abre un bloque online a través de un enlace configurado, el bloque offline correspondiente queda bloqueado y con ello se impiden las modificaciones simultáneas en el mismo.

10.1.5 Instrucciones de la vista general de elementos de programa

La ficha "Elementos de programa" de la ventana correspondiente a la vista general comprende elementos de los lenguajes KOP y FUP, así como multiinstancias ya declaradas, bloques ya listos y bloques de librerías. La ficha se puede llamar con el comando de menú **Ver > Vista general**. Los elementos de programa también se pueden adoptar en el área de instrucciones eligiendo el comando de menú **Insertar > Elementos de programa**.

Ejemplo de la ficha "Elementos de programa" en KOP

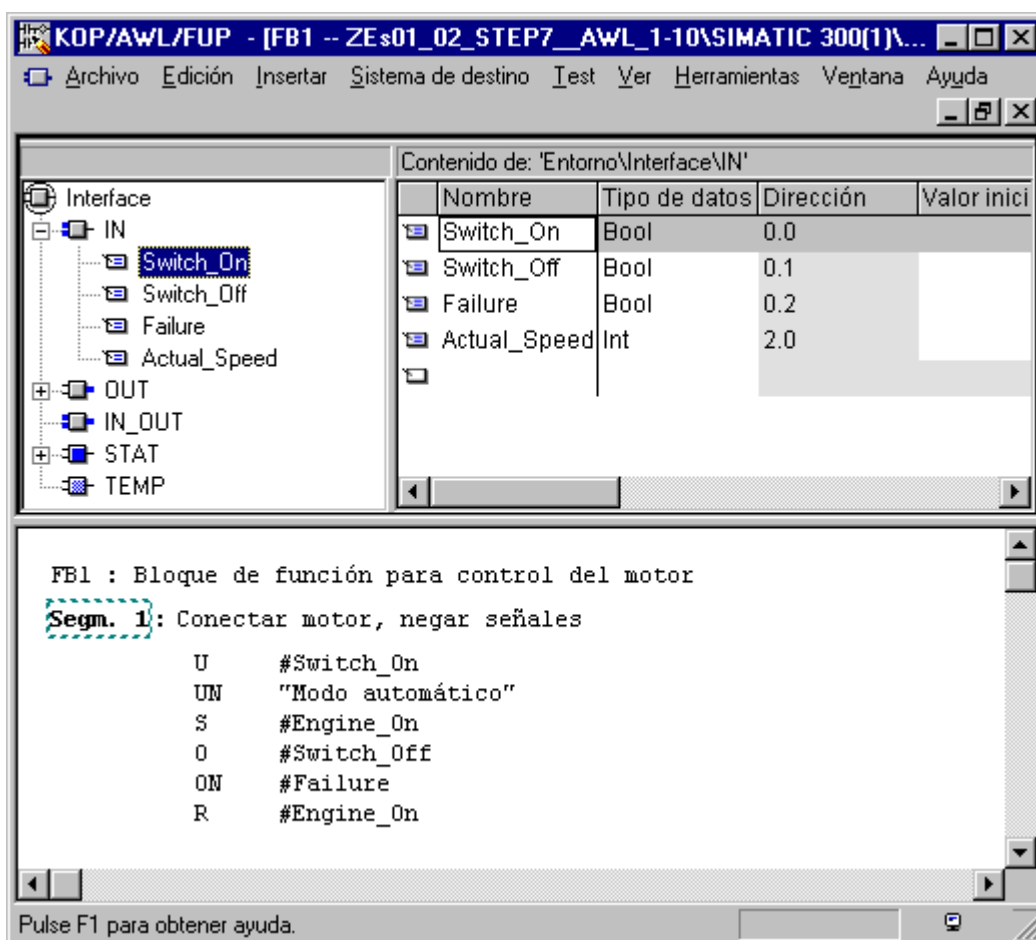


10.2 Editar la declaración de variables

10.2.1 Utilización de la declaración de variables en los bloques lógicos

Tras abrir un bloque lógico aparece una ventana con la vista general de variables y la vista detallada de variables de dicho bloque en la parte superior, así como el área de instrucciones en la parte inferior, donde se edita la lógica del bloque en sí.

Ejemplo: Vistas de variables y área de instrucciones en AWL



En la vista detallada de variables se definen las variables locales, incluyendo los parámetros formales del bloque y los atributos de sistema de los parámetros. Las consecuencias son, entre otras, las siguientes:

- Al declarar las variables se reserva suficiente espacio de memoria en la pila de datos locales para las variables temporales y, en el caso de los bloques de función, para las variables estáticas del DB de instancia que se asociará posteriormente.
- Al definir los parámetros de entrada, salida y de entrada/salida se fija también el "interface" que llamará al bloque en el programa.
- Al declarar variables en un bloque de función, dichas variables determinan (con excepción de las variables temporales) también la estructura de todos los DBs de instancia que se asocien al FB.
- Al fijar los atributos de sistema se les asignan a los parámetros determinadas propiedades, por ejemplo para la configuración de avisos y de enlaces, para funciones de manejo y visualización o para la configuración del control de procesos.

10.2.2 Interacción entre la declaración de variables y el área de instrucciones

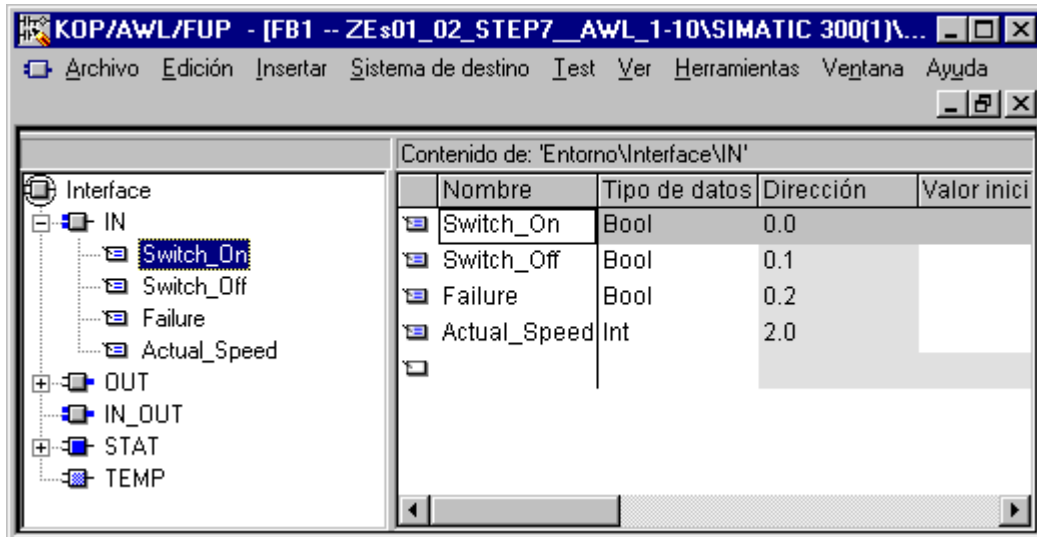
La declaración de variables y el área de instrucciones de los bloques lógicos están estrechamente relacionadas, puesto que al programar en esta última se utilizan los nombres que aparecen en la declaración de variables. Por tanto, los cambios que se hagan en la declaración de variables tienen efecto en toda el área de instrucciones.

Acción en la declaración de variables	Reacción en el área de instrucciones
Nueva entrada correcta	Si existe un código no válido, una variable que no se haya declarado antes será válida ahora.
Cambio correcto de nombre sin modificación de tipo	El símbolo se representará inmediatamente con su nuevo nombre en todas partes.
Cambio de un nombre correcto por un nombre no válido	El código no se modificará.
Cambio de un nombre no válido por un nombre correcto	Si existe un código no válido, éste se convertirá en uno válido.
Cambio de tipo	Si existe un código no válido, éste se convertirá en uno válido. Si existe un código válido, es posible que se convierta en un código no válido.
Borrado de una variable (símbolo) utilizada en el código.	El código válido se convierte en un código no válido.

Los cambios de los comentarios, la entrada incorrecta de una nueva variable, el cambio de un valor inicial o el borrado de una variable no utilizada no afectan al área de instrucciones.

10.2.3 Estructura de la ventana de declaración de variables

La ventana de declaración de variables se compone de la vista general de variables y la vista detallada de variables.



Cuando abre un bloque lógico recién creado, se muestra una vista general de variables predeterminada. Dicha vista sólo contiene los tipos de declaración válidos para el tipo de bloque seleccionado (in, out, in_out, stat, temp) en el orden predefinido. Cuando se crea un nuevo OB se muestra una declaración de variables estándar cuyos valores pueden modificarse.

Los tipos de datos admisibles para los datos locales de los diversos tipos de bloques se indican en el tema Asignar tipos de datos a los datos locales de bloques lógicos.

10.3 Multiinstancias en la tabla de declaración

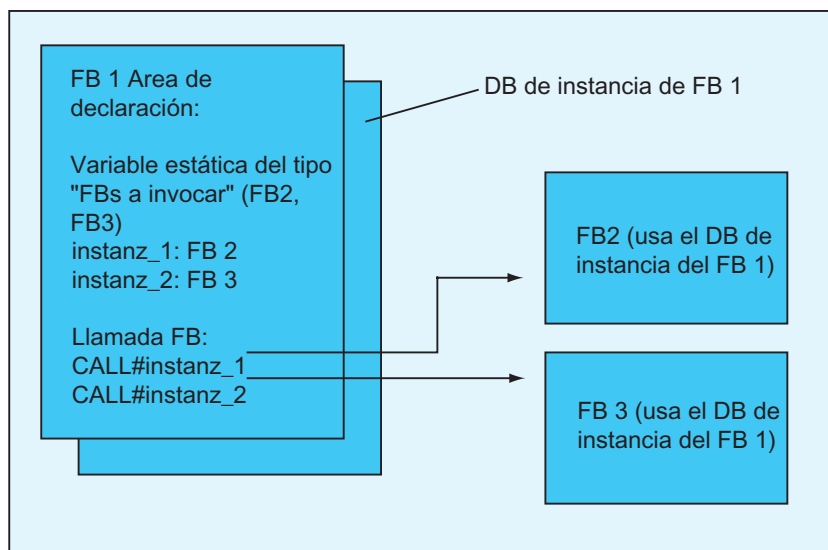
10.3.1 Utilización de multiinstancias

Es posible que, debido a los datos característicos (p.ej. capacidad de memoria) de las CPUs S7 utilizadas, sólo pueda o desee utilizar una cantidad limitada de DBs para los datos de instancia. Si en un FB de su programa de usuario se invocan otros bloques de función ya existentes (jerarquía de llamada de FBs), éstos últimos se pueden llamar sin DBs de instancia propios (es decir, adicionales).

Para solucionarlo se puede proceder de la manera siguiente:

- incorpore los FBs a llamar en forma de variables estáticas en la declaración de variables del FB invocante;
- en este último FB se llaman otros bloques de función sin DBs de instancia propios (es decir, adicionales),
- lo que permite concentrar los datos de instancia en un solo bloque de datos de instancia, pudiéndose así aprovechar mejor el número disponible de DBs.

El siguiente ejemplo muestra la solución descrita: El FB 2 y el FB 3 aprovechan el DB de instancia del FB 1 que es el bloque de función invocante.



Única condición: Es preciso "comunicarle" al bloque de función invocante qué instancias se han de llamar, así como el tipo (de FB) de dichas instancias. Las entradas se deben efectuar en la ventana de declaración del FB invocante. El FB a utilizar debe tener por lo menos una variable o un parámetro del área de datos (es decir, que no sea VAR_TEMP).

No utilice bloques de datos multiinstancia si espera realizar cambios online con la CPU en marcha. Para evitar que el proceso automatizado se vea perturbado por la carga del programa modificado es necesario utilizar bloques de datos de instancia.

10.3.2 Reglas para declarar multiinstancias

Para declarar multiinstancias se aplican las siguientes reglas: se pueden

- Las multiinstancias sólo se pueden declarar en bloques de función creados a partir de la versión 2 de STEP 7 (v. el atributo de bloque en las propiedades del FB).
- Para poder declarar las multiinstancias, el bloque de función se deberá haber creado como FB apto para multiinstancias (preajuste a partir de la versión x.x de STEP 7; deseleccionable en el editor mediante el comando de menú **Herramientas > Preferencias**).
- Al bloque de función en el que se declara una multiinstancia debe asociarse un DB de instancia.
- Una multiinstancia sólo se puede declarar como variable estática (tipo de declaración "stat").

Nota

- También se pueden crear multiinstancias para bloques de función de sistema.
 - Si el FB creado no es apto para multiinstancias y, sin embargo, Ud. precisa esta característica, puede generar una fuente del FB, borrar de ella la propiedad CODE_VERSION1 y volver a compilar el FB.
-

10.3.3 Introducir la multiinstancia en la ventana de declaración de variables

1. Abra el FB desde el que se deben invocar los FBs subordinados.
2. Al declarar las variables del FB invocante, defina una variable estática para cada llamada de un bloque de función, si no desea indicar el bloque de datos de la instancia correspondiente.
 - En la vista general de variables, seleccione el nivel jerárquico "STAT".
 - En la vista detallada de variables, introduzca en la columna "Nombre" una denominación para la llamada del FB.
 - En la columna "Tipo de datos", introduzca la dirección absoluta del FB a llamar, o bien su nombre simbólico.
 - En la columna "Comentario" se pueden introducir explicaciones opcionales.

Llamadas en el área de instrucciones

En caso de declarar multiinstancias, las llamadas de FBs se pueden utilizar sin indicar el DB de instancia.

Ejemplo: Si se ha definido la variable estática "Nombre: Motor_1 , tipo de datos: FB20", la instancia se puede llamar de la siguiente manera:

Call Motor_1 // Llamada del FB 20 sin DB de instancia

10.4 Indicaciones generales para introducir instrucciones y comentarios

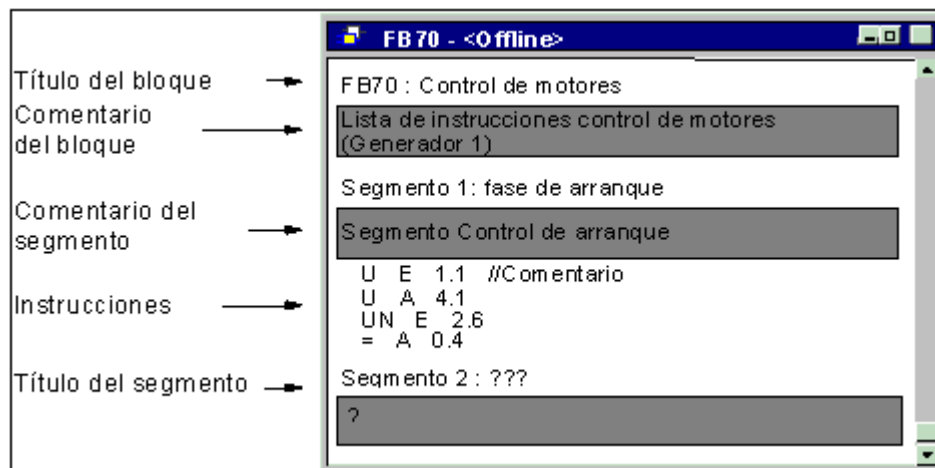
10.4.1 Estructura del área de instrucciones

En el área de instrucciones se programa la lógica del bloque. Dependiendo del lenguaje de programación elegido, deberá introducir instrucciones en segmentos. Tras introducirse una instrucción, el editor realiza inmediatamente una comprobación de sintaxis, indicando los posibles errores en letra cursiva roja.

En la mayoría de los casos, el área de instrucciones de un bloque lógico se compone de varios segmentos que, a su vez, comprenden una serie de instrucciones.

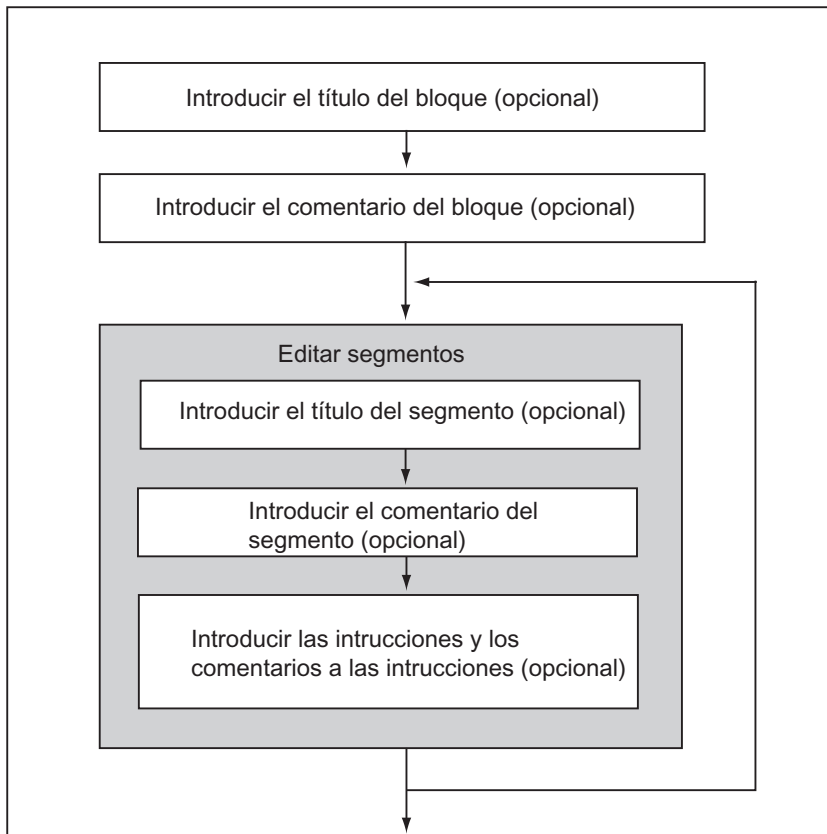
En el área de instrucciones se pueden editar los títulos y los comentarios de los bloques y de los segmentos, así como las líneas de instrucciones en los segmentos.

Estructura del área de instrucciones tomando como ejemplo el lenguaje de programación AWL



10.4.2 Procedimiento para introducir instrucciones

Por regla general, los componentes del área de instrucciones se pueden editar en un orden cualquiera. Al programar un bloque por primera vez se recomienda el siguiente procedimiento:



Además del modo de inserción, los cambios se pueden efectuar también en el modo de sobrescritura. Utilice la tecla **INS** para conmutar entre ambos modos.

10.4.3 Introducir símbolos globales en un programa

Utilice el comando de menú **Insertar > Símbolo** para insertar símbolos en el área de instrucciones de su programa. Si el cursor se encuentra al comienzo, al final o dentro de una cadena de caracteres, ya estará seleccionado el símbolo con el que comienza la cadena (si el símbolo ya existe). Si modifica la cadena de caracteres, el símbolo seleccionado en la lista se corregirá automáticamente.

Los caracteres separadores que indican el comienzo y el final de una cadena son p.ej. un carácter en blanco, un punto o dos puntos. Los caracteres separadores no se interpretan dentro de los símbolos globales.

Para insertar símbolos puede proceder de la siguiente forma:

1. Introduzca en el programa la letra inicial del símbolo deseado.
2. Pulse simultáneamente las teclas CTRL y J para visualizar la lista de símbolos. Aparecerá seleccionado el primer símbolo que tenga la letra inicial indicada.
3. Adopte el símbolo pulsando la tecla INTRO o elija un símbolo diferente.

El símbolo entre comillas quedará insertado, sustituyendo la letra inicial

Regla general: Si el cursor se encuentra al comienzo, al final o dentro de una cadena de caracteres, dicha cadena se reemplazará con el símbolo entre comillas cuando éste sea insertado.

10.4.4 Títulos y comentarios de bloques y segmentos

Los comentarios facilitan la lectura del programa de usuario, aumentando así la eficacia del funcionamiento y la búsqueda de errores. Los comentarios son una parte importante de la documentación del programa y deberían utilizarse siempre.

Comentarios para programas KOP, FUP y AWL:

Se dispone de los siguientes comentarios:

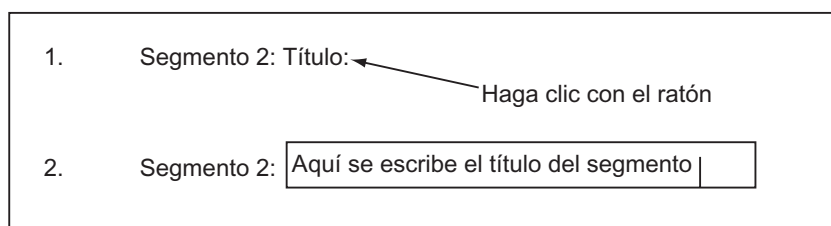
- Título del bloque: título de un bloque (máx. 64 caracteres).
- Comentario del bloque: documenta todo el bloque lógico, como p.ej., la función del bloque.
- Título del segmento: título de un segmento (máx. 64 caracteres).
- Comentario de segmento: documenta la función de los diferentes segmentos.
- Columna de comentario de la vista detallada de variables: comentarios acerca de los datos locales declarados.
- Comentario del símbolo: comentarios que se introdujeron en la tabla de símbolos para un operando en la definición del nombre.
Puede visualizar dichos comentarios con el comando de menú **Ver > Mostrar > Información del símbolo**.

En el área de instrucciones del bloque lógico se pueden introducir los títulos y los comentarios de los bloques y de los segmentos.

Títulos de bloques o de segmentos

Para introducir títulos de bloques o de segmentos, sitúe el cursor en la palabra "Título" que aparecen a la derecha del nombre del bloque o del segmento , p.ej. Segmento 1 : Título). Se abrirá un campo de texto para introducir el título. Éste puede comprender 64 caracteres como máximo.

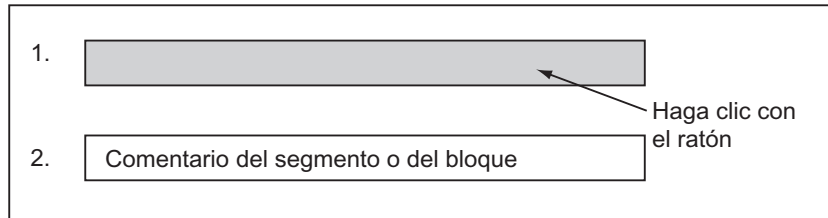
Los comentarios de bloques se refieren al bloque lógico entero. En ellos se puede describir la función de éste último. Los comentarios de segmento se refieren a los diferentes segmentos y describen sus particularidades.



Para asignar títulos de segmento automáticamente, seleccione el comando de menú **Herramientas > Preferencias** y en la ficha "General" haga clic sobre la opción "Asignación automática del título de segmento ". En ese caso se asignará el comentario simbólico del primer operando introducido como título del segmento.

Comentarios de bloques y de segmentos

Los campos de comentario grises se pueden mostrar u ocultar eligiendo el comando de menú **Ver > Mostrar > Comentario**. Haciendo doble clic en uno de dichos campos se abre el cuadro de texto donde se puede introducir el comentario. Por cada bloque dispone de 64 KB para comentarios de bloques y de segmentos.



10.4.5 Introducir comentarios de bloques/segmentos

1. Con el comando de menú **Ver > Mostrar > Comentario**, active la visualización de los comentarios (delante del comando de menú aparece una marca de verificación).
2. Sitúe el cursor con un clic del ratón en el campo gris debajo del nombre del bloque o segmento. El campo del comentario, anteriormente gris, aparecerá representado en blanco y resaltado con un margen.
3. Introduzca el comentario en el cuadro de texto abierto. Por cada bloque dispone de un total de 64 KB para comentarios de bloques y de segmentos.
4. Para salir del cuadro de texto, haga clic con el ratón fuera de dicho cuadro, o bien, pulse la tecla TAB o las teclas MAYÚS + TAB.
5. Si así lo desea, desactive nuevamente la visualización de comentarios con el comando de menú **Ver > Mostrar > Comentario**.

10.4.6 Trabajar con plantillas de segmentos

Si al programar bloques desea utilizar segmentos de forma múltiple, puede guardar los segmentos en forma de plantilla o guardarlos en una librería con comodines (p.ej. para operandos). La librería debe existir antes de crear una plantilla de segmento.

Crear una plantilla de segmento

En caso necesario cree una librería nueva en el administrador SIMATIC e inserte un programa en la librería con el comando de menú **Insertar > Programa > Programa S7**.

1. Abra el bloque que contenga el o los segmentos con los que desea crear una plantilla.
2. En caso necesario, sustituya por comodines el título, el comentario y/o los operandos. Como comodines puede utilizar las cadenas de caracteres %00 hasta %99. Los comodines para los operandos se representan en rojo. En este caso no importa ya que no va a guardar el bloque después de haber creado la plantilla. Podrá volver a sustituir los comodines por operandos adecuados al insertar la plantilla en un bloque.
3. Seleccione el "<N°> del segmento o segmentos donde se deba insertar la plantilla.
4. Elija el comando de menú **Edición > Crear plantilla de segmento**.
5. En el cuadro de diálogo que se visualiza entonces, introduzca un comentario explicativo para cada uno de los comodines utilizados.
6. Haga clic en "Aceptar" para cerrar el cuadro de diálogo.
7. En el cuadro de selección que aparece luego, seleccione la carpeta de fuentes del programa S7 en su librería de plantillas de segmentos e indique un nombre para la plantilla.
8. Confirme con el botón "Aceptar". La plantilla de segmento se depositará en la librería seleccionada.
9. Cierre el bloque sin guardarlo.

Insertar un segmento a partir de una plantilla

1. Abra el bloque en el que desea insertar el nuevo segmento.
2. En el bloque, haga clic en el segmento después del cual desea insertar el segmento de acuerdo con la plantilla.
3. Abra la ficha "Elementos de programa" (eligiendo el comando de menú **Insertar > Elementos de programa**).
4. Abra la carpeta "Programa S7" de la librería en cuestión.
5. Haga doble clic en la plantilla.
6. En el cuadro de diálogo visualizado, introduzca los valores que deban sustituir a los comodines de la plantilla.
7. Haga clic en el botón "Aceptar". La plantilla se inserta entonces detrás del segmento actual.

Nota

La plantilla también se puede arrastrar mediante Drag & Drop (arrastrar y soltar) desde la ficha hasta la ventana del editor.

10.4.7 Función de búsqueda de errores en el área de instrucciones

Los errores se reconocen fácilmente en el área de instrucciones por estar destacados en rojo. Para desplazarse más fácilmente a los errores que se encuentren fuera del área visible, el editor incorpora las dos funciones de búsqueda **Edición > Ir a > Error precedente/error siguiente**.

La búsqueda de los errores se efectúa en todos los segmentos. Por tanto, se busca en toda el área de instrucciones y no sólo en un solo segmento o en el área visible en ese momento.

Activando la barra de estado con el comando de menú **Ver > Barra de estado** se visualizan allí observaciones relativas a los errores.

Las correcciones de errores y los cambios también se pueden efectuar en el modo de sobreescritura. Utilice la tecla INS para conmutar del modo de inserción al modo de sobreescritura y viceversa.

10.5 Editar instrucciones KOP en el área de instrucciones

10.5.1 Ajustes para el lenguaje de programación KOP

Ajustar el diseño de página de KOP

Para crear el programa en lenguaje KOP, puede definir el diseño de página. El diseño seleccionado (formato vertical DIN A4/formato horizontal/tamaño máximo) determinará el número de elementos KOP representables en la rama.

1. Elija el comando de menú **Herramientas > Preferencias**.
2. En el cuadro de diálogo que aparece a continuación, elija la ficha "KOP/FUP".
3. Elija el formato deseado en el cuadro de lista "Diseño de página". Introduzca el formato deseado.

Ajustes para imprimir:

Si desea imprimir el área de instrucciones KOP, deberá ajustar el diseño de página adecuado antes de crear dicha área.

Ajustes en la ficha "KOP/FUP"

En la ficha "KOP/FUP", a la que se accede mediante el comando de menú **Herramientas > Preferencias**, se pueden efectuar ajustes básicos, p.ej. el diseño de página y el ancho del campo de operandos.

10.5.2 Reglas para introducir operaciones KOP

El lenguaje de programación "KOP" se describe en el manual "KOP para S7-300/400 - Programación de bloques" o en la Ayuda en pantalla de KOP.

Un segmento KOP puede comprender varios elementos en diversas ramas. Todos los elementos y ramas deben estar unidos entre sí, sin que la barra de alimentación izquierda se considere una unión (IEC 1131-3).

Al programar en KOP es preciso tener en cuenta ciertas reglas. En caso de detectarse un error se visualizará el correspondiente mensaje.

Finalizar un segmento KOP

Cada segmento KOP debe terminar con una bobina o con un cuadro. No obstante, los siguientes elementos KOP no se pueden utilizar para finalizar un segmento:

- Cuadros de comparación
- Bobinas para conectores $_/(#)_ /$
- Bobinas para la evaluación positiva $_/(P)_ /$ o negativa $_/(N)_ /$ de flancos

Emplazamiento de los cuadros

El punto inicial de la rama para la conexión de un cuadro debe ser siempre la barra de alimentación izquierda. No obstante, en la rama que precede al cuadro pueden encontrarse operaciones lógicas u otros cuadros.

Emplazamiento de las bobinas

Las bobinas se emplazan automáticamente en el borde derecho del segmento, constituyendo allí el final de una rama.

Excepciones: Las bobinas para conectores $_{/(\#)_/}$, así como la evaluación positiva $_{/(P)_/}$ o negativa $_{/(N)_/}$ de flancos no se pueden emplazar ni en el extremo izquierdo ni en el derecho. Tampoco son admisibles en ramas paralelas.

Algunas de las bobinas exigen una operación booleana, en tanto que otras no permiten dicha operación.

- Bobinas que exigen una operación booleana:
 - Salida $_{/ ()}$, Activar $_{/(S)}$, Desactivar $_{/(R)}$
 - Conector $_{/(\#)_/}$, Flanco positivo $_{/(P)_/}$, Flanco negativo $_{/(N)_/}$
 - Todas las bobinas de contadores y temporizadores
 - Saltar si es 0 $_{/(JMPN)}$
 - Conectar Master Control Relay $_{/(MCR<)}$
 - Cargar RLO en registro RB $_{/(SAVE)}$
 - Retorno $_{/(RET)}$
- Bobinas que no permiten una operación booleana:
 - Inicio Master Control Relay $_{/(MCRA)}$
 - Fin Master Control Relay $_{/(MCRD)}$
 - Abrir bloque de datos $_{/(OPN)}$
 - Desconectar Master Control Relay $_{/(MCR>)}$

Todas las demás bobinas pueden tener o no una operación booleana.

Las siguientes bobinas no se pueden utilizar **en calidad de salida paralela**:

- Saltar si es 0 $_{/(JMPN)}$
- Saltar si es 1 $_{/(JMP)}$
- Llamada de bloque $_{/(CALL)}$
- Retorno $_{/(RET)}$

Entrada/salida de habilitación

La entrada de habilitación "EN" y/o la salida de habilitación "ENO" de los cuadros se puede conectar, sin que ello sea absolutamente necesario.

Retirar y cambiar

Si una rama se compone de un solo elemento, al retirarse éste se retirará toda la rama.

Al retirarse un cuadro, se retirarán también todas las ramas unidas al mismo con entradas booleanas, con excepción de la rama principal.

Para la sustitución simple de elementos de un mismo tipo se puede utilizar el modo de sobreescritura.

Ramas paralelas

- Las ramas O se añaden de izquierda a derecha.
- Las ramas paralelas se abren hacia abajo y se cierran hacia arriba.
- Las ramas paralelas se abren siempre detrás del elemento KOP seleccionado.
- Las ramas paralelas se cierran siempre detrás del elemento KOP seleccionado.
- Para borrar una rama paralela deberá borrar primero todos los elementos KOP de la rama. Al borrar el último elemento KOP, se borrará también el resto de la misma.

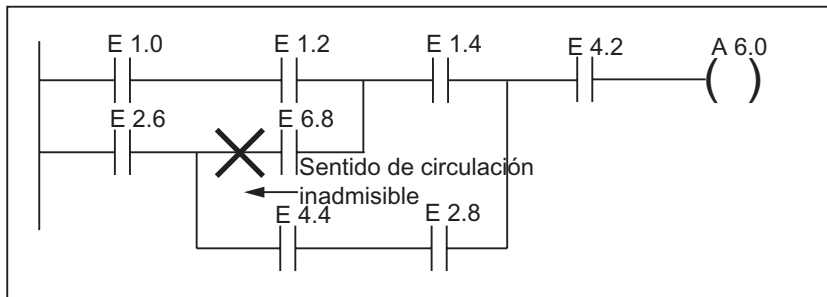
Constantes

Las vinculaciones binarias no pueden ocuparse con constantes (p.ej. TRUE o FALSE). En su lugar debe utilizar operandos con el tipo de datos BOOL.

10.5.3 Interconexiones no admisibles en KOP

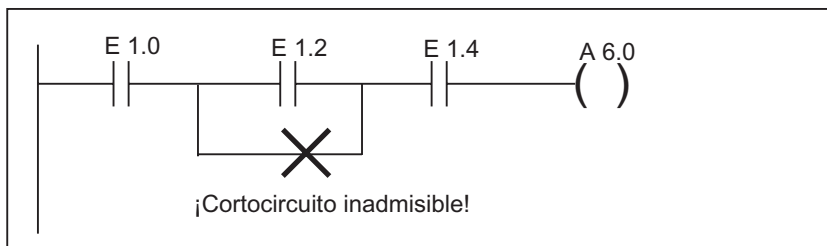
Circulación de corriente de derecha a izquierda

No se pueden editar ramas que pudieran hacer que la corriente fluya en sentido contrario. La figura siguiente muestra un ejemplo al respecto. Un estado de señal "0" en E 1.4 causaría un flujo de corriente de derecha a izquierda en E 6.8, lo cual no es admisible.



Cortocircuito

No se pueden editar ramas que causen un cortocircuito. La figura siguiente muestra un ejemplo al respecto:



10.6 Editar instrucciones FUP en el área de instrucciones

10.6.1 Ajustes para el lenguaje de programación FUP

Ajustar el diseño de página de FUP

Para crear el programa en lenguaje FUP se puede definir el diseño de página. El diseño seleccionado (formato vertical DIN A4/formato horizontal/tamaño máximo) repercute en el número de elementos FUP representables en una rama.

1. Elija el comando de menú **Herramientas > Preferencias**.
2. En el cuadro de diálogo que aparece a continuación, elija la ficha "KOP/FUP".
3. Elija el formato deseado en el cuadro de lista "Diseño de página". Introduzca el formato deseado.

Notas respecto a la impresión

Si desea imprimir el área de instrucciones FUP, deberá ajustar el diseño de página adecuado antes de crear dicha área.

Ficha KOP/FUP en Herramientas > Preferencias

En la ficha "KOP/FUP", a la que se accede mediante el comando de menú **Herramientas > Preferencias**, se pueden efectuar ajustes básicos, por ejemplo el diseño de página y el ancho del campo de operandos.

10.6.2 Reglas para introducir operaciones FUP

El lenguaje de programación "FUP" se describe en el manual "FUP para S7-300/400 - Programación de bloques" o en la Ayuda en pantalla de FUP.

Un segmento FUP puede comprender varios elementos. Todos los elementos deben estar interconectados (IEC 1131-3).

Al programar en FUP es preciso tener en cuenta ciertas reglas. En caso de detectarse un error se visualizará el correspondiente mensaje.

Insertar y editar direcciones y parámetros

Al insertar un elemento FUP se introducen las secuencias de caracteres "???" y "..." como comodines para direcciones o parámetros.

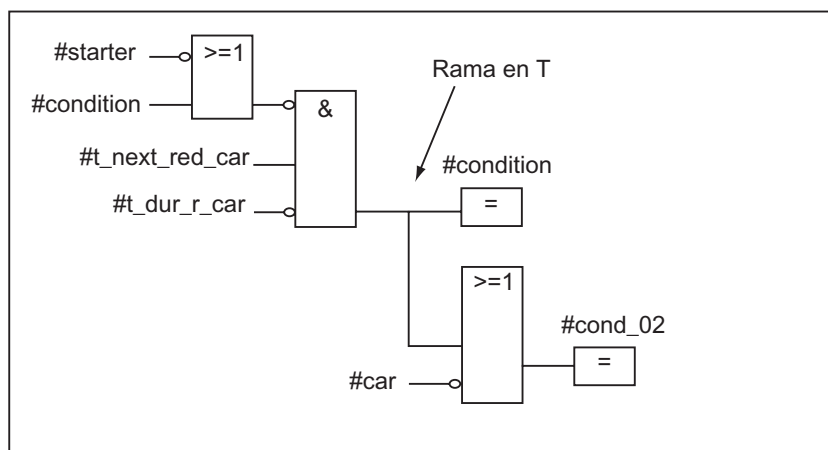
- La cadena de caracteres representada en rojo "???" indica direcciones y parámetros que se deben asignar.
- La cadena de caracteres representada en negro "..." indica direcciones y parámetros que se pueden asignar.

Si mueve el puntero del ratón en el comodín, se mostrará el tipo de datos deseado.

Emplazamiento de los cuadros

A los cuadros con operaciones binarias (&, >=1, XOR) se pueden añadir cuadros estándar (flipflops, contadores, temporizadores, operaciones aritméticas, etc.). Esta regla excluye los cuadros de comparación.

En un segmento no se pueden programar diferentes operaciones lógicas con salidas por separado. No obstante, mediante una rama de una cadena de operaciones lógicas se pueden asociar varias asignaciones. La figura siguiente muestra un segmento con dos asignaciones.



Los cuadros siguientes sólo se pueden emplazar en el borde derecho de la cadena de operaciones lógicas, constituyendo allí el final de la misma:

- Poner contador al valor inicial
- Incrementar contador, decrementar contador
- Arrancar temporizador como impulso, arrancar temporizador como impulso prolongado
- Arrancar temporizador como retardo a la conexión/desconexión.

Algunos de los cuadros exigen una operación booleana, en tanto que otros no permiten dicha operación.

Cuadros que exigen una operación booleana:

- Salida, activar, desactivar $_/[R]$
- Conector $_/#_/$, flanco positivo $_/P_/$, flanco negativo $_/N_/$
- Todos los cuadros de contadores y temporizadores
- Saltar si es 0 $_/JMPN]$
- Conectar Master Control Relay $_/MCR<]$
- Cargar RLO en registro RB $_/SAVE]$
- Retorno $_/RET]$

Cuadros que no permiten una operación booleana:

- Inicio Master Control Relay [MCRA]
- Fin Master Control Relay [MCRD]
- Abrir bloque de datos [OPN]
- Desconectar Master Control Relay [MCR>]

Todos los demás cuadros pueden tener o no una operación booleana.

Entrada/salida de habilitación

La entrada de habilitación "EN" y/o la salida de habilitación "ENO" de los cuadros se puede conectar, sin que ello sea absolutamente necesario.

Retirar y cambiar

Al retirarse un cuadro, se retirarán también todas las ramas unidas al mismo con entradas booleanas, con excepción de la rama principal.

Para la sustitución simple de elementos de un mismo tipo se puede utilizar el modo de sobreescritura.

Constantes

Las vinculaciones binarias no pueden ocuparse con constantes (p.ej. TRUE o FALSE). En su lugar debe utilizar operandos con el tipo de datos BOOL.

10.7 Editar instrucciones AWL en el área de instrucciones

10.7.1 Ajustes para el lenguaje de programación AWL

Ajustar la nemotécnica

Se puede elegir entre dos ajustes:

- Alemán o
- Inglés

La nemotécnica no se puede cambiar mientras se está editando el bloque, por lo que antes de abrir un bloque en el Administrador SIMATIC, puede ajustarla con el comando de menú **Herramientas > Preferencias** en la ficha "Lenguaje".

Las **propiedades de bloques** se editan en un cuadro de diálogo por separado.

En el editor es posible tener abiertos simultáneamente varios bloques que se pueden editar uno tras otro a voluntad.

10.7.2 Reglas para introducir instrucciones AWL

El lenguaje de programación "AWL" se describe en el manual "AWL para S7-300/400 - Programación de bloques" o en la Ayuda en pantalla de AWL (Descripción de lenguajes).

Para utilizar el método de entrada incremental de instrucciones AWL deberá respetar las siguientes reglas básicas:

- Aténgase al orden de programación de los bloques. Los bloques llamados deben programarse antes de los bloques invocantes.
- Toda instrucción se compone de una marca de salto (opcional), una operación, un operando y un comentario (opcional).
Ejemplo: M001: U E1.0 //Comentario
- Cada instrucción ocupa una línea propia.
- Por cada bloque se puede introducir un máximo de 999 segmentos.
- Por cada segmento se pueden introducir unas 2000 líneas. Si utiliza el zoom (ampliar, reducir), aumentará o disminuirá el número posible de líneas.
- Al introducir operaciones o direcciones absolutas no se distingue entre mayúsculas y minúsculas.

10.8 Actualizar llamadas a bloques

Con el comando de menú **Edición > Llamada > Actualizar** en "KOP/AWL/FUP: Programar bloques" se pueden actualizar automáticamente las llamadas de bloques que ya no sean válidas. Después de cambiar los siguientes interfaces deberá tener lugar una actualización:

- Insertar nuevos parámetros formales
- Borrar parámetros formales
- Cambiar el nombre de parámetros formales
- Cambiar el tipo de parámetros formales
- Cambiar el orden de parámetros formales (copiándolos a otro lugar)

Al asignar el lado formal y el actual se procede conforme a las siguientes reglas en el orden indicado:

1. **Nombres idénticos de parámetros:**
Los parámetros actuales se asignan automáticamente si el nombre del parámetro formal no ha cambiado.
Excepción: La precombinación lógica de parámetros de entrada binarios sólo se puede asignar automáticamente en KOP y FUP si no cambia el tipo de datos (BOOL). Si el tipo de datos ha cambiado, la precombinación lógica previa se conservará en calidad de rama abierta.
2. **Tipos de parámetros idénticos:**
Después de asignar los parámetros de igual nombre, los parámetros actuales aún no asignados se asignarán a los parámetros formales que tengan el mismo tipo de datos que los "antiguos" parámetros formales.
3. **Posición idéntica de los parámetros:**
Los parámetros actuales aún no asignados conforme a las reglas 1 y 2, se asignarán a los nuevos parámetros formales conforme a su posición en el "antiguo" interface.
4. Si los parámetros actuales no se pueden asignar conforme a las tres reglas mencionadas, se borrarán o se conservarán en calidad de ramas abiertas al tratarse de precombinaciones lógicas binarias en KOP o FUP.

Tras ejecutar esta función, los cambios efectuados se deberán comprobar en la tabla de declaración de variables y en el área de instrucciones del programa.

10.8.1 Cambiar interfaces

En los bloques offline que han sido editados con la versión 5 de STEP 7, se pueden cambiar los interfaces en el editor incremental:

1. Asegúrese de que todos los bloques han sido compilados con la versión 5 de STEP 7. Genere para ello una fuente con todos los bloques y compílela.
2. Cambia el interface del bloque afectado.
3. Abra sucesivamente todas las llamadas - las llamadas se representan en rojo.
4. Utilice la función **Edición > Llamada > Actualizar....**
5. Vuelva a generar los bloques de datos de instancia afectados.

Nota

- Los cambios que se realicen en los interfaces de un bloque abierto en modo online pueden hacer cambiar la CPU a STOP.
 - Reasignar llamadas de bloques
Cambie primero los números de los bloques llamados y ejecute después la función "Recablear" para adaptar las llamadas.
-

10.9 Guardar bloques lógicos

Para adoptar bloques nuevos o bien los cambios realizados en el área de instrucciones de bloques lógicos o en tablas de declaración de la gestión de datos de la PG, deberá guardar el bloque correspondiente, escribiendo los datos en el disco duro de la PG.

Para guardar bloques en el disco duro de la PG:

1. Active la ventana de trabajo del bloque que desea guardar.
2. Elija:
 - el comando de menú **Archivo > Guardar** si desea guardar el bloque con el mismo nombre
 - el comando de menú **Archivo > Guardar como** si desea guardar el bloque en otro programa de usuario S7, o bien con otro nombre. Introduzca la nueva ruta o el nuevo nombre del bloque en el cuadro de diálogo que aparece a continuación.

En ambos casos, el bloque se guarda solamente si la sintaxis es correcta. Los errores sintácticos se detectan inmediatamente al crear el bloque y se destacan en rojo. Dichos errores deben corregirse antes de poder guardar el bloque.

Nota

- También puede guardar los bloques o las fuentes en otros proyectos o librerías utilizando también el Administrador SIMATIC (p.ej., con arrastrar y soltar).
 - Los bloques o los programas de usuario completos sólo se pueden guardar en una Memory Card con ayuda del Administrador SIMATIC.
 - Si surgen problemas al guardar o compilar bloques, tendrá que reorganizar el proyecto. Para ello elija en el Administrador SIMATIC el comando de menú **Archivo > Reorganizar**. Después vuelva a guardar o a compilar el bloque.
-

11 Crear bloques de datos

11.1 Nociones básicas para crear bloques de datos

En el bloque de datos se depositan valores a los que deba acceder su máquina o instalación. Contrariamente al bloque lógico, programado con uno de los lenguajes de programación (KOP, FUP o AWL), el bloque de datos sólo contiene la parte de declaración de variables. No requiere un área de instrucciones y, por tanto, tampoco la programación de segmentos.

Tras abrir un bloque de datos, éste se puede visualizar en la vista "Declaración" o "Datos". Para conmutar de una vista a otra, elija los comandos de menú **Ver > Declaración** y **Ver > Datos**.

Vista "Declaración"

Elija la vista "Declaración" para

- leer o fijar la estructura de los DBs globales,
- leer la estructura de los DBs con tipo de datos de usuario (UDT) asociado,
- leer la estructura de los DBs con bloque de función (FB) asociado.

La estructura de los bloques de datos asociados a un FB o a un UDT no se puede cambiar. A tal efecto sería preciso modificar antes el FB o el UDT en cuestión y crear luego nuevamente el DB.

Vista "Datos"

Elija la vista "Datos" para modificar los datos. Solamente en la vista "Datos" es posible visualizar, introducir o cambiar el valor actual de cada uno de los elementos. Si se trata de bloques de datos cuyas variables son tipos de datos compuestos, los elementos se listarán individualmente con su nombre completo.

Diferencia entre un bloque de datos de instancia y un bloque de datos global

El bloque de datos global no se encuentra asociado a ningún bloque lógico. Contiene valores requeridos por la instalación o por la máquina y se puede llamar directamente desde cualquier lugar del programa.

El bloque de datos de instancia se encuentra asociado directamente a un bloque lógico, p.ej. a un bloque de función. El bloque de datos de instancia contiene los datos depositados en la tabla de declaración de variables de un bloque de función.

11.2 Ver declaración de bloques de datos

En los bloques de datos no globales no se puede modificar la vista "Declaración".

Columna	Explicación
Dirección	Indica la dirección que STEP 7 asigna automáticamente a la variable al terminar de introducir una declaración.
Declaración	Esta columna se representa sólo para los bloques de datos de instancia. Ella le informa de cómo se han definido las variables en la declaración de variables del FB: <ul style="list-style-type: none"> • parámetros de entrada (IN) • parámetros de salida (OUT) • parámetros de entrada/salida (IN_OUT) • datos estáticos (STAT)
Designación	Introduzca el nombre que debe asignar a cada variable.
Tipo	Introduzca el tipo de datos de la variable (BOOL, INT, WORD, ARRAY, etc.). Las variables pueden pertenecer a los tipos de datos simples, a los tipos de datos compuestos, o bien, a los tipos de datos de usuario.
Valor inic.	Indique el valor inicial, en caso de que el software no deba tomar el valor predeterminado del tipo de datos introducido. Todos los valores introducidos deben ser compatibles con los tipos de datos. Cuando guarde por primera vez el bloque de datos, el valor inicial será adoptado como valor actual de la variable, a menos que defina expresamente su valor actual. Recuerde que los valores iniciales no pueden cargarse en la CPU.
Comentario	En este campo puede introducir un comentario para documentar la variable. El comentario no debe tener más de 79 caracteres.

11.3 Ver datos de los bloques de datos

En la vista "Datos" se representan los valores actuales de todas las variables del bloque de datos. Estos valores sólo pueden ser modificados en la vista "Datos". La forma de representar tablas con dicha vista es igual para todos los bloques de datos globales. En el caso de los bloques de datos de instancia, se visualiza adicionalmente la columna "Declaración".

En el caso de las variables con tipos de datos compuestos o con tipos de datos de usuario, cada uno de los elementos se representa en la vista "Datos" en una línea propia con nombre completo. Si los elementos se encuentran en el área IN_OUT de un bloque de datos de instancia, en la columna "Valor actual" aparecerá el puntero en el tipo de datos compuesto o de usuario.

La vista "Datos" muestra las siguientes columnas:

Columna	Explicación
Dirección	Indica la dirección que STEP 7 asigna automáticamente a la variable.
Declaración	Esta columna se representa sólo para los DBs de instancia. Ella le informa de cómo se han definido las variables en la declaración de variables del FB: <ul style="list-style-type: none"> • parámetros de entrada (IN) • parámetros de salida (OUT) • parámetros de entrada/salida (IN_OUT) • datos estáticos (STAT)
Designación	Se trata del nombre que se ha definido para la variable. Este campo no se puede editar estando en la vista "Datos".
Tipo	Se trata del tipo de datos definido para la variable. Puesto que, en la vista "Datos" de las variables con tipos de datos compuestos o de usuario, los elementos se listan individualmente, se indican en el caso de un bloque de datos global sólo los tipos de datos simples. En el caso de un bloque de datos de instancia también se indicarán los tipos de datos Parámetro; en el caso de parámetros de entrada/salida (INOUT) con tipo de datos compuesto o de usuario el puntero aparecerá - en la columna "Valor actual" - el puntero en el tipo de datos.
Valor inic.	Se trata del valor inicial que usted ha definido para la variable, en caso de que el software no deba tomar el valor predefinido para el tipo de datos indicado. Cuando guarde por primera vez el bloque de datos, el valor inicial se adoptará como valor actual de la variable, a menos que defina expresamente su valor actual. Recuerde que, al contrario que en el caso de los valores actuales, los valores iniciales no pueden cargarse en la CPU.
Valor actual	Offline: Se trata del valor que tenía la variable al abrir el bloque de datos o después del último cambio que haya sido almacenado (aun cuando haya abierto el DB online, este valor no se actualiza). Online: El valor actual al abrir el bloque de datos se muestra, pero no se actualiza. Para actualizar la visualización debe pulsar la tecla F5. Este campo se puede editar, si no pertenece a un parámetro de entrada/salida (INOUT) con tipo de datos compuesto o de usuario. Todos los valores introducidos deben ser compatibles con los tipos de datos.
Comentario	Se trata del comentario introducido al documentar la variable. Este campo no se puede editar estando en la vista "Datos".

11.4 Editar bloques de datos y guardar

11.4.1 Introducir la estructura de los bloques de datos globales

Si ha abierto un bloque de datos que no esté asociado a ningún UDT o FB, puede definir su estructura en la vista Declaración del bloque de datos. En los bloques de datos no globales no se puede modificar la vista "Declaración".

1. Abra un bloque de datos global, es decir, un bloque no asociado a ningún UDT o FB.
2. Active la vista "Declaración" del bloque de datos en caso de que ésta no se visualice todavía.
3. Defina la estructura rellenando la tabla visualizada conforme a los datos que se indican a continuación.

En los bloques de datos no globales no se puede modificar la vista "Declaración".

Columna	Explicación
Dirección	Indica la dirección que STEP 7 asigna automáticamente a la variable al terminar de introducir una declaración.
Designación	Introduzca el nombre que debe asignar a cada variable.
Tipo	Introduzca el tipo de datos de la variable (BOOL, INT, WORD, ARRAY, etc.). Las variables pueden pertenecer a los tipos de datos simples, a los tipos de datos compuestos, o bien, a los tipos de datos de usuario.
Valor inic.	Indique el valor inicial, en caso de que el software no deba tomar el valor predeterminado del tipo de datos introducido. Todos los valores introducidos deben ser compatibles con los tipos de datos. Cuando guarde por primera vez el bloque de datos, el valor inicial será adoptado como valor actual de la variable, a menos que defina expresamente su valor actual.
Comentario	En este campo puede introducir un comentario para documentar la variable. El comentario no debe tener más de 80 caracteres.

11.4.2 Introducir / visualizar la estructura de bloques de datos con FB asociado (DBs de instancia)

Introducción

Si asocia un bloque de datos a un FB (FB de instancia), la declaración de variables del FB define la estructura del DB. Si desea realizar cambios, sólo podrá hacerlo en el FB asociado.

1. Abra el bloque de función asociado.
2. Edite la declaración de variables del bloque de función.
3. Cree nuevamente el bloque de datos.

Visualización

En la vista "Declaración" del DB de instancia puede apreciar cómo se declararon las variables en el FB.

1. Abra el bloque de datos.
2. Active la vista "Declaración" del bloque de datos en caso de que ésta no se visualice todavía.
3. Más abajo se proporcionan explicaciones respecto a la tabla visualizada.

En los bloques de datos no globales no se puede modificar la vista "Declaración".

Columna	Explicación
Dirección	Indica la dirección que STEP 7 asigna automáticamente a la variable.
Declaración	Esta columna muestra cómo están declaradas las variables en la declaración del FB: <ul style="list-style-type: none"> • parámetros de entrada (IN) • parámetros de salida (OUT) • parámetros de entrada/salida (IN_OUT) • datos estáticos (STAT) Los datos temporales declarados del FB no están en el bloque de datos de instancia.
Nombre:	Indica el nombre definido para la variable en la declaración del FB.
Tipo	Indica los tipos de datos definidos para la variable en la declaración del FB. Las variables pueden pertenecer a los tipos de datos simples, a los tipos de datos compuestos, o bien, a los tipos de datos de usuario. Si dentro del FB se llama a más bloques de función para cuya llamada se hayan declarado variables estáticas, se puede indicar allí también un FB o bloque de función del sistema (SFB) como tipo de datos.
Valor inicial:	Se trata del valor predeterminado que usted ha definido en la declaración del FB para la variable, en caso de que el software no deba tomar el valor predefinido para el tipo de datos. Cuando guarde por primera vez el bloque de datos, el valor inicial se adoptará como valor actual de la variable, a menos que defina expresamente su valor actual.
Comentario	Se trata del comentario escrito en la declaración de variables del FB para documentar el elemento de datos. Este campo no se puede editar.

Nota

En el caso de los bloques de datos asociados a un FB sólo podrá editar los valores actuales de las variables. La entrada de valores actuales de las variables se efectúa en la vista "Datos" de los bloques de datos.

11.4.3 Introducir la estructura de los tipos de datos de usuario (UDT)

1. Abra el tipo de datos de usuario asociado (UDT).
2. Active la vista "Declaración" del bloque de datos en caso de que ésta no se visualice todavía.
3. La estructura del tipo de datos de usuario se define fijando el orden de las variables, su tipo de datos y, en determinados casos, un valor inicial conforme a los datos siguientes.
4. Finalice la entrada de una variable pulsando la tecla TAB o la TECLA INTRO.

Columna	Explicación
Dirección:	Indica la dirección que STEP 7 asigna automáticamente a la variable al terminar de introducir una declaración.
Designación	Introduzca el nombre que debe asignar a cada variable.
Tipo	Introduzca el tipo de datos de la variable (BOOL, INT, WORD, ARRAY, etc.). Las variables pueden pertenecer a los tipos de datos simples, a los tipos de datos compuestos, o bien, a los tipos de datos de usuario.
Valor inic.	Indique el valor inicial cuando el software no deba adoptar el valor predefinido para el tipo de datos introducido. Todos los valores introducidos deben ser compatibles con los tipos de datos. Cuando guarde por primera vez una instancia del UDT (de una variable o de un bloque de datos), el valor inicial se adoptará como valor actual de la variable, a menos que defina expresamente su valor actual.
Comentario	En este campo puede introducir un comentario para documentar la variable. El comentario no debe tener más de 80 caracteres.

11.4.4 Introducir / visualizar la estructura de bloques de datos con UDT asociado

Introducción

Si asocia un bloque de datos a un UDT, la estructura de datos del UDT definirá la estructura del DB. Los cambios sólo se pueden efectuar en el UDT asociado:

1. Abra el tipo de datos de usuario asociado (UDT).
2. Edite la estructura del tipo de datos de usuario.
3. Cree nuevamente el bloque de datos.

Visualización

En la vista "Declaración" del DB sólo se puede apreciar cómo se declararon las variables en el UDT:

1. Abra el bloque de datos.
2. Active la vista "Declaración" del bloque de datos en caso de que ésta no haya sido todavía visualizada.
3. Encontrará al final más información respecto a la tabla visualizada.

La vista "Declaración" no se puede modificar. Los cambios sólo se pueden efectuar en el UDT asociado.

Columna	Explicación
Dirección	Indica la dirección que STEP 7 asigna automáticamente a la variable.
Designación	Indica el nombre definido por el UDT para la variable.
Tipo:	Indica los tipos de datos definidos en el UDT. Las variables pueden pertenecer a los tipos de datos simples, a los tipos de datos compuestos, o bien, a los tipos de datos de usuario.
Valor inic.	Indica el valor predeterminado que se ha definido para la variable en el UDT, en caso de que el software no deba tomar el valor predefinido para el tipo de datos. Cuando guarde por primera vez el bloque de datos, el valor inicial se adoptará como valor actual de la variable, a menos que defina expresamente su valor actual.
Comentario	Se trata del comentario definido por el UDT para documentar el elemento de datos.

Nota

En el caso de los bloques de datos asociados a un UDT sólo podrá editar los valores actuales de las variables. La entrada de valores actuales de las variables se efectúa en la vista "Datos" de los bloques de datos.

11.4.5 Cambiar valores de datos en la vista "Datos"

Los valores actuales sólo se pueden editar en la vista "Datos" de bloques de datos.

1. En caso necesario, cambie la representación de la tabla a la vista "Datos" con el comando de menú **Ver > Datos**.
2. Introduzca los valores actuales deseados para los elementos de datos en los campos de la columna "Valor actual". Los valores actuales deberán ser compatibles con el tipo de datos de los elementos.

Las entradas erróneas se detectan inmediatamente y se resaltan en rojo (p.ej., cuando el valor actual introducido no sea compatible con el tipo de datos). Dichos errores deben eliminarse antes de guardar el bloque de datos.

Nota

Los cambios realizados se almacenan al guardar los bloques de datos.

11.4.6 Inicializar los valores de datos

La inicialización sólo es posible en la vista "Datos" de los bloques de datos.

1. En caso necesario, cambie la representación de la tabla a la vista "Datos" con el comando de menú **Ver > Datos**.
2. Elija para ello el comando de menú **Edición > Inicializar bloque de datos**.

Todas las variables vuelven a recibir el valor inicial previsto, es decir, los valores actuales de todas las variables son reemplazados por el valor inicial correspondiente.

Nota

Los cambios realizados se almacenan al guardar los bloques de datos.

11.4.7 Guardar bloques de datos

Para adoptar en la gestión de datos de la PG bloques de datos nuevos o valores modificados de bloques de datos, deberá guardar el bloque correspondiente, escribiendo los datos en el disco duro de la PG.

Para guardar bloques en el disco duro de la PG:

1. Active la ventana de trabajo del bloque que desea guardar.
2. Elija:
 - el comando de menú **Archivo > Guardar** si desea guardar el bloque con el mismo nombre, o
 - el comando de menú **Archivo > Guardar como** si desea guardar el bloque en otro programa de usuario S7, o bien con otro nombre. Introduzca la nueva ruta o el nuevo nombre del bloque en el cuadro de diálogo que aparece a continuación. Tenga en cuenta que el DB0 no se puede usar como número de DB, puesto que dicho número está reservado para el sistema.

En ambos casos, el bloque sólo se guardará si la sintaxis es correcta. Los errores sintácticos se detectan inmediatamente al crear el bloque y se destacan en rojo. Dichos errores deben corregirse antes de guardar el bloque.

Nota

Los bloques o las fuentes también pueden guardarse en otros proyectos o librerías desde el Administrador SIMATIC (p.ej., con arrastrar y soltar).

Los bloques o los programas de usuario completos sólo se pueden guardar en una Memory Card desde el Administrador SIMATIC.

Si surgen problemas al guardar o compilar bloques, tendrá que reorganizar el proyecto. Para ello elija en el Administrador SIMATIC el comando de menú **Archivo > Reorganizar**. Después vuelva a guardar o a compilar el bloque.

12 Parametrizar bloques de datos

La función "Parametrizar bloques de datos" le permite realizar las siguientes acciones fuera del editor de programas KOP/AWL/FUP

- Editar valores actuales de bloques de datos de instancia y cargarlos en el sistema de destino sin necesidad de cargar todo el bloque de datos.
- Observar online bloques de datos de instancia.
- En la vista de parametrización, parametrizar cómodamente y observar online bloques de datos de instancia y multiinstancias con el atributo de sistema S7_techparam (funciones tecnológicas).

Proceda de la siguiente manera:

1. En el administrador SIMATIC, abra el bloque de datos de instancia con un doble clic.
2. Confirme con "Sí" el mensaje que le pregunta si desea abrir la función "Parametrizar bloques de datos". **Resultado:** El DB de instancia se abre en la aplicación "Parametrizar bloques de datos".
3. Con el comando de menú **Ver > Datos** o bien **Ver > Declaración** seleccione la vista en la que desea que se muestre el bloque de datos. En bloques de datos de instancia o multiinstancias con el atributo de sistema "S7_techparam" se abre automáticamente la vista de parametrización tecnológica.
4. Edite el bloque de datos de instancia de acuerdo con sus necesidades. Cualquier información, advertencia o error se muestra en la ventana de aviso. Haciendo doble clic sobre una advertencia o sobre un error accede al punto de error en cuestión.
5. Cargue los valores actuales modificados con el comando de menú **Sistema de destino > Transferir datos de parámetros** desde la PG a la CPU que ha asignado al programa S7 actual.
6. Con el comando de menú **Test > Observar** seleccione la visualización del estado del programa para el bloque abierto y observe online la edición de los valores actuales cargados.

Nota

Los bloques que tienen el atributo de sistema "S7_techparam" se reconocen seleccionando el bloque en el administrador SIMATIC, ejecutando el comando de menú **Edición > Propiedades del objeto** y abriendo la ficha "Atributos".

12.1 Parametrizar funciones tecnológicas

Con la función "Parametrizar bloques de datos" puede parametrizar y observar online cómodamente los bloques de regulación de temperatura FB 58 "TCONT_CP" y FB 59 "TCONT_S" incluidos en la librería estándar "PID Control Blocks".

Para ello, siga estos pasos:

1. En el Administrador SIMATIC, abra la librería estándar de STEP 7 con **Archivo > Abrir > Librerías**.
2. Seleccione "PID Control Blocks" y haga clic en "Blocks". Aquí encontrará los siguientes bloques de función con el atributo "S7_techparam":
 - **FB 58 "TCONT_CP"**: regulador de temperatura para accionadores con señal de entrada continua o en forma de impulso
 - **FB 59 "TCONT_S"**: regulador de temperatura para accionadores de efecto integral
3. Copie el bloque de función que desee (FB 58 o FB 59) desde la librería estándar en el proyecto.
4. Utilice **Insertar > Bloque S7 > Bloque de datos** para crear un DB de instancia en el FB que desee.
5. En el Administrador SIMATIC, abra el DB de instancia con un doble clic para iniciar la función "Parametrizar bloques de datos".
Resultado: El DB de instancia se abrirá en la vista tecnológica. Ahora puede parametrizar y observar online el DB de instancia cómodamente.
6. Introduzca en la vista tecnológica los valores que desee para el regulador. Las notificaciones, advertencias y errores que pudieran producirse aparecerán en la ventana de notificación. Si hace clic en una advertencia o en un error, podrá acceder al error.

Nota

Para reconocer los bloques que tienen en atributo de sistema "S7_techparam", seleccione el bloque en el Administrador SIMATIC, elija el comando de menú **Edición > Propiedades del objeto** y abra la ficha "Atributos".

13 Crear fuentes AWL

13.1 Nociones básicas para programar en fuentes AWL

Existe la posibilidad de introducir el programa o partes del mismo en forma de fuente AWL y compilarlos a continuación en bloques. La fuente puede contener el código para varios bloques, los cuales se compilarán seguidamente de una sola pasada como bloques.

La creación de un programa mediante una fuente se tiene las siguientes ventajas:

- Permite crear y editar la fuente con cualquier editor ASCII, importarla a continuación y compilarla con esta aplicación en bloques individuales. Al compilar el archivo fuente se crearán los bloques que luego se depositarán en el programa de usuario S7.
- Se pueden programar varios bloques en una sola fuente.
- Las fuentes se pueden guardar aunque contengan errores de sintaxis. Ello no es posible al crear bloques lógicos con comprobación incremental de sintaxis. No obstante, hasta que no compile la fuente no se dará cuenta de los errores que contiene.

La fuente se crea respetando la sintaxis del lenguaje de programación "lista de instrucciones (AWL)". Se estructura en bloques, declaración de variables o segmentos mediante las palabras clave.

Al crear bloques en fuentes AWL se debe considerar lo siguiente:

- reglas de programación para fuentes AWL
- sintaxis y formatos permitidos por las fuentes AWL
- estructura de bloque permitida por las fuentes AWL

13.2 Reglas para programar en fuentes AWL

13.2.1 Reglas para introducir instrucciones en fuentes AWL

En principio, una fuente AWL se compone de texto continuo. Para poder compilarla en bloques se deben respetar ciertas estructuras y reglas de sintaxis.

Para crear programas de usuario en forma de fuentes AWL se deben respetar las siguientes reglas generales:

Tema	Regla
Sintaxis	La sintaxis de las instrucciones AWL es la misma que en el editor AWL incremental, a excepción de la instrucción de llamada CALL.
CALL	<p>En las fuentes, los parámetros se introducen entre paréntesis. Los parámetros se separan unos de otros utilizando comas.</p> <p>Ejemplo de llamada FC (una línea): CALL FC 10 (param1 :=E0.0,param2 :=E0.1);</p> <p>Ejemplo de llamada FB (una línea): CALL FB10, DB100 (para1 :=E0.0,para2 :=E0.1);</p> <p>Ejemplo de llamada FB (varias líneas): CALL FB10, DB100 (para1 :=E0.0, para2 :=E0.1);</p> <p>Nota: En el editor ASCII, en las llamadas de bloques hay que transferir los parámetros en el orden definido. De lo contrario no coincidirán los comentarios asignados en estas líneas al visualizar el programa en AWL o en forma de fuente.</p>
Mayúsculas / minúsculas	Por lo general, el editor de esta aplicación no distingue las mayúsculas de las minúsculas, excepto cuando se trata de atributos del sistema y metas de salto. Al introducir cadenas de caracteres (tipo de datos STRING) se deben tener en cuenta también las mayúsculas y las minúsculas. Las palabras clave se representan en mayúsculas. No obstante, al compilar no se tienen en cuenta las mayúsculas y las minúsculas. Por tanto, las palabras clave se pueden indicar tanto en mayúsculas como en minúsculas, o bien, mezclando ambas.
Punto y coma	Finalice todas las instrucciones AWL y todas las declaraciones de variables con un punto y coma. Se puede introducir más de una instrucción por línea.
Dos barras inclinadas (//)	Comience todos los comentarios con dos barras inclinadas (//) y finalícelos pulsando la tecla INTRO.

13.2.2 Reglas para declarar variables en fuentes AWL

Para cada bloque contenido en la fuente se deben declarar las correspondientes variables.

La declaración de variables se encuentra antes del área de instrucciones del bloque.

Las variables existentes deben declararse respetando el orden de los tipos de declaración. Así figurarán juntas todas las variables de un mismo tipo.

Mientras que con KOP, FUP o AWL es necesario rellenar una tabla de declaración de variables, aquí debe utilizar palabras clave.

Palabras clave en la tabla de declaración de variables

Tipo de declaración	Palabras clave	rige para ...
Parámetros de entrada	"VAR_INPUT" Lista de declaración "END_VAR"	FBs, FCs
Parámetros de salida	"VAR_OUTPUT" Lista de declaración "END_VAR"	FBs, FCs
Parámetros de entrada/salida	"VAR_IN_OUT" Lista de declaración "END_VAR"	FBs, FCs
Variables estáticas	"VAR" Lista de declaración "END_VAR"	FBs
Variables temporales	"VAR_TEMP" Lista de declaración "END_VAR"	OBs, FBs, FCs

Con la palabra clave END_VAR se termina la lista de declaración.

La lista de declaración es la lista de variables de un tipo de declaración. La lista permite preasignar valores a las variables (excepción: VAR_TEMP). El ejemplo siguiente muestra la estructura de una entrada en la lista de declaración:

```

Duración_Motor1      :      S5TIME      :=      S5T#1H_30M  ;
Variable              Tipo de datos      Valor preasignado
    
```

Nota

- El nombre de la variable debe comenzar con una letra o con un carácter de subrayado. No se pueden dar nombres que correspondan a una de las palabras clave reservadas.
- Si los nombres de las variables de las declaraciones locales son idénticos a los de la tabla de símbolos, añada el signo # antes del nombre de las variables locales y ponga las variables de la tabla de símbolos entre comillas. En caso contrario, el bloque interpretará la variable como variable local.

13.2.3 Orden de los bloques en fuentes AWL

Los bloques llamados deben figurar delante de los bloques invocantes, es decir:

- El bloque OB1, que es el que más se usa para llamar otros bloques, debe figurar al final. Los bloques llamados por los bloques que invoca el OB1 tienen que figurar antes de éstos, etc.
- Los tipos de datos de usuario (UDTs) deben figurar delante de los bloques donde van a ser utilizados.
- Los bloques de datos con un tipo de datos de usuario asociado (UDT) deben figurar detrás del UDT.
- Los bloques de datos globales deben figurar delante de todos los bloques que los invocan.
- Un bloque de datos de instancia figura detrás del bloque de función asociado.
- El DB 0 se encuentra preasignado. Por tanto, no se puede crear ningún otro DB con dicho nombre.

13.2.4 Reglas para definir los atributos de sistema en fuentes AWL

Los atributos de sistema pueden ser asociados a bloques y parámetros. Se controla la configuración de avisos y enlaces, las funciones de manejo y visualización y la configuración para control de procesos.

Para la entrada en la fuente hay que tener en cuenta:

- Las palabras clave de los atributos de sistema empiezan siempre con S7_.
- Los atributos de sistema se especifican entre llaves.
- Sintaxis: {S7_identifier := 'string'}
Los diversos identificadores están separados por ";".
- Los atributos de sistema para bloques se encuentran delante de las propiedades de bloque y tras las palabras clave ORGANIZATION_ o TITLE.
- Los atributos de sistema para parámetros se encuentran en la declaración de parámetros, es decir, delante de los dos puntos de la declaración de datos.
- Para introducir los atributos de sistema se diferencia entre las mayúsculas y las minúsculas.

Los atributos de sistema para bloques pueden ser controlados y/o modificados cuando se esté trabajando con el método de entrada incremental con el comando de menú **Archivo > Propiedades** en la ficha "Atributos".

Los atributos de sistema para parámetros pueden ser controlados y/o modificados cuando se esté trabajando con el método de entrada incremental con el comando de menú **Edición > Propiedades del objeto**. Para ello, el cursor debe encontrarse en el campo "Nombre" de la declaración de parámetros.

13.2.5 Reglas para definir las propiedades de bloques en fuentes AWL

Mediante las propiedades de bloques podrá identificar mejor los bloques que ha creado (p.ej., en las diferentes versiones) o proteger los bloques contra cambios no autorizados.

Las propiedades de bloques pueden ser controladas y/o modificadas cuando se está trabajando con el método de entrada incremental con el comando de menú **Archivo > Propiedades** en las fichas "General 1a. parte" y "General 2a. parte".

Todas las demás propiedades sólo se pueden introducir en la fuente.

Indicaciones para la fuente:

- Introduzca las propiedades de bloques antes de la tabla de declaración de variables.
- A cada propiedad le corresponde una línea propia.
- Al final de la línea no puede haber un punto y coma.
- Las propiedades de bloques se indican mediante palabras clave.
- Si desea introducir propiedades de bloques, es necesario respetar el orden de los mismos conforme a la tabla de propiedades de bloques.
- Las propiedades posibles para cada tipo de bloque se indican en la asignación de propiedades de bloques a tipos de bloques.

Nota

Las propiedades de los bloques se indican también en el Administrador SIMATIC, en las propiedades de objeto de un bloque. Allí pueden editarse también las propiedades AUTHOR, FAMILY, NAME y VERSION.

Propiedades de bloques y orden

Si desea introducir propiedades de bloques, deberá respetar el orden indicado en la tabla que aparece a continuación.

Orden	Palabra clave / propiedad	Significado	Ejemplo
1.	[KNOW_HOW_PROTECT]	Protección de bloques: los bloques compilados con esta opción no permiten ver el área de instrucciones. La interfaz del bloque puede visualizarse pero no puede modificarse.	KNOW_HOW_PROTECT
2.	[AUTHOR:]	Nombre del autor: nombre de la empresa, del departamento u otros nombres (máx. 8 caracteres, sin espacios)	AUTHOR : Siemens, pero ninguna palabra clave
3.	[FAMILY:]	Nombre de la familia de bloques: p.ej. regulador. (máx. 8 caracteres, sin espacios).	FAMILY : Reguladores, pero ninguna palabra clave
4.	[NAME:]	Nombre del bloque (máx. 8 caracteres)	NAME : PID, pero ninguna palabra clave
5.	[VERSION: int1 . int2]	Número de versión del bloque (ambos números entre 0 y 15, es decir, 0.0 - 15.15)	VERSION : 3.10
6.	[CODE_VERSION1]	Indica si un FB es multiinstancia. Si desea declarar multiinstancias, el FB no puede llevar dicha propiedad.	CODE_VERSION1
7.	[UNLINKED] sólo para DBs	Los bloques de datos con la propiedad UNLINKED sólo se guardan en la memoria de carga pero no ocupan espacio en la memoria de trabajo ni se incluyen en el proyecto. A ellos sólo puede accederse con comandos MC7. Los contenidos de estos DBs sólo pueden transferirse a la memoria de trabajo con SFC 20 BLKMOV (S7-300. S7-400) ó SFC 83 READ_DBL (S7-300C).	
8.	[READ_ONLY] sólo para DBs	Protección contra escritura para bloques de datos; sus datos sólo pueden leerse, pero no modificarse.	FAMILY= ejemplos VERSION= 3.10 READ_ONLY

13.2.6 Propiedades válidas para los diferentes tipos de bloques

La siguiente tabla muestra qué propiedades pueden asignarse a qué tipos de bloques.

Propiedad	OB	FB	FC	DB	UDT
KNOW_HOW_PROTECT	•	•	•	•	–
AUTHOR	•	•	•	•	–
FAMILY	•	•	•	•	–
DESIGNACION	•	•	•	•	–
VERSION	•	•	•	•	–
UNLINKED	–	–	–	•	–
READ_ONLY	–	–	–	•	–

Protección de bloques (KNOW_HOW_PROTECT)

Puede proteger sus bloques indicando la palabra clave KNOW_HOW_PROTECT al programar el bloque en la fuente AWL.

La protección de bloques tiene las consecuencias siguientes:

- Si desea visualizar posteriormente el bloque compilado en el editor AWL, FUP o KOP incremental, no podrá ver el área de instrucciones del bloque.
- En la tabla de declaración de variables del bloque se visualizan sólo las variables de tipo var_in, var_out y var_in_out. Las variables de tipo var_stat y var_temp permanecerán ocultas.
- Hay que introducir la palabra clave KNOW_HOW_PROTECT antes que las demás propiedades.

Protección contra escritura de bloques de datos (READ_ONLY)

Para los bloques de datos puede activar una protección para evitar que éstos se sobrescriban al ejecutarse el programa. A tal efecto, el bloque de datos deberá existir como fuente AWL.

Introduzca en la fuente la palabra clave READ_ONLY. Esta palabra deberá figurar en una línea propia inmediatamente antes de la declaración de variables.

13.3 Estructura de bloques en fuentes AWL

Para estructurar los bloques en fuentes AWL se utilizan palabras clave. Dependiendo del tipo de bloque se diferencia la estructura de

- Bloques lógicos
- Bloques de datos
- Tipos de datos de usuario

13.3.1 Estructura de los bloques lógicos en fuentes AWL

Un bloque lógico se compone de las siguientes áreas, caracterizadas por las correspondientes palabras clave:

- Comienzo del bloque,
- Caracterizado por la palabra clave y el número o el nombre de bloque, como p.ej.
 - "ORGANIZATION_BLOCK OB 1" para un bloque de organización,
 - "FUNCTION_BLOCK FB6" para un bloque de función,
 - "FUNCTION FC 1 : INT" para una función. En el caso de las funciones se indica adicionalmente el tipo de función. Su tipo de datos puede ser simple o compuesto (excepto ARRAY y STRUCT), determinando el tipo de datos del valor de respuesta (RET_VAL). Si no se ha de obtener nuevamente ningún valor, hay que indicar la palabra clave VOID.
- Título del bloque (opcional), comenzando con la palabra clave "TITLE (longitud máxima del título: 64 caracteres).
- Comentario adicional, comenzando con dos barras inclinadas //.
- Indicación de las propiedades (opcional).
- Tabla de declaración de variables.
- Área de instrucciones, comenzando con BEGIN. El área de instrucciones se compone a su vez de uno o varios segmentos caracterizados con NETWORK. No se permite introducir números de segmento.
- Título opcional para cada segmento implementado, comenzando con la palabra clave "TITLE =" (longitud máxima del título: 64 caracteres).
- Comentarios adicionales para cada segmento, comenzando con dos barras inclinadas //.
- Fin del bloque, caracterizado por END_ORGANIZATION_BLOCK, END_FUNCTION_BLOCK o END_FUNCTION.
- Entre el tipo y el número de bloque debe haber un espacio en blanco. El nombre simbólico del bloque puede caracterizarse con comillas para no confundir los nombres de las variables locales con los nombres de la tabla de símbolos.

13.3.2 Estructura de bloques de datos en fuentes AWL

Un bloque de datos se compone de las siguientes áreas que comienzan con la palabra clave correspondiente:

- Comienzo del bloque, caracterizado por la palabra clave y el número o nombre de bloque, p.ej. DATA_BLOCK DB 26
- Indicación del UDT o del FB asociado al DB (opcional).
- Título opcional del bloque, comenzando con la palabra clave "TITLE =" (los títulos de más de 64 caracteres se cortan).
- Comentario opcional del bloque, comenzando con "//".
- Indicación de las propiedades (opcional).
- Tabla de declaración de variables (opcional)
- Area de asignación con valores preasignados, comenzando con BEGIN (opcional)
- Fin del bloque, comenzando con END_DATA_BLOCK.

Hay tres tipos de bloques de datos:

- Bloques de datos de usuario
- Bloques de datos con tipo de datos de usuario asociado (UDT)
- Bloques de datos asociados a un FB (DB de instancia)

13.3.3 Estructura de tipos de datos de usuario en fuentes AWL

Un tipo de datos de usuario se compone de las siguientes áreas, que comienzan con las correspondientes palabras clave:

- Comienzo del bloque, caracterizado por la palabra clave TYPE y el número o nombre, p.ej. TYPE UDT 20
- Indicación de un tipo de datos estructurado
- Fin del bloque, caracterizado por END_TYPE.

Los tipos de datos de usuario tienen que introducirse antes que los bloques en los que se van a utilizar.

13.4 Sintaxis y formatos de bloques en fuentes AWL

Las tablas indican la sintaxis y los formatos que se debe tener en cuenta al programar fuentes AWL. La sintaxis se representa de la manera siguiente:

- Cada elemento se describe en la columna derecha.
- Los elementos imprescindibles se representan entre comillas simples.
- Los corchetes [...] significan que no es necesario indicar el contenido de los mismos.
- Las palabras clave se indican en mayúsculas.

13.4.1 Formato de los OBs

La siguiente tabla muestra un resumen del formato de los bloques de organización en una fuente AWL:

Estructura	Descripción
"ORGANIZATION_BLOCK" ob_nr u ob_name	ob_nr es el número del bloque, p.ej., OB 1; ob_name es el nombre simbólico del bloque tal como figura en la tabla de símbolos
[TITLE=]	Comentario hasta <RETURN>; los títulos de más de 64 caracteres quedan cortados.
[Comentario de bloque]	Los comentarios se introducen comenzando con "/"
[Atributos de sistema para bloques]	Atributos de sistema para bloques
[Propiedades de bloques]	Propiedades de bloques
Tabla de declaración de variables	Declaración de variables temporales
"BEGIN"	Palabra clave para separar las instrucciones AWL de la tabla de declaración de variables
NETWORK	Comienzo de un segmento
[TITLE=]	Título de segmento (máx. 64 caracteres).
[Comentario de segmento]	Los comentarios se introducen comenzando con "/"
Lista de instrucciones AWL	Instrucciones del bloque
"END_ORGANIZATION_BLOCK"	Palabra clave para finalizar el bloque de organización

13.4.2 Formato de los FBs

La siguiente tabla muestra un resumen del formato de los bloques de función en una fuente AWL:

Estructura	Descripción
"FUNCTION_BLOCK" fb_nr ó fb_name	fb_nr es el número del bloque, p.ej., FB 6; fb_name es el nombre simbólico del bloque tal como figura en la tabla de símbolos
[TITLE=]	Comentario hasta <RETURN>; los títulos de más de 64 caracteres quedan cortados.
[Comentario de bloque]	Los comentarios se introducen comenzando con "/*"
[Atributos de sistema para bloques]	Atributos de sistema para bloques
[Propiedades de bloques]	Propiedades de bloques
Tabla de declaración de variables	Declaración de parámetros de entrada, de salida y de entrada/salida, así como de variables temporales o estáticas La declaración de parámetros puede contener las declaraciones de los atributos de sistema para los parámetros
"BEGIN"	Palabra clave para separar las instrucciones AWL de la tabla de declaración de variables
NETWORK	Comienzo de un segmento
[TITLE=]	Título de segmento (máx. 64 caracteres).
[Comentario de segmento]	Los comentarios se introducen comenzando con "/*".
Lista de instrucciones AWL	Instrucciones del bloque
"END_FUNCTION_BLOCK"	Palabra clave para terminar el bloque de función

13.4.3 Formato de las FCs

La siguiente tabla muestra un resumen del formato de las funciones en una fuente AWL:

Estructura	Descripción
"FUNCTION" fc_nr : fc_type ó fc_name : fc_type	fc_nr es el número del bloque, p.ej., FC 5; fc_name es el nombre simbólico del bloque tal como figura en la tabla de símbolos fc_type indica el tipo de datos al que pertenece el valor de respuesta (RET_VAL) de la función. Puede ser un tipo de datos simple o compuesto (excepto de ARRAY y STRUCT), o bien, de tipo VOID. Si desea utilizar atributos de sistema para el valor de respuesta (RET_VAL), deberá introducir los atributos de sistema de los parámetros antes de los dos puntos de la declaración de datos.
[TITLE=]	Comentario hasta <RETURN>; los títulos de más de 64 caracteres quedan cortados.
[Comentario de bloque]	Los comentarios se introducen comenzando con "/*"
[Atributos de sistema para bloques]	Atributos de sistema para bloques
[Propiedades de bloques]	Propiedades de bloques
Tabla de declaración de variables	Declaración de parámetros de entrada, de salida y de entrada/salida, así como de variables temporales
"BEGIN"	Palabra clave para separar las instrucciones AWL de la tabla de declaración de variables
NETWORK	Comienzo de un segmento
[TITLE=]	Título de segmento (máx. 64 caracteres).
[Comentario de segmento]	Los comentarios se introducen comenzando con "/*".
Lista de instrucciones AWL	Instrucciones del bloque
"END_FUNCTION"	Palabra clave para finalizar la función

13.4.4 Formato de los DBs

La siguiente tabla muestra un resumen del formato de los bloques de datos en una fuente AWL:

Estructura	Descripción
"DATA_BLOCK" db_nr ó db_name	db_nr es el número del bloque, p.ej., DB 5; db_name es el nombre simbólico del bloque tal como figura en la tabla de símbolos
[TITLE=]	Comentario hasta <RETURN>; los títulos de más de 64 caracteres quedan cortados.
[Comentario de bloque]	Los comentarios se introducen comenzando con "/*"
[Atributos de sistema para bloques]	Atributos de sistema para bloques
[Propiedades de bloques]	Propiedades de bloques
Area de declaración	Indicación del UDT o FB al que se refiere el bloque. Puede ser un número de bloque, un nombre simbólico o la indicación del tipo de datos compuesto.
"BEGIN"	Palabra clave para separar la tabla de declaración de la lista de asignación de valores.
[Asignación de valores iniciales]	A las variables se les pueden asignar determinados valores iniciales. Se pueden asignar constantes a variables o referenciar otros bloques.
"END_DATA_BLOCK"	Palabra clave para finalizar el bloque de datos

13.5 Crear fuentes AWL

13.5.1 Crear fuentes AWL

La fuente se debe crear debajo del programa S7 en una carpeta de fuentes. Las fuentes se pueden crear en el Administrador SIMATIC o en la ventana del editor.

Crear fuentes en el Administrador SIMATIC

1. Abra la carpeta "Fuentes" en cuestión, haciendo doble clic en la misma.
2. Para insertar una fuente AWL, elija el comando de menú **Insertar > Software S7 > Fuente AWL**.

Crear fuentes en la ventana del editor

1. Elija el comando **Archivo > Nuevo**.
2. En el cuadro de diálogo que aparece a continuación, elija la carpeta de fuentes del mismo programa S7, en el que se encuentra también el programa de usuario con los bloques.
3. Asígnele un nombre a la fuente que desea crear.
4. Confirme pulsando el botón "Aceptar".

La fuente se creará con el nombre que usted le haya dado y se visualizará en una ventana de trabajo para poderla editar posteriormente.

13.5.2 Editar fuentes S7

Las propiedades de las fuentes determinan en qué lenguaje de programación y con qué editor se ha de editar la fuente en cuestión. Al abrirse una fuente se arrancará siempre el editor correcto con el correspondiente lenguaje de programación. El software básico STEP 7 asiste la programación en fuentes AWL.

En el software opcional se ofrecen otros lenguajes de programación adicionales. Si ha cargado el software opcional en cuestión en el ordenador, podrá arrancar el editor correspondiente haciendo doble clic en la fuente.

Proceda de la siguiente forma:

1. Abra la carpeta "Fuentes" en cuestión, haciendo doble clic en la misma.
2. Arranque el editor necesario
 - haciendo doble clic en la fuente deseada en la mitad derecha de la ventana o
 - seleccionando la fuente en la mitad derecha de la ventana y activando el comando **Edición > Abrir objeto**.

13.5.3 Definir el diseño del texto fuente

Para aumentar la legibilidad del texto en los archivos fuente, seleccione el comando de menú **Herramientas > Preferencias** y en la ficha "Texto fuente" especifique el formato, la fuente y el color de los diferentes componentes del texto.

Puede especificar p.ej. que el número de líneas y las palabras clave deben visualizarse en mayúsculas.

13.5.4 Insertar plantillas de bloques en fuentes AWL

Para programar en fuentes AWL se le ofrecen plantillas de bloques para OBs, FBs, FCs, DBs, DBs de instancia, DBs de UDTs y UDTs. Las plantillas de bloques le ayudan a introducir y respetar la sintaxis y la estructura.

Proceda de la siguiente forma:

1. Active la ventana de trabajo de la fuente en la que desea insertar la plantilla.
2. Sitúe el cursor en la posición detrás de la cual desea insertar la plantilla de bloque.
3. Elija la plantilla deseada con el correspondiente comando de menú **Insertar > Plantilla de bloques > OB/FB/FC/DB/IDB/DB de UDT/UDT**.

La plantilla de bloques se insertará detrás de la posición del cursor.

13.5.5 Insertar el contenido de otras fuentes AWL

En su fuente AWL puede insertar también el contenido de otras fuentes.

Proceda de la siguiente forma:

1. Active la ventana de trabajo de la fuente en la que desea insertar otra fuente.
2. Sitúe el cursor en la posición detrás de la cual desea insertar el contenido de la fuente.
3. Elija el comando de menú **Insertar > Objeto > Archivo**.
4. Elija la fuente deseada en el cuadro de diálogo que aparece a continuación.

El contenido de la fuente seleccionada se insertará detrás de la posición del cursor. La separación de las líneas no se pierde.

13.5.6 Insertar en fuentes AWL el código de bloques existentes

En su fuente AWL puede insertar el correspondiente código de bloques creados en KOP, FUP o AWL. Ello es posible en el caso de los OBs, FBs, FCs, DBs y UDTs.

Proceda de la siguiente forma:

1. Active la ventana de trabajo de la fuente en la que desea insertar un bloque.
2. Sitúe el cursor en la posición detrás de la cual desea insertar el texto del bloque.
3. Elija el comando de menú **Insertar > Objeto > Bloque**.
4. En el cuadro de diálogo que aparece a continuación, elija el bloque deseado.

Partiendo del bloque se genera automáticamente una fuente. El contenido de la fuente se inserta detrás de la posición del cursor.

13.5.7 Insertar fuentes externas

Puede crear y editar su fuente con cualquier editor ASCII, importarla a continuación en un proyecto y compilarla en bloques individuales. A tal efecto es necesario importar las fuentes a la carpeta "Fuentes" del programa S7, en cuyo programa de usuario S7 se hayan de depositar los bloques creados en la compilación.

Para compilar y cargar una tabla GD:

1. Seleccione la carpeta "Fuentes" del programa S7 a la que desea importar las fuentes externas.
2. Elija el comando **Insertar > Fuente externa**.
3. En el cuadro de diálogo que aparece a continuación, indique la fuente que desea importar.

El nombre de archivo de la fuente a importar deberá tener una extensión válida. El tipo de fuente se define en STEP 7 conforme a la extensión del archivo. Por ejemplo, al importar un archivo con la extensión **.awl** se creará una fuente AWL. Las extensiones válidas se indican en el cuadro de diálogo bajo "Tipo de archivo".

Nota

Con el comando **Insertar > Fuente externa** puede insertar también las fuentes que haya creado con la versión 1 de STEP 7.

13.5.8 Generar fuentes AWL de bloques

Puede crear una fuente AWL con bloques existentes, editándola con cualquier editor de texto. La fuente generada se crea en la carpeta de fuentes del programa S7.

Para compilar y cargar una tabla GD:

1. En el editor de programas, elija el comando de menú **Archivo > Generar fuente**.
2. Elija en el cuadro de diálogo la carpeta de fuentes en la que desea crear la nueva fuente.
3. Introduzca en el cuadro de texto el nombre que desea darle a la fuente.
4. Elija en el cuadro de diálogo "Seleccionar bloques de STEP7" el o los bloques que desee generar en la fuente indicada anteriormente. Los bloques seleccionados aparecerán en el cuadro de lista a la derecha.
5. Confirme pulsando el botón "Aceptar".

Con los bloques seleccionados se genera una fuente AWL coherente que se muestra en una ventana de trabajo para su posterior edición.

13.5.9 Importar fuentes

Para importar una fuente de un directorio cualquiera a un proyecto:

1. En el Administrador SIMATIC, seleccione la carpeta de fuentes a donde desea importar la fuente.
2. Elija el comando **Insertar > Fuente externa**.
3. En el cuadro de diálogo visualizado, elija el directorio y la fuente a importar.
4. Haga clic en el botón "Abrir".

13.5.10 Exportar fuentes

Para exportar una fuente de un proyecto a un directorio de destino cualquiera:

1. Seleccione la fuente en la carpeta "Fuentes".
2. Elija en el Administrador SIMATIC el comando de menú **Edición > Exportar fuente**
3. Indique, en el cuadro de diálogo que se visualiza, el directorio de destino y el nombre del archivo.
4. Haga clic en el botón "Guardar".

Nota

Si el nombre del objeto carece de extensión, al nombre del archivo se le añadirá una extensión derivada de su tipo. Por ejemplo, una fuente AWL "**prog**" se exportará al archivo "**prog.awl**".

Si el nombre del objeto ya tiene una extensión de archivo válida, ésta se conservará y no será necesario añadirle ninguna otra. Por ejemplo, una fuente AWL "**prog.awl**" se exportará al archivo "**prog.awl**".

Si un nombre de un objeto tiene una extensión de archivo no válida (es decir, si el nombre contiene un punto), no se añadirá ninguna extensión.

Las extensiones de archivo válida se pueden apreciar en el cuadro de diálogo "Exportar fuente" bajo "Tipo de archivo".

13.6 Guardar y compilar fuentes AWL y comprobar su coherencia

13.6.1 Guardar fuentes AWL

Las fuentes AWL se pueden guardar en todo momento en su estado actual. En dicha operación no se compila el programa ni tampoco se comprueba la sintaxis, por lo que también se guardan los posibles errores.

Los errores de sintaxis se detectan y se indican al compilar la fuente o al comprobarse la coherencia.

Para guardar una fuente con el mismo nombre:

1. Active la ventana de trabajo de la fuente que desea guardar.
2. Elija el comando de menú **Archivo > Guardar**.

Para guardar una fuente con otro nombre/en otro proyecto:

1. Active la ventana de trabajo de la fuente que desea guardar.
2. Elija el comando de menú **Archivo > Guardar como**.
3. Elija en el cuadro de diálogo que aparece a continuación la carpeta de fuentes a la que se ha de asignar la fuente e introduzca su nuevo nombre.

13.6.2 Comprobar la coherencia en fuentes AWL

Con el comando de menú **Archivo > Comprobar coherencia** se pueden visualizar los posibles errores de sintaxis en la fuente AWL. A diferencia de la compilación, al comprobarse la coherencia no se generan bloques.

Una vez comprobada la coherencia aparece un cuadro de diálogo que le comunica el número total de errores localizados.

Si existen errores, éstos se listan individualmente en la parte inferior de la ventana de trabajo, indicándose la línea correspondiente. Los errores deben corregirse antes de compilar la fuente para que puedan crearse todos los bloques.

13.6.3 Buscar errores en fuentes AWL

La ventana de trabajo de las fuentes está dividida en dos. En la mitad inferior se listan los siguientes errores:

- los errores que se detectan después de iniciar un proceso de compilación con el **comando de menú Archivo > Compilar** o
- los errores que se detectan después de iniciar una comprobación de coherencia, con el comando de menú **Archivo > Comprobar coherencia**.

Para poder encontrar un error en la fuente, sitúe el cursor en el correspondiente mensaje de error en la ficha "Errores" de la ventana de resultados. Así se selecciona automáticamente la correspondiente línea de texto en el área de instrucciones. Además, el mensaje de error aparece en la barra de estado.

13.6.4 Compilar fuentes AWL

Requisitos

Para poder compilar en bloques un programa creado en una fuente, se deberán cumplir las siguientes condiciones:

- Sólo se pueden compilar fuentes que estén guardadas en la carpeta "Fuentes" bajo un programa S7.
- Además de la carpeta "Fuentes" también deberá existir bajo el programa S7 una carpeta "Bloques" en la que se guardarán los bloques que se creen en la compilación. Los bloques programados en la fuente sólo se crearán si la fuente ha sido compilada sin errores. Si una fuente comprende varios bloques se crearán sólo los que no contengan errores. Luego puede abrir, editar, cargar en la CPU y comprobar cada bloque.

Procedimiento en el editor

1. Abra la fuente que desea compilar. La fuente debe encontrarse en la carpeta de fuentes del programa S7 en cuyo programa de usuario S7 desea depositar los bloques compilados.
2. Elija el comando de menú **Archivo > Compilar**.
3. A continuación aparece el cuadro de diálogo "Informe de la compilación" que indica el número de líneas compiladas y de errores de sintaxis encontrados.

Los bloques indicados en el archivo sólo se crean si la fuente ha sido compilada sin errores. Si una fuente comprende varios bloques se crearán sólo los que no contengan errores. Las advertencias no evitan que se creen bloques.

Los errores de sintaxis que se detectan durante la compilación se representarán en la mitad inferior de la ventana de trabajo y deben corregirse para que se puedan crear también dichos bloques.

Procedimiento en el Administrador SIMATIC

1. Abra la carpeta "Fuentes" en cuestión, haciendo doble clic en la misma.
2. Elija una o varias fuentes a compilar. No es posible arrancar la compilación de una carpeta "Fuentes" cerrada para compilar todas las fuentes contenidas en la misma.
3. Elija el comando **Archivo > Compilar**, para arrancar la compilación. Dependiendo de la fuente elegida se arrancará el correspondiente compilador. Los bloques que se hayan compilado sin errores se depositarán a continuación en la carpeta "Bloques" bajo el programa S7.
Los errores de sintaxis detectados durante la compilación se indican en un cuadro de diálogo, siendo necesario corregirlos para que se puedan crear también esos bloques.

13.7 Ejemplos de fuentes AWL

13.7.1 Ejemplos de declaración de variables en fuentes AWL

Variables del tipo de datos simples

```

// Los comentarios se separan de la tabla de declaración con dos barras
// inclinadas.
VAR_INPUT // Palabra clave variable de entrada
in1 : INT; // Nombre y tipo de la variable se separan con un ":"
in3 : DWORD; // Cada declaración de variable se termina con un punto y coma.
in2 : INT := 10; // Definición opcional del valor inicial en la declaración
END_VAR // Fin de la declaración de variables de un mismo tipo de declaración
VAR_OUTPUT // Palabra clave variable de salida
out1 : WORD;
END_VAR // Palabra clave variable temporal
VAR_TEMP
temp1 : INT;
END_VAR

```

Variables de tipo de datos "Array"

```

VAR_INPUT // Variable de entrada
campo1 : ARRAY [1..20] of INT; // campo1 es un campo unidimensional
campo2 : ARRAY [1..20, 1..40] of DWORD; // campo2 es un campo bidimensional
END_VAR

```

Variables del tipo de datos "estructura"

```

VAR_OUT // Variable de salida
SALIDA1: STRUCT // SALIDA1 es del tipo de datos STRUCT
var1 : BOOL; // Elemento 1 de la estructura
var2 : DWORD; // Elemento 2 de la estructura
END_STRUCT; // Fin de la estructura
END_VAR

```

13.7.2 Ejemplo de OBs en fuentes AWL

```

ORGANIZATION_BLOCK OB 1
TITLE = Ejemplo de OB 1 con diferentes llamadas de bloques
// Los 3 segmentos representados muestran llamadas de bloques
//con y sin parámetros

{S7_pdiag := 'true'} //Atributo de sistema para bloques
AUTHOR:      :      Siemens
FAMILY      :      Ejemplo
Nombre      :      Test_OB
VERSION     :      1.1
VAR_TEMP
Valor intermedio : INT; // Memoria intermedia
END_VAR

BEGIN

NETWORK
TITLE = Llamada de una función con transferencia de parámetros
// Transferencia de parámetros en una línea
CALL FC1 (param1 :=E0.0,param2 :=E0.1);

NETWORK
TITLE = Llamada de un bloque de función con
// transferencia de parámetros
// transferencia de parámetros en varias líneas
CALL control de semáforo , DB 6 ( // Nombre del FB, bloque de datos de instancia
dur_g_p      := S5T#10S, // Asignación de valores actuales a parámetros

del_r_p      := S5T#30S,
starter      := TRUE,
t_dur_y_car  := T 2,
t_dur_g_ped  := T 3,
t_delay_y_car := T 4,
t_dur_r_car  := T 5,
t_next_red_car := T 6,
r_car        := "re_main", // Agregar comillas
y_car        := "ye_main", // Nombres de la tabla de símbolos
g_car        := "gr_main",
r_ped        := "re_int",
g_ped        := "gr_int");

NETWORK
TITLE = Llamada de un bloque de función con
// transferencia de parámetros
// indicación de parámetro en una línea
CALL FB10, DB100 (para1 :=E0.0,para2 :=E0.1);

END_ORGANIZATION_BLOCK

```

13.7.3 Ejemplo de FCs en fuentes AWL

```

FUNCTION FC 1: VOID
// ¡Sólo a causa del Call!!
VAR_INPUT
  param1 : bool;
  param2 : bool;
END_VAR
begin
end_function

FUNCTION FC2 : INT
TITLE = Aumento del número de piezas
// Mientras que el valor transferido sea < 1000, esta función aumenta el
//valor transferido. Si el número de piezas es superior a 1000, se obtiene
// nuevamente la función (RET_VAL) "-1" a través del valor de respuesta.

AUTHOR      :      Siemens
FAMILY      :      Por
Nombre      :      STUECK
VERSION     :      1.0

VAR_IN_OUT
STUECKZAHL : INT;           // Número de piezas fabricado hasta ahora
END_VAR

BEGIN

NETWORK
TITLE = Aumento del número de piezas en 1
// En tanto que el número de piezas actual sea inferior a 1000,
// puede incrementarse (+1)
L STUECKZAHL; L 1000;           // Ejemplo de varias
> I; SPB ERR;                 // instrucciones en una línea.
L 0; T RET_VAL;
L STUECKZAHL; INC 1; T STUECKZAHL; BEA;
ERR: L -1;
T RET_VAL;
END_FUNCTION

FUNCTION FC3 {S7_m_c := 'true'} : INT
TITLE = Aumento del número de piezas
// Mientras que el valor transferido sea < 1000, esta función aumenta el
//valor transferido. Si el número de piezas es
//superior a 1000, se obtiene
// nuevamente la función (RET_VAL) "-1" a través del valor de respuesta.
//
//RET_VAL tiene aquí un atributo de sistema para parámetros

```

```

AUTHOR      :      SIEMENS
FAMILY     :      POR
NOMBRE     :      STUECK
VERSION    :      1.0

VAR_IN_OUT
STUECKZAHL {S7_visible := 'true'}: INT;    // Número de piezas fabricado actualmente
//Atributos de sistema para parámetros
END_VAR

BEGIN

NETWORK
TITLE = Aumento del número de piezas en 1
// En tanto que el número de piezas actual sea inferior a 1000,
// puede incrementarse (+1)
L STUECKZAHL; L 1000;                        // Ejemplo de varias
> I; SPB ERR;                                // instrucciones en una línea.
L 0; T RET_VAL;
L STUECKZAHL; INC 1; T STUECKZAHL; BEA;
ERR: L -1;
T RET_VAL;

END_FUNCTION

```

13.7.4 Ejemplo de FBs en fuentes AWL

```

FUNCTION_BLOCK FB 6
TITLE = Control de semáforo sencillo
// Control de un semáforo para un paso de peatones
// en la calle principal

{S7_pdiag := 'true'} //Atributo de sistema para bloques
AUTHOR      :      SIEMENS
FAMILY      :      Semáforo
Nombre      :      Semáforo01
VERSION     :      1.3

VAR_INPUT

starter      :      BOOL      :=      FALSE; // Petición de un peatón de cruzar la calle
t_dur_y_car  :      TIMER;    // Duración fase verde peatón
t_next_r_car :      TIMER;    // Duración entre fases rojas para vehículos
t_dur_r_car  :      TIMER;
anz          {S7_server := 'alarm_archiv'; S7_a_type := 'alarm_8'} :DWORD;
// Número de vehículos
// anz tiene un atributo de sistema para parámetros

END_VAR
VAR_OUTPUT

g_car        :      BOOL      :=      FALSE; // VERDE para vehículos

END_VAR
VAR
condition    :      BOOL      :=      FALSE; // Petición de cambiar a rojo para
//vehículos
END_VAR

BEGIN
NETWORK
TITLE = Petición de cambiar a rojo para los vehículos
// Una vez transcurrido un tiempo mínimo,
//la señal del paso de peatones
//indicará la petición de cambiar a rojo para los vehículos.
      U(
      U      #starter; // Señal en el paso de peatones y vencimiento
      U      #t_next_r_car; // del tiempo entre dos fases rojas
      O      #condition; // o petición de cambiar a rojo ("autoparada")
      );
      UN      #t_dur_y_car; // y actualmente sin semáforo rojo
      =      #condition; // petición de cambiar a rojo
NETWORK
TITLE = Semáforo verde para los vehículos
      UN      #condition; // sin petición de cambiar a rojo para los vehículos
      =      #g_car; // VERDE para los vehículos
NETWORK
TITLE = Duración de la fase amarilla para los vehículos
      // programa adicional para realizar el
      // control del semáforo

END_FUNCTION_BLOCK
    
```

```
FUNCTION_BLOCK FB 10
```

```
VAR_INPUT
```

```
  para1 : bool;
```

```
  para2: bool;
```

```
end_var
```

```
begin
```

```
end_function_block
```

```
data_block db 10
```

```
FB 10
```

```
begin
```

```
end_data_block
```

```
data_block db 6
```

```
FB 6
```

```
begin
```

```
end_data_block
```

13.7.5 Ejemplos de DBs en fuentes AWL

Bloque de datos:

```
DATA_BLOCK DB 10
TITLE = DB Ejemplo 10
STRUCT
    aa : BOOL;      // variable aa del tipo BOOL
    bb : INT;       // variable bb del tipo INT
    cc : WORD;
END_STRUCT;
BEGIN // Asignación de valores iniciales
    aa := TRUE;
    bb := 1500;
END_DATA_BLOCK
```

DB con tipo de datos de usuario asociado:

```
DATA_BLOCK DB 20
TITLE = DB (UDT) Ejemplo
UDT 20 // Indicación de los UDTs asociados
BEGIN
    start := TRUE; // Asignación de valores iniciales
    soll := 10;
END_DATA_BLOCK
```

Nota

El UDT utilizado debe figurar en la fuente antes del bloque de datos.

DB con bloque de función asociado:

```
DATA_BLOCK DB 30
TITLE = DB (FB) Ejemplo
FB 30 // Indicación de los FBs asociados
BEGIN
    start := TRUE; // Asignación de valores iniciales
    soll := 10;
END_DATA_BLOCK
```

Nota

El FB debe figurar en la fuente antes del bloque de datos.

13.7.6 Ejemplo de UDTs en fuentes AWL

```
TYPE UDT 20
STRUCT
    arranque : BOOL;           //variable del tipo BOOL
    teórico : INT;             //variable del tipo INT
    valor : WORD;              //variable del tipo WORD
END_STRUCT;
END_TYPE
```


14 Mostrar datos de referencia

14.1 Sinopsis de los datos de referencia

Es posible crear y evaluar datos de referencia para probar y modificar cómodamente el programa de usuario. Los datos de referencia se utilizan, p.ej., como

- sinopsis de todo el programa de usuario,
- base para efectuar cambios y tests y
- complemento de la documentación del programa.

En la tabla siguiente se indica la información que figura en las diversas vistas:

Ver	Aplicación
Lista de referencias cruzadas	Panorámica de los operandos de las áreas de memoria E, A, M, P, T, Z y de los DB, FB, FC, SFB, SFC utilizados en el programa de usuario.
Plano de ocupación	Panorámica de los bits de los operandos de las áreas de memoria E, A y M, así como de los temporizadores y contadores que ya estén ocupados en el programa de usuario. Es una base importante para buscar errores o para efectuar cambios en el programa de usuario.
Estructura del programa	Jerarquía de llamada de los bloques en un programa de usuario y panorámica de los bloques utilizados y de su interdependencia.
Símbolos no utilizados	Panorámica de los símbolos definidos en la tabla de símbolos, pero que no se utilizan en las partes del programa de usuario para las que existen datos de referencia.
Operandos sin símbolo	Panorámica de las direcciones absolutas (operandos absolutos y bloques) que se utilizan en las partes del programa de usuario para las que existen datos de referencia. No obstante, para dichas direcciones no se ha definido ningún símbolo en la tabla de símbolos.

Los datos de referencia del programa de usuario seleccionado comprenden las listas mencionadas en la tabla. Es posible crear y visualizar varias listas de un programa de usuario o de diversos programas.

Visualizar varias vistas a la vez

Visualizando listas en distintas ventanas se puede, por ejemplo:

- comparar una determinada lista para distintos programas de usuario S7.
- tener en la pantalla, una junto a otra, distintas vistas de una lista optimizadas de distinta manera, p.ej., una lista de referencias cruzadas. Así puede visualizar, p.ej., en una lista de referencias cruzadas sólo las entradas y en la otra, sólo las salidas de un programa de usuario.
- abrir varias listas de un programa de usuario S7 a la vez, p.ej., la estructura del programa y la lista de referencias cruzadas.

14.1.1 Lista de referencias cruzadas

La lista de referencias cruzadas ofrece una panorámica de los operandos utilizados en el programa de usuario S7.

En la lista de referencias cruzadas se visualizan los operandos de las áreas de memoria de entradas (E), salidas (A), marcas (M), temporizadores (T), contadores (Z), bloques de función (FB), funciones (FC), bloques de función de sistema (SFB), funciones de sistema (SFC), periferia (P) y bloque de datos (DB), así como sus direcciones (dirección absoluta, nombre) y su utilización. La lista se visualiza en una ventana de trabajo. En la barra del título de esta ventana se muestra el nombre del programa de usuario al que pertenece la lista de referencias cruzadas.

Cada línea de la ventana corresponde a una entrada de la lista de referencias cruzadas. La función de búsqueda permite encontrar puntualizadamente los operandos y los símbolos.

La lista de referencias cruzadas es la representación estándar de los datos de referencia. Este ajuste predeterminado puede ser modificado.

Estructura

Cada entrada de la lista de referencias cruzadas tiene las siguientes columnas:

Columna	Contenido / significado
Operando (símbolo)	Dirección del operando
Bloque (símbolo)	Indicación del bloque donde se utiliza el operando
Acceso	Tipo de acceso al operando: de lectura (R) o de escritura (W)
Lenguaje	Información sobre el bloque, depende del lenguaje de programación del bloque
Punto de aplicación	Haga doble clic en el campo de un punto de aplicación para saltar al punto de aplicación del operando seleccionado.

Las columnas, bloque, tipo y lenguaje y punto de aplicación serán mostradas sólo si se han seleccionado las correspondientes propiedades para la lista de referencias cruzadas. Esta información sobre el bloque varía según el lenguaje en el que se creó el bloque.

Con el ratón puede modificar el ancho de las columnas de la lista de referencias cruzadas.

Ordenación

La lista de referencias cruzadas está ordenada por áreas de memoria, siendo éste el ajuste predeterminado. Si hace clic en un título de una columna, puede ordenar las entradas de la misma.

Ejemplo de la estructura de la lista de referencias cruzadas

Operando (símbolo)	Bloque (símbolo)	Acceso	Lenguaje	Punto de aplicación
E 1.0 (Motor on)	OB 2	R	AWL	NW 2 Anw 33 /O
M1.2 (Bit de marcas)	FC 2	R	KOP	Seg33
Z2 (Contadores2)	FB 2		FUP	Seg2

14.1.2 Estructura del programa









La estructura del programa describe la jerarquía de llamada de los bloques dentro de un programa de usuario S7. Además ofrece una visión general de los bloques utilizados, su interdependencia y su requerimiento de datos locales.

Con el comando de menú **Ver > Filtrar** en la ventana "Mostrar datos de referencia S7" se abre un cuadro de diálogo compuesto por varias fichas. En la ficha "Estructura del programa" puede definir la representación de la estructura del programa.

Puede elegir entre

- estructura de llamada y
- estructura de dependencias.

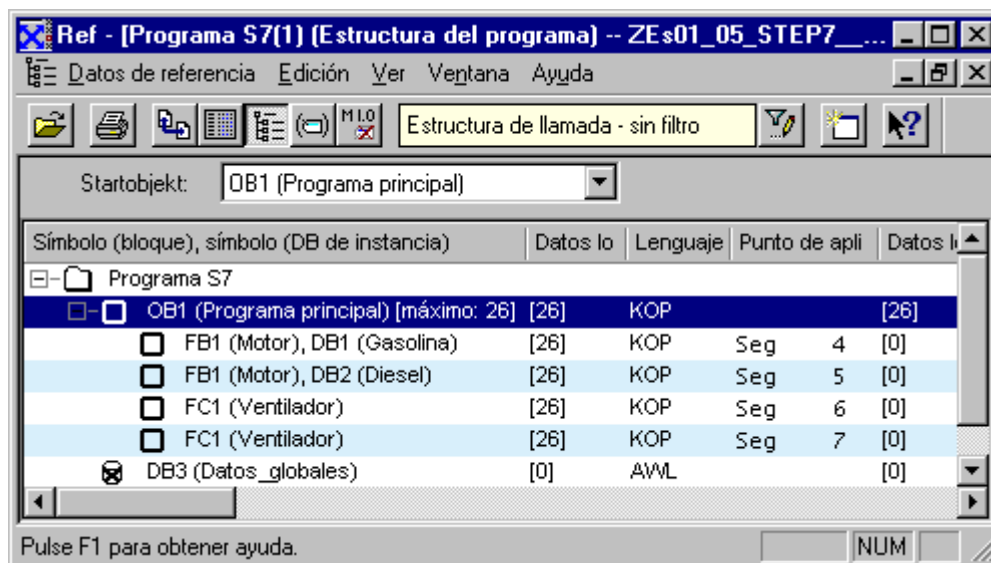
Símbolos de la estructura del programa

<u>Símbolo</u>	<u>Significado</u>
	Llamada regular de bloque (CALL FB10)
	Llamada incondicional de bloque (UC FB10)
	Llamada condicional de bloque (CC FB10)
	Bloque de datos
	Recursión
	Recursión y llamada condicional
	Recursión y llamada incondicional
	bloque no llamado

- Los bloques que se llamen repetidamente se destacan gráficamente en la estructura de llamada.
- Las recursiones dentro de la jerarquía de llamada se representan con distintos símbolos.
- Los bloques llamados regularmente (CALL), los llamados condicionalmente (CC) o los llamados incondicionalmente (UC) se caracterizan con distintos símbolos.
- Los bloques que no han sido llamados se visualizan al final de la estructura del programa, están marcados con una cruz negra y su estructura no será desglosada.

Estructura de llamada

Se representa toda la jerarquía de llamada.



Si quiere crear la estructura del programa para la totalidad de los bloques de organización (OB), pero el OB 1 no está en el programa de usuario S7, o se ha indicado un bloque de arranque que no existe en el programa, entonces se le pedirá automáticamente que especifique otro bloque para que sirva como raíz de la estructura.

Ajustando convenientemente las opciones se puede desactivar la visualización de llamadas múltiples de bloques, tanto en lo que se refiere a la estructura de llamada como a la estructura de dependencias.

Visualización del requerimiento máximo de datos locales en la estructura de llamada

Para facilitarle una panorámica rápida del requerimiento de datos locales de los OBs en el programa de usuario visualizado, puede apreciar lo siguiente en la estructura en árbol:

- el requerimiento máximo de datos locales por cada OB
- el requerimiento de datos locales por ruta

Dicha visualización se puede activar y desactivar en la ficha "Estructura del programa".

En caso de que haya OBs de errores síncronos (OB 121, OB 122), detrás del valor numérico del requerimiento máximo de datos locales aparecerá el signo "+" y el requerimiento adicional para los OBs de errores síncronos.

Estructura de dependencias

La estructura de dependencias muestra para cada bloque del proyecto las dependencias con otros bloques. En la izquierda aparece el bloque y bajo éste los bloques que llaman o utilizan el primero.

Visualización de los bloques borrados

Las líneas de los bloques borrados se destacan en rojo.

14.1.3 Plano de ocupación

El plano de ocupación muestra los operandos ya ocupados en el programa de usuario. Constituye una base importante para buscar errores o para efectuar cambios en el programa de usuario.

Al visualizar el plano de ocupación se obtiene una relación de los distintos bits y sus respectivas áreas de memoria Entradas (E), Salidas (A) o Marcas (M), así como los temporizadores (T) y contadores (Z).

El plano de ocupación se visualiza en una ventana de trabajo. La barra del título de la ventana de trabajo muestra el nombre del programa de usuario S7 al que pertenece el plano de ocupación.

Vista general E/A/M

Cada línea contiene un byte del área de memoria en el cual están identificados los ocho bits según el acceso. Igualmente se indica si el acceso se realiza por un acceso a byte, a palabra o a palabra doble.

Signos utilizados en la vista general E/A/M:

Fondo blanco	El operando no está direccionado y por tanto todavía no está asignado
X	El operando se utiliza directamente
Fondo azul	El operando se ejecuta indirectamente (acceso a byte, palabra o palabra doble)

Columnas de la vista general E/A/M

Columna	Contenido / significado
7	Número de bit del byte correspondiente
6	
5	
4	
3	
2	
1	
0	
B	El byte está ocupado por un acceso a byte
W	El byte está ocupado por un acceso a palabra
D	El byte está ocupado por un acceso a palabra doble

Ejemplo

El ejemplo siguiente muestra la estructura característica del plano de ocupación para entradas, salidas y marcas (E/A/M).

△	7	6	5	4	3	2	1	0	B	W	D
EB0		X	X	X	X	X	X				
EB1		X	X	X		X	X	X			
AB4						X	X	X			
AB5		X	X	X		X	X	X			
MB1										↑	
MB2										↑	
MB3										↑	
MB4										↑	
MB5										↑	

En la primera línea se muestra la ocupación del byte de entrada EB 0. Las entradas del operando EB 0 se utilizan directamente (acceso de bit). En las columnas "1", "2", "3", "4", "5" y "6" aparece una "X" para indicar una acceso de bit.

Sin embargo también tiene lugar un acceso de palabra a los bytes de marcas 1 y 2, 2 y 3 o 4 y 5. Por ello aparece en la columna "W" una "barra" y las líneas aparecen marcadas sobre fondo celeste. La punta negra de las barras indica el comienzo del acceso de palabra.

Vista general T/Z

En cada línea se representan 10 temporizadores o 10 contadores.

Ejemplo

	0	1	2	3	4	5	6	7	8	9
T 00-09	.	T1	T6	.	.	.
T 10-19	.	.	T12	T17	.	T19
T 20-29	T24
Z 00-09	.	.	Z2	Z7	.	.
Z 10-19	Z19
Z 20-29
Z 30-39	Z34

En este ejemplo están ocupados los temporizadores T1, T6, T12, T17, T19, T24 y los contadores Z2, Z7, Z19, Z34.

Las listas están ordenadas alfabéticamente. Las entradas se pueden ordenar haciendo clic en el encabezado de la columna en cuestión.

14.1.4 Símbolos no utilizados

Esta función ofrece una panorámica de todos los símbolos que tengan las siguientes propiedades:

- Los símbolos están definidos en la tabla de símbolos.
- No obstante, no se utilizan en las partes del programa de usuario para las que existen datos de referencia.

La lista se visualiza en una ventana de trabajo. En la barra del título de esta ventana se muestra el nombre del programa de usuario al que pertenece la lista.

Cada línea visualizada en la ventana corresponde a una entrada de la lista. En una línea se incluyen el operando, el símbolo, el tipo de datos y el comentario.

Columna	Contenido / Significado
Operando	Dirección absoluta del operando
Tipo de datos	Tipo de datos del operando
Comentario	Comentario del operando que se indica en la tabla de símbolos

Ejemplo de una lista de símbolos no utilizados

Símbolo	Operando	Tipo de datos	Comentario
MS1	E103.6	BOOL	guardamotor 1
MS2	E120.5	BOOL	guardamotor 2
MS3	E121.3	BOOL	guardamotor 3

Las entradas se pueden ordenar haciendo clic en el encabezado de la columna en cuestión.

Del mismo modo, también es posible seleccionar de la lista los símbolos ya no utilizados y eliminarlos de la tabla con la función "Borrar símbolos".

14.1.5 Operandos sin símbolo

Al visualizar la lista de operandos sin símbolo obtiene una lista de aquellos operandos que, si bien se están utilizando en el programa de usuario S7, aún no se han definido en la tabla de símbolos, es decir, no se les ha dado un nombre simbólico. La lista se visualiza en una ventana de trabajo. En la barra del título de esta ventana se muestra el nombre del programa de usuario al que pertenece la lista.

Una línea incluye el operando y el número de veces que ha sido utilizado dicho operando. Las entradas aparecen ordenadas por operandos.

Ejemplo:

Operando	Cantidad
A 2.5	4
E 23.6	3
M 34.1	20

Los operandos que aparecen sin símbolo en la lista se pueden seleccionar y asignar a nombres simbólicos mediante la función "Editar símbolos".

14.1.6 Visualizar la información del bloque en KOP, FUP, AWL

La información sobre el bloque en KOP, FUP y AWL - que comprende el lenguaje del bloque y los detalles.- se visualiza en la lista de referencias cruzadas y en la estructura del programa.

En la vista "Estructura del programa" la información sobre el bloque sólo se mostrará si en los ajustes del filtro de la ficha "Estructura del programa" se escogió la visualización "Estructura de llamadas" con las correspondientes opciones.

En la vista "Referencias cruzadas" se mostrará la información sobre el bloque al elegir el comando **Ver > Filtrar**.

- Active las casillas de verificación "Lenguaje de bloque" y "Detalles" en la ficha "Referencias cruzadas" del cuadro de diálogo "Filtrar" para visualizar la información sobre el bloque.

Dicha información, en la cual aparecen abreviaturas, varía según el lenguaje en el que se haya programado el bloque.

Lenguaje	Segmento	Instrucción	Operación
AWL	Seg	Ins	/
KOP	Seg		
FUP	Seg		

Seg e **Ins** indican en qué segmento y en qué instrucción se utiliza el operando (referencias cruzadas) o se llama el bloque (estructura del programa).

Mostrar la información sobre el bloque en lenguajes de programación opcionales

La Ayuda en pantalla sobre la información sobre el bloque se puede llamar al estar instalado el correspondiente software opcional.

14.2 Trabajar con datos de referencia

14.2.1 Métodos de visualización de los datos de referencia

Para visualizar los datos de referencia puede elegir una de las siguientes alternativas:

Visualización en el Administrador SIMATIC:

1. Seleccione en la ventana de proyecto en modo de visualización "offline" la carpeta "Bloques".
2. Elija el comando **Herramientas > Datos de referencia > Mostrar**.

Visualización en la ventana del editor:

1. En la carpeta "Bloques", abra un bloque.
2. En la ventana del correspondiente editor, elija el comando de menú **Herramientas > Datos de referencia**.

Aparece el diálogo "Preferencias". Aquí puede elegir la vista que deba aparecer siempre. La vista predeterminada es la última vista que se utilizó en la aplicación para visualizar los datos de referencia. Este cuadro de diálogo se puede desactivar para próximas sesiones.

Entonces se arranca la aplicación Mostrar datos de referencia y aparece la lista de referencias cruzadas del programa de usuario seleccionado (ajuste estándar al visualizar los datos de referencia por primera vez). Si los datos de referencia están incompletos aparecerá un cuadro de diálogo desde donde podrá actualizarlos.

Visualización directa desde el bloque compilado

Los datos de referencia de un bloque compilado se pueden visualizar directamente con el editor, obteniendo así una panorámica actual del programa de usuario.

14.2.2 Visualizar listas en ventanas de trabajo adicionales

Con el comando de menú **Ventana > Nueva ventana** se pueden abrir ventanas adicionales y visualizar allí otras listas (p.ej. lista de los símbolos no utilizados) de los datos de referencia.

Con el comando de menú **Datos de referencia > Abrir** se abre una ventana de trabajo que contiene los datos de referencia aún no visualizados.

Mediante el menú **Ver** o con el correspondiente botón de la barra de herramientas se puede cambiar a una representación diferente de los datos de referencia:

Representación de los datos de referencia	Comando de menú para visualizar dicha representación
Operandos sin símbolo	Ver > Operandos sin símbolo
Símbolos no utilizados	Ver > Símbolos no utilizados
Ocupación	Ver > Ocupación
Estructura del programa.	Ver > Estructura del programa
Referencias cruzadas	Ver > Referencias cruzadas

14.2.3 Crear y visualizar datos de referencia

Crear datos de referencia:

1. En el Administrador SIMATIC, seleccione la carpeta de bloques para la que se deben generar los datos de referencia.
2. En el Administrador SIMATIC, elija el comando **Herramientas > Datos de referencia > Generar**.

Antes de generar los datos de referencia se comprueba si existen ya algunos y si éstos son actuales.

- Los datos de referencia se generarán si no existen ya.
- Si no existen datos de referencia actuales, podrá decidir en un cuadro de diálogo si dichos datos se deben actualizar, o bien generar nuevamente.

Mostrar datos de referencia:

Con el comando **Herramientas > Datos de referencia > Mostrar** se pueden visualizar los datos de referencia.

Antes de visualizar los datos de referencia se comprueba si existen ya y si son actuales.

- Los datos de referencia se crearán si no existen ya.
- Si los datos de referencia están incompletos se visualizará un cuadro de diálogo donde se indicará que son incoherentes. Podrá decidir entonces si se deben actualizar y en qué medida desea hacerlo. A tal efecto puede elegir una de las siguientes opciones:

Elección	Significado
sólo para los bloques modificados	Se actualizan los datos de referencia de los bloques modificados y de los bloques nuevos, eliminándose las informaciones sobre los bloques que se hayan borrado.
para todos los bloques	Los datos de referencia se crean nuevamente para todos los bloques.
no actualizar	Los datos de referencia no se actualizan.

Para poder actualizar los datos de referencia se compilan de nuevo los bloques utilizando un compilador adecuado para cada bloque. Con el comando **Ver > Actualizar** se pueden actualizar en la ventana activa los datos de referencia ya visualizados.

14.2.4 Posicionamiento rápido en los puntos de aplicación del programa

Al programar utilice los datos de referencia para posicionarse en los puntos de aplicación de un operando. Para ello tiene que disponer de datos de referencia actuales. La llamada de la aplicación para mostrar los datos de referencia no es necesaria.

Procedimiento básico

1. Elija en el Administrador SIMATIC el comando de menú **Herramientas > Datos de referencia > Generar** para generar los datos de referencia actuales. Este paso solamente es necesario si no dispone de datos de referencia o si no son actuales.
2. Seleccione el operando en un bloque abierto.
3. Elija el comando de menú **Edición > Ir a > Punto de aplicación**. Aparece un cuadro de diálogo que contiene una lista con los puntos de aplicación del operando en el programa.
4. Elija la opción "Acceso solapado a las áreas de memoria", si además desea hacerse mostrar los puntos de aplicación de los operando cuyas direcciones o áreas de memoria se solapen con la del operando llamado. La tabla contendrá asimismo la columna "Operando".
5. Seleccione un punto de aplicación de la lista y haga clic en el botón de comando "Ir a".

Si al llamar el cuadro de diálogo los datos de referencia no son actuales, aparecerá un mensaje indicándoselo. En este caso puede hacer actualizar los datos de referencia.

Lista de los puntos de aplicación

La lista de los puntos de aplicación en el cuadro de diálogo contiene los siguientes datos:

- Bloque en el que se utiliza el operando
- Símbolo del bloque, si es que existe
- Detalles, es decir, informaciones sobre el punto de aplicación y la operación en lo referente al lenguaje del bloque/de la fuente (SCL)
- Informaciones sobre el bloque y lenguaje
- Tipo de acceso al operando: lectura (R), escritura (W), lectura y escritura (RW), no determinable (?).
- Lenguaje del bloque

Si lo desea puede filtrar la visualización de los puntos de aplicación, y p. ej. visualizar así sólo los accesos de escritura de un operando. Para más información sobre las opciones de entrada y visualización consulte la Ayuda en pantalla de este cuadro de diálogo.

Nota

Los datos de referencia solamente están disponibles online. Esta función opera siempre con las referencias cruzadas de los bloques offline, incluso si llama la función en un bloque online.

14.2.5 Ejemplo para trabajar con puntos de aplicación

Supongamos que desea averiguar en qué puntos del programa se activa la salida A1.0 (directo/indirecto). Como ejemplo se utiliza el siguiente código AWL en el bloque OB1:

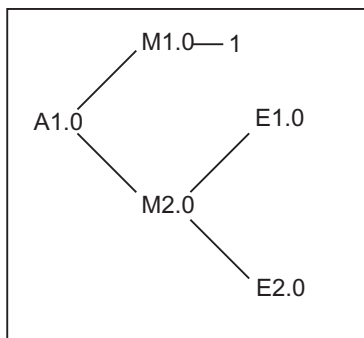
```
Segmento1: .....  
U A 1.0 // en este ejemplo  
= A 1.1 // no relevante
```

```
Segmento 2:  
  
U M1.0  
U M2.0  
  
= A 1.0 // Asignación
```

```
Segmento3:  
  
//sólo línea de comentario  
  
SET  
  
= M1.0 // Asignación
```

```
Segmento 4:  
  
U E 1.0  
U E 2.0  
  
= M2.0 // Asignación
```

Para A1.0 resulta el árbol de asignaciones siguiente:



Proceda ahora de la siguiente manera:

1. Posiciónese en el editor KOP/AWL/FUP OB 1 en A1.0 (Seg 1, Ins 1).
2. Con Edición > Ir a > Punto de aplicación o con la tecla derecha del ratón "Ir a punto de aplicación".
En el cuadro de diálogo se visualizan todas las asignaciones a A1.0:
OB1 Cycle Execution Seg 2 Ins 3 /= W AWL
OB1 Cycle Execution Seg 1 Ins 1 /U R AWL
3. Saltar mediante "Ir a" del cuadro de diálogo de "Seg 2 Ins 3" al editor:
Segmento 2:
U M1.0
U M2.0
= A 1.0
4. Ahora hay que controlar tanto las asignaciones a M1.0 como a M2.0. Posiciónese entonces en el editor KOP/AWL/FUP primero en M1.0.
5. Con Edición > Ir a > Punto de aplicación" o con la tecla derecha del ratón abra el cuadro de diálogo "Ir a punto de aplicación". Aquí se visualizan entre otras cosas todas las asignaciones a M1.0 :
OB1 Cycle Execution Seg 3 Ins 2 /= W AWL
OB1 Cycle Execution Seg 2 Ins 1 /U R AWL
6. Salte con el comando "Ir a" a "Seg 3 Ins 2" en el editor KOP/AWL/FUP.
7. En el segmento 3 del editor KOP/AWL/FUP se detecta que la ocupación de M1.0 no es interesante (por estar siempre a TRUE) y que en su lugar habría que comprobar la ocupación de M2.0.

En las versiones de STEP 7 anteriores a la versión V5 había que volver a recorrer toda la cadena de asignaciones desde el principio. Los botones de comando ">>" y "<<" simplifican el procedimiento siguiente:

8. O bien pase a primer plano el cuadro de diálogo que aún está abierto "Ir a punto de aplicación" o bien llame este cuadro de diálogo desde la posición actual del editor KOP/AWL/FUP.
9. Activar el botón de comando "<<" una o dos veces hasta que se visualicen todos los puntos de aplicación de A 1.0. El último punto "Seg 2 Ins 3" está seleccionado.
10. Salte con el comando "Ir a" (como bajo el punto 3) desde el diálogo de los puntos de aplicación "Seg 2 Ins 3" al editor:
Segmento 2:
U M1.0
U M2.0
= A 1.0
11. En el punto 4 y siguientes se ha controlado la asignación a M1.0. Ahora hay que controlar todas las asignaciones (directas/indirectas) a M2.0. Es decir, posiciónese en el editor sobre M2.0 y active la función "Ir a punto de aplicación": Se visualizarán entre otras cosas todas las asignaciones a M2.0:
OB1 Cycle Execution Seg 4 Ins 3 /= W AWL
OB1 Cycle Execution Seg 2 Ins 2 /U R AWL

12. Salte con el comando "Ir a" al "Seg 4 Ins 3" del editor KOP/AWL/FUP:

Segmento 4:

U E 1.0

U E 2.0

= M2.0

13. Ahora tiene que comprobar las asignaciones a E1.0 y E2.0. Esto ya no se explica en el presente ejemplo, pues el procedimiento a seguir es el mismo (punto 4 y siguientes)

Saltando del editor KOP/AWL/FUP al cuadro de diálogo "Puntos de aplicación" y vice versa puede determinar y analizar los puntos relevantes de su programa.

15 Comprobar coherencia del bloque y fecha y hora como propiedad del bloque

15.1 Comprobar coherencia de bloques

Introducción

Si durante o después de crear el programa hay que adaptar o completar los interfaces o el código de objetos aislados, es posible que se originen conflictos de indicaciones de fecha y hora. Los conflictos de indicaciones de fecha y hora pueden causar a su vez incoherencias de bloques entre los objetos llamantes y llamados, o en los bloques de referencia y generar de esta forma un gran esfuerzo de corrección.

La función "Comprobar coherencia del bloque" le ahorrará gran parte de este trabajo de corrección. La función "Comprobar coherencia del bloque" se ocupa automáticamente de gran parte de todos los conflictos de indicaciones de fecha y hora y de las incoherencias de bloques. En aquellos objetos, en los que no se pudieron eliminar automáticamente las incoherencias de bloques, la función le guiará a las posiciones que se desean modificar en el editor correspondiente. Allí podrá realizar las modificaciones necesarias. Paso a paso se eliminarán todas las incoherencias y se compilarán los bloques.

Requisitos

La comprobación de la coherencia de los bloques sólo es posible en los proyectos creados con la versión de STEP 7 V5.0, Service Pack 3, o posteriores. En los proyectos más antiguos primero habrá que compilar todo antes de poder iniciar la comprobación de la coherencia de los bloques (comando de menú **Programa > Compilar todo**).

En el caso de los objetos creados con un paquete opcional, éste deberá estar instalado para poder realizar la comprobación de coherencia.

Iniciar la comprobación de la coherencia de los bloques

Al iniciar la comprobación de la coherencia de los bloques se comprueba la indicación de fecha y hora de los interfaces de los objetos, y los bloques en los que es posible que se produzcan incoherencias se marcan en la ventana del árbol (árbol de llamadas: referencias / árbol de dependencias)

1. En la ventana del proyecto del administrador SIMATIC, marque la carpeta de bloques deseada e inicie la comprobación de la coherencia con el comando de menú Edición > Comprobar coherencia de bloques.
2. En "Comprobar coherencia de bloques", seleccione el comando de menú **Programa > Compilar**
STEP 7 detecta automáticamente el lenguaje en que se crearon los objetos en cuestión y llama al editor correspondiente. Los conflictos de indicaciones de fecha y hora y las incoherencias de bloques se eliminan en la medida de lo posible y los bloques se compilan. Si no se ha podido eliminar automáticamente el conflicto de indicación de fecha y hora o la incoherencia en un bloque, aparece un aviso de error en la ventana de resultados (para el procedimiento a seguir, ver el paso 3). Este proceso se ejecuta automáticamente en todos los objetos de la ventana del árbol.
3. Si en el proceso de compilación no se han podido eliminar automáticamente todas las incoherencias de los bloques, los objetos correspondientes se marcarán en la ventana de resultados como aviso de error. Posicione el puntero del ratón sobre la entrada de error correspondiente y seleccione con el botón derecho del ratón el comando del menú contextual **Mostrar errores**. El editor correspondiente se abre y se salta a las posiciones que hay que modificar. Elimine todas las incoherencias del bloque y cierre y guarde el objeto. Repita el proceso para todos los objetos marcados como errores.
4. Inicie de nuevo los pasos 2 y 3. Repita este proceso hasta que en la ventana de avisos ya no queden errores.

15.2 Fecha y hora como propiedad del bloque y conflictos de fecha y hora

Los bloques contienen dos indicaciones de fecha y hora: una de la lógica o código y otra del interface. Dichas indicaciones de fecha y hora se visualizan en el diálogo de las propiedades del bloque. Mediante las indicaciones de fecha y hora se vigila que los programas STEP 7 sean coherentes.

STEP 7 indica un conflicto de fecha y hora cuando al comparar las indicaciones de fecha y hora, se detecta una violación de las reglas.

Si no se respetan las reglas, pueden ocurrir los siguientes errores:

- El bloque llamado es más reciente que el bloque que lo llama (CALL).
- El bloque direccionado es más reciente que el bloque que lo utiliza.
- Ejemplos del último caso (segundo punto) son los siguientes:
 - Un UDT es más reciente que el bloque que lo utiliza, p. ej. un DB u otro UDT o un FC, FB, OB que utiliza el UDT en la tabla de declaración de variables.
 - Un FB es más reciente que el DB de instancia correspondiente.
 - En el FB1 hay un FB2 definido como multiinstancia y el FB2 es más reciente que el FB1.

Nota

Aunque la relación de las indicaciones de fecha y hora de los interfaces sea correcta pueden surgir conflictos:

- La definición del interface del bloque direccionado no se corresponde con el interface utilizado en su punto de aplicación.

Estos problemas se denominan conflictos de interfaces y pueden aparecer al copiar bloques de distintos programas o cuando al compilar una fuente ASCII sólo se genere una parte de los bloques del programa.

15.3 Indicación de fecha y hora de bloques lógicos

Indicación de fecha y hora del código del programa

Aquí se introduce la fecha y hora de creación del bloque. Dicha indicación se actualiza

- al cambiar la lógica del programa
- al cambiar la descripción del interface
- al cambiar el comentario
- al crearse por primera vez y al compilarse una fuente ASCII
- al cambiar las propiedades del bloque (cuadro de diálogo: Propiedades)

Indicación de fecha y hora del interface

Esta indicación de fecha y hora se actualiza

- al cambiar la descripción del interface (cambio de tipos de datos o de valores iniciales, nuevos parámetros)
- al crearse por primera vez y al compilarse una fuente ASCII, en caso de que cambie la estructura del interface.

La indicación de fecha y hora no se actualiza

- al cambiar los símbolos
- al cambiar los comentarios en la declaración de variables
- al efectuarse cambios en el área TEMP

Reglas para las llamadas de bloques

- La indicación de fecha y hora del interface del bloque llamado debe ser anterior a la indicación de fecha y hora de la lógica del bloque invocante.
- Cambie el interface de un bloque sólo cuando no esté abierto ningún otro bloque que lo llame. Si los bloques invocantes se guardan posteriormente al bloque modificado, no reconocerán la incoherencia en la indicación de fecha y hora.

Procedimiento al presentarse conflictos de fecha y hora

Al abrir el bloque invocante se visualiza un conflicto de fecha y hora. Después de modificarse un interface de una FC o de un FB, todas las llamadas a dichos bloques se representarán de forma expandida en los bloques invocantes.

Si se cambia el interface de un bloque, será preciso adaptar todos los bloques que lo llamen.

Después de cambiar un interface FB, se deberán actualizar todas las definiciones existentes de multiinstancias, así como los bloques de datos de instancia.

15.4 Indicación de fecha y hora de bloques de datos globales

Indicación de fecha y hora de la lógica

Esta indicación de fecha y hora se actualiza

- al crear por primera vez,
- al compilar una fuente ASCII,
- al efectuarse cambios en la vista "Declaración" o en la vista "Datos" del bloque.

Indicación de fecha y hora del interface

Esta indicación se actualiza

- al efectuarse cambios de la descripción del interface en la vista "Declaración" (cambio de tipos de datos o de valores iniciales, nuevos parámetros)

15.5 Indicación de fecha y hora de bloques de datos de instancia

Los bloques de datos de instancia permiten almacenar los parámetros formales y los datos estáticos de los bloques de función.

Indicación de fecha y hora de la lógica

Aquí se introduce la fecha y hora de creación del bloque de datos de instancia. Dicha indicación se actualiza al introducirse valores actuales en la vista "Datos" del bloque de datos de instancia. El usuario no puede cambiar la estructura de un bloque de datos de instancia, puesto que dicha estructura se deriva del bloque de función (FB) asignado o del bloque de función de sistema (SFB).

Indicación de fecha y hora del interface

Al crearse un bloque de datos de instancia se introduce la indicación de fecha y hora del FB o SFB asignado.

Reglas para abrir bloques sin que se presenten conflictos

Deben concordar las indicaciones de fecha y hora del interface del FB o SFB y del bloque de datos de instancia asignado.

Procedimiento al presentarse conflictos de fecha y hora

Al cambiarse el interface de un FB se actualiza la indicación de fecha y hora del interface del FB. Al abrirse un bloque de datos de instancia asignado se visualizará un conflicto de fecha y hora, puesto que la indicación de fecha y hora del bloque de datos de instancia ya no concuerda con la del FB. En el área de declaración del DB se representará el interface con los símbolos generados por el compilador (pseudosímbolos). En este caso, el bloque de datos de instancia sólo se podrá visualizar.

Para solucionar este conflicto es preciso volver a crear un DB de instancia asignado al FB modificado.

15.6 Indicación de fecha y hora de UDTs y DBs derivados de UDTs

Por ejemplo, un tipo de datos definido por el usuario (UDT) se puede utilizar para generar varios bloques de datos de la misma estructura.

Indicación de fecha y hora de la lógica

Esta indicación se actualiza cada vez que se realice un cambio.

Indicación de fecha y hora del interface

Esta indicación se actualiza al cambiar la descripción del interface (cambio de tipos de datos o de valores iniciales, nuevos parámetros).

La indicación de fecha y hora de un UDT se actualiza también al compilarse la fuente ASCII.

Reglas para abrir bloques sin que se presenten conflictos

- La indicación de fecha y hora del interface del tipo de datos de usuario (UDT) debe ser anterior a la de los bloques lógicos donde se utiliza dicho UDT.
- La indicación de fecha y hora del interface del tipo de datos de usuario (UDT) debe ser idéntica a la de un DB derivado de un UDT.
- La indicación de fecha y hora del interface del tipo de datos de usuario (UDT) debe ser anterior a la de un UDT subordinado.

Procedimiento al presentarse conflictos de fecha y hora

Si se modifica una definición de un UDT utilizada en un DB, una FC, un FB o en una definición de UDT diferente, STEP 7 visualizará un conflicto de fecha y hora al abrirse dicho bloque.

El componente UDT se representa en forma de estructura. Todos los nombres de variables se sobrescriben con valores predeterminados por el sistema.

15.7 Corregir los interfaces en una FC, un FB o un UDT

Cuando sea necesario corregir el interface de un FB, una FC o un UDT, proceda de la siguiente forma para evitar conflictos de fecha y hora:

1. Genere una fuente AWL con el bloque a modificar y con todos los bloques referenciados directa o indirectamente.
2. Guarde los cambios en la fuente creada.
3. Compile nuevamente en los bloques la fuente adaptada.

Ahora puede guardar o cambiar el cambio del interface.

15.8 Evitar errores al llamar bloques

STEP 7 sobrescribe los datos en el registro DB.

STEP 7 modifica los registros de las CPUs S7-300/S7-400 en diferentes operaciones. Por ejemplo, el contenido de los registros DB y DI se intercambia al efectuar la llamada de un FB. De esta manera es posible abrir el DB de instancia del FB llamado sin perder la dirección del DB de instancia anterior.

Si se trabaja con direccionamiento absoluto, pueden producirse errores al acceder a los datos memorizados en los registros: en algunos casos se podrían sobrescribir las direcciones contenidas en los registros AR1 (registro de direcciones 1) y en el registro DB, lo que podría provocar el que se leyera o escribiera en direcciones erróneas.



Peligro

Peligro de daños materiales o personales al utilizar:

1. CALL FC, CALL FB, CALL multiinstancia
2. Accesos a DBs con la dirección completa (p.ej., DB20.DBW10)
3. Acceso a variables de un tipo de datos compuestos

Podría cambiar el contenido del registro DB (DB y DI), del registro de direcciones (AR1, AR2) y de los acumuladores (ACU1, AC2).

Al efectuar las llamadas CALL FB/FC tampoco se puede utilizar el resultado lógico (RLO) como parámetro adicional (implícito).

Si se utilizan las posibilidades de programación anteriores, el usuario deberá tomar las precauciones necesarias para restablecer los contenidos y así evitar fallos.

Memorizar datos correctos

Para la memorización puede resultar problemático utilizar el contenido del registro DB si se accede a los datos utilizando el formato abreviado de las direcciones absolutas. Por ejemplo, si se presupone que el DB20 está abierto (y su número está memorizado en el registro del DB), entonces se indicará DBX0.2 para acceder a los datos memorizados en el bit 2 del byte 0 del DB, cuya dirección esté depositada en el registro DB (o sea, DB20). Si el registro DB contiene, no obstante, otro número de DB, entonces se accederá erróneamente a otros datos.

Para evitar errores que se producen al acceder a datos del registro DB dirija los datos:

- utilizando direccionamiento simbólico, o bien
- utilizando la dirección absoluta completa (p. ej., *DB20.DBX0.2*)

Con cualquiera de estos métodos de direccionamiento, STEP 7 abrirá automáticamente el DB correcto. Si el registro AR1 se utiliza para direccionamiento indirecto, se ha de cargar siempre la dirección directa en AR1.

Situaciones en las que se modifican registros

La manipulación de los registros de direcciones para fines de direccionamiento indirecto sólo es importante en AWL. En efecto, los otros lenguajes de programación no ofrecen prestaciones de acceso indirecto a los registros de direcciones.

En todos los lenguajes de programación es necesario que el compilador adapte el registro DB a fin de garantizar una transferencia correcta de parámetros al efectuar cualquier llamada a un bloque.

El contenido del registro de direcciones AR1 y del registro DB del bloque invocante se sobrescribe cuando se presentan las situaciones siguientes:

Situación	Explicación
En parámetros actuales procedentes de un DB	<ul style="list-style-type: none"> Una vez que se ha asignado a un bloque un parámetro actual almacenado en un DB (p. ej., DB20.DBX0.2), STEP 7 abre dicho DB (DB20) y adapta el contenido del registro DB. Tras la llamada al bloque, el programa trabaja con el DB adaptado.
Al llamar a bloques si se utilizan tipos de datos compuestos	<ul style="list-style-type: none"> Tras la llamada a un bloque desde una FC que transfiere a dicho bloque invocante un componente de un parámetro formal con formato de datos compuesto (String, Array, Struct o UDT) se modifica el contenido de AR1 y del registro DB del bloque invocante. Esto mismo es aplicable a la llamada desde un FB siempre que el parámetro esté situado dentro del área VAR_IN_OUT del bloque invocante.
Al acceder a componentes de tipo de datos compuesto	<ul style="list-style-type: none"> Cuando un FB accede a un componente de un parámetro formal que tiene como formato un tipo de datos compuesto en el área VAR_IN_OUT (String, Array, Struct o UDT), STEP7 utiliza el registro de direcciones AR1 y el registro DB. Como consecuencia se modifican los contenidos de ambos registros. Cuando un FC accede a un componente de un parámetro formal que tiene como formato un tipo de datos compuesto (String, Array, Struct o UDT), STEP7 utiliza el registro de direcciones AR1 y el registro DB. Como consecuencia se modifican los contenidos de ambos registros.

Nota

- Cuando se llama un FB desde un bloque de la versión 1 no se transfiere correctamente el parámetro actual del primer parámetro booleano IN o IN_OUT cuando la instrucción previa a la llamada no limita el RLO. En este caso se combina con el RLO pendiente.
 - Al llamar a un FB (normal o multiinstancia) se sobrescribe el registro de direcciones AR2.
 - Si dentro de un FB se modifica el registro de direcciones AR2, por ejemplo mediante las operaciones UC, CC o CALL (llamar FC/SFC sin parámetros), no se garantiza la correcta ejecución del FB.
 - Si a un parámetro ANY no se le transfiere la dirección absoluta completa del DB, el puntero ANY no contendrá el número del DB abierto sino un 0.
-

16 Configurar avisos

16.1 El sistema de avisos

Los avisos permiten detectar, localizar y eliminar rápidamente los fallos que se presenten en los sistemas de automatización durante el proceso. Así se pueden reducir considerablemente los tiempos de parada de una instalación.

Antes de poder visualizar los avisos es preciso configurarlos.

STEP 7 permite crear, editar y compilar avisos de eventos con sus textos y atributos en visualizadores.

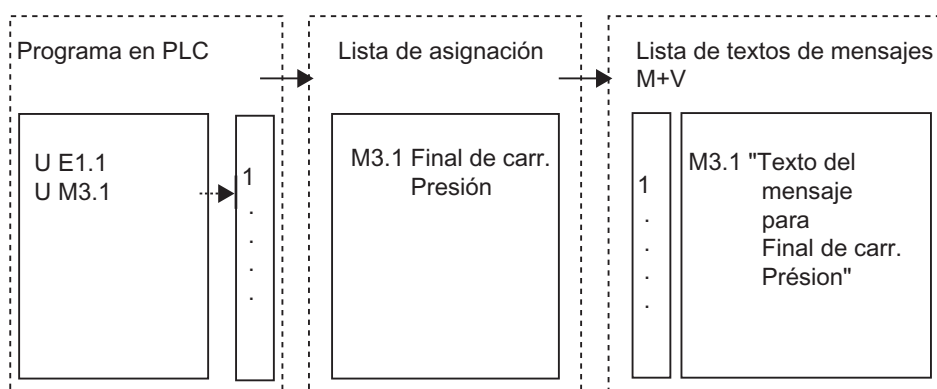
16.1.1 ¿Qué procedimientos existen para crear avisos?

Para crear avisos se dispone de distintos procedimientos.

Procedimiento de notificación por bits

El procedimiento de notificación por bits requiere 3 pasos por parte del programador:

- Crear el programa de usuario en la PG y activar el bit deseado.
- Crear con un editor de texto cualquiera una lista de asignación, en la que se asigna un texto a cada bit de notificación (p.ej., M 3.1 = Final de carrera Presión).
- Crear la lista de los textos de los avisos en el sistema de manejo a partir de la lista de asignación.

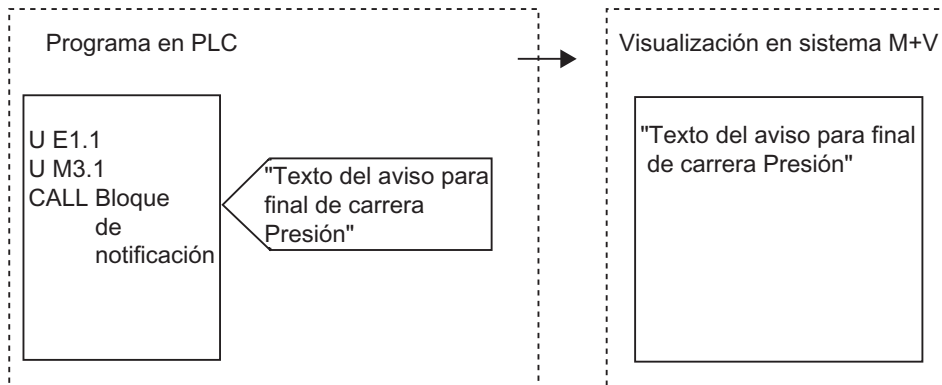


El sistema de manejo y visualización consulta cíclicamente el sistema de automatización para saber si se ha modificado o no el bit de notificación. Cuando el PLC notifica una modificación, se visualiza el aviso correspondiente. Entonces, se asigna al aviso la indicación de hora y fecha del sistema de manejo y visualización.

Procedimiento por números de avisos

El procedimiento por números de avisos sólo requiere un paso por parte del programador:

- Crear el programa de usuario en la PG, activar el bit deseado y asignar directamente al programar el texto deseado al bit.



El sistema de automatización no se consulta cíclicamente. Tan pronto como el PLC notifique una modificación, se notificará el número del aviso correspondiente al sistema de manejo y visualización y se visualizará el texto correspondiente. La indicación de hora y fecha del aviso es la misma que la del PLC, por lo que se puede determinar de forma mucho más exacta que en el caso del procedimiento de notificación por bits.

16.1.2 Selección del procedimiento de notificación

Vista general

La tabla siguiente muestra las propiedades y requisitos de los distintos procedimientos de notificación.

Procedimiento por números de avisos	Procedimiento de notificación por bits
<ul style="list-style-type: none"> Los avisos se gestionan en una base de datos común para la PG y el sistema de manejo La carga del bus es mínima (el PLC está activo) Los avisos reciben la indicación de fecha y hora del sistema de automatización 	<ul style="list-style-type: none"> No hay ninguna base de datos común para la PG y el sistema de manejo La carga del bus es alta (el sistema de manejo sondea) Los avisos reciben la indicación de fecha y hora del sistema de manejo

El procedimiento por números de avisos distingue los tres tipos siguientes de avisos:

Avisos de bloques	Avisos de símbolos	Avisos de diagnóstico personalizados
<ul style="list-style-type: none"> Síncronos con el programa Visualización mediante ProTool (sólo ALARM_S) y WinCC Posible en S7-300/S7-400 Programación con bloques de notificación: <ul style="list-style-type: none"> ALARM ALARM_8 ALARM_8P NOTIFY NOTIFY_8P ALARM_S(Q) AR_SEND ALARM_D(Q) Transferencia al sistema de manejo <ul style="list-style-type: none"> Para WinCC a través de la configuración de enlaces PLC-OS Para ProTool a través de las funciones ProTool 	<ul style="list-style-type: none"> Asíncrono con el programa Visualización mediante WinCC Posible sólo en S7-400 Configuración mediante la tabla de símbolos Transferencia al PLC a través de los bloques de datos del sistema (SDBs) Transferencia al sistema de manejo vía engineering PLC-OS 	<ul style="list-style-type: none"> Asíncrono con el programa Visualización en el búfer de diagnóstico de la PG Posible en S7-300/S7-400 Programación con bloques de notificación (función del sistema) <ul style="list-style-type: none"> WR_USMSG Sin transferencia al sistema de manejo

STEP 7 sólo soporta el procedimiento más cómodo por números de avisos, el cual se describe detalladamente a continuación. El proceso de notificación por bits se configura en los equipos HMI y se describe en ellos.

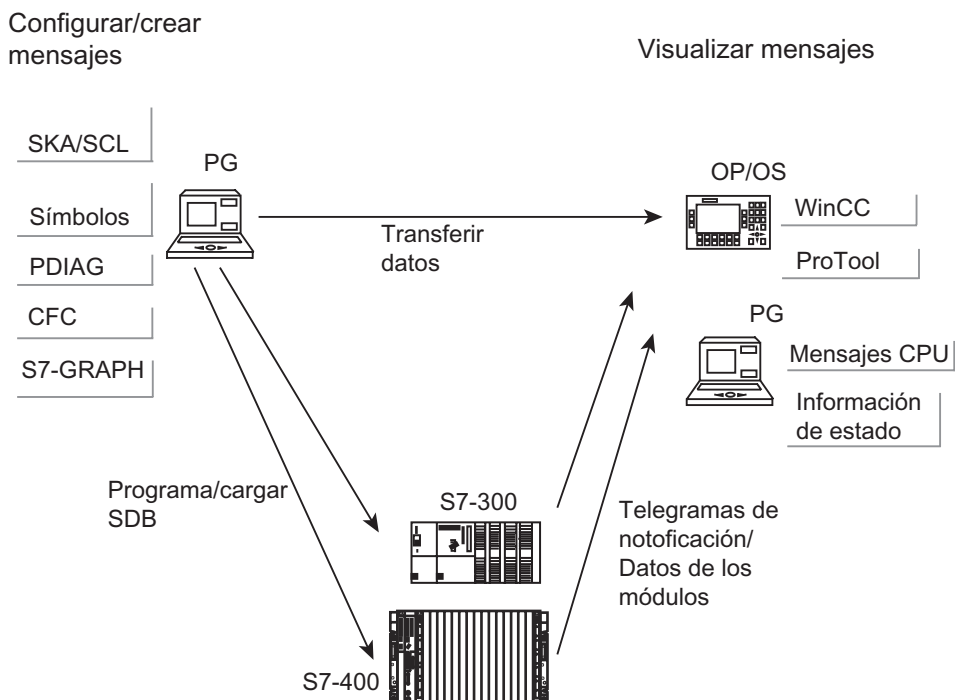
Ejemplos del procedimiento por números de avisos

Procedimiento de notificación	Campo de aplicación
Avisos de bloque	Para notificar eventos síncronos con el programa; p.ej., se debe indicar que un regulador ha alcanzado un valor límite
Avisos de símbolos	Para notificar eventos que no dependen del programa; p.ej., se debe vigilar una posición del interruptor
Avisos personalizados	Para notificar eventos de diagnóstico en el búfer de diagnóstico, en cada llamada de la SFC

16.1.3 Componentes SIMATIC

Vista general

La figura siguiente muestra, en una vista general, los componentes SIMATIC que participan en la configuración y visualización de avisos.



16.1.4 Partes de un aviso

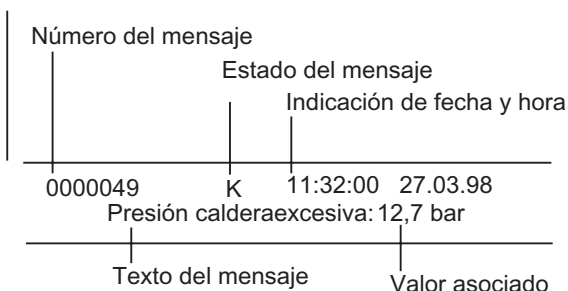
El modo de visualización de un aviso depende del procedimiento de notificación, del bloque de notificación utilizado y del visualizador.

En la tabla siguiente se listan las partes que pueden integrar un aviso:

Parte	Descripción
Indicación de fecha y hora	Se genera cuando aparece el evento de notificación en el sistema de automatización
Estado del aviso	Estados posibles: entrante, saliente, saliente sin acuse y saliente con acuse
Valor asociado	A algunos avisos se les puede asociar un valor del proceso, el cual puede ser evaluado por el bloque de notificación
Imagen	En caso de una caída del sistema, los avisos que habían aparecido pueden visualizarse posteriormente en el OS.
Número del aviso	Número asignado por el sistema, unívoco para todo el proyecto o para toda la CPU, que identifica un aviso.
Textos del aviso	Configurados por el usuario

Ejemplo

El ejemplo siguiente muestra un aviso de fallo en un panel de operador.



16.1.5 ¿Qué bloques de notificación existen?

Es posible elegir entre los siguientes bloques de notificación en los que ya está programada la correspondiente función:

- SFB 33: "ALARM"
- SFB 34: "ALARM_8"
- SFB 35 "ALARM_8P"
- SFB 36 "NOTIFY"
- SFC 18: "ALARM_S" y SFC 17: "ALARM_SQ"
- SFB 37: "AR_SEND" (para enviar archivadores; no es posible configurar textos ni atributos de avisos)
- SFB 31: "NOTIFY_8P"
- SFC 107: "ALARM_DQ"
- SFC 108: "ALARM_D".

Para obtener información más detallada consulte la ayuda de referencia sobre los Bloques .

¿Cuándo utilizar qué bloque de notificación?

La tabla siguiente facilita la elección del bloque de notificación a utilizar. Dicha elección depende de:

- el número de canales disponibles en el bloque y, por lo tanto, del número de señales que se vigilan en cada llamada del bloque.
- la posibilidad de confirmar avisos.
- la posibilidad de adjuntar valores asociados.
- los visualizadores que se desean utilizar.
- el alcance de la CPU.

Bloque de notificación	Canales	Acuse	Valores asociados	Visualización WinCC	Visualización PROTOOL	Visualización avisos de CPU/S7 Status	PLC	Particularidades
ALARM_SFB 33	1	posible	máx. 10	sí	no	no	S7-400	Envía un aviso por flanco entrante o saliente
ALARM_8_SFB 34	8	posible	no	sí	no	no	S7-400	Envía un aviso por flanco entrante o saliente de una o varias señales
ALARM_8P_SFB 35	8	posible	máx. 10	sí	no	no	S7-400	Como en ALARM_8
NOTIFY_SFB 36	1	no	máx. 10	sí	no	no	S7-400	Como en ALARM
NOTIFY_8P_SFB 31	8	no	máx. 10	sí	no	no	S7-400	Como en NOTIFY
AR_SEND_SFB 37	1	-	-	sí	no	no	S7-400	sirve para enviar un archivador. No es posible configurar textos ni atributos de avisos
ALARM_SQ_SFC 17	1	posible	1	sí	sí*	sí	S7-300/400	No es cada cambio de flanco el que genera un aviso, sino cda llamada de SFC
ALARM_S_SFC 18	1	no	1	sí	sí*	sí	S7-300/400	Como en ALARM_SQ
ALARM_DQ_SFC 107	1	posible	1	sí	sí	sí	S7-300/400	Como en ALARM_SQ
ALARM_D_SFC 108	1	no	1	sí	sí	sí	S7-300/400	Como en ALARM_SQ
* depende del tipo de OP								

16.1.6 Parámetros formales, atributos del sistema y bloques de notificación

Parámetro formal como entrada del número de aviso

Para cada aviso o grupo de avisos, en su programa necesita un parámetro formal que pueda utilizar como parámetros de entrada (IN) en la vista de variables de su programa. El parámetro formal se utiliza como entrada de número de aviso y constituye la base de un aviso.

Asignar atributos del sistema a los parámetros formales

Para poder acceder a la configuración de avisos ha tenido que haber asignado atributos del sistema a los parámetros formales:

1. Ha asignado los siguientes atributos del sistema para parámetros: "S7_server" y "S7_a_type"
2. Ha asignado a los atributos del sistema valores adecuados a los bloques de avisos que usted ha llamado en su código de programa. El valor para "S7_server" es siempre "alarm_archiv", y el valor para "S7_a_type" corresponde al bloque de avisos llamado.

Atributos del sistema y bloques de avisos correspondientes

Los objetos del servidor de avisos que se visualizan no son los propios bloques de avisos, sino los valores correspondientes a los atributos del sistema "S7_a_type". Estos valores tienen los mismos identificadores que los bloques de avisos existentes, que pueden ser SFB o SFC (excepción: "alarm_s").

S7_a_type	Bloque de mens...	Identificador	Propiedades
alarm_8	ALARM_8	SFB 34	8 canales, posibilidad de acuse, sin valores asociados
alarm_8p	ALARM_8P	SFB 35	8 canales, posibilidad de acuse, hasta 10 valores asociados por canal
notify	NOTIFY	SFB 36	1 canal, sin acuse, hasta 10 valores asociados
alarm	ALARM	SFB 33	1 canal, posibilidad de acuse, hasta 10 valores asociados
alarm_s	ALARM_S	SFC 18	1 canal, sin acuse, hasta 1 valor asociado
alarm_s	ALARM_SQ	SFC 17	1 canal, posibilidad de acuse, hasta 1 valor asociado
ar_send	AR_SEND	SFB 37	sirve para enviar un archivador
notify_8p	NOTIFY_8P	SFB 31	8 canales, sin acuse, hasta 10 valores asociados
alarm_s	ALARM_DQ	SFC 107	1 canal, posibilidad de acuse, hasta 1 valor asociado
alarm_s	ALARM_D	SFC 108	1 canal, sin posibilidad de acuse, hasta 1 valor asociado

Para obtener información más detallada consulte la ayuda de referencia acerca de los atributos del sistema

Los atributos del sistema se asignan automáticamente cuando los bloques de notificación utilizados en el programa de usuario son SFB o FB con los atributos de sistema correspondientes y se llaman en calidad de multiinstancias.

16.1.7 Plantilla y avisos

Con la configuración de avisos puede crear una plantilla o avisos en distintas fases. El procedimiento depende del bloque apto para notificación a través del cual se accede a la configuración de avisos.

El bloque apto para notificación puede ser un FB o un DB de instancia

- Si es un FB, puede crear una plantilla para otros avisos. Todas las indicaciones que haga para la plantilla son aceptadas automáticamente por los avisos. Si asigna un DB de instancia al FB, después de generarse la plantilla se generarán automáticamente avisos para el DB de instancia y se les asignarán números de aviso.
- Si se trata de un DB de instancia puede modificar para cada instancia los avisos generados a partir de la plantilla.

La diferencia visible consiste en que en los avisos ya se asignan números de aviso, mientras que en una plantilla no.

Inhibir datos en la plantilla

Al configurar avisos se indican textos y atributos de avisos controlados por eventos. Además, se puede definir cómo se visualizarán los avisos en determinados visualizadores. Para simplificar la creación de los avisos se puede operar con plantillas.

- Al introducir los datos (atributos y textos) de la plantilla se puede determinar si deben inhibirse o no. Junto al cuadro de entrada de los atributos inhibidos aparece un símbolo con una llave o se inserta una marca de verificación en la columna "Inhibido". Los textos inhibidos reciben una marca de verificación en la columna "Inhibido".
- Los datos inhibidos en la plantilla no se podrán modificar en los avisos específicos de las instancias. Únicamente se visualizarán.
- No obstante, si necesita efectuar modificaciones, debe retornar a la plantilla, cancelar la inhibición y efectuar allí la modificación. Sin embargo, las modificaciones así efectuadas no serán válidas para las instancias que se hayan generado anteriormente.

Modificar los datos en la plantilla

Si los cambios de datos en la plantilla afectan también a las instancias o no depende de si al crear el proyecto ha elegido la asignación de números de aviso para toda la CPU o para todo el proyecto.

- Asignación de números de aviso para todo el proyecto: si hace alguna modificación en la plantilla a posteriori, que también debe aplicarse a las instancias, tendrá que introducir los cambios también en las instancias.

- Asignación de números de aviso para toda la CPU: si hace alguna modificación en la plantilla a posteriori, ésta se aplicará automáticamente a las instancias
Excepciones: ya ha cambiado los datos anteriormente en las instancias o ha inhibido o desinhibido datos en el tipo de aviso con posterioridad. Si copia un bloque de función y un DB de instancia en un proyecto con asignación de números de aviso para toda la CPU desde un proyecto con asignación de números de aviso para todo el proyecto, posteriormente tendrá que modificar también en la instancia los datos que ha modificado en la plantilla.

Atención

- Si copia instancias en otro programa sin copiar también la plantilla, puede que la plantilla no se muestre por completo. En ese caso también deberá copiar la plantilla en el programa nuevo.
 - Si en una instancia los textos y atributos aparecen representados de color verde, esto significa lo siguiente: estos textos y atributos siguen siendo tal y como cuando fueron configurados. Entre tanto no han sido modificados en la instancia.
-

16.1.8 Crear una fuente AWL desde bloques notificadores

Al crear una fuente AWL desde bloques notificadores, la información sobre el proyecto de la fuente también se guarda.

La información se guarda en un pseudocomentario que comienza con `*$ALARM_SERVER` y finaliza con `**`.

Atención

Si hace una referencia simbólica a un bloque, asegúrese de que antes de compilar la fuente no pueda modificarse la tabla de símbolos.

Si la fuente contiene varios bloques, los diferentes bloques de pseudocomentarios se agrupan en un solo bloque de comentarios. No se pueden borrar bloques individuales con atributos de aviso de la fuente AWL.

16.1.9 Asignar números a los avisos

Puede decidir si desea asignar los números de aviso para todo el proyecto o para toda la CPU. La asignación de números de aviso para toda la CPU presenta la ventaja de que le permite copiar un programa sin necesidad de cambiar o volver a compilar los números de aviso. Si ha optado por números de aviso unívocos para toda la CPU y quiere verlos en un visualizador HMI, necesitará para ello el paquete de software `"WinCC V6.0"` o `"ProTool V6.0"`. Si utiliza versiones inferiores a la 6.0 de estos paquetes de software, deberá optar por números de aviso unívocos para todo el proyecto (y no para toda la CPU).

16.1.10 Diferencias entre asignar números de aviso para todo el proyecto o para toda la CPU

La tabla siguiente muestra las diferencias entre asignar números de aviso para todo el proyecto y para toda la CPU:

Para todo el proyecto	Para toda la CPU
Algunos atributos y textos del aviso dependen del equipo HMI empleado y deben configurarse en función del visualizador.	Los atributos y textos asignados son independientes del equipo HMI empleado, es decir, no hay que insertar ningún visualizador ni configurar un aviso específico del visualizador para este equipo.
Al copiar programas debe realizarse una nueva traducción.	Los programas pueden copiarse dentro de un proyecto y más allá del mismo. De todas formas, al copiar bloques individuales es necesaria una nueva traducción.
Si se modifican datos (textos y atributos) posteriormente en el tipo de aviso, estos cambios deberán transferirse a las instancias.	Si se modifican datos (textos y atributos) posteriormente en el tipo de aviso, estos cambios se adoptan automáticamente en las instancias (excepto si estos datos se han cambiado anteriormente en la instancia).
Sólo puede introducir textos de una sola línea.	Puede introducir textos de varias líneas.

16.1.11 Procedimientos para cambiar la asignación de números de aviso de un proyecto

El tipo de asignación de números de aviso puede preseleccionarse en la ficha "Números de aviso" del Administrador SIMATIC (comando de menú **Herramientas > Preferencias**) para futuros proyectos y librerías. En esta ficha se especifica si los números de aviso deben asignarse unívocamente para toda la CPU o unívocamente para todo el proyecto. Otra opción es "Consultar siempre la configuración", para efectuar la asignación posteriormente.

Si en el momento de crear un proyecto o una librería estaba activa la configuración estándar "unívoco para toda la CPU" o "unívoco para todo el proyecto", el tipo de asignación de números de aviso no puede volver a modificarse en este proyecto o en esta librería.

Si se configuró una asignación de números de aviso "unívoca para todo el proyecto" y desea una asignación "unívoca para toda la CPU", proceda del modo siguiente:

1. Marque el proyecto o la librería en cuestión en el Administrador SIMATIC.
2. Seleccione el comando de menú **Archivo > Guardar como**.
3. En el cuadro de diálogo siguiente, active la casilla de verificación "Reorganizando" e introduzca un nombre nuevo.
4. Inicie el proceso "Guardar como" con el botón "Aceptar".
5. En uno de los cuadros de diálogo siguientes puede determinarse una asignación de números de aviso "unívoca para toda la CPU".

Con el comando de menú **Archivo > Borrar** se borra el proyecto o la librería original.

16.2 Configurar avisos para todo el proyecto

16.2.1 Asignación de números de aviso para todo el proyecto

Los avisos se identifican con un número unívoco para todo el proyecto. Para ello a cada programa S7 se le asigna un margen numérico dentro del margen numérico general disponible (1 - 2097151). Si aparece un conflicto al copiar un programa, es decir, si los números de aviso de este programa ya han sido asignados en el área de destino, deberá asignarse otro margen numérico al nuevo programa. En este caso, STEP 7 abre automáticamente el cuadro de diálogo en el que se puede indicar el nuevo margen numérico.

El margen numérico de un programa S7 también se puede definir o modificar eligiendo el comando de menú **Edición > Propiedades especiales del objeto > Números de aviso**. Sin embargo esto sólo es posible si aún no se han configurado avisos para este programa.

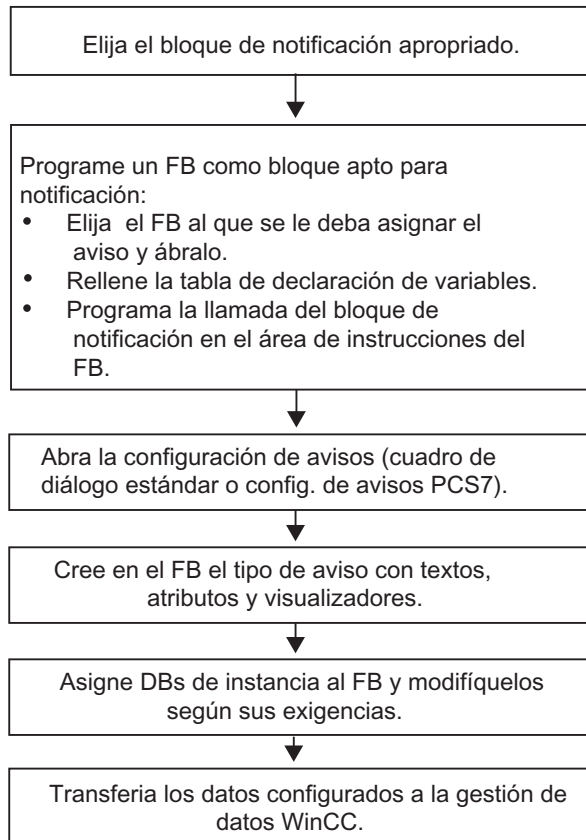
Los márgenes numéricos se asignan de forma estándar en intervalos de 20.000.

16.2.2 Asignar y editar avisos de bloque

Los avisos de bloque se asignan a un bloque (DB de instancia). Para crear un aviso de bloque se pueden utilizar bloques de funciones del sistema (SFB) y funciones del sistema (SFC).

16.2.2.1 Crear avisos de bloque (para todo el proyecto)

Procedimiento general



Programar un bloque apto para notificación (FB)

1. En el Administrador SIMATIC, elija el bloque de función (FB) para el que desea crear un aviso y ábralo haciendo doble clic en él.

Resultado: El bloque seleccionado se abre y se visualiza en la ventana "KOP/AWL/FUP".

2. Rellene la tabla de declaración de variables. Para cada bloque de notificación llamado en el FB es necesario declarar variables en el FB invocante.

Para ello introduzca las siguientes variables en la vista general de variables:

- en el parámetro "IN" un nombre simbólico para la entrada del bloque de notificación, p.ej., "Not01" (para la entrada de notificación 01), y el tipo de datos (tiene que ser "DWORD" sin valor inicial).
- en el parámetro "STAT" un nombre simbólico para el bloque de notificación invocante, p.ej., "alarma", y el tipo de datos correspondiente, aquí: "SFB33".

3. En el área de instrucciones del FB, introduzca la llamada del bloque de notificación seleccionado (en este caso "CALL alarma",) y confirme pulsando la tecla INTRO.

Resultado: En el área de instrucciones del FB se visualizarán las variables de entrada del bloque llamado, es decir, del SFB 33 en este caso.

4. Asigne a la variable "EV_ID" el nombre simbólico que ha definido en el paso 2 para la entrada del bloque de notificación, aquí: "Not01". Ahora los atributos de sistema se aplican al aviso del tipo "alarma"..

Resultado: en la columna "Nombre" con el parámetro "IN", si no está seleccionada, aparecerá una "banderilla". El bloque seleccionado se preparará para notificar avisos. Los atributos de sistema requeridos (p.ej., "S7_server" y "S7_a_type") y sus valores correspondientes se asignan automáticamente. (en algunos SFCs debe asignar los atributos de sistema para los parámetros "IN" usted mismo (comando de menú **Edición > Propiedades del sistema**, ficha "Atributos"))).

Atención: Si en lugar de un SFB llama un FB con multiinstancias en el que también se han configurador avisos, también deberá configurar los avisos del FB en el bloque que realiza la llamada.

5. Repita los pasos 2, 3 y 4 para todas las llamadas de bloques de notificación en este FB.
6. Guarde el bloque con el comando **Archivo > Guardar**.
7. Cierre la ventana "KOP/AWL/FUP".

Abrir la configuración de avisos

- Seleccione el bloque de notificación que desee y elija en el Administrador SIMATIC el comando de menú **Edición > Propiedades especiales del objeto > Aviso**.

Resultado: se abrirá el cuadro de diálogo (estándar) para la configuración de avisos en STEP 7. Si desea saber qué tiene que hacer para llamar la configuración de avisos PCS7, consulte Configuración de avisos PCS7.

Editar una plantilla

1. Seleccione el bloque de notificación deseado, abra la configuración de avisos e introduzca en las fichas "Texto" y "Atributos" el texto deseado o seleccione los atributos de sistema deseados.
Caso que haya seleccionado un bloque de notificación multicanal (p.ej., "ALARM_8"), podrá asignar a cada subnúmero textos y en parte atributos propios.
2. Asigne los visualizadores deseados a la plantilla. A tal efecto, haga clic en el botón "Nuevo visualizador". Aparecerá el cuadro de diálogo "Insertar visualizador", en el que podrá elegir los dispositivos deseados.

En las fichas posteriores, introduzca los textos y atributos deseados para los visualizadores. Salga del cuadro de diálogo haciendo clic en "Aceptar".

Nota

Para editar los textos y atributos específicos de los visualizadores, tenga en cuenta la documentación suministrada con el dispositivo en cuestión.

Crear DBs de instancia

1. Una vez creada una plantilla, puede asignarle bloques de datos de instancia (DBs) y editar los avisos para dichos DBs por instancias.
Abra en el Administrador SIMATIC el bloque que debe llamar el FB configurado previamente (p.ej.: el "OB1"), haciendo doble clic en él. En el área de instrucciones abierta en el OB, introduzca la llamada ("CALL"), así como el nombre y el número del FB que desea llamar y del DB que se ha de asignar al FB como instancia. Confirme sus entradas pulsando la tecla INTRO.

Ejemplo: Introduzca "CALL FB1, DB1". Si el DB1 no existe todavía, responda "Sí" cuando se le pregunte si desea generar el DB de instancia.

Resultado: se creará el DB de instancia. En el área de instrucciones del OB se visualizan las variables de entrada del FB correspondiente, aquí, p.ej., "Not01", y el número de aviso asignado por el sistema, aquí: el "1".

2. Con el comando **Archivo > Guardar** guarde el OB y cierre la ventana "KOP/AWL/FUP".

Editar avisos

1. Seleccione en el Administrador SIMATIC el DB de instancia generado, p.ej., el "DB1", y llame la configuración de avisos eligiendo el comando de menú **Edición > Propiedades especiales del objeto > Aviso**.

Resultado: Aparecerá el cuadro de diálogo "Configuración de avisos", en el que se visualizan el DB de instancia seleccionado y el número de aviso adjudicado por el sistema.

2. Introduzca en las fichas del cuadro de diálogo los cambios para el correspondiente DB de instancia y, en caso necesario, inserte más visualizadores. Abandone la función con "Aceptar".

Resultado: la configuración de avisos para el DB de instancia seleccionado ha concluído.

Transferir los datos de configuración

Transfiera los datos configurados a la gestión de datos de WinCC (mediante la configuración de enlaces PLC-OS) o a la gestión de datos de ProTool.

16.2.2.2 Editar avisos de bloque (para todo el proyecto)

1. En el administrador SIMATIC, seleccione el bloque que desee y el comando de menú **Editar > Propiedades especiales del objeto > Aviso**.
2. En la estructura de directorios, haga clic en la entrada de bloque de notificación o en uno de sus subnúmeros (si existen).

Resultado: Se muestra la ficha para un aviso global.

3. En la ficha "Texto" y en la ficha "Atributos" introduzca los textos y atributos.

Resultado: Se ha creado un aviso estándar que puede ser visualizado en todos los visualizadores.

4. Inserte un nuevo visualizador del tipo "ProTool" (Opx) o "WinCC" haciendo clic en el botón "Nuevo visualizador". Sólo se propondrán los visualizadores con los que se puedan visualizar realmente los avisos configurados.

Resultado: se inserta y selecciona el nuevo visualizador y se visualiza la ficha correspondiente.

5. Introduzca los atributos y textos para el aviso específico de visualización en las fichas "Textos" y "Atributos".

Resultado: Se ha creado una variante del aviso que sólo se podrá visualizar en el visualizador seleccionado.

Si desea editar otras variantes de avisos para los visualizadores ya disponibles:

- Seleccione y abra el bloque de avisos de la vista Detalle haciendo doble clic.

Resultado: el primer visualizador se selecciona automáticamente, y se puede editar las variantes de aviso específicas de la visualización.

16.2.2.3 Configuración de avisos PCS7 (para todo el proyecto)

Para editar las plantillas y avisos que se deseen visualizar en WinCC, STEP 7 ofrece -con la configuración de avisos PCS7- las siguientes posibilidades:

- configuración simplificada de visualizadores (creación automática)
- introducción simplificada de atributos y textos para avisos
- homogeneización de avisos garantizada

Llamar la configuración de avisos PCS 7

1. Seleccione en el Administrador SIMATIC el bloque (FB o DB) cuyos textos de aviso desea editar, y llame el cuadro de diálogo para la introducción de los atributos de sistema eligiendo el comando de menú **Edición > Propiedades del objeto**.
2. En la tabla que se visualiza entonces, introduzca el atributo del sistema "S7_alarm_ui" y el valor: "1" (el valor "0" desactiva la herramienta de configuración de avisos PCS 7). Los parámetros de propiedades se pueden asignar en KOP/FUP/AWL. Los DBs que se generen y se asignen posteriormente al FB adoptan automáticamente el ajuste del atributo. Este ajuste se puede modificar en los ajustes del DB independientemente de la plantilla del aviso (FB).

Nota

Al introducir los atributos del sistema se comprueba su sintaxis. Las entradas erróneas se destacan en color rojo.

3. Salga del cuadro de diálogo haciendo clic en "Aceptar".
4. Seleccione el comando de menú **Edición > Propiedades especiales del objeto > Aviso**.

Resultado: se abrirá el cuadro de diálogo para la configuración de avisos PCS 7.

Editar plantillas

1. Seleccione en el Administrador SIMATIC el FB cuyos textos de aviso desea editar y llame la configuración de avisos PCS7.

Resultado: en el cuadro de diálogo se visualizará una ficha para cada uno de los bloques de notificación para los que haya declarado una variable en el FB.

2. Rellene los cuadros de texto para las partes del aviso "Proveniencia", "Area OS" e "Ident. batch".
3. Indique la categoría y el texto de todos los eventos de los bloques de notificación utilizados y decida si se debe acusar cada evento por separado.
4. Haga clic en la marca de verificación "Inhibido" para aquellas partes del aviso que son válidas para todas las instancias y no se deben modificar.

Editar avisos

1. Seleccione en el Administrador SIMATIC el DB de instancia cuyos textos de aviso desea editar y llame la configuración de avisos PCS 7.
2. Modifique las partes específicas de cada instancia que no han sido inhibidas.

16.2.3 Asignar y editar avisos de símbolos

16.2.3.1 Asignar y editar avisos de símbolos (para todo el proyecto)

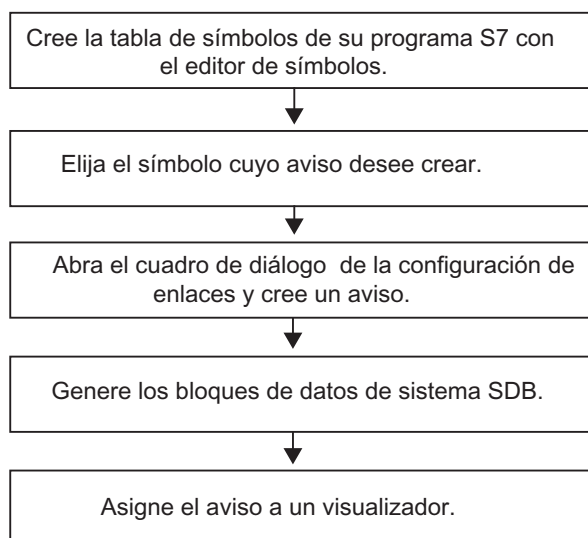
Los avisos de símbolos (SCAN) se asignan directamente a una señal en la tabla de símbolos. Se pueden utilizar todos los operandos booleanos: es decir, entradas (E), salidas (A) y marcas (M). Al configurar los avisos es posible asignar diversos atributos, textos y hasta 10 valores asociados a dichas señales. Para facilitar la elección de señales de la tabla de símbolos se pueden ajustar filtros.

Con un aviso de símbolo se puede escrutar una señal en un período predeterminado para averiguar si se ha presentado un cambio.

Nota

La duración del período depende de la CPU utilizada.

Procedimiento general



Durante la ejecución del programa se comprueban asincrónicamente las señales para las cuales se hayan configurado los avisos. Las comprobaciones se llevan a cabo en las bases de tiempo configuradas. Los avisos se visualizarán en los visualizadores asignados.

Atención

Si desea asignar o editar avisos de símbolos y, en el mismo proceso, ha copiado anteriormente símbolos entre dos tablas de símbolos, primero debe cerrar la tabla de símbolos con la que ya no quiere trabajar. De lo contrario, las configuraciones de los avisos no pueden guardarse. En determinadas circunstancias, incluso se pierden las últimas entradas realizadas en el cuadro de diálogo de configuración para avisos.

16.2.4 Crear y editar avisos de diagnóstico personalizados

Esta función permite escribir una entrada en el búfer de diagnóstico y emitir el aviso correspondiente. Los avisos de diagnóstico personalizados se realizan con la función del sistema SFC 52 (WR_USMSG, clase de error A o B) que se utiliza como bloque de notificación. La llamada de la SFC 52 se debe insertar en el programa de usuario, adjudicándole además el ID del evento.

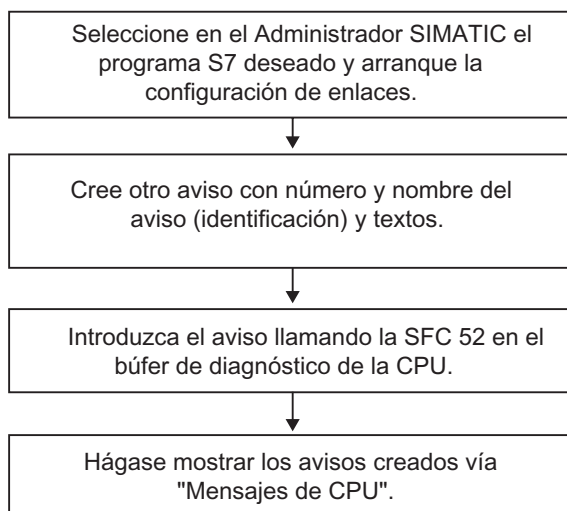
Requisitos

Para poder generar un aviso de diagnóstico personalizado es necesario haber creado previamente:

- un proyecto en el Administrador SIMATIC y
- en el proyecto, un programa S7/M7 al que se desea asignar uno o más avisos.

Procedimiento general

Para crear y visualizar avisos de diagnóstico personalizados proceda de la siguiente forma:



16.3 Configurar avisos para toda la CPU

16.3.1 Asignación de números de aviso para toda la CPU

Los avisos están identificados con un número unívoco para toda la CPU. Para ello se les asigna un margen numérico a cada una de las CPUs. Al contrario que en la asignación de números de aviso para todo el proyecto, en este caso al copiar un programa ya no tendrá que asignar al nuevo programa un nuevo margen numérico. De ese modo tampoco hace falta recompilar los programas. Esto no es aplicable para la copia de bloques individuales. En este caso sí es necesaria una recompilación para vincular en el programa los números de aviso modificados.

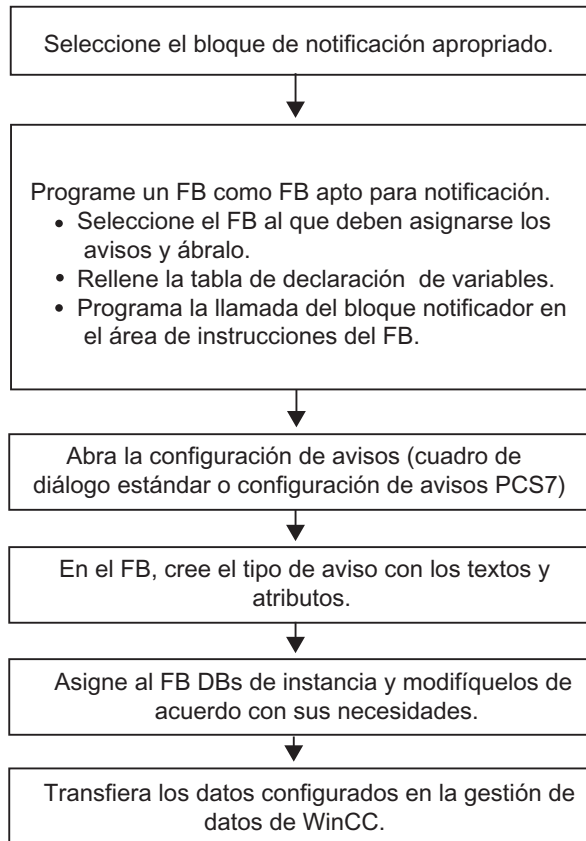
Requisitos

- WinCC V6.0
- ProTool V6.0

16.3.2 Asignar y editar avisos de bloque

16.3.2.1 Crear avisos de bloques (para toda la CPU)

Procedimiento básico



Programar un bloque apto para notificación (FB)

1. En el administrador SIMATIC, seleccione el bloque de función para el que desea crear un aviso de bloque y abra el bloque en cuestión haciendo doble clic sobre él.

Resultado: El bloque seleccionado se abre y se visualiza en la ventana "KOP/AWL/FUP".

2. Rellene la tabla de declaración de variables. Para cada bloque de notificación que se llama en el FB, debe declarar variables en el FB que hace la llamada.

Para ello debe introducir las siguientes variables en la vista general de variables:

- en el parámetro "IN" debe indicar un nombre simbólico para la entrada del bloque de notificación, p.ej. "Entr01" (para entrada de notificación 01) y el tipo de datos (debe ser "DWORD" sin valor inicial).
- en el parámetro "STAT" debe indicar un nombre simbólico para el bloque de notificación que realiza la llamada, p.ej. "alarma" y el tipo de datos correspondiente, en este caso "SFB33".

3. En el área de instrucciones del FB debe indicar la llamada correspondiente al bloque de notificación seleccionado, en este caso "CALL alarm", y finalice la introducción de los datos con la tecla INTRO.

Resultado: En el área de instrucciones del FB se visualizan las variables de entrada del bloque de notificación que realiza la llamada, en este caso SFB 33.

4. Asigne a la variable "EV_ID" el nombre simbólico que haya indicado en el paso 2 para la entrada del bloque de notificación, en este caso "Entr01".

Resultado: En la columna "Nombre", con el parámetro "IN", aparece, si no está seleccionada, una "banderilla". Esto indica que el bloque seleccionado es apto para notificación. Los atributos de sistema necesarios (p. ej. S7_server y S7_a_type) y los valores correspondientes se asignan automáticamente (en algunos SFC debe asignar los parámetros de sistema para el parámetro "IN" usted mismo (comando de menú **Edición > Propiedades del objeto**, ficha "Atributos"))).

Atención: Si en lugar de un SFB llama a un FB con multiinstancias en el que también hay configurados avisos, también debe configurar los avisos del FB con multiinstancias en el bloque que realiza la llamada.

5. Repita los pasos 2 a 4 para las llamadas de bloques de notificación en ese FB.
6. Guarde el bloque con el comando de menú **Archivo > Guardar**.
7. Cierre la ventana "KOP/AWL/FUP".

Abrir la configuración de avisos

- En el administrador SIMATIC, seleccione el bloque de notificación que desee y el comando de menú **Edición > Propiedades especiales del objeto > Aviso**.

Resultado: El cuadro de diálogo para la configuración de avisos de STEP 7 se abre. Para más información sobre el procedimiento a seguir para abrir la configuración de avisos PCS7, consulte Configuración de avisos PCS7 (para toda la CPU).

Editar plantilla de avisos

- Seleccione el bloque de notificación que desee
- En las columnas correspondientes Introduzca los textos o seleccione los atributos que desee.
También puede hacer clic sobre el botón "Ampliar" en el cuadro de diálogo "Configuración de avisos" e introducir el texto del aviso y el texto adicional que desee en la ficha "Textos estándar".
Si ha seleccionado un bloque de notificación de varios canales (p. ej. "ALARM_8"), puede asignar un a cada subnúmero textos propios y atributos en parte propios
- Si los textos o atributos no deben poder modificarse en la instancia, inhíbalos en la plantilla.

Crear DBs de instancia

1. Una vez que haya creado un tipo de aviso puede asignarle bloques de datos de instancia (DBs) y editar los avisos para esos DBs específicos para cada instancia. Para ello debe abrir en el administrador SIMATIC el bloque que debe llamar el FB que ha creado anteriormente, p.ej. "OB1" haciendo doble clic sobre él. En el área de instrucciones del OB, introduzca la llamada ("CALL") y el nombre y el número del FB que realiza la llamada y del DB que desea asignar como instancia al FB. Una vez introducidos todos los datos, pulse la tecla INTRO.

Ejemplo: Introduzca "CALL FB1, DB1". Si el DB1 todavía no existe, cuando el sistema le pregunte si desea generar el DB de instancia, confirme con "Sí".

Resultado: El DB de instancia se crea. En el área de instrucciones del OB se muestran las variables de entrada del FB correspondiente, en este caso p.ej. "Entr01" y el número de aviso asignado por el sistema, en este caso "1".

2. Guarde el OB con el comando de menú **Archivo > Guardar** y cierre la ventana "KOP/AWL/FUP".

Editar avisos

1. En el administrador SIMATIC, seleccione el DB de instancia creado, p.ej. "DB1" y abra la configuración de avisos con el comando de menú **Edición > Propiedades especiales del objeto > Aviso**.

Resultado: Se abre el cuadro de diálogo "Configuración de avisos" y se muestra el DB de instancia seleccionado con el número de aviso asignado por el sistema.

2. Introduzca las modificaciones que desee para el DB de instancia en cuestión. Cierre la función con "Aceptar".

Resultado: Con ello termina la configuración de avisos para el DB de instancia seleccionado.

Nota

Si en una instancia aparecen textos y atributos representados en color verde, esto significa lo siguiente: estos textos y atributos siguen siendo tal y como fueron configurados en la plantilla. Desde entonces no han sido modificados en la instancia.

Transferir datos de configuración

- Transfiera los datos configurados a la gestión de datos de WinCC (a través de la configuración de enlaces AS-OS) o a la de ProTool.

16.3.2.2 Editar avisos de bloques (para toda la CPU)

1. Seleccione un bloque de notificación y abra la configuración de avisos con el comando de menú **Edición > Propiedades especiales del objeto > Aviso**.
2. En las columnas "Textos estándar" y "Textos adicionales", introduzca los textos que desee.
También puede hacer clic sobre el botón "Ampliar" e introducir los textos en las fichas "Textos estándar" y "Textos adicionales" con saltos de línea.

Resultado: Ha creado un aviso estándar.

Nota

Si en una instancia aparecen textos y atributos representados en color verde, esto significa lo siguiente: estos textos y atributos siguen siendo tal y como fueron configurados en la plantilla. Desde entonces no han sido modificados en la instancia.

16.3.2.3 Configuración de avisos PCS 7 (para toda la CPU)

Para trabajar con tipos de aviso y con avisos que deben editarse en visualizadores WinCC (a partir de V6.0), con la configuración de avisos PCS 7 STEP 7 ofrece una cómoda solución para

- simplificar la configuración de visualizadores
- simplificar la introducción de atributos y textos para avisos
- garantizar la unificación de avisos

Abrir la configuración de avisos PCS7

1. En el administrador SIMATIC, seleccione el bloque (FB o DB), cuyo texto de aviso desea editar y abra el cuadro de diálogo para la introducción de atributos de sistema con el comando de menú **Edición > Propiedades del objeto**.
2. En la tabla que aparece, introduzca el atributo de sistema "S7_alarm_ui" y el valor: "1" (el valor "0" desactiva la configuración de avisos PCS 7). Los parámetros referentes a las propiedades pueden asignarse en KOP/FUP/AWL. Los DBs que se creen a continuación y que se asignen al FB en cuestión adoptarán los ajustes y podrán conmutarse a través de las definiciones de atributos propias independientemente del tipo de aviso (FB).

Nota:

Al introducir los atributos de sistema se realiza una comprobación de sintaxis y las entradas erróneas se resaltan en color rojo.

3. Cierre el cuadro de diálogo haciendo clic sobre "Aceptar".
4. Seleccione el comando de menú **Edición > Propiedades especiales del objeto > Aviso**.

Resultado: El cuadro de diálogo para la configuración de avisos PCS 7 se abre.

Editar tipos de aviso

1. En el administrador SIMATIC, seleccione el FB cuyos textos de aviso desea editar y abra la configuración de avisos PCS 7.
2. Haga clic sobre el botón "Ampliar" y en la ficha "Textos generales" rellene los campos de texto relativos a los componentes del aviso "Proveniencia", "Área OS" e "Código batch".
3. Para los resultados de los bloques de notificación utilizados, indique la clase de aviso y el texto del evento y especifique si cada evento debe acusarse individualmente.
4. Para los componentes del aviso que deben aplicarse a todas las instancias y que no puedan modificarse en ellas, haga clic en la casilla de verificación "Inhibido".

Editar avisos

1. En el administrador SIMATIC, seleccione el DB de instancia cuyos textos de aviso desea editar y abra la configuración de avisos PCS 7.
2. Modifique los componentes del aviso específicos de la instancia que no están inhibidos.

16.3.3 Asignar y editar avisos de símbolos

16.3.3.1 Asignar y editar avisos de símbolos (para toda la CPU)

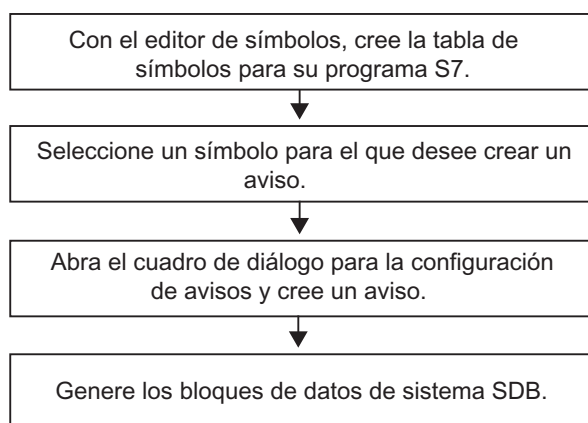
Los avisos de símbolos (SCAN) se asignan directamente a una señal de la tabla de símbolos. Las señales permitidas son todos los operandos booleanos: es decir, entradas (E), salidas (A) y marcas (M). A estas señales puede asignarles diferentes atributos, textos de aviso y hasta 10 valores asociados en la configuración de avisos. La selección de señales de la tabla de símbolos resulta más fácil si se utilizan filtros.

Un aviso de símbolos le permite explorar una señal siguiendo un modelo temporal predeterminado para saber si se ha producido un cambio de señal.

Nota:

El modelo temporal varía en función de la CPU utilizada.

Procedimiento básico



Durante el proceso, de forma asíncrona al programa se comprueban las señales para las que ha configurado avisos. Las comprobaciones se producen en el modelo temporal configurado. Los avisos se visualizan en los visualizadores.

Atención

Si desea asignar o editar avisos de símbolos y, en el mismo proceso, ha copiado anteriormente símbolos entre dos tablas de símbolos, primero debe cerrar la tabla de símbolos con la que ya no quiere trabajar. De lo contrario, las configuraciones de los avisos no pueden guardarse. En determinadas circunstancias, incluso se pierden las últimas entradas realizadas en el cuadro de diálogo de configuración para avisos.

16.3.4 Crear y editar avisos de diagnóstico personalizados

Esta función permite escribir una entrada en el búfer de diagnóstico y emitir el aviso correspondiente. Los avisos de diagnóstico personalizados se realizan con la función del sistema SFC 52 (WR_USMSG, clase de error A o B) que se utiliza como bloque de notificación. La llamada de la SFC 52 se debe insertar en el programa de usuario, adjudicándole además el ID del evento.

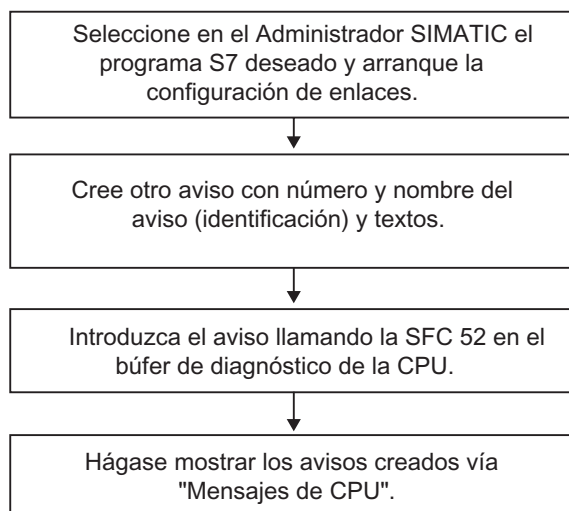
Requisitos

Para poder generar un aviso de diagnóstico personalizado es necesario haber creado previamente:

- un proyecto en el Administrador SIMATIC y
- en el proyecto, un programa S7/M7 al que se desea asignar uno o más avisos.

Procedimiento general

Para crear y visualizar avisos de diagnóstico personalizados proceda de la siguiente forma:



16.4 Trucos para editar avisos

16.4.1 Insertar valores asociados en avisos

Para añadir información actual en los avisos de bloques y de símbolos (p.ej., del proceso), puede insertar valores asociados en cualquier punto del texto de un aviso.

Para hacerlo, proceda de la siguiente forma:

1. Componga un bloque con la siguiente estructura:
`@<Nº del valor asociado><Tipo de elemento><Indicación de formato>@.`
2. Inserte este bloque en aquellos puntos del texto del aviso en los que deba visualizarse el valor asociado.

Tipo de elemento

Aquí se configura de forma unívoca el tipo de dato del valor asociado:

Tipo de elemento	Tipo de dato
Y	BYTE
W	WORD
X	DWORD
I	Integer
D	Integer
B	BOOL
C	CHAR
R	REAL

El tipo de elemento sólo identifica de forma unívoca el tipo de dato transferido al AS. No se utiliza como Casting Operator.

Indicación de formato

Defina el formato de salida del valor asociado en el visualizador. La indicación de formato está precedida del signo "%". Para textos de avisos existen las siguientes indicaciones de formato fijas:

Indicación de formato	Descripción
%[i]X	Número hexadecimal con i dígitos
%[i]u	Número decimal sin signo con i dígitos
%[i]d	Número decimal con signo con i dígitos
%[i]b	Número binario con i dígitos
%[i][.y]f	Número en coma fija Con signo, de la forma [-]dddd.dddd dddd: una o varias cifras con y dígitos tras la coma decimal e i dígitos en total
%[i]s	Cadena de caracteres (ANSI String) con i dígitos Los caracteres se imprimen hasta el primer byte 0 (00Hex).
%t#<Nombre de la librería de textos>	Acceso a la librería de textos

Aunque el formato indicado sea insuficiente, el valor se visualizará en toda su longitud.

Si el formato indicado es excesivo, delante del valor se visualizarán caracteres en blanco.

Nota

Tenga en cuenta que puede indicar "[i]", pero sin los corchetes.

Ejemplos de valores asociados:

@1%6d@: El valor representado por el valor asociado 1 se visualiza con un número decimal de un máximo de 6 dígitos.

@2R%6f@: El valor "5.4" como ejemplo del valor asociado 2 se visualiza como número de coma fija "5.4" (precedido de tres blancos).

@2R%2f@: El valor "5.4" como ejemplo del valor asociado 2 se visualiza como número de coma fija "5.4" (si el número de dígitos es pequeño no se trunca).

@1W%t#Textbib1@: Valor asociado 1- del tipo de datos WORD es el índice con el que se referencia el texto a utilizar de la librería de textos Textbib1.

Nota

En S7-PDIAG hay que indicar como tipo de elemento para CHAR siempre "C" y para REAL siempre "R". Para los restantes tipos de elementos válidos para S7-PDIAG BOOL, BYTE, WORD, INT, DWORD y DINT siempre debe indicarse una "X".

Si desea transferir más de un valor asociado a uno de los bloques ALARM_S, puede transferir un array (matriz) de 12 bytes de longitud como máximo. Por ejemplo, 12 bytes o Char como máximo, 6 Word o Int como máximo o 3 DWord, Real o DInt como máximo.

16.4.2 Integrar textos de librerías en avisos

Puede integrar en un aviso cualquier texto de un máximo de 4 librerías de textos diferentes. Los textos pueden ubicarse libremente, lo que garantiza también el uso en avisos en otros idiomas.

Proceda de la siguiente forma:

1. En el Administrador SIMATIC elija la CPU o un objeto subordinado a la CPU y elija después el comando de menú Herramientas > Librerías de textos > Librería de textos del sistema o Herramientas > Librerías de textos > Librería de textos de usuario para abrir una librería de textos.

Atención

Sólo puede integrar textos de librerías de textos de usuario en avisos si ha asignado la asignación de números de aviso para toda la CPU.

1. Defina el índice del texto que desea integrar.
2. En el lugar del aviso en el que debe aparecer el texto, introduzca un comodín de formato @[Index]#t#[Textbib]@.

Nota

[Index] = p.ej. 1W, donde 1W es el primer valor asociado del aviso del tipo WORD.

Ejemplo

Texto configurado del aviso: La presión está @2W#t#Textbib1@.

Librería de textos con el nombre "Textbib1":

Índice	alemán	español
1734	zu hoch	demasiado alta

El segundo valor asociado tiene asignado el valor 1734.

El texto del aviso que se visualiza es: La presión es demasiado alta.

16.4.3 Borrar valores asociados

Se pueden borrar valores asociados borrando del texto del aviso la cadena que representa un valor asociado.

Para hacerlo, proceda de la siguiente forma:

1. Busque en el texto del aviso el bloque que corresponde al valor asociado que desea borrar.
El bloque comienza con el carácter "@", contiene el identificador local (en el que puede reconocer el valor asociado), una indicación de formato, y termina con el carácter "@".
2. Borre del texto del aviso el bloque que ha encontrado.

16.5 Traducir y editar textos para operarios

Los textos que aparecen en los visualizadores al ejecutar el proceso se suelen mostrar en el idioma en el que se programó la solución de automatización.

Sin embargo, a menudo ocurre que el operador que debe reaccionar a estos avisos no entiende este idioma, por lo que necesita los textos en su lengua. Sólo de este modo se garantiza una rápida reacción y una respuesta adecuada a los avisos que aparezcan.

STEP 7 ofrece la posibilidad de traducir todos los textos de usuario en el idioma deseado. Para ello tiene que haberse instalado el idioma en cuestión en el proyecto (comando de menú del Administrador SIMATIC: **Herramientas > Idioma para visualizador**). El número de idiomas disponibles se define al instalar Windows (propiedad del sistema).

De esta forma se garantiza que los operadores puedan visualizar los avisos en el idioma que deseen, optimizando así la seguridad del proceso.

Son textos de usuario los textos de usuario y las librerías de textos.

16.5.1 Traducir y editar textos de usuario

Esta función permite crear textos de usuario para un proyecto entero, para programas S7, para la carpeta de bloques o para determinados bloque, así como para la tabla de símbolos, siempre que en dichos objetos puedan configurarse avisos. Contienen todos los textos y avisos que pueden visualizarse p.ej. en visualizadores. Para un proyecto puede haber varias listas de textos de usuario que se pueden traducir a los idiomas necesarios.

Puede elegir los idiomas disponibles en un proyecto (comando de menú: **Herramientas > Idioma para visualizador**). También puede añadir o borrar idiomas posteriormente.

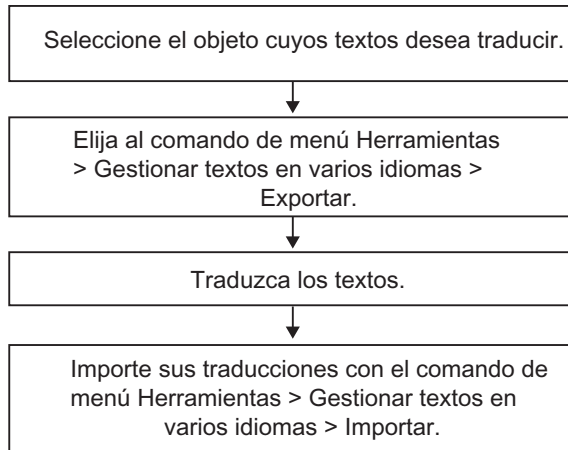
Exportar e importar textos de usuario

Los textos de usuario creados en STEP 7 pueden traducirse o editarse fuera de STEP 7. Para ello, exporte la lista visualizada de textos de usuario a archivos de exportación que pueda editar con un editor ASCII o con una herramienta de procesamiento de tablas, p.ej. la aplicación EXCEL de Microsoft (comando de menú: **Herramientas > Gestionar textos en varios idiomas > Exportar**). Una vez abiertos, en la pantalla se visualiza una tabla con una columna para cada idioma. En la primera columna se visualiza siempre el idioma ajustado como idioma estándar. Después de traducir los textos, impórtelos de nuevo a STEP 7.

Los textos de usuario sólo pueden importarse a la parte del proyecto desde la que se exportaron.

Procedimiento básico

Asegúrese de que en el Administrador SIMATIC ha ajustado con el comando de menú **Herramientas > Idioma para visualizador** los idiomas a los que van a traducirse los textos de usuario.



Nota

Los textos de usuario sólo pueden imprimirse en la aplicación en la que se traducen.

16.6 Editar y traducir librerías de textos

16.6.1 Librerías de textos de usuario

Una librería de textos de usuario permite visualizar textos o segmentos de textos de forma dinámica en función de un valor asociado. El valor asociado proporciona el índice de la librería de textos correspondiente al texto actual. En el punto en el que debe mostrarse el texto dinámico se inserta un comodín.

Puede crear librerías de textos de usuario para un programa, introducir textos en ella y asignarles un índice de selección propia. La aplicación comprueba automáticamente si el índice es unívoco dentro de la librería de textos de usuario. Una librería de textos de usuario puede referenciarse desde cualquier aviso disponible para esa CPU.

Una carpeta de librerías de textos puede contener un número ilimitado de librerías. Eso permite p. ej. utilizar el mismo programa para diferentes tareas de control simplemente adaptando las librerías de textos a las necesidades.

Atención

Si copia en otro programa un bloque que llama y que tiene una referencia a una librería de textos, también deberá copiar las librerías de textos correspondientes, crear una librería de textos nueva con el mismo nombre y contenido o modificar la referencia en el texto del aviso.

Al crear una entrada de texto se asigna siempre un índice. Al crear una nueva línea, la aplicación propone por defecto el siguiente índice libre. No se aceptan índices no unívocos dentro de una librería de textos y la aplicación los rechaza.

16.6.2 Crear librerías de textos de usuario

Para crear una librería de textos de usuario, proceda de la siguiente manera:

1. En el administrador SIMATIC, seleccione el programa o un objeto subordinado al programa para el que desee crear una librería de textos de usuario y seleccione en el administrador SIMATIC el comando de menú **Insertar > Librería de textos > Carpeta Librería de textos**.
Resultado: Se crea la carpeta "Librería de textos".
2. Ahora seleccione la carpeta "Librería de textos", seleccione el comando de menú **Insertar > Librería de textos > Librería de textos de usuario** y asigne un nombre para la librería de textos.
3. Para abrir la librería de textos, seleccione el comando de menú **Herramientas > Librerías de textos > Librería de textos de usuario**.
4. Ya puede introducir los textos que desee

Nota

Al crear una entrada de texto se fuerza la asignación de un índice. Por defecto la aplicación propone el siguiente índice libre al crear una nueva línea. La aplicación rechaza los índices duplicados dentro de una librería de textos.

16.6.3 Editar librerías de textos de usuario

Para editar librerías de textos de sistema existentes, proceda de la siguiente manera:

1. En el administrador SIMATIC, seleccione la CPU o un objeto subordinado a la CPU cuya librería de textos desea editar y seleccione en el administrador el comando de menú **Herramientas > Librerías de textos > Librerías de textos de usuario**.
2. En el cuadro de diálogo "Librerías de textos disponibles" seleccione la librería de textos de sistema que desea abrir.
3. Edite los textos mostrados. Para ello puede utilizar diferentes funciones de edición (p.ej. Buscar y reemplazar). El índice correspondiente a cada uno de los textos se genera automáticamente y no puede modificarse.
4. Imprima los textos si necesita una copia impresa de los mismos.
5. Una vez terminados todos los trabajos, guarde la librería de textos.
6. Cierre la aplicación una vez que haya finalizado todos los trabajos.

Atención

Si copia en otro programa un bloque que llama y que tiene una referencia a una librería de textos, también deberá copiar las librerías de textos correspondientes, crear una librería de textos nueva con el mismo nombre o modificar la referencia en el texto del aviso.

Si cambia el nombre de una librería de textos ya existente, todos los valores asociados de avisos ya configurados referenciados en esa librería perderán su validez.

16.6.4 Librerías de textos de sistema

Las librerías de textos de sistema se crean automáticamente p. ej. en "Notificar errores de sistema" al generar bloques. El usuario no puede crear librerías de textos de sistema, sólo puede editar las librerías ya existentes.

Una librería de textos puede referenciarse desde cualquier aviso disponible para esa CPU.

16.6.5 Traducir librerías de textos

Las librerías de textos de usuario y de sistema proporcionan una lista de textos que pueden integrarse en avisos, actualizarse dinámicamente durante el tiempo de ejecución y visualizarse en la PG o en otros visualizadores.

Los textos de librerías de textos del sistema son proporcionados por STEP 7 o por paquetes opcionales de STEP 7. Para una CPU pueden existir varias librerías de textos que se pueden traducir a los idiomas necesarios.

En el Administrador SIMATIC puede elegir los idiomas disponibles en un proyecto (comando de menú: **Herramientas > Idioma para visualizador**). También puede añadir o borrar idiomas posteriormente.

Si desea traducir una librería de textos (comando de menú: **Herramientas > Gestionar textos en varios idiomas > Exportar**), se crea un archivo de exportación que puede editarse con Microsoft EXCEL. Una vez abierto, en la pantalla se visualiza una tabla con una columna para cada idioma.

Atención

Un archivo de exportación en formato CSV no puede abrirse con un doble clic sobre el archivo. Debe abrir el archivo con el comando de menú **Archivo > Abrir** de Microsoft EXCEL.

Nota

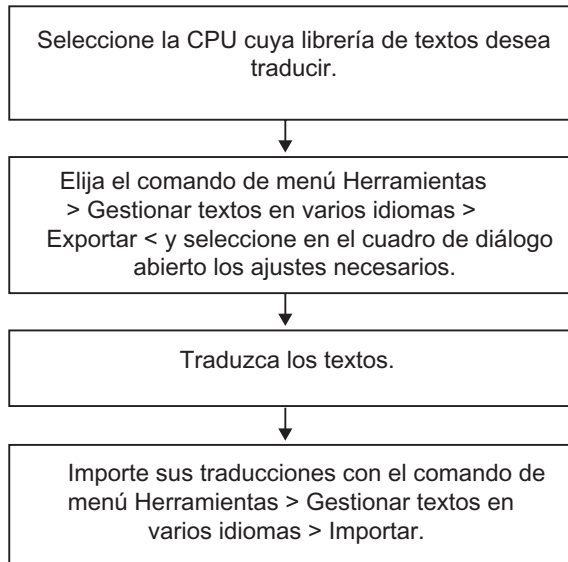
Los textos de usuario sólo pueden imprimirse en la aplicación en la que se traducen.

Ejemplo de un archivo de exportación

alemán	español
ausgefallen	no responde
gestört	averiado
Parametrierfehler	error de parametrización

Procedimiento básico

Asegúrese de que en el Administrador SIMATIC ha ajustado con el comando de menú **Herramientas > Idioma para visualizador** los idiomas a los que va a traducirse la librería de textos.



16.7 Transferir los datos de configuración al sistema de destino

Introducción

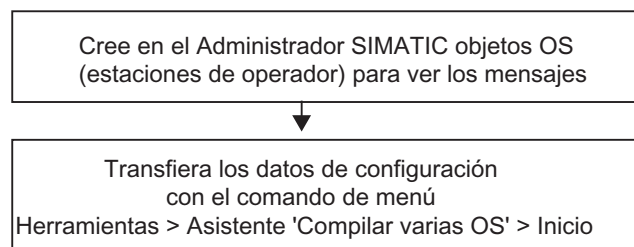
Los datos de configuración para manejo y visualización se transfieren con el programa "Ingeniería para PLC-OS" a la gestión de datos de WinCC.

Requisitos

Antes de transferir los datos se deberán cumplir los siguientes requisitos:

- Se deberá haber instalado el programa "Engineering PLC-OS".
- Se deberán haber generado los datos de configuración para crear avisos.

Procedimiento general



16.8 Visualizar avisos de CPU y avisos de diagnóstico personalizados

Con la función "Avisos de CPU" (comando de menú **Sistema de destino > Avisos de CPU**) es posible visualizar avisos de bloques ALARM_S (SFC 18 y SFC 108 para la generación de avisos siempre acusados así como SFC 17 y SFC 107 para la generación de avisos de bloque acusables).

Además, si en la ventana "Avisos de CPU" se elige el comando **Edición > Aviso > Diagnóstico personalizado**, es posible iniciar la configuración y crear avisos de diagnóstico personalizados.

Posibilidades de visualización

La función "Avisos de CPU" permite ajustar la visualización de los avisos de las CPUs seleccionadas y qué avisos se deben visualizar:

- **"Resaltar en la barra de tareas"**: cuando se recibe un aviso y la ventana no se encuentra en primer plano, la aplicación "Avisos de CPU" se resalta en la barra de tareas de Windows.
- **"Segundo plano"**: los avisos de CPU se reciben en segundo plano. La ventana permanece en segundo plano aunque se reciban nuevos avisos, pudiéndose visualizar en caso necesario.
- **"Ignorar aviso"**: los avisos de CPU nuevos **no** se visualizan y, contrariamente a los dos primeros casos, **no** se archivan.

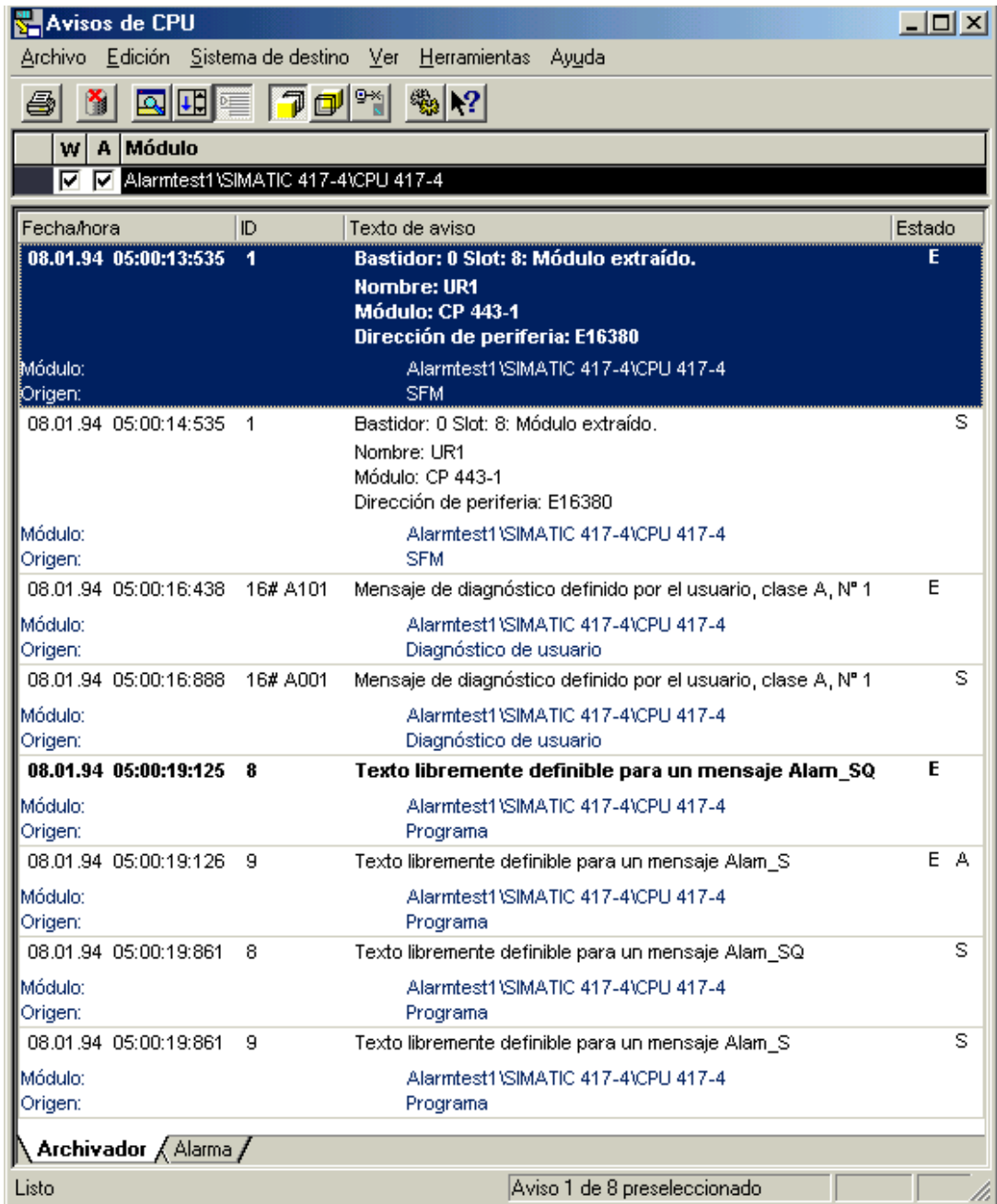
En la ventana "Avisos de CPU" se pueden seleccionar las fichas "Archivador" y "Alarma". En ambas fichas puede escoger con el comando de menú **Ver > Mostrar texto informativo** si los avisos deben visualizarse con o sin texto informativo.

Ficha "Archivador"

En esta ficha se muestran y archivan los avisos entrantes en orden cronológico de aparición. Puede definir el tamaño del archivador (entre 40 y 3000 avisos de CPU) con el comando de menú **Herramientas > Preferencias** en el cuadro de diálogo "Preferencias - Visualizar avisos de CPU". Si se sobrepasa el tamaño definido para el archivador, se borra el aviso de mayor antigüedad.

Los avisos acusables (ALARM_SQ y ALARM_DQ) se muestran en negrita y pueden acusarse con el comando de menú **Edición > Acusar aviso de CPU**.

En la siguiente imagen se representan algunos ejemplos



Ficha "Alarma"

Los avisos de bloques ALARM_S que todavía no han salido o no han sido acusados se muestran de forma adicional en la ficha "Alarma" con el correspondiente estado.

Con el comando de menú **Ver > Avisos en varias líneas** selecciona si los avisos deben mostrarse en una o varias líneas. También puede ordenar las columnas como desee.

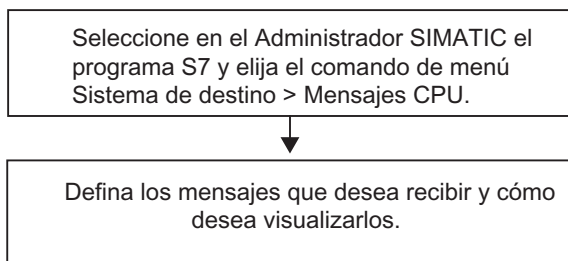
Actualizar avisos de bloques ALARM_S

Al actualizar avisos ALARM_S/SQ se vuelven a registrar en el archivador los avisos que no han salido o los que todavía no se han acusado. Los avisos se actualizan en los siguientes casos:

- al rearrancar el módulo al que se refieren los avisos (pero no al realizar un re arranque completo)
- haciendo clic en el campo "A" para avisos de bloques ALARM_S en la lista de módulos.

Procedimiento general

Para configurar avisos de CPU para los módulos seleccionados:



16.8.1 Configurar avisos de CPU

Para configurar avisos de CPU para determinados módulos proceda como sigue:

1. En el Administrador SIMATIC, inicie la aplicación "Avisos de CPU" desde un proyecto online. Para ello seleccione online un programa S7 y, con el comando **Sistema de destino > Avisos de CPU**, llame la aplicación "Avisos de CPU" para la CPU seleccionada.

Resultado: Aparecerá la ventana de la aplicación "Avisos de CPU", donde figura la CPU dada de alta.

2. Si es necesario, seleccione otras CPUs; para ello repita el primer paso para otros programas o interfaces.
3. Haga clic en las casillas de verificación que aparecen delante de las entradas de la lista y determine qué avisos habrá de recibir el módulo:

A: activa avisos de bloques ALARM_S (SFC 18 y SFC 108 para la creación de avisos de bloques siempre acusados y SFC 17 y SFC 107 para la creación de avisos de bloques acusables), p. ej., avisos de diagnóstico de proceso de S7-PDIAG; S7-GRAPH o 'Notificar errores de sistema'.

W: activa eventos de diagnóstico
Determine el tamaño del archivador.

4. **Resultado:** Cuando se presenten los avisos mencionados, éstos se escribirán en el archivador y se visualizarán conforme al modo elegido.

Nota

En la lista de módulos dados de alta que aparece en la ventana de la aplicación "Avisos de CPU" están registradas las CPUs para las que eligió el comando de menú **Sistema de destino > Avisos de CPU**. Las entradas de esta lista no se borran hasta que no se borren del cuadro de diálogo.

16.8.2 Visualizar los avisos de CPU guardados

Los avisos de la CPU se guardan siempre en el archivador, a menos que se haya elegido el comando de menú **Ver > Ignorar aviso**. Todos estos avisos archivados se visualizarán siempre.

16.9 Configurar 'Notificación de errores del sistema'

Introducción

En caso de error del sistema, los componentes S7 y los "esclavos DP normalizados" (esclavos cuyas propiedades están determinadas por su archivo GSD) pueden provocar llamadas a un bloque de organización.

Ejemplo: En caso de rotura de hilo, un módulo apto para diagnóstico puede disparar la alarma de diagnóstico (OB 82).

Los componentes S7 proporcionan información sobre los errores del sistema que se han producido. La información del evento de arranque, es decir, los datos locales del OB asignado (contienen, entre otros, el registro 0), proporcionan información general sobre el lugar (p. ej., dirección lógica del módulo) y el tipo del error (p. ej., error de canal o fallo del respaldo de la pila).

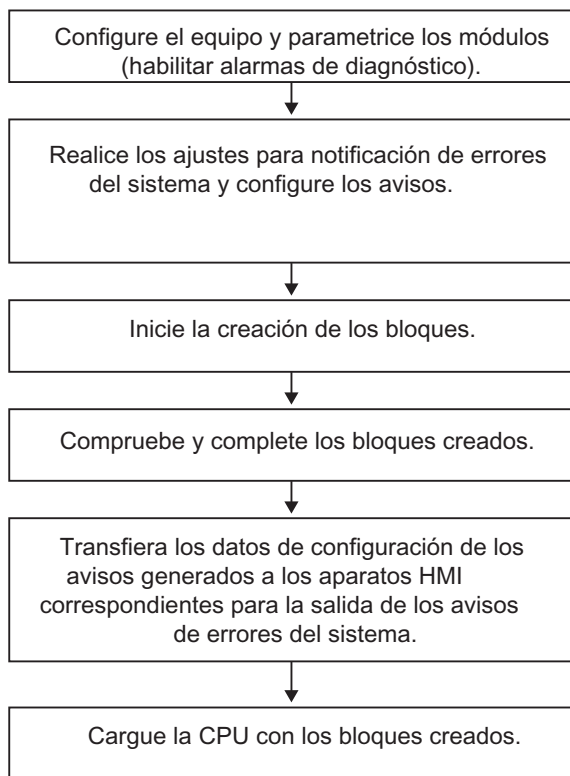
Además, informaciones de diagnóstico adicionales (lectura del registro 1 con SFC51 o lectura del telegrama de diagnóstico del esclavo normalizado DP con SFC 13) permiten especificar con mayor precisión el error: p. ej., canal 0 o 1, rotura de hilo o rebase por exceso del margen de medida.

Con la función "Notificar errores del sistema", STEP 7 ofrece una forma cómoda de ver en forma de avisos la información de diagnóstico proporcionada por los componentes.

Los bloques y textos de aviso necesarios para ello son generados automáticamente por STEP 7. El usuario sólo tiene que cargar en la CPU los bloques generados y transferir los textos a los aparatos HMI conectados.

Encontrará una visión general de la información de diagnóstico que ofrecen los diferentes esclavos en el apartado Componentes compatibles y funcionalidad.

Procedimiento básico



Los avisos se envían a través de la ruta de notificación estándar ALARM_S/SQ a "Avisos de CPU" en la PG o a los aparatos HMI conectados.

16.9.1 Componentes compatibles y su funcionalidad

Los componentes de los equipos S7-300, S7-400, esclavos DP y WinAC son compatibles con la herramienta "Notificar errores de sistema", siempre que soporten funciones tales como alarma de diagnóstico, alarma de presencia de módulo y diagnóstico específico de un canal.

Los siguientes componentes **no** son compatibles con "Notificar errores del sistema":

- Configuraciones M7, C7 y PROFIBUS-DP a través de interfases maestras DP (CP 342-5 DP) en equipos S7-300

En caso de un arranque normal hay que tener en cuenta, además, que pueden faltar avisos de alarma. Esto se debe a que la memoria de acuse de avisos de la CPU no se borra en el arranque normal, pero la herramienta "Notificar errores del sistema" restaura los datos internos.

En las dos tablas siguientes se indican todos los bloques de diagnóstico de los distintos esclavos PROFIBUS soportados por la función "Notificar errores del sistema".

Bloque de diagnóstico	De código (Slot erróneo)	De canal (error de canal) 1)	Estado del módulo (error de módulo, mód. erróneo/sin mód.	Denom. aparato
ID encabez.2)	0x01	0x10	0x00 Tipo 0x82	0x00 + 1 Byte info diag.
ET 200 S	Aviso: 'Consultar el diagnóstico'	Aviso en texto explícito	Aviso en texto explícito	- 3)
ET 200 M	no se evalúa	no se evalúa	no se evalúa	-
ET 200 X	Aviso: 'Consultar el diagnóstico'	-	-	-
ET 200 X DESINA	Aviso: 'Consultar el diagnóstico'	Aviso en texto explícito	Aviso en texto explícito	-
ET 200 L	no se evalúa	-	-	-
ET 200 B Digital	-	-	-	Aviso: 'Módulo defectuoso'
ET 200 B Analog	-	-	-	-
ET 200 C Digital	-	-	-	-
ET 200 C Analog	Aviso: 'Consultar el diagnóstico'	-	-	Aviso: 'Módulo defectuoso'
ET 200 U	Aviso: 'Consultar el diagnóstico'	-	-	Aviso: 'Módulo defectuoso'
ET 200 iS	Aviso: 'Consultar el diagnóstico'	Aviso en texto explícito	Aviso en texto explícito	-
ET 200 eco	-	-	-	Aviso en texto explícito
Enlace DP AS-i	Aviso: 'Consultar el diagnóstico'	-	Aviso en texto explícito	-

Bloque de diagnóstico	DS0/DS1 1)	otras variantes
ID encabezado 2)	0x00 Tipo 0x01	0x00 otros tipos
ET 200 S	-	-
ET 200 M	Aviso en texto explícito	no se evalúa
ET 200 X	-	-
ET 200 X DESINA	Aviso en texto explícito	-
ET 200 L	Aviso en texto explícito	-
ET 200 B Digital	-	-
ET 200 B Analog	Aviso en texto explícito	-
ET 200 C Analog	Aviso en texto explícito	-
ET 200 iS	Aviso en texto explícito	-
ET 200 eco	-	-
Enlace DP AS-i	Aviso: "Fallo de módulo"	-

En las tablas siguientes encontrará todos los bloques de diagnóstico de los diferentes esclavos PROFINET que son soportados por "Notificar errores de sistema":

Bloque de diagnóstico	De código (slot erróneo)	De canal (error de canal) 1)	Estado del módulo (error de módulo, módulo incorrecto/no hay módulo)	Nombre del dispositivo
ID encabezado 2)	0x01	0x10	0x00 tipo 0x82	0x00 + 1 byte de información de diagnóstico
ET 200 S	Aviso: "Hay diagnóstico"	Aviso con texto explícito	Aviso con texto explícito	-
SCALANCE Switches	-	-	Aviso con texto explícito	-

Bloque de diagnóstico	DS0/DS1 1)	sonst. Ausprägung
ID encabezado 2)	0x00 tipo 0x01	0x00 otro tipo
ET 200 S	-	-
SCALANCE Switches	Aviso con texto explícito	No se evalúa

1) DS0: Diagnóstico estándar, p. ej., fallo del módulo, falta tensión auxiliar externa o el conector frontal, tamaño 4 bytes, reside en los datos locales de OB 82.
 DS1: Error de canal, definido de forma distinta para cada tipo de canal, legible en el programa de usuario mediante SFC 51.
 Los textos provienen del diagnóstico HW S7.

2) ID encabez.: ID contenido en el telegrama de diagnóstico que identifica distintas partes del diagnóstico.

El telegrama de diagnóstico (también denominado telegrama de esclavo normalizado) está formado por los bloques de diagnóstico mencionados anteriormente y se puede leer en el programa de usuario a través de SFC 13.

En STEP 7 se visualiza el telegrama de diagnóstico a través de la llamada del estado del módulo en la ventana online "HW Config" (Diagnosticar hardware) en la ficha "Diagnóstico de esclavo DP" bajo "Representación hexadecimal".

Repetidor de diagnóstico: Los avisos del repetidor de diagnóstico se muestran en texto explícito. Los textos se leen del archivo GSD.

PROFINET

- En PROFINET IO, el diagnóstico de canal se indica en forma de texto explícito.
- ET 200S: se soportan esclavos con direcciones comprimidas.
- En los dispositivos PROFINET IO se soporta el diagnóstico específico del fabricante.

16.9.2 Ajustes para notificar errores del sistema

Dispone de varias posibilidades de abrir el diálogo para definir los ajustes:

- Seleccione en HW Config la CPU para la que desea configurar la notificación de errores del sistema. Elija a continuación el comando de menú **Herramientas > Notificar errores del sistema**.
- Si ya ha creado bloques para notificar errores del sistema, puede abrir el diálogo haciendo doble clic sobre uno de los bloques creados (FB, DB).
- En el diálogo de propiedades del equipo, seleccione la opción para abrir automáticamente al guardar y compilar la configuración.

Puede acceder a la opción para abrir automáticamente al guardar y compilar:

1. Seleccionando el equipo correspondiente en el Administrador SIMATIC.
2. Seleccionando el comando de menú **Edición > Propiedades del objeto**.
3. Seleccionando la ficha "Preferencias".

Nota

La ficha "Preferencias" del diálogo de propiedades también se puede abrir en HW Config con el comando de menú **Equipo > Propiedades**.

En el diálogo, introduzca:

- el FB y el DB de instancia asignado que debe crearse
- si deben generarse datos de referencia
- si se deben visualizar advertencias durante la generación de Notificar errores del sistema
- si debe mostrarse el diálogo si se llama automáticamente Notificar errores del sistema después de guardar y compilar la configuración (ver ajuste más arriba)
- si en el programa S7 se deben o no generar OBs de errores no existentes y en qué OBs debe ejecutarse la función "Notificar errores de sistema"
- el comportamiento de la CPU en caso de error: puede seleccionar qué tipos de errores harán que la CPU se ponga en STOP
- el aspecto de los avisos (estructura y disposición de los textos posibles)
- si los avisos deben ser acusables
- los parámetros que deben estar incluidos en el interface del bloque de usuario

Puede consultar informaciones detalladas en la Ayuda del diálogo abierto.

16.9.3 Generar bloques para la notificación de errores del sistema

Después de realizar los ajustes para notificar errores del sistema, pueden crearse los bloques necesarios (FB y DB, dependiendo de la configuración, también OBs aún no existentes). Para ello, haga clic en el botón de comando "Generar" del cuadro de diálogo "Notificar errores del sistema".

Se generarán los siguientes bloques:

- FB de diagnóstico (ajuste predeterminado: FB 49)
- DB de instancia para el FB de diagnóstico (ajuste predeterminado: DB 49)
- OBs de error (si se ha marcado esta opción en la ficha "Configuración de OBs")
- Bloque de usuario opcional que será llamado por el FB de diagnóstico

16.9.4 FB, DB generados

El FB generado evalúa los datos locales del OB de error, lee, dado el caso, las informaciones de diagnóstico adicionales de los componentes S7 que originan los errores y genera automáticamente el aviso correspondiente.

Sus propiedades son:

- Lenguaje SFM ("Notificar errores del sistema", también válido para el DB de instancia generado)
- Protección Know-How activada (también válido para el DB de instancia generado)
- Retarda las alarmas entrantes durante el tiempo de ejecución
- Mediante doble clic abre el diálogo para ajustar la función "Notificar errores del sistema" (también válido para el DB de instancia generado).

Bloque de usuario

Dado que el FB de diagnóstico tiene activada la protección Know-how, no se puede editar. El FB proporciona un interface para el programa de usuario, de manera que se tiene acceso, por ejemplo, al estado del error o al número de aviso.

El bloque para la evaluación en el programa de usuario (que puede ajustarse en la ficha "Bloque de usuario" del diálogo) se llama desde el FB generado con los parámetros seleccionados. Se pueden elegir los siguientes parámetros:

Nombre	Tipo de datos	Comentario
EV_C	BOOL	//aviso entrante (TRUE) o saliente (FALSE)
EV_ID	DWORD	//número de aviso generado
IO_Flag	BYTE	//módulo de entrada: B#16#54 módulo de salida: B#16#55
logAdr	WORD	//dirección lógica
TextlistId	WORD	//ID de la librería de textos (librería estándar = 1)
ErrorNo	WORD	//número de error generado
Channel_Error	BOOL	//error de canal (TRUE)
ChannelNo	WORD	//número de canal
ErrClass	WORD	//Clase de error
HErrClass	WORD	//Clase de error de sistemas H

Si el FB de usuario no existe lo generará SFM (=Notificar errores de sistema) con los parámetros seleccionados.

Los textos de error geneados para errores estándar responden al siguiente orden:

Número de error (decimal)		OB de error afectado	Código de error en el OB	
desde	hasta		desde	hasta
1	86	OB 72	B#16#1	B#16#56
162	163	OB 70	B#16#A2	B#16#A3
193	194	OB 72	B#16#C1	B#16#C2
224		OB 73	B#16#E0	
289	307	OB 81	B#16#21	B#16#33
513	540	OB 82		
865	900	OB 83	B#16#61	B#16#84
1729	1763	OB 86	B#16#C1	B#16#C8

Los números de error por encima de 12288 corresponden a errores de canal. Si visualiza el número de error como números hexadecimal, puede calacular el tipo de canal y detectar el bit de error. Debe consultar la descripción detallada en la ayuda correspondiente al módulo o al canal en cuestión.

Ejemplo:

12288 = W#16#3000 -> Highbyte 0x30 - 0x10 = Tipo de canal 0x20 (interfaz de CP);
 Lowbyte 0x00, es decir, bit de error 0

32774 = W#16#8006 -> Highbyte 0x80 - 0x10 = Tipo de canal 0x70 (introducción digital);
 Lowbyte 0x06, es decir bit de error 6

16.9.5 Crear textos de aviso en otros idiomas en 'Notificar errores de sistema'

Los avisos configurados en "Notificar errores de sistema" pueden visualizarse en los idiomas en los que ha instalado STEP 7.

Procedimiento:

1. Elija en el Administrador SIMATIC el comando de menú **Herramientas > Idioma para visualizadores...** y añada en el cuadro de diálogo que aparece el idioma deseado para el proyecto.
2. Confirme las entradas realizadas con Aceptar.
3. En HW Config elija el comando de menú **Herramientas > Notificar errores de sistema...** y haga clic en el botón "Generar" del cuadro de diálogo que aparece a continuación "Generieren".

Resultado: Los textos de aviso se generan en todos los idiomas en los que se haya instalado STEP 7, pero se visualizan solamente en el idioma que se haya definido como estándar en el cuadro de diálogo "Agregar y quitar idioma y ajustar idioma estándar" mediante un clic en el botón "Estándar".

Ejemplo

Ha instalado STEP 7 en alemán, inglés y francés y estos idiomas están definidos en su proyecto. Genere ahora los textos de aviso como se describe más arriba. Para mostrar los textos de aviso en el idioma en cuestión, defina el idioma deseado como idioma estándar en el cuadro de diálogo "Agregar y quitar idioma y ajustar idioma estándar".

17 Configurar variables para manejo y visualización

17.1 Configurar variables para manejo y visualización

Sinopsis

STEP 7 ofrece un método cómodo para manejar y visualizar magnitudes variables del proceso o del sistema de automatización con WinCC.

La ventaja frente a los métodos habituales es que los datos no se deben configurar individualmente para cada estación de operador (OS), sino una sólo vez con STEP7. Los datos creados con STEP 7 durante la configuración se pueden transferir mediante el programa 'AS-OS Engineering' (que forma parte del paquete de software "Sistema de control de procesos PCS7") a la gestión de datos de WinCC. En dicho proceso se comprueba la coherencia y la compatibilidad de los datos con el sistema de visualización. WinCC utiliza los datos en bloques de imágenes y en gráficos.

STEP 7 permite configurar y/o cambiar los atributos de manejo y visualización (M+V) de las siguientes variables:

- Parámetros de entrada, de salida y de entrada/salida de bloques de función
- Marcas y señales de E/S
- Parámetros de bloques CFC en esquemas CFC

Procedimiento general

El procedimiento para configurar variables que se puedan manejar y visualizar depende del lenguaje de programación o de configuración elegido, así como del tipo de variables que se deseen manejar y visualizar. No obstante, el procedimiento general comprende siempre los siguientes pasos:

1. Asigne atributos de manejo y visualización (M+V) a los parámetros de un bloque de función o a los símbolos de una tabla de símbolos.

Este paso no es necesario en CFC, puesto que los bloques se pueden extraer ya listos de una librería.

2. En un cuadro de edición, asigne a las variables que desea manejar y visualizar los atributos M+V necesarios (S7_m_c). En el cuadro de diálogo "Manejo y visualización" (comando de menú **Edición > Propiedades especiales del objeto > Manejo y visualización**) podrá modificar los atributos de WinCC tales como los límites, los valores de sustitución, las propiedades de los informes, etc.
3. Con el programa 'Ingeniería para PLC-OS', transfiera los datos de configuración creados con STEP 7 al sistema de visualización (WinCC).

Convenciones relativas a los nombres

Para que los datos de configuración se puedan guardar y transferir a WinCC, se almacenan con un nombre unívoco adjudicado automáticamente por STEP 7. Los nombres de las variables que se pueden manejar y visualizar, así como de los esquemas CFC y de los programas S7, forman parte de dicho nombre unívoco, por lo que se deben respetar determinadas convenciones:

- Los nombres de los programas S7 contenidos en un proyecto S7 deben ser unívocos (en diferentes equipos no puede haber programas S7 con un mismo nombre).
- Los nombres de las variables, de los programas S7 y de los esquemas CFC no pueden contener caracteres de subrayado, blancos o los siguientes caracteres especiales: ['] [.] [%] [-] [/] [*] [+].

17.2 Configurar atributos M+V con AWL, KOP y FUP

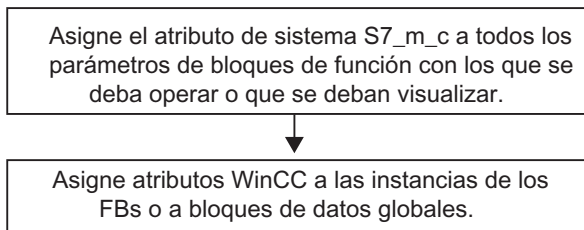
Introducción

Con el procedimiento aquí descrito es posible manejar y visualizar los parámetros de bloques de función y asignar los atributos M+V (para manejo y visualización) necesarios a los bloques de datos (DBs) de instancia o globales asociados en el programa de usuario.

Requisito

Se deberán haber creado previamente un proyecto STEP 7, un programa S7 y un FB.

Procedimiento general



17.3 Configurar atributos M+V a través de la tabla de símbolos

17.3.1 Configurar atributos M+V desde de la tabla de símbolos

Introducción

Independientemente del lenguaje de programación utilizado, con el procedimiento descrito abajo es posible configurar las siguientes variables:

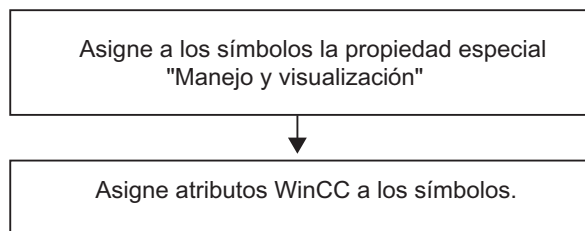
- Marcas
- Señales de E/S

Requisito

Antes de comenzar con la configuración se deberán cumplir los siguientes requisitos:

- Se deberá haber creado un proyecto en el Administrador SIMATIC.
- En dicho proyecto deberá existir un programa S7 con una tabla de símbolos.
- La tabla de símbolos deberá estar abierta.

Procedimiento general



17.4 Modificar atributos para manejo y visualización con CFC

Introducción

En CFC los programas de usuario se crean seleccionando de una librería los bloques destinados a manejo y visualización y posicionándolos e interconectándolos en un esquema.

Requisito

Antes será preciso insertar un programa S7 en un proyecto de STEP7, generar un esquema CFC y colocar allí los bloques.

Procedimiento general

Edite las propiedades de los bloques
(Propiedades del objeto)

Nota

Si utiliza bloques creados por usted y a los que ha asignado el atributo de sistema S7_m_c en el lenguaje de programación, puede asignar a dichos bloques los atributos M+V (para manejo y visualización) necesarios, activando la opción "Manejo y visualización" en el cuadro de diálogo "Manejo y visualización" (comando de menú **Edición > Propiedades especiales del objeto > Manejo y visualización**).

17.5 Transferir los datos de configuración al sistema de destino M+V

Introducción

Los datos de configuración para manejo y visualización se transfieren con el programa 'Configuración de enlaces PLC-OS' a la gestión de datos de WinCC.

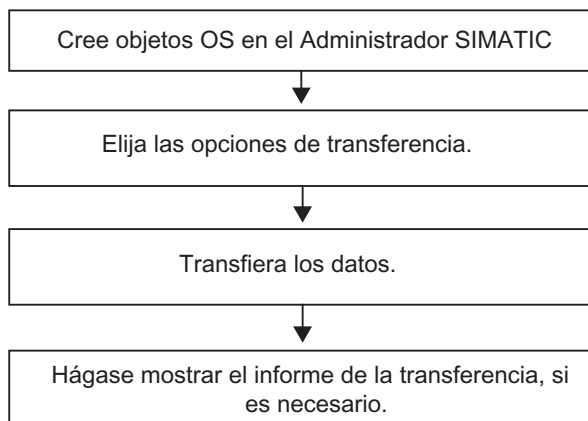
Requisito

Antes de transferir los datos se deberán cumplir los siguientes requisitos:

- Se deberá haber instalado el programa "Engineering PLC-OS".
- Se deberán haber generado los datos de configuración para manejo y visualización.

Procedimiento general

Para transferir los datos de configuración para manejo y visualización a la gestión de datos de WinCC proceda como sigue:



18 Establecer enlaces online y ajustar la CPU

18.1 Establecer enlaces online

Un enlace online entre el sistema de origen y el sistema de destino se requiere para cargar programas de usuario S7 y bloques del sistema de origen en el sistema de destino y para cargar bloques del sistema de destino S7 en el sistema de origen, así como para las siguientes actividades:

- Comprobar programas de usuario
- Visualizar y cambiar el estado operativo de la CPU
- Visualizar y ajustar la fecha y la hora de la CPU
- Visualizar el estado de los módulos
- Comparar bloques online/offline
- Diagnosticar el hardware

Para poder establecer un enlace online, el sistema de origen y el sistema de destino deberán estar comunicados mediante un interface apropiado (p.ej. interface multipunto (MPI)). Desde la ventana online del proyecto o desde la ventana "Estaciones accesibles" se podrá acceder entonces al sistema de destino.

18.1.1 Establecer un enlace online desde la ventana "Estaciones accesibles"

Esta alternativa le permite acceder rápidamente al sistema de destino, p.ej., para efectos de mantenimiento. Puede acceder a todos los módulos programables accesibles en la red. Elija este método si no existen datos de proyecto para los sistemas de destino en su PG.

A la ventana "Estaciones accesibles" se accede con el comando de menú **Sistema de destino > Estaciones accesibles**. El objeto "Estaciones accesibles" permite visualizar todas las estaciones accesibles por red con su respectiva dirección.

También pueden visualizarse estaciones no programables con STEP 7 (p.ej.: unidades de programación o paneles de operador).

La siguiente información adicional puede visualizarse entre paréntesis:

- (directa): esta estación está directamente conectada al sistema de origen (PG o PC).
- (pasiva): no es posible programar ni estado/forzar a través de PROFIBUS-DP en esta estación
- (en espera): no es posible la comunicación con esta estación porque su configuración no se adapta a los ajustes de la red.

Determinar las estaciones conectadas directamente

La información adicional "directa" no es compatible con las estaciones PROFINET. No obstante, para determinar las estaciones conectadas directamente puede utilizar el comando de menú **Sistema de destino > Diagnóstico/Configuración > Test de parpadeo de estaciones**.

En el cuadro de diálogo que aparece entonces puede ajustar la duración de parpadeo e iniciar el test. La estación conectada directamente se identifica por el hecho de que el LED FORCE parpadea.

El test de parpadeo no puede ejecutarse si la función FORCEN está activada.

18.1.2 Establecer un enlace online desde la ventana online del proyecto

Elija este método si ha configurado el sistema de destino en un proyecto en su PG/PC. Para acceder a la ventana online desde el Administrador SIMATIC, elija el comando de menú **Ver > Online**. Allí se indican los datos del proyecto contenidos en el sistema de destino (contrariamente a la ventana offline, donde aparecen los datos del proyecto contenidos en la PG o en el PC). En la ventana online se visualizan los datos tanto del programa S7 como del programa M7 que se encuentran depositados en el sistema de destino.

Esta vista se utiliza para acceder al sistema de destino. Algunas funciones del menú "Sistema de destino" del Administrador SIMATIC pueden llamarse en el modo de visualización "online", pero no en el modo "offline".

Se distingue entre:

- **Acceso con hardware configurado**

Así puede acceder sólo a los módulos configurados offline. De la dirección MPI que se haya ajustado al configurar el módulo programable depende a qué módulos es posible acceder online.

- **Acceso sin hardware configurado**

Para este tipo de acceso se requiere un programa S7 o un programa M7 que se haya creado independientemente del hardware (es decir, directamente bajo el proyecto). De la dirección MPI indicada en las propiedades del programa S7 o M7, respectivamente, depende a qué módulos es posible acceder online.

Este tipo de acceso combina los datos del sistema de destino con los del sistema de origen. Si, por ejemplo, abre online un bloque S7 bajo un proyecto, se muestra lo siguiente:

- área de código del bloque de la CPU del sistema de destino S7 y
- comentarios y símbolos de la gestión de datos del sistema de origen (siempre que existan offline). Si abre bloques directamente en la CPU conectada sin que exista una estructura del proyecto, se visualizan tal y como se encuentran en la CPU, es decir, sin símbolos ni comentarios.

18.1.3 Acceso online a sistemas de destino desde el multiproyecto

Accesos multiproyecto con PG/PC asignado

La función "Asignar PG/PC" para los objetos "PG/PC" y "Equipo PC SIMATIC" también está disponible en el multiproyecto.

El módulo de destino para el acceso online puede seleccionarse de un proyecto cualquiera del multiproyecto. El procedimiento es el mismo que para trabajar con un solo proyecto.

Requisitos

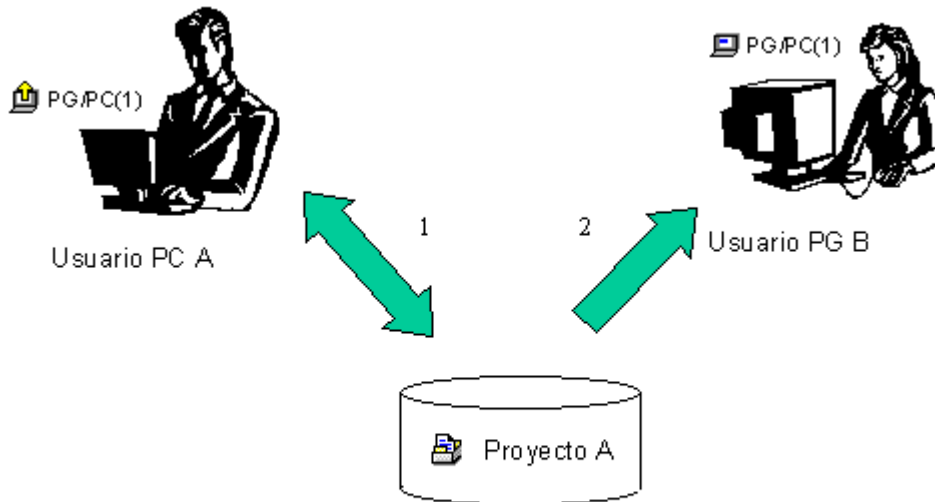
- Para PGs/PCs y equipos PC con los que deba accederse online a sistemas de destino, debe existir la asignación en un proyecto cualquiera dentro del multiproyecto.
Nota: cuando el proyecto en cuestión está abierto, la PG/el PC o el equipo PC asignado están resaltados en amarillo.
Una asignación de PG/PC sólo es visible cuando la asignación para la PG que abre el proyecto es correcta.
- Las subredes multiproyecto deben estar agrupadas.
- Todos los proyectos del multiproyecto deben estar compilados y la información de configuración debe estar cargada en los equipos participantes, de modo que, por ejemplo, esté disponible información routing de todos los módulos afectados para la configuración del enlace entre PG/PC y módulo de destino.
- El módulo de destino debe ser accesible por redes.

Problemas posibles en proyectos divididos

Si se cambian las asignaciones de los proyectos y un proyecto es abierto por una PG/un PC en la/el que no se ha sido creado, la asignación de PG/PC no es visible.

Con todo, el objeto configurado PG/PC todavía tiene la propiedad "asignado", aunque con la PG/el PC "erróneo".

En este caso, primero hay que eliminar la asignación existente y, posteriormente, asignar de nuevo el objeto PG/PC. A continuación, puede accederse online a los módulos accesibles dentro del multiproyecto sin problema alguno.



1. Guardar proyecto A con PG/PC asignado en la red
2. Abrir el mismo proyecto A con otro ordenador

Consejo para trabajar en proyectos divididos

Cuando varios usuarios acceden online a sistemas de destino con PGs, es aconsejable configurar en el multiproyecto un objeto "PG/PC" o "Equipo PC SIMATIC" para cada una de las PGs y, a continuación, crear una asignación para cada PG.

Según la PG desde la que se abra el proyecto se marcará con una flecha amarilla en el Administrador SIMATIC sólo el objeto asignado a la PG que abre.

18.1.4 Protección por contraseña para acceder a sistemas de destino

Con la contraseña es posible

- proteger el programa de usuario en la CPU y sus datos contra cambios no deseados (protección contra escritura)
- preservar el know how contenido en su programa de usuario (protección contra lectura)
- impedir la ejecución de funciones online que pudieran influir negativamente en el proceso

Para poder proteger un módulo o el contenido de la MMC (p. ej.: de una CPU 31xC) por contraseña es necesario que el módulo soporte dicha función.

Si desea proteger un módulo o el contenido de la MMC con una contraseña deberá definir el nivel de protección y la contraseña en la parametrización del módulo y cargar luego ésta última en el mismo.

Si se requiere una contraseña para poder ejecutar una función online o acceder al contenido de la MMC, se visualizará el cuadro de diálogo "Introducir contraseña". Introduciendo la contraseña debida se obtiene un permiso de acceso a los módulos para los cuales se ha definido un nivel de protección en la parametrización. Ello le permitirá establecer enlaces online con el módulo protegido y ejecutar las funciones online que corresponden al nivel de protección.

Con el comando de menú **Sistema de destino > Permiso de acceso > Crear** puede llamar directamente el cuadro de diálogo para introducir la contraseña. Así, p.ej., es posible introducir una vez la contraseña al principio de la sesión, con lo cual ya no se le pedirá que la introduzca cada vez que acceda online. La contraseña introducida permanecerá activa hasta que se vuelva a salir del Administrador SIMATIC o bien hasta ser anulada con el comando de menú **Sistema de destino > Permiso de acceso > Cancelar**.

Parámetros de la CPU	Notas
Test/Proceso (no para S7-400 ni CPU 318-2)	Configurable en la ficha "Protección". En el modo de proceso, las funciones de test tales como el estado del programa o la observación y el forzado de variables se limitan de manera tal que no se exceda el tiempo de ciclo admisible ajustado. Por ejemplo, en el estado de programa no se permiten las condiciones de llamada y la visualización del estado de un bucle programado se interrumpe en el punto de retorno. El test con puntos de parada y la ejecución del programa paso a paso no se pueden efectuar en el modo de proceso. En el modo Test se pueden utilizar ilimitadamente todas las funciones de test desde la PG o el PC, incluyendo las que pudieran prolongar en gran medida el tiempo de ciclo.
Nivel de protección	Configurable en la ficha "Protección". Los accesos de escritura y/o de lectura a la CPU pueden someterse a la previa introducción de una contraseña. Dicha contraseña se parametriza en esa ficha.

18.1.5 Nota respecto a la actualización del contenido de ventanas

Tenga en cuenta lo siguiente:

- Los cambios en la ventana online de un proyecto debidos a acciones efectuadas por el usuario (p.ej. cargar o borrar bloques) no se adoptan automáticamente en la ventana "Estaciones accesibles" que pudiese estar abierta.
- Los cambios que se hagan en la ventana "Estaciones accesibles" tampoco se adoptan automáticamente en una ventana online de un proyecto que pudiese estar abierta.

Con objeto de adoptar la visualización actual en otra ventana abierta simultáneamente, ésta se deberá actualizar también (mediante el correspondiente comando de menú o la tecla de función F5).

18.2 Visualizar y cambiar el estado operativo

Utilizando esta función puede cambiar la CPU nuevamente al estado operativo "RUN" tras haber corregido un error.

Visualizar el estado operativo

1. Abra su proyecto y seleccione un programa S7 o M7, o bien elija el comando de menú **Sistema de destino > Mostrar estaciones accesibles** para abrir la ventana "Estaciones accesibles" y seleccione una estación ("MPI=...").
2. Elija el comando Sistema de destino > Diagnóstico/Preferencias > Estado operativo.

Este cuadro de diálogo muestra el estado operativo actual y el último, así como la posición actual del selector del módulo. Para los módulos cuya posición actual de selector no pueda ser leída aparece el texto "sin definir".

Cambiar el estado operativo

Mediante los botones de comando puede cambiar el estado operativo del módulo CPU. Sólo estarán activos los botones de comando que se puedan seleccionar en el estado operativo actual.

18.3 Visualizar y ajustar la fecha y la hora

18.3.1 Relojes de la CPU con ajuste de zonas horarias y horario de verano/invierno

A partir de STEP 7 V5.1, Servicepack 2, en las nuevas CPUs (a partir de la versión de firmware 3) se pueden realizar y evaluar los siguientes ajustes, además de los de fecha y hora:

- Horario de verano/invierno
- Factor de corrección para la representación de zonas horarias

Representación de zonas horarias

Para toda la instalación existe una sola hora que avanza sin interrupción: la hora del módulo

En el sistema de automatización se puede calcular localmente una hora adicional diferente de la hora del módulo para utilizarla en el programa de usuario. La hora local no se introduce directamente, sino que se calcula a partir de la hora del módulo, sumándole o restándole un factor de corrección (desfase de la hora local con respecto a la hora del módulo).

Horario de verano/invierno

Al ajustar la hora y la fecha también se puede ajustar el horario de verano y de invierno. Al cambiar del horario de verano al de invierno, por ejemplo, por programa de usuario, esto sólo se considerará en el desfase con respecto a la hora del módulo. Puede realizar este cambio en el programa de usuario mediante un bloque que está disponible en Internet.

Lectura y ajuste de la hora y del estado horario

El estado horario contiene el código de la horario de verano y de invierno y el desfase con respecto a la hora del módulo.

Existen las siguientes posibilidades para leer y ajustar la hora y el estado horario:

Con STEP 7 (online)

- Mediante el comando de menú **Sistema de destino > Diagnóstico/Preferencias > Ajustar la hora** (leer y ajustar)
- Mediante el diálogo "Información del módulo", ficha "Sistema de reloj" (sólo leer)

En el programa de usuario

- SFC 100 "SET_CLKS" (leer y ajustar)
- SFC 51 "RDSYSST" con SZL 132, índice 8 (sólo leer)

Indicación de fecha y hora en el búfer de diagnóstico, avisos e Información de arranque de OB

Las indicaciones de fecha y hora se generan con la hora del módulo.

Alarma horaria

Si al cambiar del horario de invierno al de verano no se ha disparado la alarma horaria por este "salto temporal", se ejecutará el OB 80.

Al cambiar del horario de verano al de invierno, en las alarmas horarias con periodicidad de cada minuto y de cada hora se conserva la periodicidad.

Sincronización horaria

Una CPU parametrizada como reloj maestro (p. ej., en la ficha de la CPU "Diagnóstico / Reloj") sincroniza los demás relojes con la hora del módulo y su estado horario actual.

18.4 Actualizar el firmware

18.4.1 Actualizar el firmware de módulos y submódulos online

A partir de STEP 7 V5.1, Servicepack 3, podrá actualizar online de forma unificada módulos o submódulos de un equipo. A continuación se describe el procedimiento.

Concepto

Para actualizar el firmware de un módulo, como p.ej. una CPU, un CP o una IM, dispone de ficheros con la extensión *.UPD con el firmware actualizado.

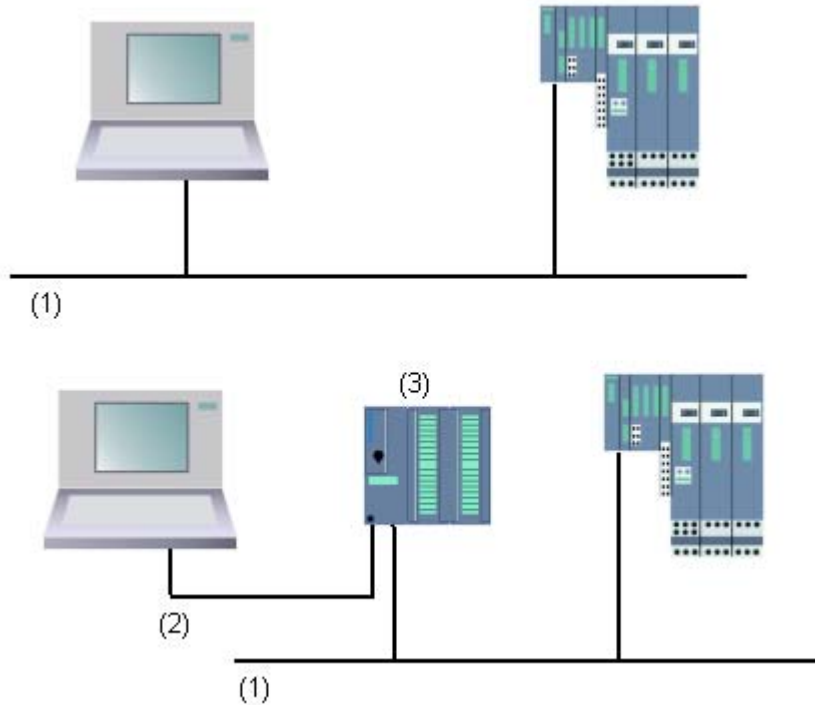
Seleccione uno de estos archivos y cárguelo en el módulo (menú Sistema de destino).

Requisitos

El módulo del equipo cuyo firmware desee actualizar deberá estar disponibles online, por ejemplo cuando el PG está conectado al mismo PROFIBUS o Ethernet que el módulo cuyo Firmware desee actualizar. También se puede actualizar el firmware, si el PG está conectado a la interfaz MPI de la CPU del maestro DP y el módulo, cuyo firmware desea actualizar, está conectado a la PROFIBUS de la interfaz DP o a la Ethernet de la interfaz PN. Además, la CPU deberá soportar el routing S7 entre las interfaces MPI y DP, así como entre las interfaces MPI y PN.

El módulo también deberá ser compatible con la versión actualizada del firmware.

Los archivos con las versiones de firmware actualizadas deberán estar disponibles en el sistema de archivos de su PG o PC. En una carpeta sólo puede haber archivos de **una** versión de firmware.



(1) Subred PROFIBUS o Ethernet

(2) Subred MPI

(3) CPU con interfaz MPI y DP o interfaz PN (con routing S7)

Procedimiento en HW Config

1. Abra el equipo con el módulo que desee actualizar.
2. Marque el módulo.
Si trabaja con interfaces DP PROFIBUS, como p.ej. IM 151, marque el símbolo del esclavo DP, en este caso el ET 200S correspondiente.
3. Seleccione el comando de menú **Sistema de destino > Actualizar firmware**. Este comando de menú sólo puede activarse si el módulo o esclavo DP marcado soporta la función "Actualizar firmware".
4. Aparecerá el cuadro de diálogo "Actualizar firmware". Seleccione con el botón "Examinar" la ruta para los archivos de actualización de firmware (*.UPD).
5. Una vez haya seleccionado un archivo, en los campos inferiores del cuadro de diálogo "Actualizar firmware" aparecerá la información indicando para qué módulos resulta indicado el archivo y a partir de qué versión de firmware.

- Haga clic en el botón "Ejecutar".
STEP 7 comprobará si el módulo puede interpretar el archivo seleccionado y, en caso afirmativo, cargará el fichero en el módulo.
En caso de que para hacerlo se tenga que modificar el estado operativo de la CPU, aparecerán cuadros de diálogo que le solicitarán dicha modificación.
A continuación, el módulo ejecutará la actualización de firmware de forma independiente.
Nota: Para la actualización de firmware p.ej. en una CPU 317-2 PN/DP generalmente se establece un enlace propio con la CPU. En ese caso el proceso no puede interrumpirse. Si no hay recursos disponibles para otro enlace, se utiliza automáticamente un enlace ya existente. En ese caso el proceso no puede interrumpirse, el botón "Cancelar" del cuadro de diálogo de transferencia aparece en gris y no puede utilizarse.
- Compruebe con STEP 7 (mediante la lectura del búfer de diagnóstico de la CPU) si el módulo funciona correctamente con el nuevo firmware.

Procedimiento en SIMATIC Manager

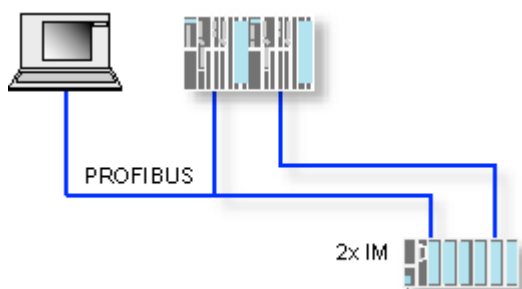
La función puede activarse si se abre el cuadro de diálogo "Estaciones accesibles". El procedimiento a seguir es idéntico al de HW Config y el comando de menú se denomina igualmente **Sistema de destino > Actualizar firmware**. Sin embargo, en el momento de la ejecución STEP 7 comprueba si el módulo soporta esta función.

Actualización del firmware de módulos en funcionamiento redundante

A partir de STEP 7 V5.4 se soporta la actualización del firmware de módulos en funcionamiento redundante, p. ej. del IM 153-2BA00 con bus posterior activo en un equipo H. Para los IM redundantes se puede llevar a cabo una actualización de firmware de una pasada; el IM redundante recibe automáticamente la nueva versión de firmware.

Requisitos: La PG debe estar conectada al mismo PROFIBUS que uno de los IMs y la actualización de firmware debe realizarse en el SIMATIC Manager desde "Estaciones accesibles".

Principio



Consecuencias de la actualización del firmware durante el funcionamiento

En el cuadro de diálogo "Actualizar firmware" hay una opción que permite determinar que el nuevo firmware se active directamente después de la actualización.

Si esta opción está marcada, el equipo ejecuta un re arranque como después de una desconexión/conexión (POWER OFF/POWER ON). En ese caso una CPU puede p.ej. permanecer en estado operativo STOP o el desarrollo del programa de usuario puede verse afectado. Debe tomar las medidas oportunas para estas limitaciones en el funcionamiento.

En un re arranque fallan p.ej. todos los módulos del equipo, incluso la periferia F que pueda haber.

En caso de desconexión (POWER OFF), la periferia F comunica un error de comunicación al módulo de interfaz y se desconecta orientada a la seguridad, es decir, se pasiva. Esta pasivación no se elimina reiniciando el módulo de interfaz. Cada módulo debe depasivarse individualmente y esto provoca que las aplicaciones orientadas a la seguridad no funcionen.

19 Cargar

19.1 Cargar en el sistema de destino desde la PG

19.1.1 Requisitos para cargar

Requisitos para cargar en el sistema de destino

- Haber establecido un enlace entre la PG y la CPU del sistema de destino (p.ej., vía el interface MPI).
- Poder acceder al sistema de destino.
- Al cargar desde bloques al sistema de destino, en las propiedades del objeto del proyecto debe haber seleccionado la entrada "STEP 7" para "Utilización".
- El programa que se desea cargar se deberá haber compilado sin errores.
- La CPU se tiene que encontrar en un estado operativo en el que se pueda cargar (STOP o RUN-P).
No obstante, si se cargan en modo RUN-P, el programa se transferirá de bloque en bloque. Si sobrescribe un antiguo programa de CPU, pueden aparecer conflictos, p.ej., si se han modificado parámetros de bloques. Entonces, al procesarse el ciclo, la CPU pasa al estado operativo STOP. Es por ello que se recomienda cambiar la CPU a "STOP" antes de cargar.
- Caso que haya abierto el bloque offline y lo desee cargar, el módulo CPU tendrá que estar asignado a un programa de usuario online en el Administrador SIMATIC.
- Antes de cargar el programa de usuario en la CPU es recomendable efectuar un borrado total de la misma para evitar que contenga bloques "antiguos".

Estado operativo STOP

Cambie el estado operativo de RUN a STOP antes de

- cargar todo el programa de usuario o partes del mismo en la CPU,
- activar el borrado total de la CPU,
- comprimir la memoria de usuario.

Rearranque completo (en caliente) (cambio al estado operativo RUN)

Si realiza un reارئانque completo (en caliente) partiendo del modo "STOP", se reinicia el programa arrancando primero en modo "ARRANQUE" (en el bloque OB100). Si el arranque ha tenido éxito, la CPU cambia a modo RUN. Un reارئانque completo (en caliente) es necesario después de:

- efectuar el borrado total de la CPU,
- cargar el programa de usuario en el estado STOP,

19.1.2 Diferencia entre guardar y cargar bloques

Por principio, se diferencia entre guardar y cargar bloques.

	Guardar	Cargar
Comandos de menú	Archivo > Guardar Archivo > Guardar como	Sistema de destino > Cargar
Función	El estado actual del bloque en el editor se guarda en el disco duro de la PG.	El estado actual del bloque en el editor se carga sólo en la CPU.
Comprobación de sintaxis	Se comprueba la sintaxis. En caso de detectarse errores, éstos se visualizarán en cuadros de diálogo donde se indican la causa y el lugar de los errores. Dichos errores deberán corregirse antes de guardar o cargar el bloque. Si la sintaxis es correcta, el bloque se compila a continuación en código máquina y se guarda o se carga.	Se comprueba la sintaxis. En caso de detectarse errores, éstos se visualizarán en cuadros de diálogo donde se indican la causa y el lugar de los errores. Dichos errores deberán corregirse antes de guardar o cargar el bloque. Si la sintaxis es correcta, el bloque se compila a continuación en código máquina y se guarda o se carga.

La tabla será válida independiente de si el bloque se ha abierto online u offline.

Nota respecto a los cambios en bloques - guardar primero y cargar luego

Para adoptar bloques nuevos o bien los cambios realizados en el área de instrucciones de bloques lógicos, en tablas de declaración, o bien, en valores de bloques de datos, deberá guardar el bloque correspondiente. Los cambios que se efectúen en el editor y que se transfieran a la CPU con el comando de menú **Sistema de destino > Cargar** – p.ej., para comprobar cambios pequeños – se deberán guardar en todo caso también en el disco duro de la PG antes de salir del editor. En caso contrario, el estado del programa de usuario en la CPU será diferente de la versión contenida en la PG. En general, es aconsejable guardar primero los cambios y cargar a continuación.

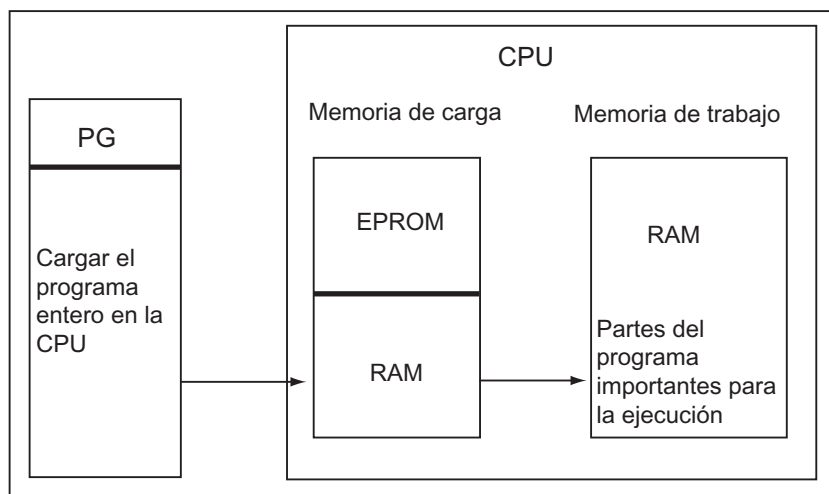
19.1.3 Memorias de carga y de trabajo de la CPU

Una vez concluida la configuración, la parametrización y la creación de programas, así como el establecimiento del enlace online, es posible transferir programas de usuario enteros o bloques individuales a un sistema de destino. Para comprobar bloques individuales es preciso cargar por lo menos un OB, así como los FBs y las FCs llamados por dicho OB, así como los DBs utilizados. Para poder transferir al sistema de destino los datos de sistema que se han generado después de configurar el hardware y las redes, o bien, después de haber creado una tabla de enlaces, hay que cargar el objeto "Bloques de datos del sistema".

Utilizando el Administrador SIMATIC, los programas de usuario se cargan en un sistema de destino, p.ej. en la fase final de la comprobación del programa o para ejecutar el programa de usuario ya listo.

Interacción de las memorias de carga y de trabajo en la CPU

El programa de usuario entero se transfiere a la memoria de carga, en tanto que las partes ejecutables del mismo se cargan también en la memoria de trabajo.



Memoria de carga de la CPU

- En la memoria de carga se guarda el programa de usuario sin la tabla de símbolos y sin comentarios (éstos permanecen en el área de memoria de la PG).
- Los bloques caracterizados como no relevantes para la ejecución se memorizan exclusivamente en la memoria de carga.
- La memoria de carga puede ser una RAM, una ROM o una EPROM.
- En el caso de los S7-300 la memoria de carga puede integrar una parte de memoria RAM y una parte de memoria EEPROM (p. ej., CPU312 IFM y CPU314 IFM).
- En el caso de los S7-400 para poder ampliar la memoria de usuario es imprescindible utilizar una Memory Card (RAM o EEPROM).

Memoria de trabajo de la CPU

En la memoria de trabajo (RAM integrada) se guardan las partes ejecutables del programa de usuario.

Procedimientos posibles al cargar

El programa de usuario o los objetos cargables (p.ej. los bloques) se pueden cargar en el sistema de destino mediante la función de carga. Si un bloque ya existe en la RAM de la CPU, se le consulta, antes del proceso de carga, si desea sobrescribirlo o no.

- Es posible seleccionar los objetos mencionados en la ventana del proyecto y cargarlos desde el Administrador SIMATIC (comando: **Sistema de destino > Cargar**).
- Al configurar el hardware y las redes, así como al programar bloques, el objeto recién editado se puede cargar directamente mediante el menú de la ventana principal correspondiente a la actividad en cuestión (con el comando **Sistema de destino > Cargar**).
- Otra posibilidad consiste en abrir una ventana online con vista al sistema de destino (p.ej. utilizando los comandos de menú **Ver > Online** o **Sistema de destino > Estaciones accesibles**) y copiar luego en la ventana online el objeto a cargar.

Con esta función también se puede cargar en la PG el contenido actual de los bloques de la memoria de carga RAM de la CPU.

19.1.4 Procedimientos de carga en función de la memoria de carga

La división de la memoria de carga en un área RAM y un área EEPROM tiene consecuencias al cargar el programa de usuario o bloques del mismo. Para cargar los datos en la CPU se dispone de las siguientes posibilidades:

Memoria de carga	Posibilidades de carga	Tipo de comunicación entre los sistemas de origen y de destino
RAM	Cargar y borrar bloques individuales	Enlace online PG - sistema de destino
	Cargar y borrar un programa de usuario entero	Enlace online PG - sistema de destino
	Recargar bloques individuales	Enlace online PG - sistema de destino
EPROM integrada (sólo en el S7-300) o enchufable	Cargar programas de usuario enteros	Enlace online PG - sistema de destino
EPROM enchufable	Cargar programas de usuario enteros	Cargar la EPROM externamente e insertar la Memory Card o bien acceder online a la EPROM que está insertada en el sistema de destino.

Cargar en la RAM mediante un enlace online

Al producirse un corte de alimentación, los datos no se guardan en el sistema de destino si la RAM no tiene respaldo. En este caso, se pierden los datos contenidos en la RAM.

Guardar en una Memory Card EPROM

Los bloques o el programa de usuario se guardan en una Memory Card EPROM que se enchufa luego en una ranura de la CPU.

Las Memory Cards son soportes de datos portátiles. Se escriben en el sistema de origen, siendo enchufadas luego en el correspondiente receptáculo de la CPU.

Los datos que se guardan allí se conservan aún en caso de un corte de alimentación o de un borrado total de la CPU. Tras un borrado total de la CPU o a un fallo de la alimentación, si la RAM no tiene respaldo, el contenido de la EPROM se copia nuevamente en el área RAM de la memoria de la CPU cuando retorna la alimentación.

Guardar en la EPROM integrada

En el caso de la CPU 312 existe también la posibilidad de copiar el contenido de la RAM en la EPROM integrada. Los datos contenidos en dicha EPROM permanecen remanentes aunque se produzca un corte de la alimentación. Tras un borrado total de la CPU o un corte de alimentación, si la RAM no tiene respaldo, el contenido de la EPROM integrada se copia nuevamente en el área RAM de la memoria de la CPU al retornar la alimentación.

19.1.5 Cargar el programa en una CPU S7

19.1.5.1 Cargar con gestión de proyectos

1. Seleccione en la ventana del proyecto el programa de usuario o los bloques que desea cargar.
2. Cargue los objetos seleccionados en el sistema de destino eligiendo el comando **Sistema de destino > Cargar**.

Alternativa (con "arrastrar y soltar")

1. Abra en su proyecto una ventana offline y otra online.
2. Seleccione en la ventana offline los objetos a cargar y arrástrelos hasta la ventana online.

19.1.5.2 Cargar sin gestión de proyectos

1. Abra la ventana "Estaciones accesibles" con el comando **Sistema de destino > Estaciones accesibles** o haciendo clic en el botón correspondiente de la barra de herramientas.
2. En la ventana "Estaciones accesibles" haga doble clic en las estaciones correspondientes ("MPI=..."), para ver la carpeta "Bloques".
3. Abra la librería o el proyecto de donde desea cargar el programa de usuario o los bloques en el sistema de destino. Para ello elija el comando **Archivo > Abrir**.
4. Seleccione en la ventana abierta del proyecto o de la librería los objetos que desea cargar.
5. Cargue los objetos en el sistema de destino, copiándolos con arrastrar y soltar (drag&drop), en la carpeta "Bloques" de la ventana "Estaciones accesibles".

19.1.5.3 Recargar bloques en el sistema de destino

Puede sobrescribir los bloques existentes en la memoria de carga (RAM) o en la memoria de trabajo de la CPU del sistema de destino S7. La nueva versión del bloque reemplazará el contenido anterior.

El procedimiento de sobrescritura es similar al de cargar bloques S7. Sólo se le pregunta si desea existir sobrescribir el bloque existente.

No es posible borrar un bloque que esté guardado en la EPROM, pero éste se declara no válido durante la función de sobrescritura. El bloque que lo sustituye se carga en la RAM. Así se crean vacíos en la memoria de carga o en la de trabajo. Si, a causa de dichos vacíos, no es posible cargar más bloques nuevos, se deberá comprimir la memoria.

Nota

Cuando retorne la tensión después de un corte de alimentación, si la RAM no tiene respaldo, o bien, después de un borrado total de la CPU, los "antiguos" bloques de la EPROM serán los que se admitan como válidos y se cargarán.

19.1.5.4 Guardar en la EPROM integrada los bloques cargados

En los módulos CPU que poseen una EPROM integrada (p.ej., CPU 312) es posible guardar bloques de la RAM en la EPROM integrada para que los datos no se pierdan en caso de un corte de alimentación o de un borrado total.

1. Elija el comando de menú **Ver > Online** para visualizar una ventana con la vista online de un proyecto abierto o acceda a la ventana "Estaciones accesibles", haciendo clic en el botón "Estaciones accesibles" en la barra de herramientas, o bien, con el comando de menú **Sistema de destino > Estaciones accesibles**.
2. Seleccione el programa S7 o M7 en la vista online de la ventana del proyecto, o bien, la correspondiente estación en la ventana "Estaciones accesibles".
3. Seleccione la carpeta "Bloques" que desea guardar de la CPU:
 - en la ventana online del proyecto, si desea trabajar con gestión de proyectos.
 - en la ventana "Estaciones accesibles", si trabaja sin gestión de proyectos.
4. Elija el comando Sistema de destino > Copiar RAM en ROM.

19.1.5.5 Cargar mediante Memory Cards EPROM

Requisitos

Para poder acceder en el sistema de origen a Memory Cards EPROM previstas para un sistema de destino S7, se necesitan los correspondientes drivers EPROM. Para poder acceder a Memory Cards EPROM previstas para un sistema de destino M7 se deberá haber instalado el Flash File System (ésto sólo es posible en las PGs 720, 740 y 760, Field PG y Power PG). Los drivers EPROM y el Flash File System se ofrecen opcionalmente al instalar el paquete básico de STEP 7. Si utiliza un PC, para guardar en la Memory Cards EPROM necesita además un Prommer externo.

Los drivers también se pueden instalar posteriormente. A tal efecto, abra el cuadro de diálogo correspondiente, partiendo de la barra de inicio (**Inicio > Simatic > STEP 7 > Parametrizar Memory Card**) o del Panel de control (haciendo doble clic en el icono "Parameterizar Memory Card").

Guardar en una Memory Card

Para guardar bloques o programas de usuario en una Memory Card, proceda de la forma siguiente:

1. Enchufe la Memory Card (tarjeta de memoria en formato de tarjeta de crédito) en la correspondiente ranura del sistema de origen.
2. Abra la ventana "Memory Card S7"
 - haciendo clic en el botón de las "Memory Cards S7" de la barra de herramientas. En caso necesario, active la visualización de la misma con el comando Ver > Barra de herramientas o
 - Eligiendo el comando de menú Archivo > Memory Card S7 > Abrir.
3. Abra o active una ventana visualizando los bloques que desea guardar. Puede escoger entre una de las ventanas siguientes:
 - ventana de proyecto, vista "ONLINE"
 - ventana de proyecto, vista "OFFLINE"
 - ventana de librería
 - ventana "Estaciones accesibles"
4. Seleccione la carpeta "Bloques" o los bloques que desea guardar y cópielo(s) en la ventana "Memory Card S7".
5. Si en la Memory Card ya existe un bloque, aparecerá un mensaje de error. En este caso, borre el contenido de la misma y repita el proceso a partir del 2º paso.

19.2 Compilar varios objetos de la PG y cargar

19.2.1 Requisitos y observaciones para cargar

Cargar carpetas de bloques

Con las carpetas de bloques sólo pueden cargarse bloques lógicos. Los demás objetos de la carpeta de bloques, como datos del sistema (SDBs), etc. no pueden cargarse en este punto. Los SDBs se cargan a través del objeto "Hardware".

Nota

En proyectos PCS 7 no es posible cargar bloques desde el cuadro de diálogo "Compilar objetos y cargar", como tampoco lo es desde el Administrador SIMATIC. Para proyectos PCS 7 recuerde que los sistemas de destino sólo se cargan a través CFC para garantizar un orden de carga correcto y evitar así una parada de la CPU.

Puede consultar si se trata de un proyecto PCS 7 en las propiedades del objeto del proyecto.

Cargar partes F de autómatas seguros

Por motivos de seguridad, las partes F de autómatas seguros sólo pueden cargarse después de entrar una contraseña. Por este motivo, en "Compilar objetos y cargar" el proceso de carga se interrumpe con un aviso de error. Cargue las partes correspondientes del programa en el sistema de destino con el paquete opcional.

Cargar el hardware

La carga del hardware (es decir, la carga de los SDBs offline) a través de "Compilar objetos y cargar" sólo funciona sin interrupciones para todos los objetos marcados cuando no se emiten avisos de error o consultas. El apartado siguiente ofrece información sobre cómo evitar avisos de error y consultas.

Requisitos para cargar el hardware

- Las CPUs deben estar en estado operativo STOP.
- Los enlaces online con las CPUs deben ser posibles; las CPUs protegidas con contraseña requieren un enlace legitimado o la entrada de la contraseña con el botón "Edición" con la CPU marcada o con la carpeta de bloques marcada antes de iniciar "Compilar objetos y cargar".
- La interfaz del sistema de destino a través de la cual se realiza la carga no puede reparametrizarse a fondo durante la carga:
 - La dirección de la interfaz no puede modificarse.
 - Si se modifican los ajustes de la red es posible que no pueda accederse a todos los módulos.
- En las CPUs H puede seleccionar la CPU que debe cargarse CPU H 0 o CPU H 1) antes de iniciar "Compilar objetos y cargar" (marque el objeto "CPU" y haga clic en el botón "Edición").

- Los siguientes parámetros de la CPU no pueden haber cambiado:
 - El tamaño máximo de los datos locales y los recursos de comunicación de la CPU (ficha "Memoria").
 - La protección con contraseña de la CPU F (ficha "Protección").
- Para cada módulo configurado deben cumplirse las condiciones siguientes:
 - La referencia del módulo configurado debe ser idéntica a la referencia del módulo insertado realmente.
 - La versión de firmware del módulo configurado no puede ser superior a la versión de firmware del módulo insertado realmente.
 - El nombre del equipo, el nombre del módulo y el código de subsistema no pueden haber cambiado desde la última carga. Está permitido asignar un nuevo código de subsistema.

Observaciones sobre el proceso de carga

- Se cargan todos los SDBs offline (es decir, además de la configuración del hardware, los SDBs de enlace y los SDBs que han surgido mediante la configuración de datos globales).
- La carga sólo se ejecuta si no se ha producido ningún error en la compilación previa.
- Durante la carga no aparecen indicaciones; si hubiera escasez de memoria en la CPU, se comprimirá sin previa consulta.
- Tras la carga, los módulos cargados se mantienen en estado STOP (excepto los módulos que se detuvieron y reiniciaron automáticamente sin previa consulta).

Consejo

Si después de cargar aparece el mensaje que indica que el objeto se ha cargado con advertencias, debería consultar el informe. Puede que el objeto no se haya cargado o no se haya cargado completamente.

19.2.2 Compilar objetos y cargar

En el cuadro de diálogo "Compilar objetos y cargar" se preparan los objetos seleccionables de su proyecto o multiproyecto para transferirlos al sistema de destino y, posteriormente, se cargan al sistema de destino de acuerdo con sus necesidades. El cuadro de diálogo puede aplicarse a objetos de un equipo, de un proyecto o de un multiproyecto.

Dependiendo del objeto seleccionado algunas informaciones determinadas no pueden mostrarse o para esos objetos no están disponibles todas las funciones abajo descritas. Especialmente pueden aparecer limitaciones con objetos generados con paquetes opcionales.

En bloques de una carpeta de bloques, "compilar" significa que se comprueba la coherencia de los bloques. Para facilitar la comprensión, la comprobación de la coherencia de bloques se describe a continuación como proceso de compilación.

Proceda de la forma siguiente:

1. En el Administrador SIMATIC, seleccione el objeto que desea compilar o compilar y cargar. En el administrador SIMATIC puede seleccionar los siguientes objetos:
 - Multiproyecto
 - Proyecto
 - Equipo
 - Programa S7 sin asignación de equipo
2. En el Administrador SIMATIC seleccione el comando de menú **Sistema de destino > Compilar objetos y cargar**.
3. Seleccione "Sólo compilar" si desea comprobar los bloques pero no cargarlos en el sistema de destino. Utilice esta opción si no desea cargar ninguno de los objetos en el sistema de destino.
4. Si desea impedir que los equipos se carguen completamente por fallos de compilación, active la casilla de verificación "No cargar si hay errores de compilación. En ese caso no se cargará nada.
Si la casilla de verificación no está activada, se cargarán todos los objetos compilados sin ningún error. Los objetos que han provocado un error de compilación no se cargan.
5. Si desea compilar y cargar enlaces, active las casillas de verificación correspondientes en el objeto "Enlaces".
6. Especialmente el multiproyecto es adecuado como objeto de acceso ya que desde ese objeto pueden cargarse también todos los interlocutores de enlaces de varios proyectos automáticamente.
7. En las columnas "Compilar" y "Cargar", seleccione los objetos que desea compilar o cargar. Una marca de verificación indica que el objeto está seleccionado. Si en el paso 3 ha seleccionado "Sólo compilar", la columna "Cargar" estará oculta.
8. Inicie el proceso de compilación con el botón "Iniciar".
9. Siga las instrucciones que aparecen en la pantalla.

Después del proceso de compilación y carga se muestra automáticamente un informe completo. Sin embargo, el informe completo o individual se puede abrir en cualquier momento:

- Haga clic en el botón "Total", si desea ver un informe completo del proceso.
- Haga clic en el botón "Objeto individual", si desea ver un informe del objeto seleccionado en la lista.

Particularidades de la compilación y carga de enlaces

Si en un módulo se marca el objeto "Enlaces" para la **compilación**, entonces STEP 7 marcará automáticamente los correspondientes objetos "Enlaces" de los interlocutores. De este modo, STEP 7 generará siempre datos de configuración coherentes (bloques de datos de sistema). Los objetos marcados automáticamente no se pueden deseleccionar manualmente de forma directa. Sin embargo, la marca se borrará automáticamente cuando se vuelva a deseleccionar el objeto "Enlaces" que se había marcado.

Si en un módulo se marca el objeto "Enlaces" para la **carga**, entonces STEP 7 también marcará automáticamente la casilla de verificación "Compilar". STEP 7 marcará además las casillas de verificación "Compilar" y "Cargar" de todos los interlocutores. Si solamente han sido marcados objetos del tipo "Enlaces", podrá cargar los enlaces también en el estado operativo RUN-P de la CPU.

Puede cargar los diferentes enlaces con ayuda de NetPro.

Compilar y cargar hardware: efectos sobre los enlaces

Si marca el objeto "Hardware" para la compilación o carga, automáticamente se marcarán los objetos "Enlaces" para la compilación o carga bajo el hardware marcado. Sin embargo, en este caso **no** se marcarán automáticamente los objetos de enlace de los interlocutores.

19.3 Cargar en la PG desde el sistema de destino

Esta función le asiste en las siguientes acciones:

- Guardar informaciones del sistema de destino (p.ej. para fines de mantenimiento)
- Configuración y edición rápida de un equipo cuando los componentes del hardware ya están disponibles al empezar a configurar.

Salvaguardar informaciones del sistema de destino

Esta medida puede ser necesaria, p. ej. cuando los datos del proyecto offline de la versión que se esté ejecutando en la CPU no existan o existan sólo en parte. En este caso puede acceder por lo menos a los aquella parte de los datos del proyecto a la que se puede acceder online en su PG.

Configuración rápida

Para editar con mayor facilidad la configuración de su equipo puede cargar los datos de configuración del sistema de destino en su PG tras haber montado el hardware y haber realizado un rearranque completo (en caliente) del equipo. Entonces obtendrá la configuración del equipo con las indicaciones de tipo de los distintos módulos. Después basta que especifique (referencia) y parametrice los distintos módulos.

En la PG se cargan las siguientes informaciones:

- S7-300: configuración para el bastidor central y posibles bastidores de ampliación
- S7-400: configuración para el bastidor central con una CPU y módulos de seal sin bastidores de ampliación
- Los datos de configuración para la periferia descentralizada no se puede cargar en la PG.

Este volumen de información se carga cuando el sistema de destino no contiene datos de configuración, p.ej. tras realizar un borrado total del sistema. Por lo demás, la función "Cargar en la PG" da mejores resultados.

En los sistemas S7-300 sin periferia descentralizada sólo tiene que especificar (referencia) y parametrizar los módulos.

Nota

Al cargar en la PG (sin que exista una configuración offline), STEP 7 no puede determinar todas las referencias de los componentes.

Las referencias que no aparecen se pueden introducir al configurar el hardware con el comando de menú **Herramientas >Especificar módulo**. De esta manera se pueden parametrizar módulos desconocidos por STEP 7 (es decir, que no se parametrizan en la ventana "Catálogo de hardware"); sin embargo no se verifica si se respetan las reglas de parametrización.

Limitaciones al cargar desde el sistema de destino

Para los datos que se cargan en la PG desde el sistema de destino rigen las siguientes limitaciones:

- Los bloques no contienen nombres simbólicos para parámetros, variables y marcas.
- Los bloques no contienen comentarios.
- En la PG se carga todo el programa con todos los datos del sistema. Sin embargo, sólo se pueden seguir editando de forma habitual aquellos datos del sistema que son parte de "HW-Config: Configurar el hardware".
- Los datos relacionados con los temas "Comunicación mediante datos globales (GD)", "Configuración de avisos de símbolos" ya no se pueden modificar.
- Las peticiones de forzado permanente no se cargan en la PG. Estas peticiones se tienen que guardar por separado en forma de tabla de variables (VAT) en el diálogo de la petición.
- Los comentarios que se hagan en los diálogos de los módulos no se cargan.
- Los nombres de los módulos sólo se visualizan si ha seleccionado la opción correspondiente en la configuración (HW-Config: Herramientas > Preferencias: "Guardar nombre del objeto en sistema de destino")

19.3.1 Cargar un equipo en la PG

Utilice el comando de menú **Sistema de destino > Cargar equipo en PG** para cargar en la PG la configuración actual y todos los bloques del sistema de automatización a elegir.

A tal efecto, STEP 7 crea un nuevo equipo en el proyecto actual, bajo el cual se guarda la configuración. El nombre estándar del equipo insertado se puede cambiar (p.ej.. "Equipo SIMATIC 300(1)"). El equipo insertado se visualiza tanto en la vista "online" como en la vista "offline".

Para elegir este comando de menú tiene que estar abierto un proyecto. No es necesario seleccionar un objeto en la ventana de proyecto o en la vista (online u offline) para ejecutar el comando de menú.

Utilice esta función para que la configuración resulte más fácil.

- En los sistemas de destino S7-300 se carga la configuración del sistema de automatización utilizado incluidos los bastidores de ampliación pero sin la periferia descentralizada (DP).
- En los sistemas de destino S7-400 se carga la configuración del bastidor sin bastidores de ampliación ni periferia descentralizada.

En los sistemas S7-300 sin periferia descentralizada basta con especificar los módulos (indicando su referencia) y parametrizarlos.

Limitaciones en cuanto a la carga de un equipo en la PG

Para los datos cargados en la PG rigen las siguientes limitaciones:

- Los bloques no contienen nombres simbólicos para parámetros formales, variables temporales y marcas.
- Los bloques no contienen comentarios.
- En la PG se carga todo el programa con todos los datos del sistema. Sin embargo, no todos los datos pueden ser procesados por la misma.
- Los datos relacionados con los temas "Comunicación mediante datos globales (GD)", "Configurar avisos de símbolos" y "Configurar redes" ya no se pueden modificar.
- Las peticiones de forzado permanente no se cargan en la PG.

19.3.2 Cargar bloques desde una CPU S7

Con el Administrador SIMATIC puede cargar bloques S7 de la CPU en el disco duro del sistema de origen. La posibilidad de cargar bloques en la PG desde la CPU puede servirle, p.ej., para:

- hacer una copia de seguridad del programa de usuario que está cargado actualmente en la CPU. Por ejemplo, después de trabajos de mantenimiento, como p.ej. después de un borrado total de la CPU, el personal encargado puede transferir nuevamente dicha copia de seguridad.
- Además, puede cargar el programa de usuario de la CPU en el sistema de origen y editarlo allí, como p.ej. para buscar errores a efectos de mantenimiento. En este caso, no dispone de símbolos ni de comentarios para documentar el programa. Por lo tanto, este procedimiento se recomienda sólo a efectos de mantenimiento.

19.3.3 Editar los bloques cargados en la PG/en el PC

La posibilidad de editar bloques en la PG ofrece las siguientes ventajas:

- Corregir el bloque directamente en la CPU durante la fase de test y documentar el resultado.
- Cargar en la PG el contenido actual de los bloques de la memoria de carga RAM de la CPU.

Nota

Conflicto de la indicación de hora y fecha al trabajar online y offline

Las acciones descritas a continuación pueden causar conflictos de la indicación de hora y fecha, por lo que deberán evitarse.

Dichos conflictos se presentan al abrir un bloque online si

- los cambios efectuados online no se han guardado offline en el programa de usuario S7
- los cambios efectuados offline no se han cargado en la CPU.

Dichos conflictos se presentan al abrir un bloque offline si

- un bloque online con conflicto de indicación de hora y fecha se copia offline en el programa de usuario S7, abriéndose luego el bloque offline.
-

Distinguir dos casos diferentes

Al cargar bloques de la CPU en la unidad de programación es preciso distinguir dos casos:

1. el programa de usuario al que pertenecen los bloques se encuentra en la unidad de programación.
2. el programa de usuario al que pertenecen los bloques no se encuentra en la unidad de programación.

Por consiguiente, no se dispone de aquellos componentes del programa que no se pueden cargar en la CPU. Estas son:

- la tabla de símbolos con los nombres simbólicos de los operandos y los comentarios,
- los comentarios de los segmentos de un programa FUP o KOP,
- los comentarios de las líneas de un programa AWL y
- tipos de datos de usuario,

19.3.3.1 Editar bloques cargados si la PGo el PC contiene el programa de usuario

Para editar bloques de la CPU:

1. En el Administrador SIMATIC, abra la ventana online del proyecto.
2. Seleccione en la ventana online una carpeta "Bloques". Entonces se visualiza la lista de los bloques cargados.
3. Seleccione, abra y modifique los bloques que desee.
4. Elija el comando de menú **Archivo > Guardar** para almacenar el cambio offline en la PG.
5. Cargue los objetos modificados en el sistema de destino eligiendo el comando **Sistema de destino > Cargar**.

19.3.3.2 Editar bloques cargados si la PGo el PC no contiene el programa de usuario

Para editar bloques de la CPU:

1. En el Administrador SIMATIC, pulse el botón "Estaciones accesibles" o elija el comando **Sistema de destino > Estaciones accesibles**.
2. En la lista que aparece entonces, elija las estaciones (objeto "MPI=...") y abra la carpeta "Bloques" para visualizar los bloques.
3. Entonces podrá abrirlos y editarlos, supervisarlos o copiarlos, según sea necesario.
4. Elija el comando **Archivo > Guardar como ...** e introduzca en el correspondiente cuadro de diálogo la ruta donde desee guardarlos en la PG. ,
5. Cargue los objetos modificados en el sistema de destino eligiendo el comando **Sistema de destino > Cargar**.

19.4 Borrar en el sistema de destino

19.4.1 Borrar las memorias de carga/de trabajo y borrado total de la CPU

Antes de cargar su programa de usuario en el sistema de destino S7 es recomendable que lleve a cabo un borrado total de la CPU para garantizar que en la CPU no se encuentren más bloques "antiguos".

Requisito para el borrado total

Para poder llevar a cabo un borrado total, la CPU deberá estar en estado operativo STOP (selector en STOP o en RUN-P y cambio del estado operativo a STOP mediante el comando **Sistema de destino > Diagnóstico/Preferencias > Estado operativo**).

Borrado total de CPUs S7

En un borrado total de una CPU S7 sucede lo siguiente:

- La CPU se pone a cero.
- Se borran todos los datos de usuario (bloques y SDBs, exceptuando los parámetros del interface MPI)
- La CPU deshace todos los enlaces existentes.
- Si existen datos en una EPROM (Memory Card o EPROM integrada), la CPU vuelve a copiar su contenido en la memoria RAM después del borrado total.

Se conservan el contenido del búfer de diagnóstico y los parámetros del interface MPI.

Borrado total de CPUs/FMs M7

En un borrado total de una CPU M7 o de un FM M7 sucede lo siguiente:

- Se restablece el estado inicial.
- Se borran los SDBs, exceptuando los parámetros del interface MPI.
- La CPU/FM deshace todos los enlaces existentes. Los programas de usuario no se pierden y siguen corriendo tras cambiar de STOP a RUN.

Si se producen errores muy graves, con la función Borrado total puede restablecer el estado que tenía la CPU/el FM M7 en su origen. Para ello borre de la memoria de trabajo los bloques de datos de sistema (SDB) actuales y vuelva a cargar los SDBs que se encuentran en la memoria ROM. En algunos casos es necesario además un rearranque (arranque en caliente) del sistema operativo. Para ello debe borrar por completo el M7 con el selector de modos de operación (posición MRES). Con el selector de modos de operación de las CPUs/de los FMs SIMATIC M7 sólo se podrá efectuar un reset si se utiliza el sistema operativo RMOS32.

19.4.2 Borrar bloques S7 en el sistema de destino

Durante la fase de comprobación del programa de CPU puede ser necesario borrar bloques en la CPU. Los bloques están guardados en la memoria de usuario de la CPU bien sea en la EPROM, o bien, en la RAM (dependiendo de la CPU y del proceso de carga).

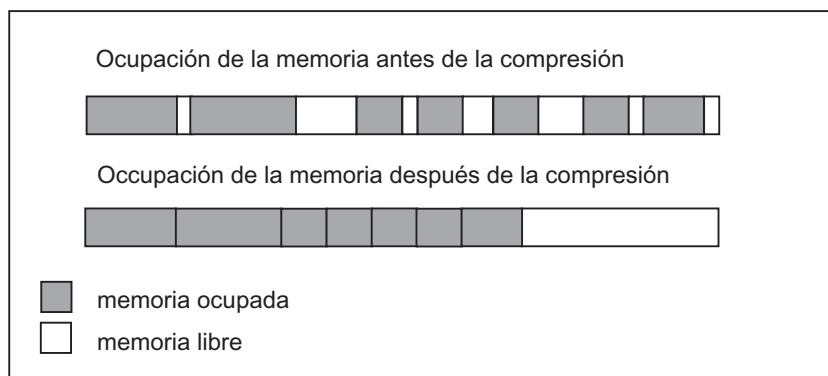
- Los bloques que estén cargados en la RAM se pueden borrar directamente. El espacio de memoria ocupado se libera en las memorias de carga y de trabajo.
- Los bloques que estén guardados en la EPROM integrada se copian en la memoria RAM siempre después de haberse borrado totalmente la CPU. Las copias pueden borrarse directamente en la memoria RAM. En la EPROM, los bloques borrados son declarados no válidos hasta el próximo borrado total o hasta el próximo corte de tensión con memoria RAM no respaldada. Después de un corte de tensión, si la memoria RAM no tiene respaldo, o bien, después de un borrado total de la CPU, los bloques "borrados" se copian nuevamente de la EPROM en la RAM, siendo entonces los bloques activos. Los bloques contenidos en la EPROM integrada (p.ej. en el caso de la CPU 312) se borran al sobrescribirse con el nuevo contenido de la RAM.
- Las Memory Cards EPROM se deben borrar en el sistema de origen.

19.5 Comprimir la memoria de usuario (RAM)

19.5.1 Creación de huecos en la memoria de usuario (RAM)

Al borrar y sobrescribir bloques con frecuencia, se forman huecos en la memoria de usuario (memorias de carga y de trabajo) que reducen el espacio de memoria aprovechable. Comprimiendo la memoria de usuario es posible repositonar los bloques existentes uno junto a otro, creándose un espacio contiguo de memoria libre.

En la figura siguiente se muestra de forma esquemática cómo se desplazan los bloques de memoria ocupados al activar la función "Comprimir memoria".



En lo posible, se deberá comprimir en estado operativo "STOP".

Para poder rellenar todos los huecos de la memoria es preciso comprimirla en el estado operativo "STOP". Si se comprime en modo RUN-P (posición del selector de modos de operación), no se desplazan los bloques que se están ejecutando actualmente, puesto que están abiertos. En modo RUN (posición del selector de modos de operación) no es posible llevar a cabo la compresión (debido a la protección contra escritura).

19.5.2 Comprimir el contenido de la memoria de una CPU S7

Alternativas para comprimir

Hay dos alternativas para comprimir la memoria de usuario:

- Si al cargar falta espacio de memoria en el sistema de destino, aparece un cuadro de diálogo donde se indica el error. Haciendo clic en el correspondiente botón es posible comprimir la memoria.
- Como medida preventiva puede visualizar el grado de utilización de la memoria (elija a tal efecto el comando de menú **Sistema de destino > Diagnóstico/Preferencias > Información del módulo.../ficha "Memoria"**) y, en caso necesario, iniciar la compresión.

Procedimiento

1. Seleccione el programa S7 en la vista online o en la ventana "Estaciones accesibles".
2. Elija el comando **Sistema de destino > Diagnóstico/Preferencias > Información del módulo**.
3. En el cuadro de diálogo siguiente elija la ficha "Memoria", donde encontrará el botón de comando "Comprimir" para comprimir la memoria en el caso de que la CPU soporte esta función.

20 Test

20.1 Introducción al test con tablas de variables

Las tablas de variables ofrecen la ventaja de que pueden guardar distintos entornos de test. Así se pueden reproducir sin esfuerzo tests y observaciones durante la puesta en marcha o para operaciones de servicio técnico o de mantenimiento. El número de tablas de variables que se pueden almacenar es ilimitado.

Para comprobar el programa con tablas de variables se dispone de las siguientes funciones:

- **Observar variables**
Esta función permite visualizar en la PG/el PC los valores actuales de determinadas variables de un programa de usuario o de una CPU.
- Forzar variables
Esta función permite asignar valores fijos a determinadas variables de un programa de usuario o de una CPU. El test con el estado del programa posibilita asimismo un forzado único e inmediato.
- Desbloquear salidas y Activar valores forzados
Ambas funciones permiten asignar valores fijos a determinadas salidas de la periferia de una CPU en el estado operativo STOP.
- Forzar variables de forma permanente
Utilice esta función para asignar valores permanentes (fijos) a las variables de un programa de usuario o de una CPU de manera que el programa de usuario no los pueda sobrescribir.

Puede determinar o visualizar los valores de las siguientes variables:

- Entradas, salidas, marcas, temporizadores y contadores
- Contenido de bloques de datos
- Periferia

Elabore una tabla con aquellas variables cuyos valores desee visualizar o forzar.

Definiendo el punto de disparo y la condición de disparo se define cuándo se deben observar o sobrescribir las variables y con qué frecuencia.

20.2 Procedimiento básico para observar y forzar con tablas de variables

Para poder utilizar las funciones **Observar** y **Forzar**:

1. Cree una tabla de variables nueva o abra una ya existente.
2. Edite o compruebe la tabla de variables.
3. Con el comando **Sistema de destino > Establecer enlace con**, establezca un enlace online entre la tabla de variables actual y la CPU deseada.
4. Con el comando **Variable > Condiciones de disparo**, elija un punto de disparo apropiado y ajuste la condición de disparo.
5. Con los comandos **Variable > Observar** y **Variable > Forzar** es posible activar o desactivar la función en cuestión.
6. Guarde la tabla de variables creada utilizando los comandos **Tabla > Guardar** o **Tabla > Guardar como** para poder llamarla de nuevo en cualquier momento.

20.3 Editar y guardar tablas de variables

20.3.1 Crear y abrir una tabla de variables

Antes de poder observar o forzar variables, tiene que crear una tabla de variables (VAT) e introducir las variables deseadas. Para crear una tabla de variables puede elegir una de las alternativas siguientes:

En el Administrador SIMATIC:

- Seleccione la carpeta "Bloques" y elija el comando de menú **Insertar > Bloque S7> Tabla de variables**. En el cuadro de diálogo que aparece a continuación puede asignar un nombre a la tabla (cuadro de texto "Nombre simbólico"). Este nombre aparecerá en la ventana de proyecto. Para abrir la tabla de variables, haga doble clic en el objeto.
- Seleccione un enlace de la lista de las estaciones accesibles o un programa S7 o M7 de la vista online. Con el comando **Sistema de destino > Observar/forzar variable** se puede crear una tabla de variables sin nombre.

En "Observar/forzar variables":

- Eligiendo el comando **Tabla > Nueva** puede crear una tabla que no esté asociada todavía a ningún programa S7/M7. Para acceder a las tablas ya existentes, elija el comando **Tabla > Abrir**.
- Puede usar los botones de la barra de herramientas para crear o abrir tablas de variables.

La tabla de variables se puede guardar, imprimir y utilizar en repetidas ocasiones para observar y forzar.

20.3.2 Copiar/mover tablas de variables

Las tablas de variables se pueden copiar o mover en las carpetas de bloques de programas S7/M7.

Al copiar y mover tablas de variables debe tener en cuenta los siguientes puntos:

- En la tabla de símbolos del programa de destino se actualizan los símbolos que ya figuran en el programa.
- Al mover una tabla de variables, también se mueven los símbolos de la tabla del programa de origen a la tabla de símbolos del programa de destino.
- Al borrar la tabla de variables de una carpeta de bloques, también se borran los símbolos correspondientes de la tabla de símbolos del programa S7/M7.
- Si el programa de destino ya contiene una tabla de variables con el mismo nombre, al copiar se dispone de la posibilidad de cambiarle el nombre a la tabla de variables (de forma predeterminada se le añade un número al nombre existente).

20.3.3 Guardar una tabla de variables

Puede guardar esta tabla de variables para poderla utilizar en un nuevo test de su programa para observar y forzar variables.

1. Guarde la tabla de variables con el comando de menú **Tabla > Guardar**.
2. Si se ha generado una nueva tabla de variables debe indicar un nombre para la tabla, p.ej., "Test_1".

Al guardar la tabla de variables se almacenan todos los ajustes actuales, así como el formato de la tabla. Por tanto, se guardarán todos los ajustes efectuados mediante el comando de menú "Condiciones de disparo".

20.4 Introducir variables en tablas de variables

20.4.1 Insertar operandos o símbolos en una tabla de variables

Defina las variables cuyos valores desea predeterminar u observar, e introdúzcalas en la tabla de variables. Al hacerlo proceda de "fuera" a "dentro", es decir, seleccione primero las entradas y después las variables sobre las que influyen las entradas, o que influyen en las salidas, y finalmente seleccione las salidas.

Por ejemplo, si desea observar el bit de entrada 1.0, la palabra de marcas 5 y el byte de salidas 0, en la columna de operando introduzca lo siguiente:

Ejemplo:

```
E 1.0
MW 5
AB 0
```

Ejemplo de una tabla de variables rellena

La figura siguiente muestra una tabla de variables con las siguientes columnas: operando, símbolo, formato de visualización, valor de estado y valor de forzado

	Operando	Símbolo	Formato de vi	V. de estad	V. de fuerza
1	//OB1 Segmento 1				
2	E 0.1	"Pulsador 1"	BOOL	true	
3	E 0.2	"Pulsador 2"	BOOL	true	
4	A 4.0	"Lampara verde"	BOOL	false	
5	//OB1 Segmento 3				
6	E 0.5	"Modo Autom. ON"	BOOL	true	
7	E 0.6	"Modo Manual ON"	BOOL	true	
8	A 4.2	"Modo automático"	BOOL	true	true
9	//OB1 Llamada FB1 para encender motor de gasolina				
10	E 1.0	"Encender_MG"	BOOL	false	
11	E 1.1	"Apagar_MG"	BOOL	false	
12	E 1.2	"Fallo_MG"	BOOL	false	
13	A 5.1	"Vel_teor_alcanzada"	BOOL	false	
14	A 5.0	"MG_ON"	BOOL		true
15	//OB1 Llamada FB1 para encender motor diese				
16	E 1.4	"Encender_MD"	BOOL	false	
17	E 1.5	"Apagar_MD"	BOOL		

MPI = 3 (directo) Run

Notas para insertar símbolos

- La variable a forzar se indica con su operando o con su nombre simbólico. Los símbolos y los operandos se pueden registrar en las columnas denominadas "Símbolo" y "Operando". El registro se escribe automáticamente en la columna adecuada. Si se ha definido el símbolo correspondiente en la tabla de símbolos, la entrada en la columna "Símbolo" u "Operando" se actualiza automáticamente.
- Solamente se pueden registrar aquellos símbolos que ya hayan sido definidos en la tabla de símbolos.
- Cuando introduzca el símbolo regístrelo exactamente igual que como está definido en la tabla de símbolos.
- Los nombres simbólicos que contengan caracteres especiales tienen que escribirse entre comillas (p.ej.: "Motor.on", "Motor+on", "Motor-on").
- Si quiere definir nuevos símbolos en la tabla de símbolos, seleccione el comando **Herramientas > Tabla de símbolos**. También puede copiar los símbolos en la tabla de símbolos y pegarlos en una tabla de variables.

Comprobación de sintaxis

Al introducir las variables en la tabla, la sintaxis se comprueba antes de abandonar cada línea. Las entradas erróneas se destacan en rojo.

Si se coloca el cursor en una línea marcada en rojo, una información rápida le muestra la causa del error. Pulsando la tecla F1 se obtendrá información acerca de cómo subsanar el mismo.

Nota

Si desea editar la tabla de variables preferentemente con el teclado (sin ratón), debe haber activado "Mostrar información breve en caso de utilizar el teclado".

Si es necesario, cambie el ajuste de la tabla de variables con el comando de menú **Herramientas > Preferencias** en la ficha "General".

Tamaño máximo

El tamaño máximo de una tabla es de 1024 líneas y cada línea puede incluir como máximo 255 caracteres. Si se sobrepasa el límite de caracteres el programa le impedirá el cambio a la línea siguiente.

20.4.2 Insertar un área de operandos conexos en una tabla de variables

1. Abra una tabla de variables.
2. Posicione el cursor en la línea tras la cual desee insertar el área de operandos conexos.
3. Elija el comando de menú **Insertar > Área**. Se visualizará el campo de diálogo "Área".
4. En el campo "Desde operando " introduzca una dirección de operando como dirección inicial.
5. En el campo "Cantidad" introduzca el número de las líneas a insertar.
6. En la lista visualizada escoja el formato de visualización que desee.
7. Haga clic en "Aceptar".

El área se inserta en la tabla de variables.

20.4.3 Insertar valores de forzado

Valor de forzado como comentario

Si desea dejar sin efecto el "Valor de forzado" de una variable utilice el comando de menú **Variable > Valor de forzado como comentario**. Un carácter iniciador de comentario "//" antes del valor que va a forzarse de una variable indica que dicho valor pierde su validez. En lugar de la llamada del comando de menú también se puede iniciar el comentario con el carácter "/*". También puede cambiarse la invalidez del "valor de forzado" activando de nuevo el comando de menú **Variable > Valor de forzado como comentario** o eliminando el carácter de comentario.

20.4.4 Valores máximos para introducir temporizadores

Al introducir los valores de los temporizadores, tenga en cuenta los siguientes valores máximos:

W#16#3999 (valor máximo en el formato BCD)

Ejemplos:

Operando	Formato de visualización	Introducción	Mostrar valor de forzado	Explicación
T 1	TIEMPO_SIMATIC	137	S5TIME#130MS	Conversión a milisegundos
MW	TIEMPO_SIMATIC	137	W#16#0089	Posibilidad de visualizar en formato BCD
MW	HEX	137	W#16#0089	Posibilidad de visualizar en formato BCD
MW	HEX	157	W#16#009D	Visualización en formato BCD imposible, por lo que no se puede seleccionar el formato de visualización TIEMPO_SIMATIC

Nota

- Si bien los valores de tiempo se pueden introducir tomando como unidad el milisegundo, el valor que se haya especificado será adaptado conforme a un determinado periodo. La magnitud de este periodo dependerá de la magnitud del valor de tiempo especificado (137 se redondea, resultando así 130 ms).
- Los valores de forzado de los operandos del tipo de datos WORD (p.ej. EW 1), se convierten al formato BCD. ¡No todas las configuraciones binarias son números BCD válidos! Si en el caso de un operando del tipo de datos WORD no es posible representar la entrada en forma de TIEMPO_SIMATIC, se cambiará automáticamente al formato predeterminado (en este caso: HEX, véase "Elegir formato de estado", preajuste en el menú Ver), de manera que el valor introducido se pueda visualizar.

Formato BCD para variables en formato TIEMPO_SIMATIC

Los valores de variables en formato TIEMPO_SIMATIC se introducen en formato BCD. Los 16 bits tienen el siguiente significado:

| 0 0 x x | h h h h | z z z z | e e e e |

Bits 15 y 14 son siempre cero.

Bits 13 y 12 (señalados con xx) definen el multiplicador para los bits 0 a 11:

- 00 => Multiplicador 10 milisegundos
- 01 => Multiplicador 100 milisegundos
- 10 => Multiplicador 1 segundo
- 11 => Multiplicador 10 segundos

Bits 11 a 8 Centenas (hhhh)

Bits 7 a 4 Decenas (zzzz)

Bits 3 a 0 Unidades (eeee)

20.4.5 Valores máximos para introducir contadores

Al introducir los valores de los contadores, tenga en cuenta los siguientes límites:

Límites superiores para contadores: C#999
W#16#0999 (valor máximo en el formato BCD)

Ejemplos:

Operando		Formato de visualización	Introducción	Mostrar valor de forzado	Explicación
C	1	COUNTER	137	C#137	Conversión
MW	4	COUNTER	137	C#89	Posibilidad de visualizar en formato BCD
MW	4	HEX	137	W#16#0089	Posibilidad de visualizar en formato BCD
MW	6	HEX	157	W#16#009D	Visualización en formato BCD imposible, por lo que no se puede seleccionar el formato de visualización CONTADOR


Nota

- Si introduce en un contador un número decimal y no identifica el valor con C#, dicho valor se convertirá automáticamente al formato BCD (137 se convierte en C#137).
- Los valores de forzado de los operandos del tipo de datos WORD (p.ej. EW 1), se convierten al formato BCD. ¡No todas las configuraciones binarias son números BCD válidos! Si en el caso de un operando del tipo de datos WORD no es posible representar la entrada en forma de COUNTER, se cambiará automáticamente al formato predeterminado (en este caso: HEX, véase "Elegir formato de estado", preajuste en el menú Ver), de manera que el valor introducido se pueda visualizar.

20.4.6 Introducir líneas de comentario

Las líneas de comentario se introducen con dos barras inclinadas "//".

Si desea dejar sin efecto una o varias líneas de la tabla de variables utilice el comando

Edición > Desactivar línea o el correspondiente botón  de la barra de herramientas.

20.4.7 Ejemplos

20.4.7.1 Ejemplo de introducción de operandos en tablas de variables

Operando permitido:	Tipo de datos:	Ejemplo (nematécnica SIMATIC):
Entrada Salida Marca	BOOL	E 1.0 A 1.7 M 10.1
Entrada Salida Marca	BYTE	EB 1 AB 10 MB 100
Entrada Salida Marca	WORD	EW 1 AW 10 MW 100
Entrada Salida Marca	DWORD	ED 1 AD 10 MD 100
Periferia (Entrada Salida)	BYTE	PEB 0 PAB 1
Periferia (Entrada Salida)	WORD	PEW 0 PAW 1
Periferia (Entrada Salida)	DWORD	PED 0 PAD 1
Temporizadores	TIMER	T 1
Contadores	COUNTER	Z 1
Bloque de datos	BOOL	DB1.DBX 1.0
Bloque de datos	BYTE	DB1.DBB 1
Bloque de datos	WORD	DB1.DBW 1
Bloque de datos	DWORD	DB1.DBD 1

Nota

Por razones de utilización interna no está permitida la entrada "DB0...".

Ventana "Valores de forzado"

Los módulos S7-300 solamente permiten un forzado permanente de entradas, salidas y periferia (salidas).

Los módulos S7-400 solamente permiten un forzado permanente de entradas, salidas, marcas y periferia (entradas/salidas).

20.4.7.2 Ejemplo de introducción de un área de operandos conexos

Abra una tabla de variables y visualice el cuadro de diálogo "Insertar área" con el comando de menú **Insertar > Área**.

Rellene el cuadro de diálogo indicando el número de líneas, a partir de qué operando y el formato en el que se introducirán en la tabla de variables, como por ejemplo:

- Desde operando: M 3.0
- Cantidad: 10
- Formato de visualización: BIN

Operando	Formato de estado
M 3.0	BIN
M 3.1	BIN
M 3.2	BIN
M 3.3	BIN
M 3.4	BIN
M 3.5	BIN
M 3.6	BIN
M 3.7	BIN
M 4.0	BIN
M 4.1	BIN

En este ejemplo puede observar cómo varía la denominación en la columna "Operando" después de la octava entrada.

20.4.7.3 Ejemplos de introducción de valores de forzado normal y de forzado permanente

Operandos de bit

Operandos de bit posibles	Valores permitidos de forzado normal y forzado permanente
E1.0	True
M1.7	False
A10.7	0
DB1.DBX1.1	1
E1.1	2#0
M1.6	2#1

Operandos de byte

Operandos de byte posibles	Valores permitidos de forzado / de forzado permanente
EB 1	2#00110011
MB 12	b#16#1F
MB 14	1F
AB 10	'a'
DB1.DBB 1	10
PAB 2	-12

Operandos de palabra

Operandos de palabra posibles	Valores permitidos de forzado normal y forzado permanente
EW 1	2#0011001100110011
MW 12	w#16#ABCD
MW 14	ABCD
AW 10	b#(12,34)
DB1.DBW 1	'ab'
PAW 2	-12345
MW 3	12345
MW 5	s5t#12s340ms
MW 7	0.3s ó 0,3s
MW 9	c#123
MW 11	d#1990-12-31

Operandos de palabra doble

Operandos de palabra doble posibles	Valores permitidos de forzado / de forzado permanente
ED 1	2#0011001100110011001100110011
MD 0	1.23e4
MD 4	1.2
AD 10	dw#16#abcdef10
AD 12	ABCDEF10
DB1.DBD 1	b#(12,34,56,78)
PAD 2	'abcd'
MD 8	L# -12
MD 12	L#12
MD 16	123456789
MD 20	123456789
MD 24	T#12s345ms
MD 28	Tod#1:2:34.567
MD 32	p#e0.0

Temporizador

Operandos posibles del tipo temporizador	Valores permitidos de forzado normal y forzado permanente	Explicación
T 1	0	Valor temporizado en milisegundos (ms)
T 12	20	Valor temporizado en milisegundos
T 14	12345	Valor temporizado en milisegundos
T 16	s5t#12s340ms	Valor temporizado 12s 340ms
T 18	1.3	Valor temporizado 1s 300 ms
T 20	1.3s	Valor temporizado 1s 300 ms

El forzado de un temporizador sólo afecta su valor, mas no su estado. Por tanto, el temporizador T1 se puede forzar al valor 0, pero el resultado lógico en U T1 no se modifica.

Las secuencias de caracteres "s5t" y "s5time" se pueden escribir tanto en mayúsculas como en minúsculas.

Contadores

Operandos posibles del tipo contador	Valores permitidos de forzado / de forzado permanente
Z 1	0
Z 14	20
Z 16	c#123

El forzado de un contador sólo afecta a su valor, pero no a su estado. Por tanto, el contador T1 se puede forzar al valor 0, pero el resultado lógico en U Z1 no se modifica.

20.5 Establecer un enlace con la CPU

Para poder observar y forzar las variables que ha reunido en su tabla de variables (VAT) actual, debe establecer un enlace con la CPU correspondiente. Cualquier tabla de variables se puede enlazar con otra CPU.

Visualizar el enlace online

Si existe un enlace online, en la barra de título de la ventana de la tabla de variables aparecerá la palabra "ONLINE". En la barra de estado se muestran, dependiendo de la CPU utilizada, los estados operativos "RUN", "STOP", "DESCONECTADO" o "CONECTADO".

Establecer un enlace online con la CPU

Si no existe un enlace online con la CPU deseada, éste se deberá definir con el comando **Sistema de destino > Establecer enlace con > ...**, para poder observar o forzar las variables.

Deshacer el enlace online con la CPU

Para deshacer un enlace online entre la tabla de variables y la CPU, elija el comando **Sistema de destino > Deshacer enlace**.

Nota

Si ha creado una tabla de variables sin nombre con el comando **Tabla > Nueva**, sólo podrá establecer un enlace con la última CPU configurada si está definida.

20.6 Observar variables

20.6.1 Introducción a la observación de variables

Para observar variables puede elegir una de las alternativas siguientes:

- Active la función "Observar" con el comando **Variable > Observar**. Dependiendo del punto y de la condición de disparo que se hayan ajustado, los valores de las variables seleccionadas se visualizan en la tabla de variables. Si ha ajustado la condición de disparo "Cíclico", desactive nuevamente la función "Observar", con el comando **Variable > Observar**.
- Con el comando **Variable > Actualizar valores de estado** se actualizan inmediatamente los valores de las variables seleccionadas una única vez. Los valores actuales de las variables seleccionadas se visualizan en la tabla de variables.

Cancelar la observación con la tecla ESC

Si pulsa la tecla ESC con la función "Observar" en curso, saldrá de "Observar" sin consulta previa.

20.6.2 Ajustar el disparo para observar variables

Para observar variables puede visualizar en la PG los valores actuales de variables de un programa de usuario en un punto predeterminado de la ejecución del programa (el punto de disparo).

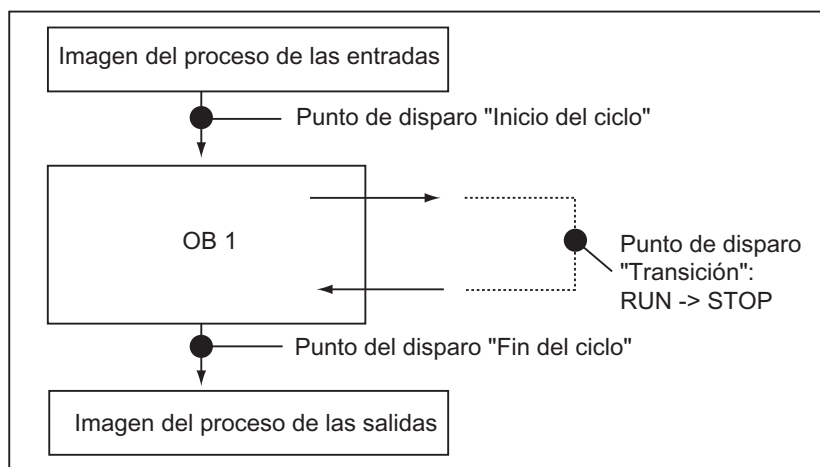
Mediante la elección del punto de disparo se determina el momento en el que se visualizan los valores de estado de las variables.

Con el comando **Variable > Condiciones de disparo** es posible ajustar los puntos y las condiciones de disparo

Condiciones de disparo	Posibilidades de ajuste
Punto de disparo	Inicio del ciclo Fin del ciclo Transición de RUN a STOP.
Condición de disparo	Unico Permanente

Punto de disparo

La figura siguiente ilustra la posición de los puntos de disparo.



Para poder visualizar el valor forzado, el punto de disparo para "Observar" se deberá ajustar en "Inicio del ciclo", en tanto que el punto de disparo para "Forzar" se deberá definir en "Fin del ciclo".

Disparo inmediato

Los valores de variables seleccionadas se pueden actualizar con el comando de menú **Variable > Actualizar valores de estado**. Esta petición se ejecuta una sola vez tan pronto como sea posible, sin tener relación con una posición determinada en el programa de usuario. Estas funciones se utilizan principalmente en modo STOP para observar y forzar variables.

Condición de disparo

La siguiente tabla muestra el efecto que tiene la condición de disparo ajustada en la observación de variables:

	Condición de disparo única	Condición de disparo cíclica
Observar variables	Actualización única dependiendo del punto de disparo	Observar con condición de disparo definida Al comprobar un bloque se puede observar exactamente el procesamiento ulterior.

20.7 Forzar variables

20.7.1 Introducción al forzado de variables

Para forzar variables puede elegir una de las alternativas siguientes:

- Active la función "Forzar" con el comando **Variable > Forzar**. Dependiendo del punto y de la condición de disparo que se hayan ajustado, el programa de usuario adopta los valores de forzado de las variables seleccionadas en la tabla de variables. Si ha ajustado la condición de disparo "Cíclico", desactive nuevamente la función "Forzar" con el comando **Variable > Forzar**.
- Con el comando **Variable > Activar valores de forzado** se actualizan inmediatamente los valores de las variables seleccionadas una única vez.

Las funciones Forzado permanente y Desbloquear salidas ofrecen más posibilidades al respecto.

Tenga en cuenta lo siguiente al forzar:

- Sólo se forzarán los operandos visibles en la tabla de variables al comienzo del forzado. Si después de iniciar el forzado se reduce el área visible de la tabla de variables, podrían forzarse operandos que han dejado de ser visibles. Si se amplía el área visible de la tabla de variables, puede haber operandos que ya no se puedan forzar.
- El proceso de forzado es irreversible (p. ej., con **Edición > Deshacer**).



Peligro

Si los valores de las variables se modifican estando la instalación en servicio y se presentan fallos de funcionamiento o errores del programa, pueden producirse serios daños materiales o personales.

Antes de ejecutar la función "Forzar", asegúrese de que no puedan presentarse situaciones peligrosas.

Cancelar el forzado con la tecla ESC

Si, con la función "Forzar" en curso, pulsa la tecla ESC, saldrá de "Forzar" sin consulta previa.

20.7.2 Ajustar las condiciones de disparo para forzar variables

En un punto predeterminado en la ejecución del programa de usuario (punto de disparo) se pueden asignar valores únicos o cíclicos a variables individuales de dicho programa.

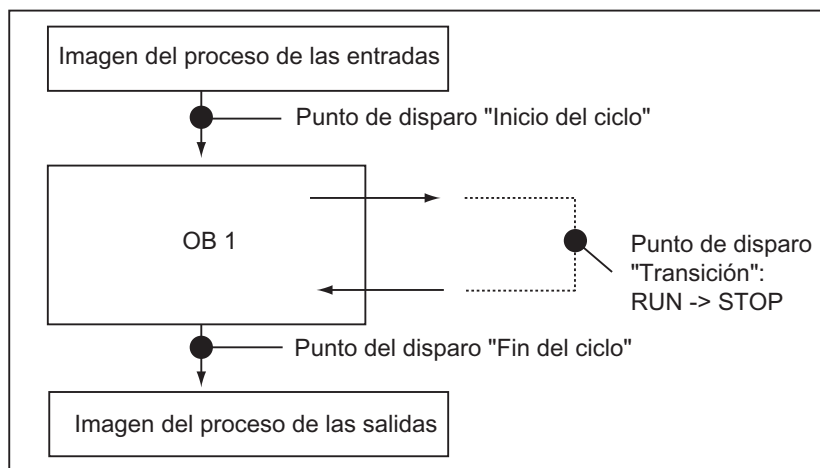
Mediante la elección del punto de disparo se determina el momento en el que los valores de forzado se asignan a las variables.

Con el comando **Variable > Condiciones de disparo** es posible ajustar los puntos y las condiciones de disparo

Condiciones de disparo	Posibilidades de ajuste
Punto de disparo	Inicio del ciclo Fin del ciclo Transición a STOP.
Condición de disparo	Unico Cíclico

Punto de disparo

La figura siguiente muestra la posición de los puntos de disparo.



De la posición de los puntos de disparo se desprende lo siguiente:

- el forzado de entradas sólo tiene sentido si se ha seleccionado el punto de disparo "Inicio del ciclo" (equivale al principio del programa de usuario OB 1), ya que de lo contrario se actualiza la imagen de proceso de las entradas y, por consiguiente, se sobrescribe.
- el forzado de salidas sólo tiene sentido si se ha seleccionado el punto de disparo "Fin de ciclo" (equivale al fin del programa de usuario OB 1), ya que de lo contrario el programa de usuario podría sobrescribir la imagen de proceso de las salidas.

Para que el valor forzado aparezca en la columna "Valor de estado", es necesario ajustar el punto de disparo para observar al "Inicio del ciclo" y el punto de disparo para forzar al "Final del ciclo"

En caso de forzar variables hay que ajustar los puntos de disparo de la siguiente manera:

- Si ha seleccionado la condición de disparo "Único", aparecerá un aviso en caso de que no sea posible forzar las variables seleccionadas.
- Si ha seleccionado la condición de disparo "Permanente" no aparecerá ningún aviso.

Disparo inmediato

Con el comando de menú **Variable > Activar valores de estado** puede forzar los valores de las variables. Esta petición se ejecuta una sola vez tan pronto como sea posible, sin tener relación con una posición determinada en el programa de usuario. Esta función se utiliza principalmente en modo STOP para forzar variables.

Condiciones de disparo

La siguiente tabla muestra el efecto que tiene la condición de disparo ajustada en el forzado de variables:

	Condición de disparo única	Condición de disparo cíclica
Forzar variables	<p><i>Activación única (forzar variables)</i></p> <p>Dependiendo del punto de disparo, a las variables se les pueden asignar valores una sola vez.</p>	<p><i>Forzar con condición de disparo definida</i></p> <p>Asignando valores fijos puede simular determinadas situaciones para su programa de usuario, comprobando así las funciones programadas.</p>

20.8 Forzado permanente de variables

20.8.1 Reglas de seguridad para el forzado permanente de variables



Prevenir daños personales y materiales

Al ejecutar la función "Forzado permanente" tenga en cuenta que una acción errónea puede:

- poner en peligro la vida o la salud de personas, o
- provocar daños en la máquina o la instalación completa.



Cuidado

- Antes de arrancar la función "Forzado permanente", debe cerciorarse de que nadie ejecute simultáneamente la misma función en la misma CPU.
- Sólo puede borrar o salir de una petición de forzado permanente con el comando de menú **Variable > Anular forzado permanente**. Con sólo cerrar la ventana "Valores de forzado permanente" o salir de la aplicación "Observar y forzar variable" no se anula la petición de forzado permanente.
- El "Forzado permanente" no puede deshacerse con el comando de menú **Edición > Deshacer**.
- Infórmese sobre las diferencias entre forzado permanente y forzado normal de variables.
- Si una CPU no soporta la función de forzado permanente, no pueden seleccionarse ninguno de los comandos de menú relativos al forzado permanente en el menú Variable.

Cuando se ha anulado el bloqueo de salidas usando el comando de menú **Variable > Desbloquear salidas**, todos los módulos de salidas forzados devuelven su valor de forzado permanente.

20.8.2 Introducción al forzado permanente de variables

Es posible asignar valores permanentes (fijos) a las variables de un programa de usuario, de manera que el programa que se ejecute en la CPU no los pueda cambiar ni sobrescribir. Para ello es necesario que la CPU ofrezca esta prestación (p. ej., las CPUs S7-400). Puede ajustar diversas situaciones para predefinir valores fijos en variables para su programa de usuario, y hacer así un test de las funciones programadas.

Ventana "Valores de forzado":

Sólo pueden seleccionarse los comandos de menú para forzado permanente si está activa la ventana "Valores de forzado permanente".

Para visualizar dicha ventana, elija el comando de menú **Variable > Mostrar valores de forzado permanente**.

Para una CPU se deberá abrir una sola ventana "Valores de forzado permanente". En dicha ventana se visualizan todas las variables con los valores de la petición activa.

Ejemplo de una ventana de forzado permanente

	Operand	Símbol	Formato de vis	Valor de forz
1	EB 0		HEX	B#16#10
2	A 0.1		BOOL	true
3	A 1.2		BOOL	true
4				

En la **barra de título** figura el nombre del enlace online actual.

En la **barra de estado** figura el momento (fecha y hora) en el que se ha leído en la CPU la petición de forzado permanente.

Cuando no hay ninguna petición de forzado activa, la ventana está vacía.

Las distintas formas de **mostrar variables** en la ventana Valores de forzado permanente tienen los siguientes significados:

Visualización	Significado
Negrita	VARIABLES predefinidas con carácter fijo con un valor en la CPU.
Normal:	VARIABLES recién editadas.
Atenuado	VARIABLES de un módulo no existente / no enchufado. o VARIABLES con error de direccionamiento; se muestra un aviso de error.

Aceptar los operandos de la tabla de variables que se puedan forzar permanentemente

Si desea adoptar las variables de una tabla de variables en la ventana "Valores de forzado permanente", seleccione la tabla y marque las variables deseadas. A continuación, abra la ventana "Valores de forzado permanente" con el comando de menú **Variable > Mostrar valores de forzado permanente**. Las variables susceptibles de ser forzadas permanentemente se transferirán a la ventana "Valores de forzado permanente".

Aceptar la petición de forzado permanente de la CPU o crear una nueva petición de forzado permanente

Cuando la ventana "Valores de forzado permanente" está abierta y activa se muestra un segundo aviso:

- Si confirma dicho aviso, las modificaciones efectuadas en la ventana "Valores de forzado permanente" son sobrescritas con la petición de forzado permanente existente en la CPU. Podrá restablecer el contenido anterior de la ventana usando el comando **Edición > Deshacer**.
- Si cancela el aviso, se mantiene el contenido actual de la ventana "Valores de forzado permanente". Después puede guardar el contenido en la ventana "Valores de forzado permanente" como tabla de variables (usando el comando de menú **Tabla > Guardar como**), o seleccionar el comando de menú **Variable > Forzado permanente**: al hacerlo, el contenido actual de la ventana "Valores de forzado permanente" se escribe como nueva petición de forzado permanente en la CPU.

En la ventana "Valores de forzado permanente" no se pueden observar ni forzar variables. Tan sólo es posible en la tabla de variables.

Borrar valores de forzado permanente

Con el comando de menú **Variable > Mostrar valores de forzado permanente** se abre la ventana "Valores de forzado permanente". A continuación, con el comando de menú **Variable > Anular forzado permanente** se borran los valores de forzado permanente en la CPU seleccionada.

Guardar el contenido de la ventana "Valores de forzado permanente"

El contenido de la ventana "Valores de forzado permanente" se puede guardar en una tabla de variables. Con el comando de menú **Insertar > Tabla de variables** se puede insertar el contenido guardado en la ventana Valores de forzado permanente.

Notas respecto a los símbolos en la ventana "Valores de forzado permanente"

Los símbolos de la última ventana activa se importan siempre, a no ser que se haya iniciado la función "Observar y forzar variable", desde una aplicación que no tenga símbolos.

Si no se puede introducir ningún nombre simbólico, la columna "Símbolo" está oculta. Entonces tampoco puede seleccionarse el comando de menú **Herramientas > Tabla de símbolos**.

20.8.3 Diferencias entre el forzado normal y el forzado permanente

En la panorámica siguiente se resumen las diferencias entre el forzado normal y el forzado permanente:

Característica / función	Forzado permanente con S7-400 (incl. CPU 318-2DP)	Forzado permanente con S7-300 (sin CPU 318-2DP)	Forzado normal
Marcas (M)	sí	–	sí
Temporizadores y contadores (T, Z)	–	–	sí
Bloques de datos (DB)	–	–	sí
Entradas periféricas (PEB, PEW, PED)	sí	–	–
Salidas de la periferia (PAB, PAW, PAD)	sí	–	sí
Entradas y salidas (E, A)	sí	sí	sí
El programa de usuario puede sobrescribir los valores de forzado normal o permanente.	–	sí	sí
El valor de forzado permanente se mantiene activo sin interrupciones.	sí	sí	–
Al salir de la aplicación, las variables conservan sus valores.	sí	sí	–
Al deshacer el enlace con la CPU, las variables conservan sus valores.	sí	sí	–
Se permiten errores de direccionamiento: p ej., EW1 Valor de forzado normal/permanente: 1 EW1 Valor de forzado normal/permanente: 0	–	–	el último tiene efecto
Ajustar las condiciones de disparo	siempre disparo inmediato	siempre disparo inmediato	único o cíclico
La función tiene efecto sólo en la variable que está en el área visible de la ventana activa.	tiene efecto en todos los valores de forzado perman.	tiene efecto en todos los valores de forzado perman.	sí

Nota

- Al "Desbloquear salidas", los valores de forzado permanente de las salidas periféricas forzadas tendrán efecto en los módulos de salidas. Por el contrario, los valores de forzado de las salidas periféricas forzadas permanentemente no tendrán efecto en ellos.
- En el forzado permanente, la variable conserva siempre el valor forzado. Dicho valor se lee en cada acceso de lectura en el programa de usuario. Ningún acceso de escritura será válido.
- Los accesos de escritura del programa tienen efecto durante el forzado permanente del programa, permaneciendo efectivos hasta el siguiente punto de disparo.

21 Test con el estado del programa

El programa se puede comprobar visualizando para cada instrucción el estado del programa (RLO, bit de estado) o el contenido de la ficha en cuestión. El volumen de la información visualizada se define en la ficha "AWL" del cuadro de diálogo "Preferencias". Dicho cuadro de diálogo se abre con el comando de menú **Herramientas > Preferencias** en la ventana "KOP/AWL/FUP: Programar bloques".



Precaución

Si el test se realiza con la instalación en marcha y se presentan fallos en el funcionamiento del sistema o errores del programa, pueden producirse serios daños materiales o personales.

Antes de ejecutar esta función asegúrese de que no pueden presentarse situaciones peligrosas.

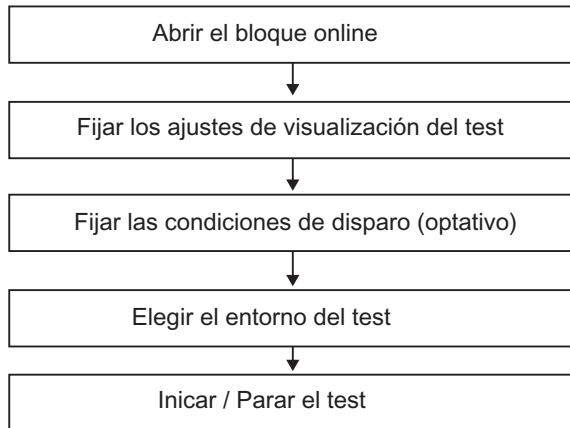
Requisitos

Para poder visualizar el estado del programa se deberán cumplir los siguientes requisitos:

- El bloque se deberá haber guardado sin errores, cargándose posteriormente en la CPU.
- La CPU deberá estar en servicio y el programa de usuario se deberá estar ejecutando.

Procedimiento básico para observar el estado del programa:

No es aconsejable llamar y comprobar inmediatamente todo el programa, sino llamar y comprobar los bloques uno por uno. Comience con los bloques lógicos subordinados, es decir, compruebe los bloques que se encuentran en el último nivel de la jerarquía de llamada, p.ej. llamándolos en el OB1 y preparando el entorno del test para el bloque con la función Observar y forzar variables.



Para posicionar puntos de parada y ejecutar el programa por etapas tiene que estar ajustado el modo de funcionamiento "Test" (véase el comando de menú **Test > Modo de funcionamiento**). Dichas funciones de test no se pueden llevar a cabo en el modo "Proceso".

21.1 Visualización del estado de programas

La visualización del **estado de programa** se actualiza cíclicamente. Comienza con el segmento seleccionado.

Identificadores preajustados en KOP y FUP

- El estado se cumple: líneas verdes continuas
- El estado no se cumple: líneas azules punteadas
- El estado es desconocido: líneas negras continuas

Este ajuste (tipo de línea y color) se puede modificar seleccionando el comando de menú **Herramientas > Preferencias / Ficha KOP/FUP**.

Estado de los elementos

- El estado de un contacto
 - se cumple si el valor del operando es "1",
 - no se cumple si el valor del operando es "0",
 - es desconocido si el valor del operando es desconocido.
- El estado de elementos con una salida de habilitación (ENO) corresponde al estado de un contacto con el valor de la salida ENO como operando.
- El estado de elementos con salida Q corresponde al estado de un contacto con el valor del operando.
- El estado en CALLs se cumple si tras la llamada se ha puesto a 1 el bit RB.
- El estado de una operación de salto se cumple si se realiza el salto; es decir, si la condición del salto se cumple.
- Los elementos con salida de habilitación (ENO) se representan en negro si la salida de habilitación no está asignada a una conexión.

Estado de las líneas

- Las líneas son negras si no ha circulado corriente por ellas o si el estado es desconocido.
- El estado de aquellas líneas que comiencen en la barra de alimentación se cumple siempre ("1").
- El estado de aquellas líneas que se encuentren al comienzo de ramas paralelas se cumple siempre ("1").
- El estado de la línea situada tras un elemento se cumple si se cumplen tanto el estado de la línea situada antes del elemento como el estado del elemento.
- El estado de la línea situada tras NOT se cumple si no se cumple el estado de la línea situada antes de NOT (y viceversa).
- El estado de la línea situada **tras** la confluencia de varias líneas se cumple si
 - se cumplen tanto el estado de como mínimo una línea situada **antes** de la confluencia de líneas
 - como el estado de la línea situada antes de la rama.

Estado de los parámetros

- Los valores de los parámetros **en negrita** son actuales.
- Los valores de los parámetros en letra normal provienen de un ciclo anterior; este punto del programa no se ha ejecutado en el ciclo actual.

21.2 Información importante para comprobar en modo Etapa individual / Puntos de parada

Para realizar un test en modo Etapa individual puede

- Editar programas instrucción por instrucción (= etapa individual)
- Posicionar puntos de parada

La función "Test en modo Etapa individual" no se puede llevar a cabo en todos los sistemas de automatización (v. la documentación del sistema de automatización correspondiente)



Requisitos

- Deberá estar ajustado el modo de funcionamiento Test. En el modo de funcionamiento Proceso no se puede realizar el test en modo Etapa individual (v. comando de menú **Test > Modo de funcionamiento**).
- El test en modo Etapa individual sólo se puede realizar en AWL. Para realizar el test de bloques en KOP o FUP tendrá que cambiar de vista (modo de visualización) mediante el comando de menú **Ver > AWL**.
- El bloque no puede estar protegido.
- El bloque debe estar abierto online.
- El bloque abierto no puede ser modificado con el Editor

Número de puntos de parada:

El número de puntos de parada es variable y depende del:

- número de puntos de parada ya posicionados
- número de estados de las variables en curso
- número de estados de los programas en curso

Lea, por favor, en la documentación de su sistema de automatización si éste admite el test en modo Etapa individual.

Los comandos de menú para posicionar, activar o borrar los puntos de parada se encuentran en el menú "Test". Además, dichos comandos de menú se pueden elegir mediante los botones de la barra de puntos de parada. Para visualizar ésta última, elija el comando de menú **Ver > Barra de puntos de parada**.

Funciones de test permitidas:

- Observar/forzar variables
- Información del módulo
- Estado operativo



Peligro

Atención: evite situaciones peligrosas en la instalación durante el estado operativo "PARADA".

21.3 Información importante sobre el estado operativo PARADA

Si el programa llega a un punto de parada, el sistema de automatización cambia al estado operativo "PARADA".

Situación de los diodos luminosos (LEDs) en el estado operativo Parada:

- LED RUN intermitente
- LED STOP encendido

Ejecución de programas en el estado operativo "PARADA":

- En el estado operativo "PARADA" no se puede ejecutar lógica S7, es decir, todos los niveles de ejecución quedan sin reprocesar.
- Se congelan todos los temporizadores:
 - no se ejecutan las celdas de los temporizadores
 - se detienen todos los tiempos de vigilancia
 - se detienen los períodos básicos de los niveles controlados por tiempo
- Se pone en marcha el reloj de tiempo real
- Por razones de seguridad, en el estado operativo "PARADA" se desconectan siempre las salidas (comp. Módulos de salidas "output disable")

Comportamiento durante un corte de alimentación en el estado operativo "PARADA":

- Si se produce un corte de alimentación, al retornar la alimentación de la red eléctrica, los sistemas de automatización respaldados que se encuentran en el estado operativo "PARADA", entran en el estado operativo "STOP" y se mantienen así. La CPU no vuelve a arrancar automáticamente. Partiendo de STOP podrá proseguir con los demás procedimientos (p.ej., activar/desactivar puntos de parada, arrancar manualmente).
- Los sistemas de automatización que no están respaldados no tienen "memoria" y, por eso, cuando retorna la alimentación reaccionan automáticamente (en caliente), independientemente del estado operativo previo.

21.4 Observar el estado de bloques de datos

A partir de la versión 5 de STEP 7 es posible observar un bloque de datos en la vista "Datos" online. La visualización se puede activar tanto desde un bloque de datos online como desde uno offline. En ambos casos se representa el contenido del bloque de datos online del sistema de destino.

El bloque de datos no puede ser modificado antes de iniciar el estado del programa (status). Si hay diferencias estructurales (declaración) entre el bloque de datos online y el bloque de datos offline, es posible cargar el bloque de datos offline en el sistema de destino. El programa visualiza en este caso un mensaje.

El bloque de datos se tiene que encontrar en la vista "Datos" para que se puedan representar los valores online en la columna "Valor actual". Solamente se actualiza aquella parte del bloque de datos que esté visible en la pantalla. Mientras está activo el estado (status) no se puede cambiar a la vista "Declaración".

Durante la actualización se ve una barra verde en la barra de estado del bloque de datos así como el estado operativo del sistema.

Los valores se indican en el formato del tipo de datos que corresponda, no siendo posible cambiar el formato.

Al salir del estado del programa, se vuelve a visualizar en la columna de valores actuales el valor que era válido antes de iniciar el estado del programa. No es posible adoptar los valores online actualizados en el bloque de datos offline.

Actualización de tipos de datos:

Todos los tipos de datos simples se actualizan tanto en un DB global como en todas las declaraciones (in/out/inout/stat) de un bloque de datos de instancia.

Algunos tipos de datos no se pueden actualizar. Estando activado el estado del programa, estos campos se visualizan sobre fondo gris en la columna "Valor actual" e indican valores no actualizados.

- Los tipos de datos compuestos DATE_AND_TIME y STRING no se actualizan.
- En los tipos de datos compuestos ARRAY, STRUCT, UDT, FB, SFB sólo se actualizan aquellos elementos que son tipos de datos simples.
- En la declaración INOUT de un bloque de datos de instancia solamente se representa el puntero que señala al tipo de datos compuesto, pero no sus elementos. El puntero no se actualiza.
- Los tipos de parámetros no se actualizan.

21.5 Definir la visualización del estado del programa

La forma de visualizar el estado del programa puede ser definida en un bloque AWL, FUP o KOP de la siguiente forma:

1. Elija el comando de menú Herramientas > Preferencias.
2. Elija en el cuadro de diálogo "Preferencias" la ficha "AWL" o la ficha "KOP/FUP".
3. Seleccione las opciones deseadas para el test. Se pueden visualizar los siguientes campos de estado:

Al marcar ...	se visualizará:
Bit de estado	Bit de estado, es decir, el bit 2 de la palabra de estado
Resultado lógico	Bit 1 de la palabra de estado; indica el resultado de una operación lógica o de una comparación aritmética.
Estado estándar	Contenido del ACU1.
Registros de direcciones 1/2	Contenido del correspondiente registro de direcciones en el direccionamiento indirecto por registro (intraárea o interárea)
ACU2	Contenido del ACU2
Registros DB 1/2	Contenido del registro del bloque de datos, bien sea del primer o del segundo bloque de datos abierto.
Modo indirecto	Referencia de memoria indirecta; indicación del puntero (dirección), no del contenido de la dirección; posible sólo en el direccionamiento indirecto por memoria, no en el direccionamiento indirecto por registro contenido de una palabra de temporización o de contaje si en la instrucción aparecen las operaciones correspondientes.
Palabra de estado	Todos los bits de estado de la palabra de estado

21.6 Definir el modo de funcionamiento para el test

Procedimiento

1. Hágase mostrar el entorno de test ajustado activando para ello el comando de menú **Test > Modo de funcionamiento**.
2. Elija el modo de funcionamiento deseado. Puede escoger entre "Test" y "Proceso".

Modo de funcionamiento	Explicación
Modo de funcionamiento "Test"	Se pueden realizar todas las funciones de test ilimitadamente. Se puede prolongar considerablemente el tiempo de ciclo de la CPU, ya que por ejemplo cada vez que se recorren los bucles programados se define el estado de las instrucciones que contienen.
Modo de funcionamiento "Proceso"	Se limita la función de test Estado del programa para no sobrecargar excesivamente el tiempo de ciclo. <ul style="list-style-type: none">• Por tanto, no se permite p.ej. ninguna condición de llamada.• La visualización del estado de un bucle programado se interrumpe en el punto de retorno.• Tampoco se pueden ejecutar las funciones de test "PARADA" y la ejecución del programa a pasos individuales.

Nota

Si el modo de funcionamiento ha sido definido al parametrizar la CPU, solamente se podrá modificar cambiando la parametrización de la CPU. En caso contrario puede cambiarlo en el cuadro de diálogo visualizado.

22 Test con el programa de simulación S7-PLCSIM (software opcional)

22.1 Test con el programa de simulación (software opcional)

Con el paquete opcional Simulación puede ejecutar y comprobar su programa en un sistema de automatización simulado existente en su ordenador o unidad de programación (p. ej., Power PG). Dado que la simulación tiene lugar por completo en el software STEP 7, no necesitará hardware S7 (CPU o módulos de señales). Con la CPU S7 simulada puede comprobar programas para CPUs S7-300 y S7-400 y eliminar errores.

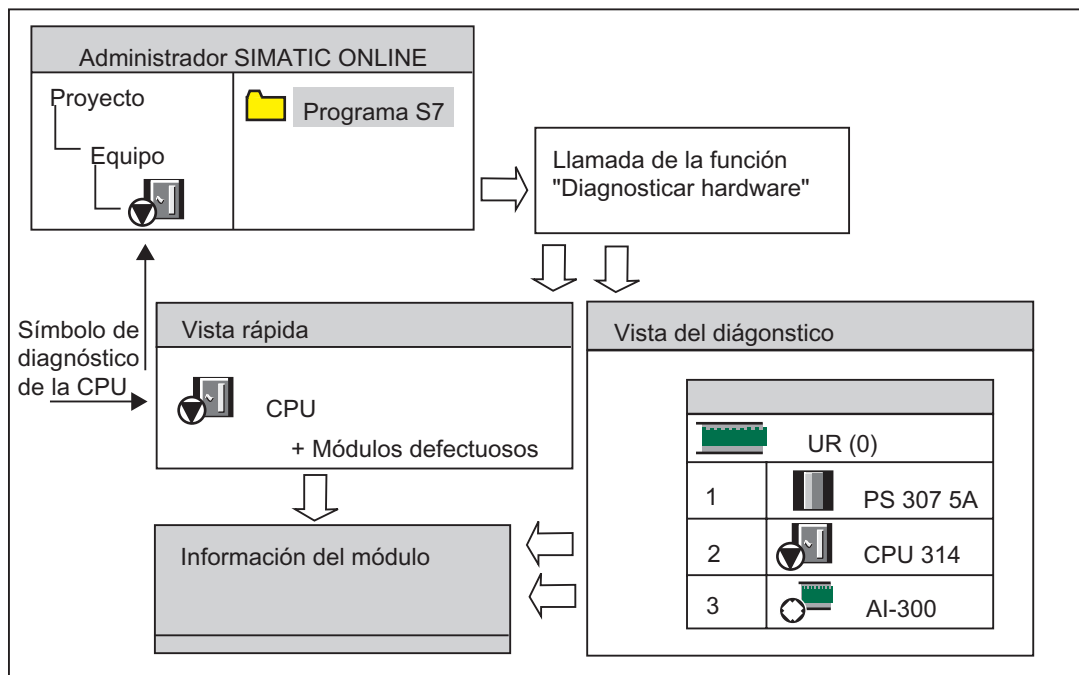
La aplicación dispone de un sencillo interface de usuario para supervisar y modificar los distintos parámetros que se usan en su programa (p.ej. para activar y desactivar las entradas). Además podrá utilizar otras aplicaciones del software de STEP 7 mientras su programa se ejecuta desde la CPU simulada. Así podrá, por ejemplo, observar y modificar variables con la tabla de variables.

23 Diagnóstico

23.1 Diagnóstico del hardware y búsqueda de errores

Los símbolos de diagnóstico indican si existe información de diagnóstico para un módulo determinado. Estos símbolos muestran el estado del módulo correspondiente y, tratándose de CPUs, también indican su estado operativo.

Los símbolos de diagnóstico se visualizan en la ventana del proyecto en la vista "Online" así como tras llamar la función "Diagnosticar hardware" en la vista rápida (ajuste por defecto) o en la vista del diagnóstico. Si requiere información de diagnóstico más detallada vaya a la "Información del módulo" a la que se accede haciendo doble clic en un símbolo de diagnóstico de la vista rápida o de la vista del diagnóstico.



Procedimiento para localizar fallos

1. Abra la ventana online del proyecto con el comando de menú **Ver > Online**.
2. Abra todos los equipos, de manera que se vean los módulos programables que tienen configurados.
3. Determine las CPUs en las que aparece un símbolo de diagnóstico que indique un error o fallo. Acceda con la tecla F1 a la página de la Ayuda que explica el significado de los distintos símbolos.
4. Seleccione el equipo que desea analizar.
5. Elija el comando de menú **Sistema de destino > Diagnóstico/Preferencias > Información del módulo** para hacerse mostrar informaciones sobre el estado de la CPU de este equipo.
6. Elija el comando de menú **Sistema de destino > Diagnóstico/Preferencias > Diagnosticar hardware** para hacerse mostrar la "Vista rápida" con la CPU y los módulos defectuosos de este equipo. La vista rápida aparece por defecto (comando de menú **Herramientas > Preferencias**, ficha "Vista").
7. Seleccione un módulo defectuoso en la vista rápida.
8. Haga clic en el botón de comando "Información del módulo", para obtener información sobre este módulo.
9. Estando en la vista rápida, haga clic en el botón de comando "Abrir equipo online" para hacerse mostrar la vista del diagnóstico. Esta contiene todos los módulos del equipo en el orden en que están enchufados en los slots.
10. Estando en la vista del diagnóstico, haga doble clic en un módulo para hacerse mostrar informaciones sobre el estado del mismo. También obtendrá información sobre aquellos módulos que no están defectuosos y que por consiguiente no aparecen en la vista rápida.

No es necesario realizar todos los pasos, sino que basta realizar los suficientes para acceder a la información que necesita.




23.2 Símbolos de diagnóstico en la vista online

Los símbolos de diagnóstico se visualizan en la vista online del proyecto y en la ventana "Configurar hardware" con la vista online de las tablas de configuración.






Los símbolos de diagnóstico facilitan la detección de fallos. Basta con echar un vistazo al símbolo del módulo para detectar si existen informaciones de diagnóstico al respecto. Si no han ocurrido fallos, los símbolos de los módulos se representarán sin símbolos adicionales de diagnóstico.

Si existe información de diagnóstico para un módulo en su símbolo, se visualizará también un símbolo de diagnóstico. En caso contrario, el icono del módulo aparecerá atenuado.


Símbolos de diagnóstico de los módulos (ejemplo: FM / CPU)

Símbolo	Significado
	La configuración real difiere de la teórica: El módulo configurado no existe o se ha insertado un módulo de otro tipo.
	Error: fallo en el módulo. Posibles causas: detección de una alarma de diagnóstico, de un error de acceso a periferia o de un LED de error.
	El diagnóstico no es posible porque no hay ningún enlace online o porque la CPU no proporciona informaciones de diagnóstico del módulo (p. ej., fuente de alimentación, submódulos).

Símbolos de diagnóstico de los estados operativos (ejemplo: CPU)

Símbolo	Estado operativo
	ARRANQUE
	STOP
	STOP debido al estado operativo "STOP" de otra CPU en modo multiprocesador
	RUN
	PARADA

Símbolo de diagnóstico para el forzado permanente

Símbolo	Estado operativo
	<p>En este módulo se está efectuando un forzado permanente de las variables, es decir que a las variables del respectivo programa de usuario se les han asignado valores fijos que el programa no puede modificar.</p> <p>El símbolo de forzado permanente puede aparecer también junto con otros (aquí junto con el símbolo del estado operativo "RUN").</p>

Actualizar los símbolos de diagnóstico

Tiene que estar activada la ventana en cuestión.

- Pulse la tecla de función F5 o
- Elija en la ventana el comando de menú **Ver > Actualizar**.

23.3 Diagnosticar el hardware: Vista rápida

23.3.1 Acceder a la vista rápida

La vista rápida ofrece un acceso rápido a la función "Diagnosticar hardware" con información menos detallada que la que se indica en la vista del diagnóstico de HWConfig. La vista rápida aparece de forma estándar al llamar la función "Diagnosticar hardware".

Mostrar la vista rápida

Esta función se activa desde el Administrador SIMATIC con el comando de menú **Sistema de destino > Diagnóstico/Preferencias > Diagnosticar hardware**.

El comando de menú se utiliza de la manera siguiente:

- en la ventana online del proyecto, cuando hay un módulo o un programa S7/M7 seleccionado.
- en la ventana "Estaciones accesibles" cuando hay una estación seleccionada ("MPI=...") y esta entrada pertenece a un módulo CPU.

Así podrá seleccionar módulos de las tablas de configuración visualizadas y hacerse mostrar información sobre su estado.

23.3.2 Funciones de información de la vista rápida

En la vista rápida se visualizan las siguientes informaciones:

- Datos sobre el enlace online con la CPU
- Símbolo de diagnóstico de la CPU
- Símbolo de diagnóstico de los módulos en los que la CPU ha detectado un fallo (p. ej. alarma de diagnóstico, error de acceso a la periferia).
- El tipo y la dirección del módulo (bastidor, slot, sistema maestro DP con número de equipo)

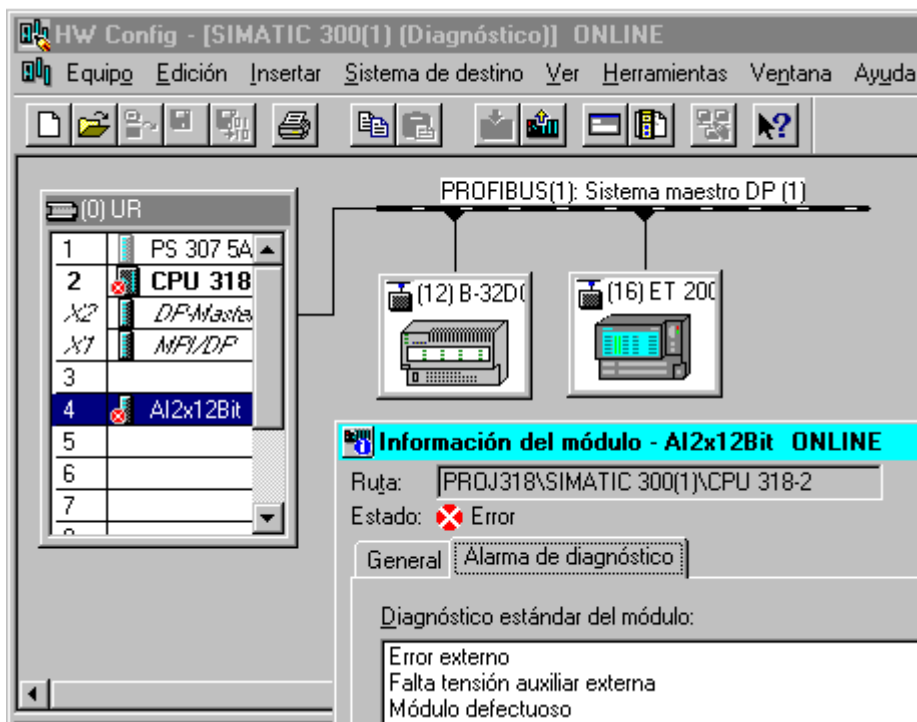
Otras posibilidades de diagnóstico en la vista rápida

- **Información del módulo**
Con el botón de comando "Información del módulo" se abre este cuadro de diálogo. Dependiendo de las funciones de diagnóstico que ofrezca el módulo se visualizarán informaciones detalladas de diagnóstico del módulo que esté seleccionado. Especialmente puede hacerse mostrar las entradas del búfer de diagnóstico.
- **Vista del diagnóstico**
Con el botón de comando "Abrir equipo online" abra el cuadro de diálogo que contiene - a diferencia de la vista rápida - un gráfico con todo el equipo así como con los datos de configuración. Se posiciona en el módulo que esté seleccionado en la lista "CPU / Módulos defectuosos".

23.4 Diagnosticar el hardware: Vista del diagnóstico

23.4.1 Acceder a la vista de diagnóstico de HW Config

Utilizando esta alternativa se puede visualizar el diálogo "Información del módulo" para todos los módulos del bastidor. La vista de diagnóstico (tabla de configuración) muestra la configuración real de un equipo a nivel de bastidor, así como las estaciones DP con sus módulos.



Nota

- Si la tabla de configuración ya está abierta offline, puede acceder a la vista online de las tablas de configuración con el comando de menú **Equipo > Abrir online**.
- El número de fichas visualizadas en el cuadro de diálogo "Información del módulo" depende de si el módulo es apto para diagnóstico o no.
- En la ventana "Estaciones accesibles" se visualizan solamente los módulos con dirección de estación propia (dirección Ethernet, MPI o PROFIBUS).

Llamada en el Administrador SIMATIC desde la vista ONLINE de un proyecto

1. En la vista del proyecto en el Administrador SIMATIC, establezca un enlace online con el sistema de destino, eligiendo el comando de menú **Ver > Online**.
2. Seleccione un equipo y ábralo haciendo doble clic en el mismo.
3. Abra seguidamente el objeto "Hardware" que contiene. Se abrirá la vista de diagnóstico.

Allí podrá seleccionar un módulo y acceder a la información del mismo, eligiendo el comando de menú **Sistema de destino > Diagnóstico/Preferencias > Información del módulo**.

Llamada en el Administrador SIMATIC desde la vista OFFLINE de un proyecto

Proceda como sigue:

1. En la vista del proyecto en el Administrador SIMATIC, seleccione un equipo y ábralo haciendo doble clic en el mismo.
2. Abra seguidamente el objeto "Hardware" que contiene. Se abrirá la tabla de configuración.
3. Elija el comando Equipo > Abrir online.
4. Se abrirá la vista de diagnóstico de HW Config con la configuración del equipo tal y como ha sido determinada en los módulos (p.ej. CPU). El estado de los módulos se representa mediante símbolos. El significado de dichos símbolos se explica en la Ayuda en pantalla. Si hay módulos defectuosos o si faltan módulos configurados, se indican en un cuadro de diálogo aparte. Desde dicho cuadro de diálogo se puede navegar directamente a uno de los módulos visualizados (botón "Ir a").
5. Haga doble clic en el símbolo del módulo cuyo estado le interesa saber. En un cuadro de diálogo compuesto por varias fichas (dependiendo del tipo de módulo) es posible analizar detalladamente su estado.

Llamada en el Administrador SIMATIC desde la ventana "Estaciones accesibles"

Proceda como sigue:

1. Vaya al Administrador SIMATIC y abra la ventana "Estaciones accesibles" con el comando **Sistema de destino > Mostrar estaciones accesibles**.
2. En la ventana "Estaciones accesibles", seleccione una estación.
3. Seleccione a continuación el comando de menú Sistema de destino > Diagnóstico/Preferencias > Diagnosticar hardware.

Nota

En la ventana "Estaciones accesibles" se visualizan solamente los módulos con dirección de estación propia (dirección Ethernet, MPI o PROFIBUS).

23.4.2 Funciones de información de la vista de diagnóstico

En la vista online se visualiza toda la configuración del equipo accesible online. Esta abarca:

- Configuración de los bastidores.
- Símbolos de diagnóstico de **todos** los módulos configurados.
De los símbolos se deduce el estado del módulo en cuestión, y si el módulo es una CPU se indica además su estado operativo.
- Tipo y referencia del módulo así como direcciones y comentarios sobre la configuración.

Otras posibilidades de diagnóstico en la vista del diagnóstico

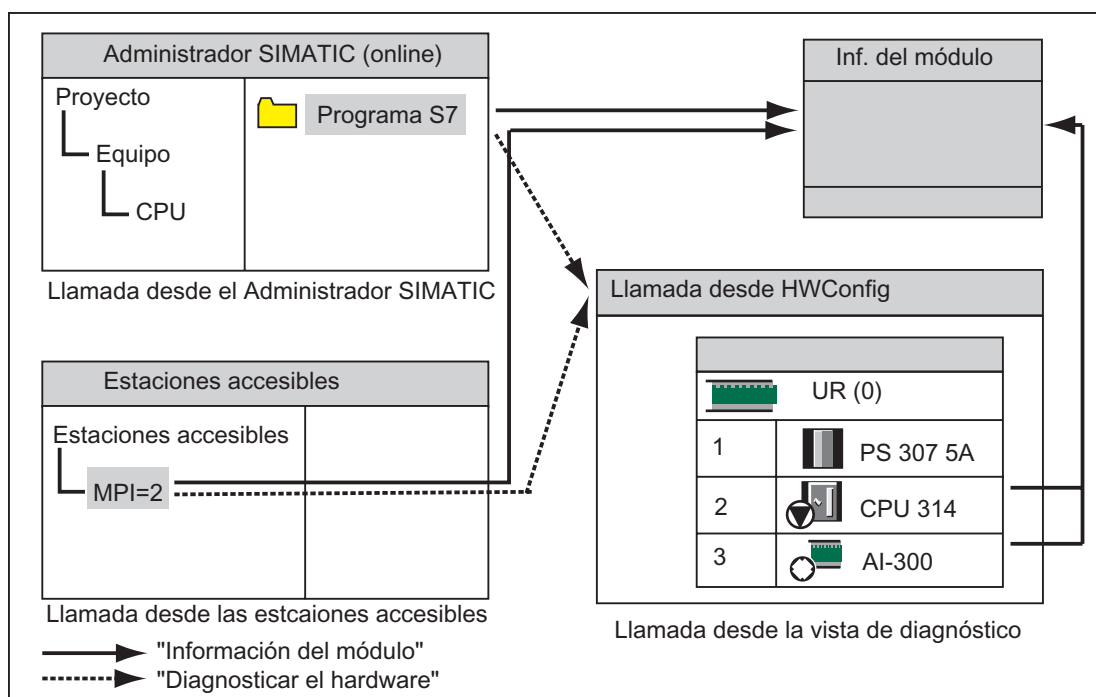
Haciendo doble clic en un módulo puede hacerse mostrar información sobre su estado.

23.5 Información del módulo

23.5.1 Métodos para acceder a la información del módulo

Es posible acceder al cuadro de diálogo "Información del módulo" desde diversos puntos. A continuación se proponen algunos procedimientos como ejemplo para distintos tipos de aplicación de frecuente uso:

- Llamada en el Administrador SIMATIC desde una ventana del proyecto "online" u "offline"
- Llamada en el Administrador SIMATIC desde una ventana "Estaciones accesibles"
- Llamada desde la vista de diagnóstico de HW Config



Para averiguar el estado de un **módulo con dirección de estación propia** necesita un enlace online con el sistema de destino. Lo podrá establecer con el modo de visualización online del proyecto o en la ventana "Estaciones accesibles".

23.5.2 Funciones de la información del módulo

Las funciones de información se encuentran a la disposición en la ficha de ese mismo nombre en el cuadro de diálogo "Información del módulo". En un caso concreto se visualizan únicamente las fichas apropiadas para el módulo seleccionado.

Función	Información	Aplicación
General	Datos de identificación del módulo seleccionado, p.ej. tipo, referencia, versión, estado, slot en el bastidor.	La información online del módulo físico se puede comparar con los datos del módulo configurado.
Búfer de diagnóstico	Panorámica de los eventos residentes en el búfer de diagnóstico e informaciones detalladas sobre el evento seleccionado..	Para evaluar la causa del STOP de una CPU y los eventos que se han presentado en el módulo seleccionado. Gracias al búfer de diagnóstico, los errores de sistema se pueden evaluar incluso al cabo de un tiempo prolongado para averiguar la causa de un STOP o para determinar e interpretar la aparición de determinados eventos de diagnóstico.
Alarma de diagnóstico	Datos de diagnóstico del módulo seleccionado	Para averiguar la causa del fallo de un módulo.
Diagnóstico del esclavo DP	Datos de diagnóstico del esclavo DP seleccionado (según la norma EN50170).	Para averiguar la causa del fallo de un esclavo DP.
Memoria	Ampliación de la memoria, Utilización actual de la memoria de trabajo, de la memoria de carga y de la memoria remanente de la CPU seleccionada o del módulo de función (FM) M7.	Antes de transferir bloques nuevos o ampliados a una CPU, con objeto de comprobar si en dicha CPU/FM se dispone de suficiente memoria de carga, así como para comprimir el contenido de la memoria.
Tiempo de ciclo	Duración del ciclo máximo, mínimo y del último ciclo de la CPU seleccionada o del módulo de función (FM) M7.	Para comprobar el tiempo de ciclo mínimo parametrizado, así como los tiempos de ciclo máximo y actual.
Sistema de reloj	Hora actual, horas de funcionamiento e informaciones respecto a la sincronización del reloj (intervalos de sincronización).	Para visualizar la hora y la fecha de un módulo y para comprobar la sincronización del reloj.
Datos característicos	Áreas de operandos y bloques disponibles del módulo (CPU o FM) seleccionado.	Antes y durante la creación de un programa de usuario y para comprobar si la CPU cumple los requisitos para poder ejecutar dicho programa, p.ej. respecto al tamaño de la imagen del proceso.
	Visualización de todos los tipos de bloques que ofrece el módulo seleccionado. Lista de los OBs, SFBs y SFCs que se pueden utilizar en el módulo en cuestión.	Para comprobar qué bloques estándar puede contener o llamar su programa de usuario para poder ejecutarse en la CPU seleccionada.
Comunicación	Las velocidades de transferencia, los enlaces, la carga de la comunicación, así como el tamaño máximo de los telegramas en el bus K del módulo seleccionado.	Para determinar cuántos y qué enlaces de la CPU o del FM M7 son posibles o están ocupados.

Función	Información	Aplicación
Pilas	Ficha Pilas : sólo se puede activar en estado operativo STOP o PARADA. Se muestra la pila BSTACK del módulo seleccionado. Puede leer además las pilas USTACK y LSTACK y saltar a la posición donde se ha producido el error en el bloque interrumpido.	Para averiguar la causa de un cambio a modo STOP y para corregir un bloque.

Informaciones adicionales

En cada ficha se visualizan las siguientes informaciones adicionales:

- Ruta online del módulo seleccionado
- Estado operativo de la CPU correspondiente (p. ej., RUN, STOP)
- Estado del módulo seleccionado (p. ej., error, OK)
- Estado operativo del módulo seleccionado (p.ej. RUN, STOP), en tanto que dicho módulo disponga de un estado operativo propio (p.ej. CP342-5).

El estado operativo de la CPU y el estado del módulo seleccionado no se pueden visualizar si el cuadro de diálogo "Información del módulo" de un módulo que no sea una CPU se abre desde la ventana "Estaciones accesibles".

Visualizar varios módulos a la vez

Si lo desea, puede hacerse mostrar información sobre distintos módulos a la vez. Para ello deberá cambiar de módulo, seleccionar un módulo diferente y llamar la correspondiente información del mismo. Entonces aparecerá otro grupo de fichas. No obstante, sólo es posible abrir un grupo de fichas por módulo.

Actualizar los indicadores de la información del módulo

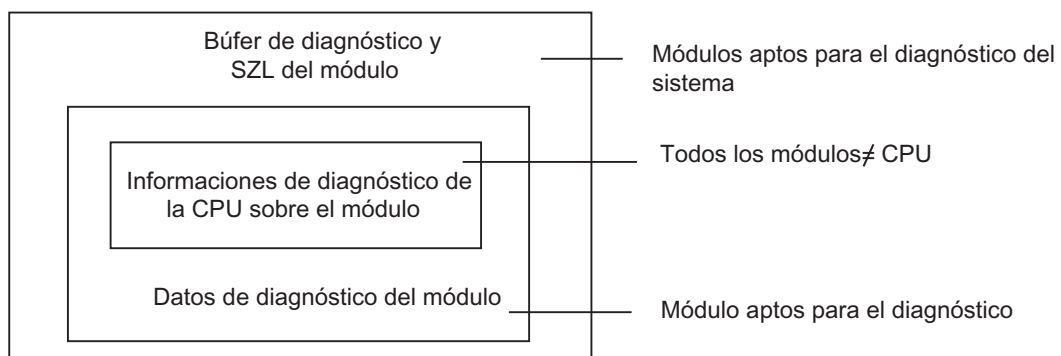
Cada vez que se cambia a otra ficha del cuadro de diálogo "Información del módulo" se vuelven a leer los datos actuales del módulo. Sin embargo, mientras se está leyendo una de las fichas no se actualizan automáticamente los valores. Haciendo clic en el botón de comando "Actualizar" se vuelven a leer los datos del módulo sin cambiar de ficha.

23.5.3 Volumen de información del módulo en función del tipo de módulo

El volumen de información que se puede evaluar y visualizar depende:

- del módulo seleccionado y
- del punto de vista desde el que se consulta la información del módulo.
El volumen total de información se obtiene haciendo la llamada desde la vista online de la tabla de configuración o desde la ventana del proyecto.
Se puede alcanzar un ámbito limitado de información con la ventana "Estaciones accesibles".

Según el volumen de información disponible, los módulos se dividen en módulos aptos para el diagnóstico del sistema, aptos para el diagnóstico y no aptos para el diagnóstico. El gráfico siguiente ilustra esta relación:



- Módulos aptos para el diagnóstico del sistema: FM351 y FM354 (por ejemplo).
- Módulos aptos para el diagnóstico: la mayoría de los módulos de señales (SM) analógicas.
- Módulos no aptos para el diagnóstico: la mayoría de los módulos de señales (SM) digitales.

Fichas disponibles

La tabla indica qué fichas del cuadro de diálogo "Información del módulo" están disponibles para cada tipo.

Ficha	CPU o FM M7	Módulo apto para el diagnóstico del sistema	Módulo apto para el diagnóstico	Módulo no apto para el diagnóstico	Esclavo DP
General	sí	sí	sí	sí	sí
Búfer de diagnóstico	sí	sí	–	–	–
Alarma de diagnóstico	–	sí	sí	–	–
Memoria	sí	–	–	–	–
Tiempo de ciclo	sí	–	–	–	–
Sistema de reloj	sí	–	–	–	–
Datos característicos	sí	–	–	–	–
Pilas	sí	–	–	–	–
Comunicación	sí	–	–	–	–
Diagnóstico del esclavo DP	–	–	–	–	sí
Estado H ¹⁾	sí	–	–	–	–

¹⁾ sólo en CPUs de sistemas H

Además de las informaciones que figuran en las fichas, se visualiza también el estado operativo de los módulos. Si el cuadro de diálogo se llama online desde las tablas de configuración, el estado del módulo se indicará desde el punto de vista de la CPU (p. ej., OK, Error, Módulo no existente).

23.5.4 Visualizar la información de módulo de aparatos de campo PA y esclavos DP a través de un enlace Y

A partir de STEP 7 V5.1, Servicepack 3, es posible evaluar la información del módulo de esclavos DP y aparatos de campo PA "a través" de un enlace DP/PA (IM 157).

Esto afecta a las siguientes configuraciones:

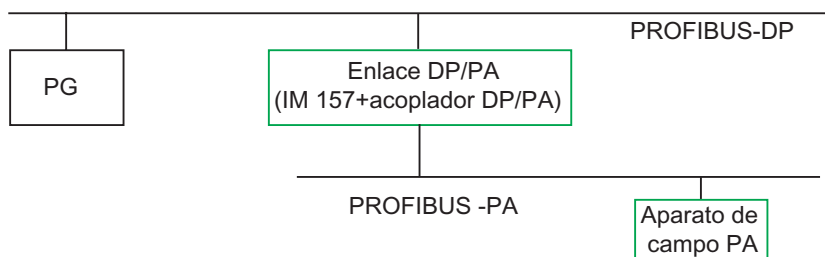
- IM 157 con acopladores DP/PA para la conexión del PA PROFIBUS
- IM 157 como módulo de interconexión expandible redundante para la conexión de un DP PROFIBUS no redundante ("enlace Y")

Con estos ajustes, la PG estará conectada a la misma subred PROFIBUS que el enlace DP/PA.

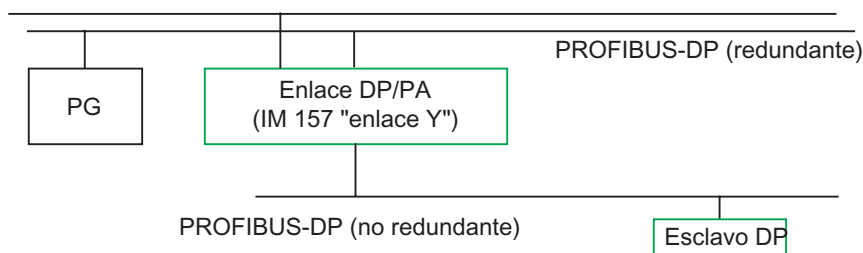
Existe además una posibilidad de configuración. En ella se conecta la PG a Industrial Ethernet y se enruta un equipo S7-400 a la subred PROFIBUS.

En la siguiente figura se muestran los requisitos que se han de cumplir.

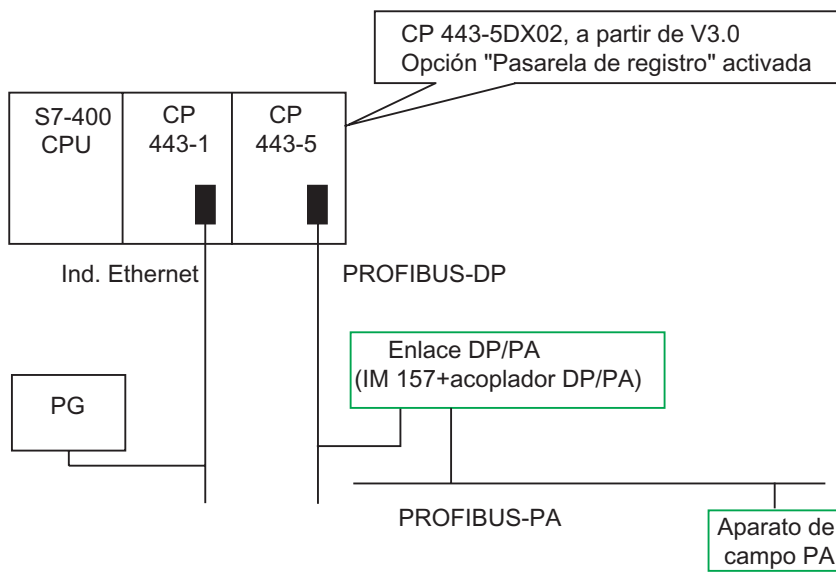
IM 157 con acopladores DP/PA para la conexión del PA PROFIBUS



IM 157 como enlace Y



PG en Industrial Ethernet



23.6 Diagnóstico en el estado operativo STOP

23.6.1 Procedimiento básico para averiguar la causa de un STOP

Para averiguar por qué la CPU ha pasado a STOP proceda como sigue:

1. Seleccione la CPU que ha pasado al estado operativo STOP.
2. Elija el comando Sistema de destino > Diagnóstico/Preferencias > Información del módulo.
3. Seleccione la ficha "Búfer de diagnóstico".
4. A través de las últimas entradas podrá averiguar la causa del STOP.

En caso de errores de programación:

1. La entrada "STOP por OB de error de programación no cargado" significa p.ej. que la CPU ha detectado un error de programación, habiendo intentado arrancar el OB (no existente) para remediar dicho error. La entrada precedente indica el error de programación en sí.
2. Seleccione el aviso relativo al error de programación.
3. Haga clic en el botón "Abrir bloque".
4. Seleccione la ficha "Pilas".

23.6.2 Contenido de las pilas en estado operativo STOP

Sirviéndose del búfer de diagnóstico y el contenido de las pilas podrá averiguar la causa de los distintos errores de ejecución de un programa de usuario.

Si la CPU ha pasado al estado operativo "STOP" p.ej. debido a un error de programación o al comando de parada, en la ficha "Pilas" de la información del módulo se visualizará la pila BSTACK (pila de bloques). El contenido de las demás pilas se puede visualizar mediante los botones "USTACK", "LSTACK" y "Pila de paréntesis". Los contenidos de las pilas indican qué instrucción en qué bloque ha causado el STOP de la CPU.

Contenido de la pila BSTACK

En la pila BSTACK se indican los bloques que se llamaron antes de que la CPU cambiara al modo STOP y que todavía no se han terminado de ejecutar.

Contenido de la pila USTACK

Si hace clic en el botón "USTACK", se visualizan los datos del punto de interrupción. La pila de interrupción (USTACK) contiene los datos o estados que eran válidos cuando se produjo la interrupción, como p.ej.

- contenido de los acumuladores y de los registros
- DBs abiertos y su tamaño
- contenido de la palabra de estado
- prioridad
- bloque interrumpido
- bloque donde continuaría la ejecución del programa después de la interrupción.

Contenido de la pila LSTACK

Es posible seleccionar un bloque cualquiera de la pila BSTACK y, haciendo clic en el botón "LSTACK", visualizar los datos locales correspondientes.

La pila de datos locales (LSTACK) contiene los valores de los datos locales de los bloques que ha utilizado el programa de usuario hasta su interrupción.

Para poder interpretar y evaluar los datos locales visualizados es necesario conocer muy bien el sistema. La primera parte de los datos visualizados corresponde a las variables temporales del bloque.

Contenido de la pila de paréntesis

Activando el botón "Pila de paréntesis" se visualiza el contenido de la pila de paréntesis en el punto de interrupción.

La pila de paréntesis es un área de memoria que utilizan las operaciones lógicas **U(**, **UN(**, **O(**, **ON(**, **X(** y **XN(**.

El botón de comando sólo se activa cuando aún quedan paréntesis sin cerrar al producirse la interrupción.

23.7 Control de los tiempos de ciclo para evitar errores de tiempo

La ficha "Tiempo de ciclo" de la información del módulo indica los tiempos de ciclo del programa de usuario.

Si la duración del ciclo más largo se aproxima al tiempo de vigilancia, es posible que las fluctuaciones del tiempo de ciclo causen un error de tiempo. Para evitarlo, prolongue el tiempo de ciclo máximo del programa de usuario.

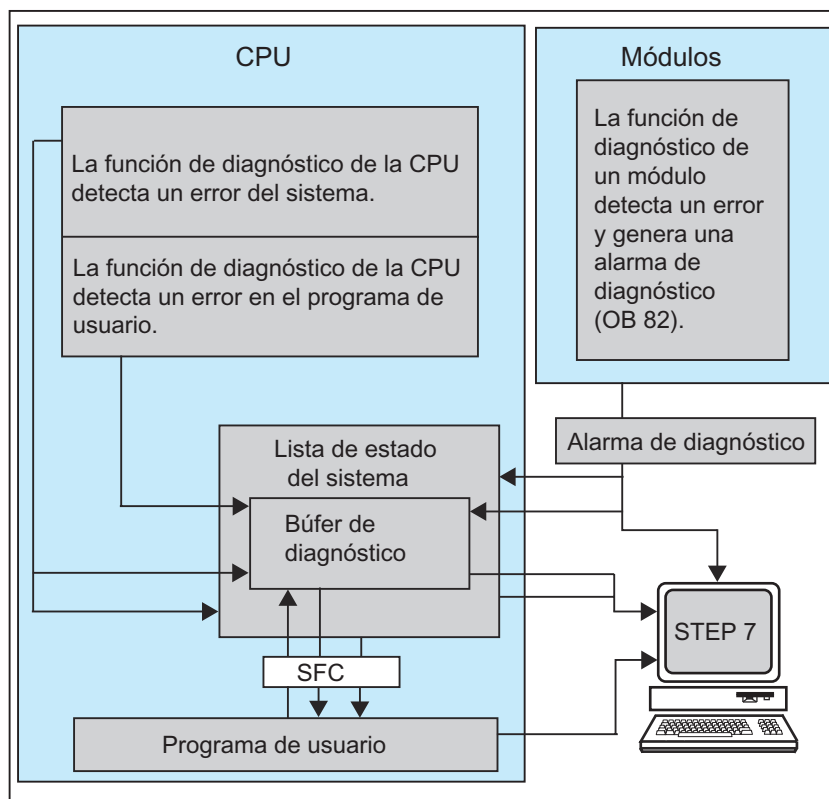
Si el ciclo dura menos que el tiempo mínimo parametrizado, la CPU o el FM prolongará automáticamente el ciclo al tiempo mínimo parametrizado. Si se trata de una CPU, durante el tiempo que se prolongue el ciclo se ejecutará el OB de tarea no prioritaria (OB 90), si es que está cargado.

Ajustar el tiempo de ciclo

Los tiempos de ciclo máximo y mínimo se pueden ajustar al configurar el hardware. A tal efecto, en la vista offline de la tabla de configuración, haga doble clic en la CPU (o en el FM) para definir sus propiedades. Los valores deseados se pueden ajustar en la ficha "Ciclo/marca de ciclo".

23.8 Transmisión de informaciones de diagnóstico

La figura siguiente muestra la transmisión de informaciones de diagnóstico en SIMATIC S7.



Leer las informaciones de diagnóstico

Desde el programa de usuario es posible leer las entradas de diagnóstico utilizando la SFC 51 RDSYSST o visualizar en texto explícito los mensajes de diagnóstico con STEP 7.

Estos ofrecen informaciones sobre:

- dónde y cuándo ocurrió el error
- a qué tipo de eventos de diagnóstico pertenece la entrada (evento de diagnóstico de usuario, error síncrono/asíncrono, cambio de estado operativo).

Generar mensajes colectivos del sistema de control

La CPU inscribe los eventos de diagnóstico estándar y de diagnóstico ampliado en el búfer de diagnóstico. Además, en el caso de eventos de diagnóstico estándar genera un mensaje del sistema de control siempre que se cumplan las condiciones siguientes:

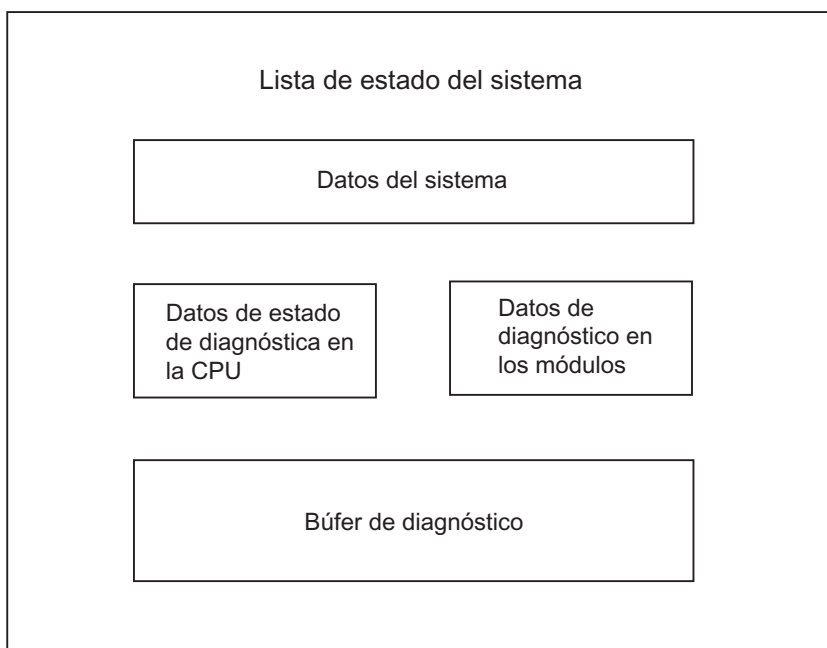
- Con STEP 7 se ha definido la generación de mensajes colectivos del sistema de control.
- Para la CPU está registrado como mínimo un visualizador para la presentación en el mismo de los mensajes colectivos del sistema de control.
- Los mensajes colectivos del sistema de control sólo se forman cuando aún no existe ningún mensaje colectivo del sistema de control de la clase asociada (hay 7 clases).
- Por cada clase es posible generar un mensaje colectivo del sistema de control.

23.8.1 Lista de estado del sistema (SZL)

La lista de estado del sistema SZL describe el estado actual del sistema de automatización: ofrece una visión general sobre la configuración, la parametrización actual, los estados y procesos actuales de la CPU y los módulos asociados.

Los datos de la SZL sólo se pueden leer y no se pueden modificar. Es una lista virtual que solamente se genera cuando se solicita.

Las informaciones que se pueden emitir a través de la SZL se subdividen en cuatro grupos:



Leer la lista de estado del sistema

Se dispone de dos posibilidades para leer las informaciones de la lista de estado del sistema SZL:

- implícitamente a través de los comandos de menú de STEP 7 desde la unidad de programación (p.ej. capacidad de memoria, datos estáticos de la CPU, búfer de diagnóstico, visualizaciones de estado),
- explícitamente mediante la función de sistema SFC 51 RDSYSST desde el programa de usuario, indicando el número de referencia deseado (v. Ayuda sobre bloques).

Datos del sistema de la SZL

Los datos del sistema son datos identificadores de una CPU fijos o parametrizados. La tabla siguiente muestra qué informaciones se pueden emitir (listas parciales de la SZL) para los diferentes grupos de temas:

Grupo	Información
Identificación de módulos	Referencia, identificación de tipo y versión del módulo
Características de la CPU	Sistema de reloj, comportamiento del sistema (p.ej. modo multiprocesador) y descripción del lenguaje de la CPU
Áreas de memoria	Capacidad de memoria del módulo (p.ej. capacidad de la memoria de trabajo)
Áreas del sistema	Memoria de sistema del módulo (p.ej. cantidad de marcas, temporizadores, contadores, tipo de memoria)
Tipos de bloques	Qué tipos de bloques (OB, DB, SDB, FC, FB) están contenidos en el módulo, cantidad máxima contenida de bloques de un tipo y tamaño máximo de un tipo de bloque
Asignación de errores de alarma	Asignación de alarmas/errores a los OBs
Estado de alarmas	Generación/tratamiento de alarmas actuales
Estado de prioridades	Qué OB está en ejecución, qué prioridad ha sido inhibida por parametrización
Estado operativo y cambio de estados operativos	Qué estados operativos son posibles, último cambio, estado operativo actual

Datos de estado de diagnóstico en la CPU

Los datos de estado de diagnóstico describen el estado actual de los componentes vigilados por el diagnóstico del sistema. La tabla siguiente muestra qué informaciones se pueden emitir (listas parciales de la SZL) para los diferentes grupos de temas:

Grupo	Información
Datos de estado de comunicación	Funciones de comunicación actualmente ajustadas en el sistema
Estación de diagnóstico	Módulos diagnosticables registrados en la CPU
Lista de información de arranque del OB	Informaciones de arranque a los OBs de la CPU
Lista de eventos de arranque	Eventos de arranque y prioridades de los OBs
Información de estado de los módulos	Informaciones de estado de todos los módulos insertados, perturbados, generadores de alarmas de proceso y asociados

Datos de diagnóstico en los módulos

Además de la CPU, existen otros módulos diagnosticables (SM, CP, FM), cuyos datos de diagnóstico se registran en la lista de estado del sistema. La tabla siguiente muestra qué informaciones se pueden emitir (listas parciales de la SZL) para los diferentes grupos de temas:

Grupo	Información
Información de diagnóstico de módulos	Dirección inicial de módulos, errores internos/externos, error de canal, error de parametrización (4 bytes)
Datos de diagnóstico de módulos	Todos los datos de diagnóstico de un módulo determinado

23.8.2 Enviar avisos de diagnóstico personalizados

El diagnóstico del sistema estándar de SIMATIC S7 se puede ampliar adicionalmente a través de la función del sistema SFC 52 WR_USMSG, que permite

- registrar informaciones de diagnóstico personalizadas (p. ej., informaciones sobre la ejecución del programa de usuario) en el búfer de diagnóstico
- enviar avisos de diagnóstico personalizados a las estaciones registradas (visualizadores como PG, OP, TD).

Eventos de diagnóstico personalizados

Los eventos de diagnóstico están subdivididos en las clases de evento 1 a F. Los eventos de diagnóstico personalizados pertenecen a las clases de evento 8 a B y se subdividen en dos grupos:

- las clases de evento 8 y 9 comprenden los avisos con número predeterminado y texto predefinido que se puede consultar mediante su número asociado.
- las clases de evento A y B comprenden avisos con número (A000 a A0FF, B000 a B0FF) y texto que puede elegir a voluntad.

Enviar avisos de diagnóstico a estaciones

Además de registrar una entrada para un evento de diagnóstico personalizado en el búfer de diagnóstico, la SFC 52 WR_USMSG permite también enviar los eventos de diagnóstico personalizado a visualizadores registrados. Al llamar la SFC 52 con SEND= 1, el aviso de diagnóstico se registra en el búfer de emisión y se envía automáticamente a la estación o estaciones registradas en la CPU.

Si no se puede enviar (p.ej. porque no se ha registrado ninguna estación o porque el búfer de emisión está lleno), el evento de diagnóstico personalizado se registrará sin embargo en el búfer de diagnóstico.

Generar aviso con indicación de acuse

Si acusa un evento de diagnóstico personalizado y desea detectar dicho acuse mediante el programa, proceda de la forma siguiente:

- ponga a 1 una variable del tipo BOOL al entrar el evento y póngala a 0 cuando el evento salga.
- vigile dicha variable con ayuda del SFB 33 ALARM.

23.8.3 Funciones de diagnóstico

El diagnóstico del sistema consiste en el reconocimiento, la valoración y la notificación de errores aparecidos en el sistema de automatización. A tal efecto, cada CPU y cada módulo apto para el diagnóstico (p.ej. FM354) dispone de un búfer de diagnóstico en el que se depositan informaciones detalladas sobre todos los eventos de diagnóstico en su orden de aparición.

Eventos de diagnóstico

Los eventos de diagnóstico pueden ser, p. ej.:

- errores internos y externos de un módulo
- errores de sistema en la CPU
- cambios de estado operativo (p. ej., de RUN a STOP)
- error en el programa de usuario
- extraer/insertar módulos
- avisos personalizados introducidos con la función de sistema SFC52

El contenido del búfer de diagnóstico se mantiene incluso después del borrado total de la CPU. Gracias al búfer de diagnóstico, los errores de sistema se pueden evaluar incluso al cabo de un tiempo prolongado para averiguar la causa de un STOP o para determinar e interpretar la aparición de determinados eventos de diagnóstico.

Registro de los datos de diagnóstico

El registro de los datos de diagnóstico a través del diagnóstico del sistema no tiene que ser programado, está disponible de modo estándar y funciona automáticamente. SIMATIC S7 ofrece diferentes funciones de diagnóstico. Algunas de ellas están integradas en la CPU, otras son ofrecidas por los módulos (SM, CP y FM).

Visualización de errores

Los errores internos y externos de los módulos se visualizan a través de LED en los paneles frontales del módulo correspondiente. Las visualizaciones por LED y su evaluación se describen en los manuales sobre el hardware S7. En los sistemas de automatización S7-300, los errores internos y externos se resumen en un error colectivo.

La CPU detecta errores del sistema, así como errores en el programa de usuario, y registra los eventos de diagnóstico en la lista de estado del sistema en el búfer de diagnóstico. Estos avisos de diagnóstico se pueden leer en la PG.

Los módulos de función y de señales diagnosticables detectan errores internos y externos de módulo y generan una alarma de diagnóstico, ante la cual se puede reaccionar con un OB de alarmas.

23.9 Medidas en el programa para tratar fallos

Al detectar errores en la ejecución del programa (errores síncronos) y errores en el sistema de automatización (errores asíncronos), la CPU llama el OB de error correspondiente:

Error detectado	OB de error
Error de redundancia en periferia	OB 70
Error de redundancia en CPU	OB 72
Error de tiempo	OB 80
Fallo de alimentación	OB 81
Alarma de diagnóstico	OB 82
Alarma de extraer/insertar	OB 83
Error de hardware CPU	OB 84
Error de ejecución del programa	OB 85
Fallo en el bastidor o fallo de un equipo de la periferia descentralizada	OB 86
Error de comunicación	OB 87
Error de programación	OB 121
Errores de acceso a periferia	OB 122

Si el OB en cuestión no existe, la CPU pasará al estado operativo "STOP" (excepciones: OB 70, OB 72, OB 81, OB 87). En caso contrario, es posible programar instrucciones en el OB, indicando cómo se debe reaccionar al error. Así se pueden reducir los efectos del error o remediarlo.

Procedimiento general

Crear y abrir el OB

1. Llame la información del módulo de su CPU.
2. Elija la ficha "Datos característicos".
3. En la lista visualizada, consulte si el OB a programar es admisible para su CPU.
4. Inserte el OB en la carpeta "Bloques" de su programa y abra este último.
5. Introduzca el programa para corregir el error.
6. Cargue el OB en el sistema de destino.

Programar las medidas para el tratamiento de errores

1. Evalúe los datos locales del OB para averiguar la causa exacta del error. Las variables OB8x_FLT_ID u OB12x_SW_FLT de los datos locales contienen el código de error. Su significado se explica en el manual de referencia "Funciones estándar y funciones de sistema".
2. Bifurque a la parte del programa donde se debe reaccionar a dicho error.

En el tema "Ejemplo del diagnóstico de módulos con la SFC 51 (RDSYSST)" contenido en los temas de Ayuda de las funciones estándar y de las funciones del sistema se indica un ejemplo de cómo tratar las alarmas de diagnóstico.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

23.9.1 Evaluar el parámetro de salida RET_VAL

Cualquier función del sistema (SFC) muestra, a través del parámetro de salida RET_VAL (valor de respuesta), si la CPU la ha ejecutado correctamente.

Informaciones de error en el valor de respuesta

El valor de respuesta es de tipo entero (INT). El signo de un entero indica si se trata de un número entero positivo o negativo. La relación del valor de respuesta con respecto a "0" indica si ha ocurrido un error al ejecutarse la función (v. también tabla 11-5):

- Si al ejecutarse la función ocurre un error, el valor de respuesta es inferior a 0. El bit de signo del número entero es "1".
- Si la función se ejecuta sin error, el valor de respuesta es mayor o igual a 0. El bit de signo del entero es "0".

Ejecución de la SFC por la CPU	Valor de respuesta	Signo del número entero
errónea	inferior a "0"	negativo (el bit de signo es "1")
sin error	mayor o igual a "0"	positivo (el bit de signo es "0")

Reaccionar a informaciones de error

Si ocurre un error al ejecutar una SFC, ésta emite un código de error a través del valor de respuesta RET_VAL.

Se distingue entre:

- un código de error general, que puede ser emitido por todas las SFCs y
- un código de error específico, que puede ser emitido por una SFC conforme a sus funciones específicas.

Emisión del valor de la función

Algunas SFCs utilizan el parámetro de salida RET_VAL para emitir el valor de la función. Por ejemplo, la SFC 64 TIME_TCK emite el tiempo (hora) del sistema leído a través de RET_VAL.

Para más información sobre el parámetro RET_VAL consulte la Ayuda de los SFBs/SFCs.

23.9.2 OBs de error para reaccionar a errores detectados

Errores detectables

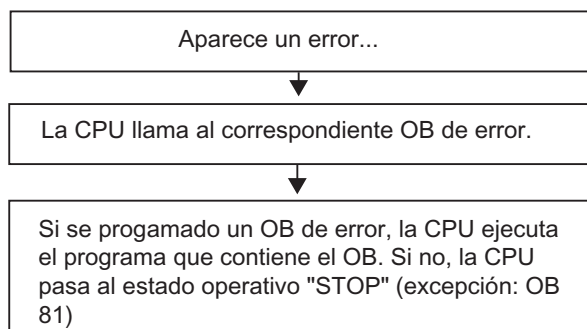
El programa del sistema puede detectar los errores siguientes:

- funcionamiento anómalo de la CPU
- error en la ejecución del programa de sistema
- error en el programa de usuario
- error en la periferia

Según el tipo de error, se conmuta la CPU al estado STOP o se llama un OB de error.

Programar las reacciones

Es posible crear programas para reaccionar a los distintos tipos de errores y definir así el comportamiento de la CPU. El programa para un error determinado se puede memorizar entonces en un OB (de tratamiento) de errores. Al llamar éste, se ejecuta entonces el programa.



OBs de error

Se distingue entre errores síncronos y asíncronos:

- Los errores síncronos se pueden asignar a una instrucción MC7 (p. ej., instrucción de carga para un módulo de señales extraído).
- Los errores asíncronos se pueden asignar a una prioridad o a todo el sistema de automatización (p. ej., desbordamiento de ciclo).

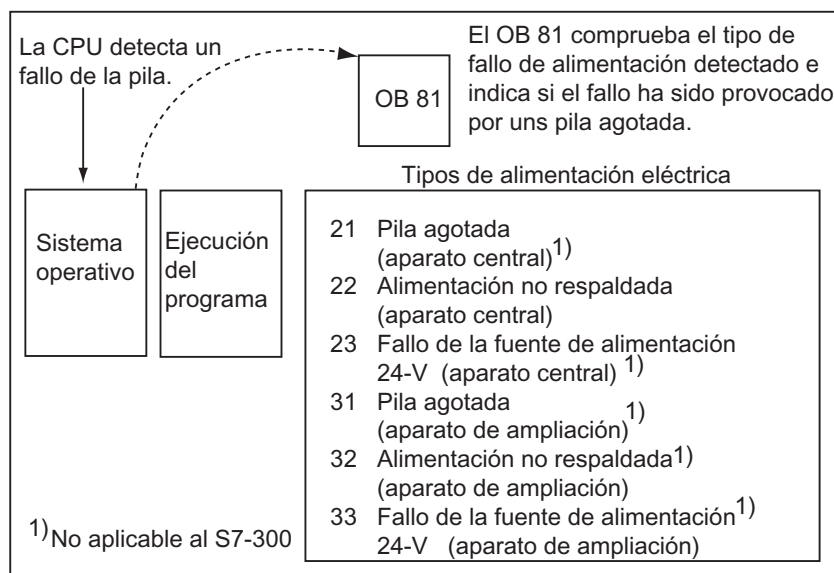
La tabla siguiente muestra qué tipos de errores ocurren generalmente. En el manual "Sistema de automatización S7-300, Configuración, instalación y datos de la CPU" o en el manual de referencia "Sistemas de automatización S7-400/M7-400, Configuración e instalación" puede consultar si su CPU asiste los OBs indicados.

Tipo de error	Clase de error	OB	Prioridad
Redundancia	Error de redundancia en periferia (sólo en CPUs H)	OB 70	25
	Error de redundancia en CPU (sólo en CPUs H)	OB 72	28
Asíncrono	Error de tiempo	OB 80	26
	Error de alimentación	OB 81	(ó 28, si el OB de error está en el
	Alarma de diagnóstico	OB 82	programa de arranque)
	Alarma de extraer/insertar	OB 83	
	Error de hardware CPU	OB 84	
	Error de ejecución del programa	OB 85	
	Error de bastidor	OB 86	
Síncrono	Error de programación	OB 121	Prioridad del OB que ha ocasionado el error
	Error de acceso	OB 122	

Ejemplo de aplicación del OB 81

Los datos locales (información de arranque) del OB de error permiten evaluar la clase del error ocurrido.

Si, p. ej., la CPU detecta un error de pila tampón, entonces el sistema operativo llama el OB 81 (v. fig.).



Es posible escribir un programa que evalúe el código del evento que ha provocado la llamada del OB 81. También es posible escribir un programa que provoque una reacción determinada, p. ej., activar una salida conectada a una lámpara de la estación de operador.

Datos locales del OB 81 de error

La tabla siguiente describe las variables temporales (TEMP) definidas en la tabla de declaración de variables del OB 81.

El símbolo *Fallo de la pila* (BOOL) también se deberá identificar como salida en la tabla de símbolos (p.ej. A 4.0), de manera que las demás partes del programa puedan acceder también a dichos datos.

Declaración	Designación	Tipo	Descripción
TEMP	OB81_EV_CLASS	BYTE	Clase de error/identificador de error 39xx
TEMP	OB81_FLT_ID	BYTE	Código de error: b#16#21 = Por lo menos una pila tampón del bastidor central vacía ¹ b#16#22 = Falta tensión de respaldo en el bastidor central b#16#23 = Fallo de la alimentación de 24 V en el bastidor central ¹ b#16#31 = Por lo menos una pila tampón vacía en un bastidor de ampliación ¹ b#16#32 = Falta tensión de respaldo en un bastidor de ampliación ¹ b#16#33 = Fallo de la alimentación de 24 V en un bastidor de alimentación ¹
TEMP	OB81_PRIORITY	BYTE	Prioridad = 26/28
TEMP	OB81_OB_NUMBR	BYTE	81 = OB 81
TEMP	OB81_RESERVED_1	BYTE	Reservado
TEMP	OB81_RESERVED_2	BYTE	Reservado
TEMP	OB81_MDL_ADDR	INT	Reservado
TEMP	OB81_RESERVED_3	BYTE	Sólo relevante para los códigos de error B#16#31, B#16#32, B#16#33
TEMP	OB81_RESERVED_4	BYTE	
TEMP	OB81_RESERVED_5	BYTE	
TEMP	OB81_RESERVED_6	BYTE	
TEMP	OB81_DATE_TIME	DATE_AND TIME	Fecha y hora del arranque del OB
¹⁾ = No en el S7-300.			

Programa de ejemplo para el OB de tratamiento de errores

Con un programa AWL se ejemplificamos cómo leer el código de error en el OB 81.

El programa está estructurado de la siguiente forma:

- El código de error en el OB 81 (OB81_FLT_ID) se lee y compara con el valor del evento "pila vacía" (B#16#3921).
- Si el código de error corresponde al código para "pila vacía", entonces el programa salta a la meta FPil y activa la salida *Error de pila*.
- Si el código de error no corresponde al código para "pila vacía", entonces el programa compara dicho código con el código para "fallo de pila".
- Si el código de error corresponde al código para "fallo de pila", entonces el programa salta a la meta FPil y activa la salida *Fallo de pila*. En caso contrario se finaliza el bloque.

AWL	Descripción
L B#16#21	//Comparar código de evento "Pila agotada"
	//(B#16#21) con
L #OB81_FLT_ID	//el código de error para el OB 81.
==I	//Si es idéntico (pila agotada),
	//saltar a BF.
SPB BF	
L B#16#22	//Comparar código de evento "Fallo de pila"
	//(b#16#22) con
==I	//el código de error para el OB 81.
SPB BF	//Si es idéntico, saltar a BF.
BEA	//No hay aviso sobre error de pila
BF: L B#16#39	//Comparar código de evento entrante
L #OB81_EV_CLASS	//con el código de error para el OB 81.
==I	//si se detecta un fallo de la pila o que la pila
	//está agotada,
S Batteriefehler	//activar el bit de error de pila
	//(Variable de la tabla de símbolos)
L B#16#38	//Comparar código de evento saliente
==I	//con el código de error para el OB 81.
R Batteriefehler	//Borrar error de pila cuando se haya eliminado.

Para más información sobre los OBs, SFBs y SFCs y explicaciones sobre las ID de eventos consulte la Ayuda de bloques.

23.9.3 Insertar valores de sustitución al detectar errores

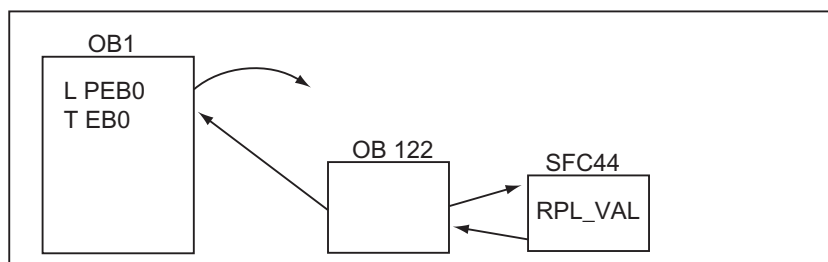
Para determinados tipos de error (p.ej. rotura de hilo en señal de entrada) se pueden asignar valores de sustitución para aquellos valores que quedan indisponibles a causa del error. Existen dos posibilidades para asignar valores de sustitución:

- Los valores de sustitución se pueden parametrizar con STEP 7 para módulos de salidas parametrizables. Los módulos de salidas no parametrizables tienen preajustado el valor de sustitución 0.
- La SFC 44 RPL_VAL permite programar valores de sustitución en OBs de errores (sólo para módulos de entradas).

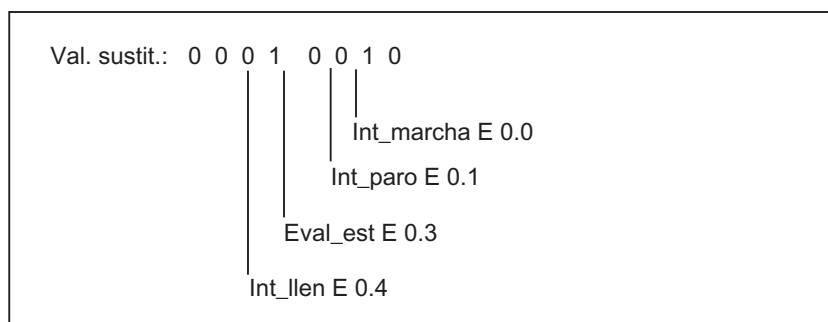
Para todas las instrucciones de carga que producen errores síncronos se puede asignar en el OB de error un valor de sustitución para el contenido del ACU.

Programa de ejemplo para sustituir un valor

En el siguiente programa de ejemplo se ofrece un valor de sustitución en la SFC 44 RPL_VAL. La figura siguiente muestra cómo se puede llamar el OB 122 cuando la CPU detecta que un módulo de entradas no reacciona.



En este ejemplo, el valor de sustitución de la figura siguiente se introduce para que el programa pueda seguir corriendo con valores razonables.



Si falla un módulo de entradas, el tratamiento de la instrucción L PEB0 genera un error síncrono y arranca el OB 122. Por ajuste estándar, la instrucción de carga lee el valor 0. No obstante, la SFC 44 permite definir valores de sustitución discretos adecuados para el proceso. La SFC sustituye el contenido del ACU por el valor de sustitución preajustado.

El siguiente programa de ejemplo podría estar memorizado en el OB 122. La siguiente tabla muestra las variables temporales que en este caso deben definirse en la tabla de declaración de variables del OB 122.

Declaración	Designación	Tipo	Descripción
TEMP	OB122_EV_CLASS	BYTE	Clase de error/identificador de error 29xx
TEMP	OB122_SW_FLT	BYTE	Código de error: 16#42, 16#43, 16#44 ¹⁾ , 16#45 ¹⁾
TEMP	OB122_PRIORITY	BYTE	Clase de prioridad = Prioridad del OB en el cual ha ocurrido el error
TEMP	OB122_OB_NUMBR	BYTE	122 = OB 122
TEMP	OB122_BLK_TYPE	BYTE	Tipo de bloque en el cual ha ocurrido el error
TEMP	OB122_MEM_AREA	BYTE	Area de memoria y tipo de acceso
TEMP	OB122_MEM_ADDR	WORD	Dirección de memoria en la cual ha ocurrido el error
TEMP	OB122_BLK_NUM	WORD	Número del bloque en el cual ha ocurrido el error
TEMP	OB122_PRG_ADDR	WORD	Dirección relativa de la instrucción que ha provocado el error
TEMP	OB122_DATE_TIME	DATE_AND_TIME	Fecha y hora del arranque del OB
TEMP	Error	INT	Memoriza el código de error de la SFC44

¹⁾ = No en el S7-300.

AWL	Descripción
L B#16#2942	Comparar el código de evento del OB 122 con el código de evento (B#16#2942) para el acuse de un error de tiempo al leer la periferia. Si es igual, saltar a "QFeh".
L #OB122_SW_FLT	
==I	Comparar el código de evento del OB 122 con el código de evento (B#16#2943) para un error de direccionamiento (escribir en un módulo que no existe). Si es diferente, saltar a "Stop".
SPB QFeh	
L B#16#2943	Meta "QFeh": transfiere DW#16#2912 (binario 10010) a la SFC44 (REPL_VAL). La SFC44 carga este valor en el ACU 1 (y sustituye el valor que ha provocado la llamada del OB 122). Memoriza el código de error de la SFC en #Error.
<> I	
SPB Stop	Compara #Error con 0 (si es igual, entonces no ha ocurrido ningún error al tratar el OB 122). Finalizar el bloque si no ha ocurrido ningún error.
QFeh: CALL "REPL_VAL"	
VAL := DW#16#2912	Meta "Stop": llama la SFC46 "STP" y lleva la CPU al estado operativo STOP.
RET_VAL := #Error	
L #Error	
L 0	
==I	
BEB	
Stop: CALL "STP"	

23.9.4 Error de redundancia en periferia (OB 70)

Descripción

El sistema operativo de la CPU H llama el OB 70 al presentarse una pérdida de redundancia en el DP PROFIBUS (p. ej. en caso de un fallo de bus del maestro DP activo o de un error en el módulo de interconexión (interfase) del esclavo DP) o al cambiar el maestro DP activo de esclavos DP con periferia conmutada.

Programar el OB 70

El OB 70 se debe generar como objeto en el programa STEP 7. Escribir el programa, que se debe ejecutar en el OB 70, en el bloque generado y cargarlo en la CPU como parte del programa de usuario.

El OB 70 se puede utilizar, p. ej., para:

- evaluar la información de arranque del OB 70 y averiguar qué evento ha causado la pérdida de redundancia en periferia;
- mediante la SFC 51 RDSYSST, averiguar el estado de su sistema (SZL_ID=B#16#71).

La CPU no cambia a estado operativo "STOP" si se presenta un error de redundancia en periferia y si el OB 70 no está programado.

Si el OB 70 está cargado y el sistema H se encuentra en modo redundante, el OB 70 se procesará en ambas CPUs. El sistema H permanece en modo redundante.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

23.9.5 Error de redundancia en CPU (OB 72)

Descripción

El sistema operativo de la CPU H llama el OB 72 al presentarse uno de los eventos siguientes:

- Pérdida de redundancia de las CPUs
- Error de comparación (p.ej., RAM, PAA)
- Conmutación reserva/maestro
- Error de sincronización
- Error en un módulo SYNC
- Interrupción del proceso de igualación de datos
- El OB 72 es ejecutado por todas las CPUs que se encuentren en estado operativo RUN o ARRANQUE tras el correspondiente evento de arranque.

Programar el OB 72

El OB 72 se debe generar como objeto en el programa STEP 7. Escriba el programa, que se debe ejecutar en el OB 72, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 72 se puede utilizar, p. ej., para:

- evaluar la información de arranque del OB 72 y averiguar qué evento ha causado la pérdida de redundancia en la CPU;
- averiguar el estado de su sistema (SZL_ID=B#16#71) con ayuda de la SFC 51 RDSYSST;
- reaccionar a la pérdida de redundancia en la CPU conforme a la instalación.

La CPU no cambia a estado operativo "STOP" si se presenta un error de redundancia en la CPU y si el OB 72 no está programado.

Encontrará información más detallada acerca de los OBs, SFBs y SFCs en la Ayuda de bloques

23.9.6 Error de tiempo (OB 80)

Descripción

El sistema operativo de la CPU llama el OB 80 cuando ha ocurrido un error de tiempo. Errores de tiempo pueden ser, p. ej.:

- exceder el tiempo de ciclo máximo
- saltar alarmas horarias por adelanto de la hora
- retardo demasiado grande al tratar una prioridad

Programar el OB 80

El OB 80 se debe generar como objeto en el programa S 7 con ayuda de STEP 7. Escribir el programa, que se debe ejecutar en el OB 80, en el bloque generado y cargarlo en la CPU como parte del programa de usuario.

El OB 80 se puede utilizar, p. ej., para:

- evaluar la información de arranque del OB 80 y averiguar qué alarmas horarias se han saltado
- desactivar con la SFC 29 CAN_TINT la alarma horaria saltada, para que ésta no se ejecutada y el tratamiento de las alarmas horarias se pueda continuar correctamente a la hora nuevamente ajustada.

Si las alarmas horarias saltadas en el OB 80 no se desactivan, entonces se ejecuta la primera alarma horaria saltada y se ignoran todas las demás.

Si no se programa el OB 80, entonces la CPU pasa al estado operativo "STOP" al detectarse un error de tiempo.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

23.9.7 Fallo de alimentación (OB 81)

Descripción

El sistema operativo de la CPU llama el OB 81 cuando en el bastidor central o en un bastidor de ampliación ha fallado

- la alimentación de 24 V,
- una pila,
- todo el respaldo por pila

o se ha eliminado una perturbación (llamada en caso de evento entrante y saliente).

Programar el OB 81

El OB 81 se debe generar como objeto en el programa S 7 con ayuda de STEP 7. Escriba el programa, que se debe ejecutar en el OB 81, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 81 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 81 y averiguar qué fallo de alimentación existe
- determinar el número del bastidor con fallo de alimentación
- activar una lámpara en una estación de operador para visualizar al personal de mantenimiento que se ha de sustituir una pila.

Si no se ha programado el OB 81, la CPU no pasará al estado operativo STOP al detectarse un fallo de alimentación. No obstante, el fallo se registra en el búfer de diagnóstico y el LED correspondiente visualiza dicho fallo en el panel frontal.

Para más información sobre los OBs, SFBs y SFCs consulte la Saltos a descripciones de lenguajes, ayuda de bloques y atributos del sistema

23.9.8 Alarma de diagnóstico (OB 82)

Descripción

El sistema operativo de la CPU llama el OB 82 cuando en un módulo diagnosticable, para el cual se ha habilitado la alarma de diagnóstico, detecta un error y al eliminarse dicho error (llamada en caso de evento entrante y saliente).

Programar el OB 82

El OB 82 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escribir el programa, que se debe ejecutar en el OB 82, en el bloque generado y cargarlo en la CPU como parte del programa de usuario.

El OB 82 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 82
- efectuar un diagnóstico exacto del error ocurrido.

Cuando se dispara una alarma de diagnóstico, el módulo averiado registra automáticamente 4 bytes de datos de diagnóstico, así como su dirección inicial, en la información de arranque del OB de tratamiento de alarmas de diagnóstico y en el búfer de diagnóstico. Esto permite saber cuándo y en qué módulo ocurrió el error.

Otros datos de diagnóstico del módulo averiado (en qué canal ha ocurrido el error, de qué tipo de error se trata) se pueden evaluar con un programa correspondiente en el OB82. La SFC 51 RDSYSST permite leer los datos de diagnóstico del módulo y con la SFC 52 WR_USRMSG se pueden registrar estas informaciones en el búfer de diagnóstico. Además, el aviso de diagnóstico autodefinido adicionalmente se puede enviar a un visualizador registrado.

Si no se programa el OB 82, entonces la CPU cambia al estado de operación "STOP" al activarse una alarma de diagnóstico.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

23.9.9 Alarma de presencia de módulo (OB 83)

Descripción

Las CPUs S7-400 vigilan cíclicamente en intervalos de aprox. 1 segundo si se han extraído e insertado módulos en el bastidor central y en los bastidores de ampliación.

Tras conectar la tensión de alimentación, la CPU comprueba si todos los módulos listados en la tabla de configuración creada usando STEP 7 están realmente insertados. Si es así, esta configuración real se memoriza y sirve como valor de referencia para la vigilancia cíclica de los módulos. En cada ciclo de consulta, se compara la nueva configuración real determinada con la configuración real anterior. En caso de diferencias, se dispara una alarma de extraer/insertar y se registra una entrada tanto en el búfer de diagnóstico como en la lista de estados del sistema. En el estado operativo RUN se arranca el OB de presencia de módulo.

Nota

En el estado de operación RUN no se pueden extraer los módulos de alimentación ni las CPUs e IMs. Entre la extracción e inserción de un módulo deben transcurrir por lo menos 2 s, para que dichas operaciones pueden ser detectadas correctamente por la CPU.

Parametrizar un módulo una vez insertado

Si se inserta un módulo durante el estado operativo RUN, la CPU comprueba si el tipo del módulo que ha sido insertado coincide con el tipo del módulo originalmente insertado. Si coinciden, se efectúa la parametrización. Se transfieren bien los parámetros predeterminados o bien los parámetros asignados con STEP 7.

Programar el OB 83

Cree primero OB 83 en el programa S7. Escriba el programa, que se debe ejecutar en el OB 83, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 83 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 83
- reparametrizar el módulo insertado con las funciones del sistema SFC 55 a 59.

Si no se programa el OB 83, la CPU cambiará de RUN al estado operativo STOP al dispararse una alarma de extraer/insertar.

Encontrará información detallada sobre OBs, SFBs y SFCs en Saltos a descripciones de lenguajes, ayudas de bloques y atributos de sistema

23.9.10 Fallo de CPU (OB 84)

Descripción

El sistema operativo de la CPU llama el OB 84 cuando se detecta un fallo en el interface a la red MPI, al bus de comunicación o al módulo de interconexión (interfase) para la periferia descentralizada, p.ej. nivel de señal erróneo en el cable o cuando se elimina el error (llamada en caso de evento entrante y saliente).

Programar el OB 84

El OB 84 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escriba el programa que se debe ejecutar en el OB 84, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 84 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 84
- enviar un aviso al búfer de diagnóstico a través de la función del sistema SFC 52 WR_USMSG.

Si no se programa el OB 84, la CPU cambiará al estado operativo STOP al detectarse un fallo de hardware de la CPU.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

23.9.11 Error de ejecución del programa (OB 85)

Descripción

El sistema operativo de la CPU llama el OB 85 cuando

- existe un evento de arranque para un OB de alarma, pero el OB no se puede ejecutar porque no ha sido cargado en la CPU
- ha ocurrido un error al acceder al bloque de datos de instancia de un bloque de funciones del sistema
- ha ocurrido un error al actualizar la imagen de proceso (módulo no existente o defectuoso).

Programar el OB 85

El OB 85 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escriba en el bloque generado el programa que se debe ejecutar en el OB 85 y cárguelo en la CPU como parte del programa de usuario.

El OB 85 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 85 y determinar qué módulo está defectuoso o falta (indicación de la dirección inicial del módulo)
- determinar el puesto/slot del módulo correspondiente con la SFC 49 LGC_GADR.

Si no se programa el OB 85, entonces la CPU pasa al estado operativo "STOP" al detectarse un error de prioridad.

23.9.12 Fallo del bastidor (OB 86)

Descripción

El sistema operativo de la CPU llama al OB 86 al detectar uno de los siguientes eventos:

- Fallo de un aparato de ampliación centralizado (no en el S7-300), p. ej. línea de conexión interrumpida, corte de alimentación de un bastidor descentralizado.
- Fallo de un sistema maestro o de un esclavo (PROFIBUS DP) o fallo de un sistema IO o de un dispositivo IO (PROFINET IO).

El OB 86 también es llamado cuando se elimina el error (llamada con evento entrante y saliente).

Programar el OB 86

Cree en primer lugar el objeto OB 86 en el programa S7. Escriba en el bloque generado el programa que se debe ejecutar en el OB 86 y cárguelo en la CPU como parte del programa de usuario.

El OB 86 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 86 y determinar qué bastidor está defectuoso o falta.
- enviar un aviso al búfer de diagnóstico y a un visualizador a través de la función del sistema SFC 52 WR_USMSG.

Si no se programa el OB 86, la CPU cambia al estado operativo STOP al detectarse un fallo de bastidor.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

23.9.13 Error de comunicación (OB 87)

Descripción

El sistema operativo de la CPU llama el OB 87 al ocurrir un error de comunicación durante el intercambio de datos a través de bloques de función para comunicaciones o de comunicación de datos globales, p. ej.,

- se detectó un identificador de telegrama erróneo durante la recepción de datos globales
- el bloque de datos para la información de estado de los datos globales no existe o es demasiado corto.

Programar el OB 87

El OB 87 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escribir el programa, que se debe ejecutar en el OB 87, en el bloque generado y cargarlo en la CPU como parte del programa de usuario.

El OB 87 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 87 y
- crear un bloque de datos cuando falta el bloque de datos para la información de estado de la comunicación de datos globales.

La CPU no cambia al estado operativo "STOP" cuando se detecta un error de comunicación y no está programado el OB 87.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

23.9.14 Error de programación (OB 121)

Descripción

El sistema operativo de la CPU llama el OB 121 cuando ocurre un error de programación, como p. ej., cuando

- no existen temporizadores direccionados
- el bloque llamado no está cargado

Programar el OB 121

El OB 121 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escriba el programa, que se debe ejecutar en el OB 121, en el bloque generado y cárguelo en la CPU como parte del programa de usuario.

El OB 121 se puede utilizar, p. ej., para

- evaluar la información de arranque del OB 121
- registrar la causa del error en un bloque de datos de avisos.

Si no se programa el OB 121, la CPU cambia al estado operativo "STOP" al detectarse un error de programación.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

23.9.15 Error de acceso a la periferia (OB 122)

Descripción

El sistema operativo de la CPU llama el OB 122 cuando se accede con una operación STEP 7 a una entrada o salida de un módulo de señales que no estaba asignada a ningún módulo en el momento de efectuar el último re arranque completo, p. ej.,

- error al acceder directamente a la periferia (módulo defectuoso o no existente)
- acceso a una dirección de periferia desconocida por la CPU.

Programar el OB 122

El OB 122 se debe generar como objeto en el programa S7 con ayuda de STEP 7. Escriba en el bloque generado el programa que se debe ejecutar en el OB 122 y cárguelo en la CPU como parte del programa de usuario.

El OB 122 puede utilizar, p. ej., para

- evaluar la información de arranque del OB 122
- llamar la función del sistema SFC 44 y asignar un valor sustitutivo para un módulo de entradas, para que el programa pueda seguir corriendo con un valor razonable, adecuado al proceso.

Si no se programa el OB 122, la CPU cambiará al estado operativo "STOP" al detectarse un error de acceso a periferia.

Para más información sobre los OBs, SFBs y SFCs consulte la Ayuda de bloques

24 Imprimir y archivar

24.1 Imprimir la documentación de un proyecto

Una vez creado el programa para la solución de automatización, es posible imprimir todos los datos importantes para documentar el proyecto utilizando la función de impresión integrada en STEP 7.

Componentes imprimibles del proyecto

El contenido de los objetos se puede imprimir bien directamente desde el Administrador SIMATIC o bien abriendo el objeto deseado y activando el comando de impresión.

Desde el Administrador SIMATIC se pueden imprimir automáticamente los siguientes componentes de un proyecto:

- árbol de objetos (estructura del proyecto/de la librería),
- listas de objetos (contenido de una carpeta de objetos),
- contenido de un objeto y
- mensajes

Abriendo el objeto en cuestión se pueden imprimir p.ej. los siguientes componentes del proyecto:

- los bloques en lenguaje KOP, FUP, AWL u otro lenguaje (software opcional),
- la tabla de símbolos con los nombres simbólicos de las direcciones absolutas,
- la tabla de configuración con la disposición de los módulos en el PLC y los parámetros de los mismos,
- el contenido del búfer de diagnóstico,
- la tabla de variables con los formatos de estado, así como los valores de estado y de forzado,
- los datos de referencia, es decir, las listas de referencias cruzadas, los planos de ocupación, las estructuras del programa, los operandos no utilizados y los operandos sin símbolo,
- la tabla de datos globales,
- los datos de los módulos e información sobre su estado,
- textos de usuario (textos de usuario y librerías de textos),
- los documentos del software opcional, p.ej. de lenguajes de programación

Paquete opcional DOCPRO

Para crear, editar e imprimir manuales de circuitos normalizados está disponible el paquete opcional DOCPRO. Dicho paquete permite crear una documentación conforme a las normas DIN y ANSI.

24.1.1 Procedimiento básico para imprimir

Para imprimir un componente de un proyecto:

1. Abra el objeto en cuestión para visualizar los datos que desea imprimir.
2. Active el cuadro de diálogo "Imprimir" con el comando **Archivo > Imprimir...** en la respectiva ventana. Dependiendo de la ventana, el primer menú de la barra de menús puede tener un nombre diferente, p.ej. "Tabla de símbolos", en vez de "Archivo".
3. En caso necesario, cambie en el cuadro de diálogo los ajustes de impresión (p.ej. impresora, zona de impresión, número de copias) y cierre luego dicho cuadro.

Algunos cuadros de diálogo contienen el botón de comando "Imprimir", p. ej., el cuadro de diálogo 'Información del módulo'. Para imprimir el contenido del cuadro de diálogo, haga clic en este botón.

No es necesario abrir los bloques. Estos se pueden imprimir directamente desde el Administrador SIMATIC con el comando **Archivo > Imprimir...**

24.1.2 Funciones para imprimir

Para imprimir objetos se dispone de las siguientes funciones complementarias:

Objetos de impresión	Comando de menú	Función	Función	Función
		Presentación preliminar	Preparar página, ficha "Formato de papel"	Preparar página, ficha, ficha "Encabezado/pie de página"
Bloques, fuentes AWL	Archivo > *	•	•	–
Información del módulo		–	•	–
Tabla de datos globales	Tabla GD > *	•	•	–
Tabla de configuración	Equipo > *	•	•	–
Objeto, carpeta de objetos	Archivo > *	–	•	•
Datos de referencia	Datos de referencia > *	•	•	–
Tabla de símbolos	Tabla > *	•	•	–
Tabla de variables	Tabla > *	–	•	–
Tabla de enlaces	Red > *	•	•	–
Listas de textos de usuario (textos de usuario y librerías de textos)	Textos > *	•	•	–

* : El carácter * es un comodín en el comando de menú para la función correspondiente (p. ej., Presentación preliminar o Prepara página)

Las instrucciones paso a paso para imprimir los objetos las encontrará bajo:

Pasos para imprimir .

Presentación preliminar

La función "Presentación preliminar" permite visualizar el documento tal y como será impreso.

Nota

La imagen de la impresión optimizada no se muestra en la presentación preliminar.

Ajustar el formato de página y los encabezados y pies de página

Con la función "Preparar página", en la ficha "Formato de papel" se ajusta el formato de página (z. B. A4, A5, Letter) y la dirección (vertical u horizontal) de los documentos a imprimir. Puede escoger si los ajustes deben aplicarse a todos el proyecto o sólo a la sesión actual.

Ajuste el formulario a imprimir al tamaño de papel deseado. Si el formulario es demasiado ancho, el margen derecho se imprimirá en la página siguiente.

Si elige el formato de página con borde (p.ej., "borde A4"), el documento se imprimirá dejando un borde en la página izquierda que se puede utilizar para el taladrado.

En la ficha "Encabezado/Pie de página" se pueden configurar los encabezados y pies de página de los documentos a imprimir de todo el proyecto o sólo de la sesión actual.

24.1.3 Particularidades al imprimir el árbol de objetos

En el cuadro de diálogo "Imprimir lista de objetos" puede seleccionar el cuadro de opción "Árbol", con objeto de imprimir no sólo dicha lista, sino también el árbol de objetos.

Si en "Imprimir" selecciona la opción "Todo", se imprimirá toda la estructura del árbol. Si se elige "Selección", se imprimirá la estructura del árbol a partir del objeto seleccionado hacia abajo.

Nota

Los ajustes efectuados en el cuadro de diálogo son válidos únicamente para la impresión de la lista o del árbol, pero no para los contenidos. Para ello se utilizan los ajustes de las aplicaciones en cuestión.

24.2 Archivar proyectos y librerías

Los proyectos y las librerías se pueden guardar de forma comprimida en un archivo comprimido. El archivero puede copiarse en disco duro o en soportes de datos portátiles (p.ej. disquetes).

Programas archiveros

Dentro de STEP 7 puede abrirse el programa archivero preferido para archivar proyectos. Los programas archiveros ARJ y PKZIP 4.0 forman parte del volumen de suministro de STEP 7. Encontrará los programas y sus descripciones en la ruta de instalación del directorio ...\\Step7\\S7bin\\.

Para poder utilizar un determinado programa archivero, se requiere la siguiente versión (o una posterior):

- PKZip Commandline V4.0 (incluido en el suministro)
- WinZip a partir de la versión 6.0
- JAR a partir de la versión 1.02
- ARJ V2.4.1a (sólo para desarchivar, incluido en el suministro)
- ARJ32 V3.x (sólo para desarchivar)
- LHArc a partir de la versión 2.13 (sólo para desarchivar)

Particularidades

A partir de la versión V5.2 de STEP 7, para archivar sólo se soportan los programas archiveros PKZip 4.0, JAR y WinZip, mientras que para desarchivar también pueden emplearse los demás programas listados arriba.

Si en versiones anteriores de STEP 7 se utilizó ARJ32 V3.x para archivar, estos archivos sólo podrán volver a desarchivarse con ARJ32 V3.x.

En unidades de red, el proceso de archivado con PKZIP V4.0 es considerablemente más largo que en unidades locales.

24.2.1 Utilización de las funciones 'Guardar / Archivar'

Guardar como

Con esta función se puede crear una **copia** del proyecto con otro nombre.

Esta función se puede utilizar

- al crear copias de seguridad
- al duplicar un proyecto ya existente para adaptarlo a otras necesidades.

Para crear una copia de la forma más rápida posible, seleccione en el cuadro de diálogo que aparece a continuación 'Guardar sin reorganizar'. Así se copiará el árbol completo del archivo desde el directorio del proyecto sin comprobar si es coherente y se guardará con otro nombre.

En el soporte de datos debe haber memoria suficiente para depositar la copia de seguridad. No intente guardar proyectos en un disquete, puesto que normalmente la memoria no es suficiente. Para transportar datos del proyecto a disquetes utilice la función "Archivar".

La función 'Guardar reorganizando' dura más, pero si no se puede copiar o guardar un objeto aparecerá un mensaje indicándolo. Las causas se pueden deber a que falta el paquete opcional o a que los datos de un objeto son defectuosos.

Archivar

Es posible guardar proyectos o librerías en archivos comprimidos en archivadores. Dicho archivo comprimido puede copiarse en disco duro o en soportes de datos portátiles (p.ej. disquetes).

Si desea transportar proyectos a disquetes, hágalo sólo en forma de archivos comprimidos. Si el proyecto es demasiado grande, elija un programa archivador con el que pueda crear archivadores de múltiples disquetes.

Los proyectos o librerías que se hayan comprimido en un archivador no se pueden utilizar. Si desea procesar nuevamente dichos proyectos o librerías, deberá descomprimir los datos, es decir desarchivar el proyecto o la librería en cuestión.

24.2.2 Requisitos para poder archivar

Para archivar un proyecto o una librería se deberán cumplir los siguientes requisitos:

- El programa archivero deberá estar instalado en su sistema. La incorporación en STEP 7 se indica en el tema de la Ayuda "Procedimiento para archivar y desarchivar".
- Sin excepción alguna, todos los datos del proyecto deberán estar depositados directamente en el directorio del proyecto o en uno de sus subdirectorios. Aunque, al trabajar en el entorno de desarrollo C, se podrían depositar los datos en otro lugar, en tal caso éstos no se incorporarían en el archivo comprimido.
- A partir de la versión V5.2 de STEP 7, para archivar sólo pueden utilizarse los programas archiveros PKZip 4.0, JAR y WinZip, aunque para desarchivar también pueden emplearse los programas ARJ y LHarc.

24.2.3 Procedimiento para archivar y desarchivar

Para archivar y desarchivar un proyecto o una librería, utilice los comandos de menú **Archivo > Archivar** y **Archivo > Desarchivar**, respectivamente.

Nota

Los proyectos o librerías que se hayan comprimido en un archivero no pueden ser utilizadas. Si desea procesar nuevamente dichos proyectos o librerías, deberá descomprimir los datos, es decir desarchivar el proyecto o la librería en cuestión.

Al desarchivar los proyectos y librerías se integran automáticamente en la lista de proyectos/librerías.

Ajustar el directorio de destino

Para ajustar el directorio de destino, elija en el Administrador SIMATIC el comando de menú **Herramientas > Preferencias** para visualizar el cuadro de diálogo "Preferencias".

En la ficha "Archivar" de este cuadro de diálogo puede activar y desactivar la opción "Consultar el directorio de destino al desarchivar".

Si dicha opción está desactivada, al desarchivar se utilizará como ruta de destino la ruta indicada en la ficha "General" de este cuadro de diálogo (Ruta para proyectos y Ruta para librerías).

Copiar un archivo comprimido en un disquete

Asimismo, es posible comprimir un proyecto o una librería y copiar luego el archivo comprimido en un disquete. Además, también se puede elegir directamente una unidad de disquete en el cuadro de diálogo "Archivar".

25 Trabajar con sistemas de automatización M7

25.1 Procedimiento para sistemas M7

Gracias a su arquitectura PC estandarizada, los microcomputadores industriales M7-300/400 constituyen una extensión programable de los sistemas de automatización SIMATIC. Los programas de usuario para SIMATIC M7 se pueden crear en un lenguaje de alto nivel, tal como C, o bien gráficamente con CFC (Continuous Function Chart).

Para crear los programas se necesita, además de STEP 7, el software de sistema M7-SYS RT para M7-300/400, así como un entorno de desarrollo para programas M7 (ProC/C++ o CFC).

Procedimiento básico

Al crear una solución de automatización con SIMATIC M7 se deben realizar los siguientes trabajos. La siguiente tabla muestra los trabajos que se deben realizar en la mayoría de los proyectos y los asigna a un procedimiento general en forma de guía de orientación. En la tabla se hace referencia a los correspondientes capítulos del presente manual o a otros manuales.

Procedimiento	Descripción
Planificar una solución de automatización	Específico de M7; Ver: PHB M7-SYS RT
Arrancar STEP 7	Igual que en STEP 7
Crear la estructura del proyecto Parametrizar el equipo Configurar el hardware	Igual que en STEP 7
Configurar los enlaces de comunicación	Igual que en STEP 7
Definir la tabla de símbolos	Igual que en STEP 7
Crear programa de usuario en C o CFC	Específico de M7; Ver: ProC/C++
Configurar el sistema operativo Instalar el sistema operativo en M7-300/400 Cargar la configuración del hardware y el programa de usuario en M7	Específico de M7; Ver: BHB M7-SYS Rt
Comprobar y depurar el programa de usuario	ProC/C++
Vigilar el funcionamiento y diagnosticar M7	Igual que en STEP 7, pero sin diagnóstico definido por el usuario
Imprimir y archivar	Igual que en STEP 7

Diferencias con respecto al S7

STEP 7 no asiste las siguientes funciones para los sistemas M7-300/400:

- Modo multiprocesador - funcionamiento síncrono de varias CPUs
- Forzado permanente
- Comunicación de datos globales
- Diagnóstico personalizado

Gestionar los sistemas de destino M7

STEP 7 le asiste especialmente en las siguientes tareas con sistemas de automatización M7:

- Instalar el sistema operativo en el M7-300/400
- Configurar el sistema operativo editando archivos de sistema
- Transferir programas de usuario al M7-300/400
- Actualizar el firmware

Para acceder a la gestión de sistemas de destino M7, elija el siguiente comando de menú desde un proyecto que contenga equipos con CPUs o FMs M7, estando seleccionada la carpeta de programas M7:

Sistema de destino > Gestionar sistema de destino M7

La Ayuda en pantalla y el manual de usuario M7-SYS RT contienen instrucciones más detalladas.

25.2 Software opcional para la programación M7

Software opcional M7

STEP 7 ofrece las funciones básicas para:

- crear y gestionar proyectos,
- configurar y parametrizar el hardware,
- configurar redes y enlaces, así como
- gestionar datos simbólicos.

Estas funciones son aplicables a los sistemas de automatización SIMATIC S7 y SIMATIC M7.

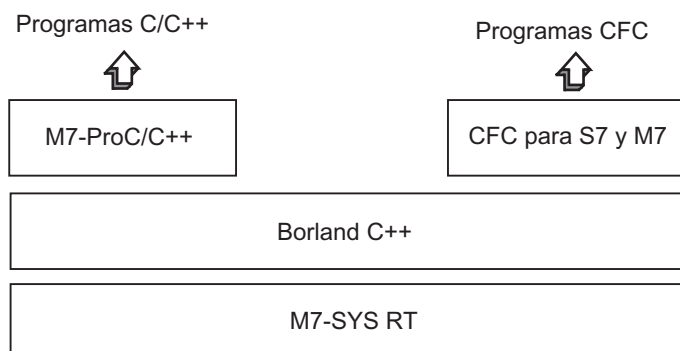
Para crear programas de usuario M7 se requiere, además de STEP 7, el software opcional M7;

Software	Indice del capítulo
M7-SYS RT	<ul style="list-style-type: none"> • Sistema operativo M7 RMOS32 • Librería de sistema M7-API • Soporte MPI
CFC para S7 y M7	Software de programación para aplicaciones CFC (CFC = Continuous Function Chart)
M7-ProC/C++	<ul style="list-style-type: none"> • Incorporación del entorno de desarrollo Borland en STEP 7 • Editor y generador de importación de símbolos • Depurador con lenguaje de alto nivel Organon xdb386
Borland C++	Entorno de desarrollo Borland C++

Junto con el software opcional M7, STEP 7 le ayuda a realizar las siguientes actividades:

- transferir datos al M7 vía MPI
- consultar informaciones sobre el sistema de automatización M7
- efectuar determinados ajustes en el sistema de automatización M7 y realizar un borrado total del M7

La figura siguiente muestra las dependencias del software opcional M7 para la programación M7.



Resumen

Para crear ...	necesita el software opcional M7 ...
Programas C/C++	2º M7-SYS RT 3º M7-ProC/C++ 4º Borland C++
Programas CFC	1º M7-SYS RT 2º CFC para S7 y M7 3º Borland C++

Para obtener ayuda

Las herramientas específicas para crear aplicaciones M7 están integradas en parte en STEP 7 y en parte en el software opcional M7.

En la tabla siguiente se indica qué ayuda ofrecen los diversos paquetes de software:

El software ...	permite ...
STEP 7	<ul style="list-style-type: none"> • instalar el sistema operativo M7, • gestionar el sistema de automatización M7, • transferir, iniciar y borrar programas de usuario M7, • llamar informaciones de estado y de diagnóstico, así como • efectuar el borrado total de la CPU
M7-SYS RT	gracias al sistema operativo y al software de sistema M7: <ul style="list-style-type: none"> • controlar la secuencia del programa, • gestionar la memoria y los recursos, • acceder al ordenador y al hardware SIMATIC, • gestionar alarmas, • realizar diagnósticos, • controlar el estado y • establecer enlaces de comunicación
M7-ProC/C++	<ul style="list-style-type: none"> • gracias a la creación de lógica integrada (integración del interface de desarrollo Borland en STEP 7) • gracias a la integración de los símbolos del proyecto en la lógica fuente e • gracias a la integración de funciones de depuración
Borland C++	<ul style="list-style-type: none"> • crear programas C y C++
CFC para S7 y M7	<ul style="list-style-type: none"> • crear, comprobar y depurar programas CFC e • iniciar y ejecutar programas CFC

25.3 Sistemas operativos para M7-300/400

Los servicios que ofrece el sistema operativo tienen importancia decisiva para las aplicaciones creadas en los lenguajes de alto nivel C y C++. El sistema operativo se encarga de las siguientes tareas:

- Acceso al hardware
- Gestión de recursos
- Integración en el sistema
- Comunicación con otros componentes del sistema

Para solucionar tareas de automatización, en los microcomputadores industriales SIMATIC M7 se utiliza el sistema operativo de tiempo real M7 RMOS32 (RMOS = **R**ealtime-**M**ultitasking-**O**perating-**S**ystem). Para poder incorporar M7 RMOS32 en el sistema SIMATIC, se le ha agregado un interface de llamada M7-API (API = **A**pplication **P**rogramming **I**nterface).

El sistema operativo de tiempo real M7 RMOS32 se utiliza para aplicaciones de 32 bits destinadas a solucionar tareas críticas de tiempo real y de multitarea. Se encuentra disponible en las siguientes configuraciones para los módulos M7:

- M7 RMOS32
- M7 RMOS32 con MS-DOS

La configuración del sistema operativo a seleccionar para el sistema de automatización M7 depende de los módulos M7 utilizados:

Configuración del sistema operativo	Módulo / memoria principal	PROFIBUS-DP y TCP/IP sí/no	Instalación en memorias de masa
M7 RMOS32	FM 356-4 / 4MB	no	Memory Card \geq 4 MB o disco duro
	FM 356-4 / 8MB	sí	
	CPU 388-4 / 8MB	sí	
	FM 456-4 / 16MB	sí	
	CPU 488-3 / 16MB	sí	
	CPU 486-3 / 16MB	sí	
M7 RMOS32 con MS-DOS	FM 356-4 / 8MB	no	Memory Card \geq 4 MB o disco duro
	CPU 388-4 / 8MB	no	
	FM 456-4 / 16MB	sí	
	CPU 488-3 / 16MB	sí	
	CPU 486-3 / 16MB	sí	

26 Consejos y trucos

26.1 Intercambiar módulos en la tabla de configuración

Si está corrigiendo una configuración de equipos con HW-Config. y desea intercambiar un módulo, p.ej. por una nueva referencia, proceda de la siguiente forma:

1. Mediante "Arrastrar y soltar" (Drag & Drop) saque el módulo de la ventana "Catálogo de hardware" sobre el módulo ("antiguo") ya ubicado.
2. Deje "caer" el nuevo módulo, que en cuanto sea posible adoptará los parámetros del previamente insertado.

Este procedimiento es más rápido que intercambiar los módulos borrando el antiguo e insertando y parametrizando a continuación el nuevo.

Esta función puede activarse o desactivarse a voluntad en HW-Config. con el comando de menú **Herramientas > Preferencias** ("Permitir sustitución de módulos").

26.2 Proyectos con un gran número de equipos en red

Si configura equipos sucesivamente y después llama a NetPro con **Herramientas > Configurar red** para configurar enlaces, los equipos se ubicarán automáticamente en la representación de la red. Este procedimiento tiene la desventaja de que posteriormente deberá ordenar por criterios topológicos los equipos y subredes.

Si su proyecto abarca un gran número de equipos en red y dese configurar enlaces entre dichos equipos, debe configurar desde el principio en la representación de la red la estructura de la instalación, para conservar una visión panorámica:

1. Cree el nuevo proyecto en el Administrador SIMATIC (comando de menú **Archivo > Nuevo**)
2. Arranque NetPro (comando de menú **Herramientas > Configurar red**)
3. En NetPro, cree un equipo tras otro:
 - Mediante "Arrastrar y soltar" (Drag & Drop) mueva el equipo de la ventana "Catálogo".
 - Haga doble clic en el equipo para iniciar HW-Config.
 - Mediante "Arrastrar y soltar" (Drag & Drop) ubique en HW-Config. los módulos aptos para comunicaciones (CPUs, CPs, FMs, módulos IF).
 - Si dese conectar dichos módulos en red: haga doble clic en las líneas correspondientes de la tabla de configuración para crear de nuevo subredes y para conectar en red los interfaces.
 - Guarde la configuración y cambie a NetPro.
 - Posicione en NetPro los equipos y subredes (mover objeto con el cursor del ratón, hasta que se alcance la posición deseada).
4. Configura los enlaces en NetPro y corrija (si fuese necesario) las conexiones a la red.

26.3 Reorganización

Cuando surgen problemas inexplicables al trabajar con STEP 7, suele ser útil reorganizar la gestión de datos del proyecto o de la librería en cuestión.

Para ello elija el comando de menú **Archivo > Reorganizar**. Al reorganizar los datos se eliminan los huecos que se fueron creando al borrar datos, es decir, se reduce el espacio de memoria requerido por los datos del proyecto o librería.

La función optimiza el almacenamiento de los datos del proyecto o de la librería, de la misma manera que un programa para desfragmentar un disco duro optimiza el almacenamiento de los datos en el mismo.

El proceso de reorganización puede prolongarse, todo depende de los datos que se tengan que desplazar en la memoria. Por esta razón, la función no se ejecuta automáticamente sino que ha de ser iniciada por el usuario cuando éste lo considere necesario.

Requisitos

Para poder reorganizar proyectos o librerías es necesario que los objetos afectados no estén ocupados, p. ej. por estar abiertos en otras aplicaciones.

26.4 Editar símbolos en varios segmentos

En el editor de programas KOP/AWL/FUP pueden visualizarse y editarse los símbolos de varios segmentos.

1. Marque un segmento haciendo clic en el nombre del segmento (p.ej. "Segmento 1").
2. Mantenga pulsada la tecla CTRL y marque otros segmentos.
3. Con el botón derecho del ratón, seleccione el comando de menú contextual **Editar símbolos**.

Para marcar todos los segmentos de un bloque, pulse CTRL+A y marque la denominación de un segmento.

26.5 Test con la tabla de variables

Para observar y forzar variables en la tabla de variables hay una serie de consejos prácticos de edición:

- Los símbolos y operandos se pueden introducir tanto en la columna "Símbolo" como en la columna "Operando". La entrada se escribirá automáticamente en la columna correspondiente.
- Para poder ver el valor forzado es necesario ajustar el punto de disparo para "Observar" al "Inicio del ciclo" y el punto de disparo para "Forzar" al "Fin del ciclo".
- Si coloca el cursor en una línea marcada en rojo, una información breve le informa sobre la causa del error. Con la tecla F1 obtendrá indicaciones para eliminar el error.
- Sólo se pueden introducir aquellos símbolos que ya estén definidos en la tabla de símbolos.
Un símbolo debe introducirse exactamente como está definido en la tabla de símbolos. Los nombres simbólicos que contienen caracteres especiales deben ir entre comillas (p.ej. "Motor.OFF", "Motor+OFF", "Motor-OFF").
- Desactivar las advertencias en la ficha "Online" (cuadro de diálogo "Preferencias")
- Cambiar de enlace sin haber deshecho otro enlace que estuviera establecido
- Ajustar el disparo para "Observar" mientras se observan variables.
- Forzar las variables que seleccione marcando las líneas correspondientes y ejecutando la función "Forzar". En este caso se forzarán solamente las variables seleccionadas que estén visibles.
- Salir sin consultar:

Si pulsa la tecla ESC cuando estén activadas "Observar", "Forzar" o "Desbloquear salidas", se saldrá de las funciones "Observar" y "Forzar" sin consultar.

- Introducir un área de operandos conexos:

Utilice el comando de menú **Insertar > Área**.

- Mostrar y ocultar columnas:

Para mostrar u ocultar columnas concretas, utilice los siguientes comandos de menú:

Símbolo: comando de menú **Ver > Símbolo**

Comentario del símbolo: comando de menú **Ver > Comentario del símbolo**

Formato de representación del valor de estado: comando de menú **Ver > Formato de visualización**

Valor de estado de la variable: comando de menú **Ver > Valor de estado**

Valor de forzado de la variable: comando de menú **Ver > Valor de forzado**

- Cambiar simultáneamente el formato de visualización de varias líneas de la tabla:
 1. Seleccione el área de la tabla en la que desee cambiar el formato de visualización arrastrando el ratón (con la tecla izquierda pulsada) por el área que desee de la tabla.
 2. Elija la representación con el comando de menú **Ver > Elegir formato de visualización**. Sólo cambia el formato de las líneas de la tabla seleccionada para las que está permitido el cambio de formato.
- Ejemplos de entrada con la tecla F1:
 - Si coloca el cursor en la columna de operandos y pulsa F1, obtendrá ejemplos de introducción de operandos.
 - Si coloca el cursor en la columna de valores de forzado y pulsa F1 obtendrá ejemplos de introducción de valores de forzado / de forzado permanente.

26.6 Forzar variables desde el editor de programas

Las entradas binarias y las marcas del editor de programas pueden proveerse de un botón que le permita forzar estos operandos rápida y fácilmente pulsando el botón con el ratón.

Requisitos

- Debe haber asignado esta propiedad al operando que debe forzarse en la tabla de símbolos mediante el comando de menú **Propiedades especiales del objeto > Accionar contacto**.
- Debe haber seleccionado la opción "Accionar contacto" en la ficha "General" del editor de programas KOP/AWL/FUP (comando de menú **Herramientas > Preferencias**).
- Debe haber seleccionado el comando de menú **Test > Observar**.

La condición de disparo es "permanente/al principio del ciclo".

El forzado de las entradas que existen realmente en la instalación dura mientras se mantiene pulsado el botón. La selección múltiple (tecla CTRL) permite también forzar varias entradas simultáneamente.

En las marcas y en las entradas no existentes, el estado pasa a 1 cuando se pulsa el botón. El estado no vuelve a 0 hasta que se fija explícitamente mediante el menú contextual o la tabla de variables, o bien cuando el operando se restaura con el programa S7.

En una entrada o marca no negada se fuerza con el valor de forzado "1" pulsando el botón, mientras que con la entrada o marca negada se fuerza con "0".

Nota para WinCC

Si el editor de programas se ha iniciado desde WinCC a través del manejo y la visualización de una variable, sólo se permite la posibilidad de forzado de WinCC. Excepto si el usuario tiene "Derechos de mantenimiento" de WinCC; en este caso se permiten ambas posibilidades de forzado.

26.7 Memoria de trabajo virtual

La memoria de trabajo virtual puede afectar al correcto funcionamiento de STEP 7.

Para trabajar con STEP 7 es necesario adaptar la memoria de trabajo virtual. Proceda como sigue:

1. Abra el panel de control, p. ej. desde el menú **Inicio > Configuración > Panel de control** y haga doble clic en el símbolo "Sistema".
Sólo XP: **Inicio > Mi PC > Propiedades > Opciones avanzadas > Rendimiento > Configuración.**
2. En Windows 2000, elija la ficha "Avanzadas" en el cuadro de diálogo que aparecerá y pulse el botón "Opciones de rendimiento".
En Windows XP/Server 2003 seleccione la ficha "Erweitert" del cuadro de diálogo "Systemeinstellungen".
3. Haga clic en el botón "Cambiar"
4. Introduzca bajo "Mínimo" o "Tamaño inicial" como mínimo 40 MBytes y bajo "Máximo" o "Tamaño máximo" como mínimo 150 MBytes.

Nota

Como la memoria virtual reside en el disco duro (por defecto C: y dinámico) hay que prever suficiente espacio de memoria para el directorio TMP o TEMP (aprox. 20 - 30 MB):

- Si el proyecto S7 se encuentra en la misma partición en la que reside la memoria virtual, se debería prever el doble de memoria de la que ocupa el proyecto S7.
 - No obstante, si el proyecto se gestiona en otra partición, no es necesario prever el doble de memoria.
-

A Anexos

A.1 Estados operativos

A.1.1 Estados operativos y cambios de estado

Estados operativos

El estado operativo describe el comportamiento de la CPU en cualquier momento. El conocimiento de los diferentes estados operativos de las CPUs sirve de ayuda para la programación del arranque, la prueba del autómata y el diagnóstico de errores.

Las CPUs S7-300 y S7-400 pueden adoptar los siguientes estados operativos:

- STOP
- ARRANQUE
- RUN
- PARADA

En el estado operativo "STOP", la CPU comprueba la existencia de los módulos configurados o de los que utilizan direcciones predeterminadas conduciendo además la periferia a un estado básico predefinido. El programa de usuario no se ejecuta en el estado operativo "STOP".

En el estado operativo "ARRANQUE" se distingue entre "Rearranque completo" (en caliente), "Arranque en frío" y "Rearranque":

- En el re arranque completo (en caliente) se vuelve a ejecutar el programa desde el principio con un "ajuste básico" de los datos del sistema y de las áreas de operandos de usuario (se inicializan los temporizadores, contadores y marcas no remanentes).
- En el arranque en frío se lee la imagen de proceso de las entradas y el programa de usuario STEP 7 se ejecuta comenzando por la primera instrucción del OB 1 (rige también para el re arranque completo (en caliente)).
 - Se borran los bloques de datos creados mediante SFCs (funciones del sistema) en la memoria de trabajo, en tanto que los demás bloques de datos adoptan el valor estándar de la memoria de carga.
 - La imagen del proceso, así como todos los temporizadores, contadores y marcas se ponen a cero, independientemente de que se hayan parametrizado como remanentes o no.

- En el re arranque, la ejecución del programa se inicia en el punto interrumpido (no se reposicionan los temporizadores, contadores y marcas). El re arranque sólo es posible en las CPUs S7-400.

En el estado operativo "RUN", la CPU ejecuta el programa de usuario, actualiza las entradas y salidas y procesa las alarmas y los avisos de error.

En el estado "PARADA" se detiene la ejecución del programa de usuario y se puede comprobar dicho programa paso a paso. Este estado sólo puede activarse en la prueba con la PG.

La CPU puede comunicarse en todos estos estados operativos a través del puerto MPI.

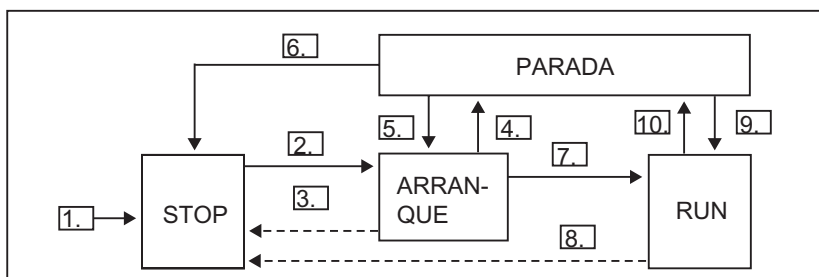
Otros estados operativos

Si la CPU no está operacional, entonces se encuentra en uno de los dos siguientes estados operativos:

- sin tensión, es decir, la tensión de red está desconectada.
- defectuosa, es decir, existe un error no eliminable.
Comprobar si la CPU está realmente defectuosa: poner la CPU en STOP, desconectar y volver a conectar con el interruptor de red. Si la CPU arranca, leer el búfer de diagnóstico para analizar el error. Si la CPU no arranca, es necesario sustituirla.

Cambios de estados operativos

La figura siguiente muestra los estados operativos y los cambios de estado operativo de las CPUs S7-300 y S7-400:



La tabla siguiente muestra bajo qué condiciones cambian los estados operativos.

Cambio	Descripción
1.	Tras conectar la tensión de alimentación, la CPU se encuentra en el estado "STOP".
2.	La CPU pasa al estado "ARRANQUE" <ul style="list-style-type: none"> • tras llevar la CPU a RUN o RUN-P con el selector de modo o la PG o • tras activarse automáticamente un tipo de arranque por CONEXION. • Al ejecutarse las funciones de comunicación "RESUME" o "START". El selector debe estar en ambos casos en RUN o RUN-P.
3.	La CPU pasa nuevamente a STOP cuando <ul style="list-style-type: none"> • se detecta un error durante el arranque • la CPU se lleva a STOP con el selector de modo o la PG • se procesa una orden de parada en el OB de arranque o • se ejecuta la función de comunicación "STOP".
4.	La CPU pasa al estado "PARADA" cuando se alcanza un punto de parada en el programa de arranque.
5.	La CPU pasa a "ARRANQUE" cuando un punto de parada estaba ajustado en el programa de arranque y se ejecuta la orden "ABANDONAR PARADA" (función de prueba).
6.	La CPU pasa nuevamente a STOP cuando <ul style="list-style-type: none"> • la CPU se lleva a STOP con el selector de modo o la PG o • se ejecuta la orden de comunicación "STOP".
7.	Si el arranque es correcto, la CPU cambia a RUN.
8.	La CPU pasa nuevamente a STOP cuando <ul style="list-style-type: none"> • se detecta un error en el estado RUN y el OB correspondiente no está cargado, • la CPU se lleva a STOP con el selector de modo o la PG, • se procesa una orden de parada en el programa de usuario o • se ejecuta la función de comunicación "STOP".
9.	La CPU pasa al estado "PARADA" cuando se alcanza un punto de parada en el programa de usuario.
10.	La CPU pasa a "RUN" cuando estaba ajustado un punto de parada y se ejecuta la orden "ABANDONAR PARADA".

Prioridad de los estados operativos

Si se solicitan simultáneamente varios cambios de estado operativo, se cambia al estado que tenga la prioridad más alta. Por ejemplo, si el selector de modos de operación de la CPU se encuentra en RUN y se intenta cambiar la CPU a STOP desde la PG, ésta pasará a STOP, puesto que éste es el estado operativo de máxima prioridad.

Prioridad	Estado operativo
más alta	STOP
	PARADA
	ARRANQUE
más baja	RUN

A.1.2 Estado operativo STOP

El programa de usuario no se ejecuta en el estado operativo "STOP". Todas las salidas se ajustan a valores sustitutos, llevando así el proceso de forma controlada a un estado operativo seguro. La CPU comprueba si

- existen problemas de hardware (p. ej., módulos no disponibles)
- debe regir el ajuste por defecto para la CPU o existen registros de parámetros
- son correctas las condiciones requeridas para el comportamiento en arranque programado
- existen problemas de software del sistema.

En el estado STOP se pueden recibir también datos globales, se puede efectuar comunicación unilateral pasiva a través de SFBs para comunicaciones para enlaces configurados y a través de SFCs para comunicaciones para enlaces no configurados.

Borrado total

En STOP se puede borrar totalmente la CPU. El borrado total puede ejecutarse manualmente a través del selector de modo (MRES) o desde la PG (p.ej., antes de cargar un programa de usuario).

Con el borrado total se conduce la CPU al "estado original", es decir

- se borra el programa de usuario completo en la memoria de trabajo y en la memoria de carga RAM, así como todas las áreas de operandos,
- se reposicionan los parámetros del sistema y los parámetros de la CPU y de módulos al ajuste por defecto, se conservan los parámetros MPI ajustados antes del borrado total.
- si hay una Memory Card (Flash-EEPROM) insertada, la CPU copia el programa de usuario de dicha Memory Card en la memoria de trabajo (incluidos los parámetros de la CPU y de módulos, si los correspondientes datos de configuración se encuentran también en la Memory Card).

El búfer de diagnóstico, los parámetros MPI, la hora y el contador de horas de funcionamiento no se reposicionan.

A.1.3 Estado operativo ARRANQUE

Antes de que la CPU comience, tras la conexión, a ejecutar el programa de usuario, se ejecuta un programa de arranque. En el programa de arranque se pueden definir determinados preajustes para el programa cíclico a través de la programación de OB de arranque.

Se dispone de tres tipos de arranque: Rearranque completo (en caliente), arranque en frío y re arranque. Por principio, un re arranque completo sólo se puede efectuar en las CPUs S7-400, debiéndose haber definido mediante STEP 7 en el juego de parámetros de la CPU.

En el estado operativo "ARRANQUE":

- se ejecuta el programa en el OB de arranque (OB 100 para re arranque completo (en caliente), OB 101 para re arranque, OB 102 para arranque en frío)
- no es posible la ejecución de programa controlada por tiempo o por alarmas
- se actualizan los temporizadores
- corre el contador de horas de funcionamiento
- las salidas digitales están bloqueadas en los módulos de señales, pero se pueden posicionar por acceso directo.

Rearranque completo (en caliente)

El re arranque completo (en caliente) es siempre posible, a no ser que el sistema haya solicitado el borrado total. En los casos siguientes, el re arranque completo (en caliente) es posible sólo tras:

- borrado total
- carga del programa de usuario en el estado STOP de la CPU
- desbordamiento de la PILA DE INTERRUPCION/PILA DE BLOQUES (USTACK/BSTACK)
- interrupción de re arranque completo (en caliente) (por DESCONEXION o a través del selector de modo)
- rebase del límite de tiempo de interrupción parametrizado para el re arranque.

Rearranque completo (en caliente) manual

El re arranque completo (en caliente) manual se puede activar:

- con el selector de modo
(el selector CRST/WRST - si existe - debe estar en CRST)
- con el correspondiente comando de menú desde la PG o a través de las funciones de comunicación
(cuando el selector de modo esté en RUN o RUN-P)

Rearranque completo (en caliente) automático

El re arranque completo (en caliente) automático se puede activar en CONEXION cuando:

- la CPU no estaba en STOP cuando se desconectó la alimentación
- el selector de modo operativo está en RUN o RUN-P
- no hay re arranque automático parametrizado tras CONEXION
- la CPU ha sido interrumpida durante el re arranque completo (en caliente) por corte de alimentación (independientemente de la parametrización del tipo de arranque).

En el re arranque completo (en caliente) automático no tiene efecto el selector CRST/WRST.

Rearranque completo (en caliente) automático sin respaldo por pila

Si la CPU opera sin pila tampón (si se requiere funcionamiento libre de mantenimiento), tras la conexión o al regresar la tensión después de OFF, se efectúa automáticamente el borrado total de la CPU y se realiza luego un re arranque completo (en caliente). A tal efecto, el programa de usuario debe estar memorizado en la Flash-EPROM (Memory Card).

Rearranque

Después de un corte de tensión en RUN y al regresar nuevamente la tensión, las CPUs S7-400 corren una rutina de inicialización y ejecutan luego automáticamente un re arranque. En caso de re arranque, el programa de usuario se continúa en el punto donde fue interrumpida la ejecución. La parte del programa de usuario que no llegó a ejecutarse antes del corte de tensión se denomina ciclo residual. El ciclo residual puede contener también partes del programa controladas por tiempo y por alarmas.

Básicamente, el re arranque sólo es admisible si el programa de usuario no ha sido modificado en el estado STOP (p. ej. recargando un bloque modificado) y si no se requiere por otras razones un re arranque completo (en caliente). Se distingue entre re arranque manual y automático.

Rearranque manual

El re arranque manual sólo es posible si se ha parametrizado correspondientemente el registro de parámetros de la CPU y tras las siguientes causas de STOP:

- el selector de modo se ha cambiado de RUN a STOP
- STOPs programados por el usuario, STOPs tras llamar OBs no cargados
- el estado STOP ha sido causado por la PG o por una función de comunicación.

El re arranque manual se puede activar:

- con el selector de modo
 - El selector CRST/WRST debe estar en WRST.
- por comando de menú desde la PG o por funciones de comunicación (si el selector de modo está en RUN o RUN-P)
- se ha parametrizado el registro de parámetros de la CPU para re arranque manual.

Rearranque automático

El rearranque automático se puede activar en CONEXION cuando:

- la CPU no estaba en estado STOP o PARADA en OFF
- el selector de modo operativo está en RUN o RUN-P
- en el registro de parámetros de la CPU se ha parametrizado rearranque automático tras CONEXION.

El selector CRST/WRST no tiene efecto en el rearranque automático.

Areas de datos remanentes tras fallar la red

Las CPUs S7-300 y S7-400 reaccionan de forma distinta al regresar la tensión tras un fallo de red.

En las CPUs S7-300 (con excepción de la CPU 318) sólo se dispone del "Rearranque completo". No obstante, con STEP 7 se puede ajustar la remanencia de marcas, temporizadores, contadores y áreas en los bloques de datos para impedir la pérdida de datos al producirse un corte de tensión. Al regresar la tensión, se efectúa un "rearranque completo automático con memoria".

Al regresar la tensión, las CPUs S7-400 reaccionan con un rearranque completo (en caliente) (tras CONEXION con o sin respaldo por pila) o con un rearranque (sólo tras CONEXION con respaldo por pila).

La tabla A-1 muestra el comportamiento de remanencia de las CPUs S7-300 y S7-400 en caso de un rearranque completo (en caliente), un arranque en frío y un rearranque:

X	significa que:	se conservan los datos
VC	significa que:	se conserva el bloque lógico en la EPROM, un bloque lógico sobrecargado se pierde
VX	significa que:	el bloque de datos sólo se conserva si existe en la EPROM, los datos remanentes se adoptan de la RAM NV (los bloques de datos cargados o creados en la RAM se pierden)
0	significa que:	los datos se inicializan o se borran (contenido de DBs)
V	significa que:	los datos se ajustan al valor de inicialización, a partir de la memoria EPROM
---	significa que:	no es posible, porque no existe una RAM NV

Comportamiento de remanencia en la memoria de trabajo (en la memoria de carga EPROM y RAM)

EPROM (Memory Card o integrada)									
CPU con respaldo					CPU sin respaldo por pila				
Datos	Bloques en memoria de carga	DB en memoria de trabajo	Marcas, temp., contadores	Marcas, temp., contadores	Bloques en memoria de carga	DB en memoria de trabajo	DB en memoria de trabajo	Marcas, temp., contadores	Marcas, temp., contadores
			param. como remanentes	param. como no remanentes		param. como remanentes	param. como no remanentes	param. como remanentes	param. como no remanentes
Rearranque completo en S7-300	X	X	X	0	VC	VX	V	X	0
Rearranque completo en S7-400	X	X	X	0	VC	---	V	0	0
Arranque en frío en S7-300	X	0	0	0	VC	V	V	0	0
Arranque en frío en S7-400	X	0	0	0	VC	---	V	0	0
Rearranque en S7-400	X	X	X	X	Sólo se permite rearranque completo				

Actividades en el arranque

La tabla siguiente muestra qué actividades efectúa la CPU durante el arranque:

Actividades durante la secuencia de ejecución	en el re arranque completo (en caliente)	en el arranque en frío	en el re arranque
Borrar pila de interrupción/pila de bloques	X	X	0
Borrar marcas, temporizadores, contadores no remanentes	X	0	0
Borrar todas las marcas, temporizadores y contadores	0	X	0
Borrar imagen de proceso de las salidas	X	X	parametrizable
Reponer salidas de los módulos de señales	X	X	parametrizable
Rechazar alarmas de proceso	X	X	0
Rechazar alarmas de retardo	X	X	0
Rechazar alarmas de diagnóstico	X	X	X
Actualizar lista de estado del sistema (SZL)	X	X	X
Evaluar parámetros de módulos y transferirlos a los módulos o entregar valores por defecto	X	X	X
Ejecutar el OB de arranque correspondiente	X	X	X
Procesar ciclo residual (parte del programa de usuario cuya ejecución no pudo continuar en virtud de una desconexión de alimentación)	0	0	X
Actualizar imagen de proceso de las entradas	X	X	X
Habilitar salidas digitales (desactivar señal OD) tras cambio de estado operativo a RUN	X	X	X
X significa que se realiza 0 significa que no se realiza			

Interrumpir un arranque

Si ocurren errores durante el arranque, éste se interrumpe y la CPU pasa o permanece en STOP.

Un re arranque completo (en caliente) interrumpido debe ser repetido. Tras un re arranque interrumpido, es posible tanto un re arranque completo (en caliente) como un re arranque normal.

Un arranque (re arranque completo (en caliente) o re arranque) no se ejecuta, o se interrumpe, si

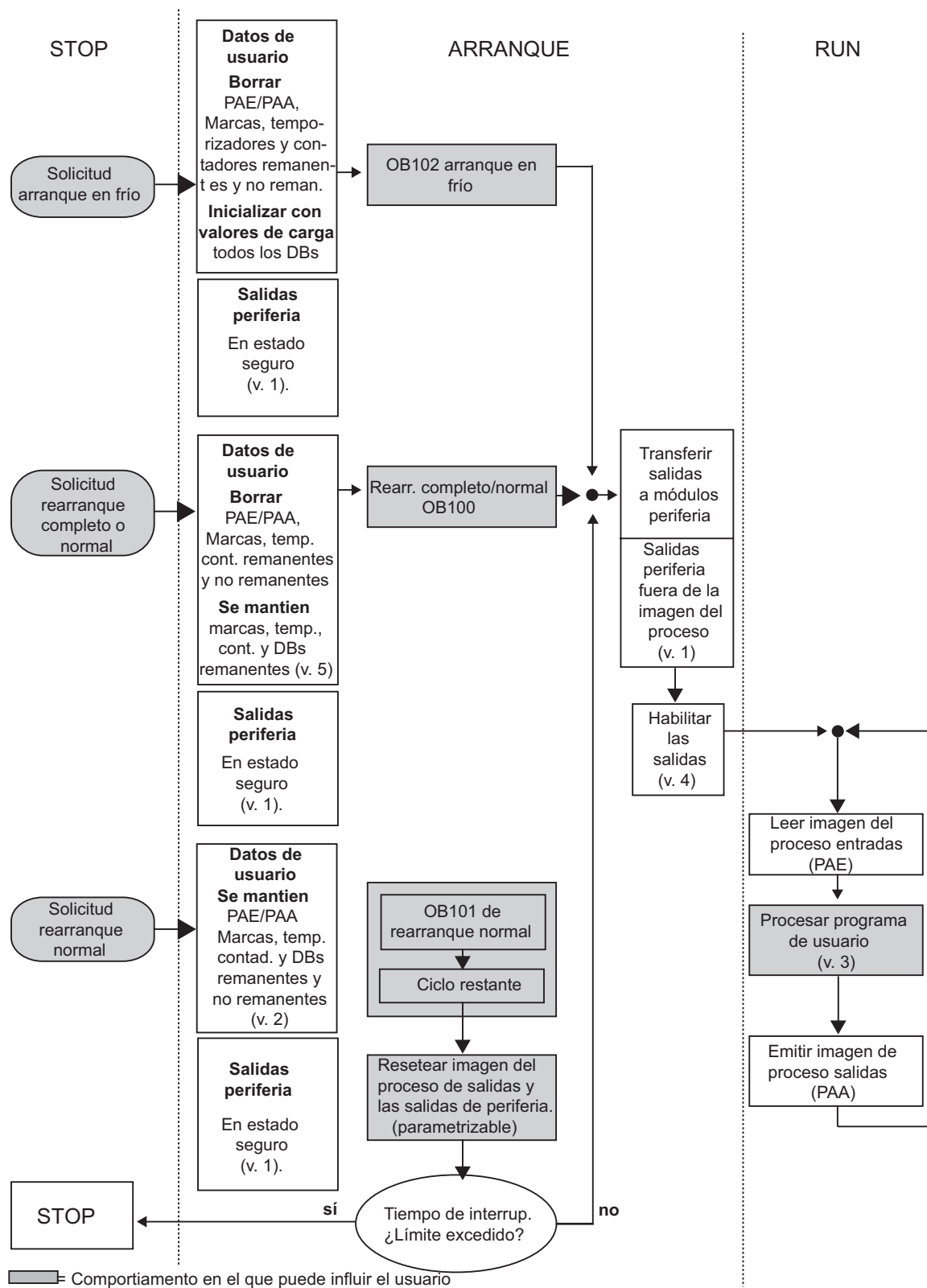
- el selector de modo de la CPU está en STOP
- se solicita borrado total
- hay insertada una Memory Card cuya identificación de aplicación no es admisible para STEP 7 (p. ej. STEP 5)
- en modo monoprocesador hay más de una CPU insertada
- el programa de usuario contiene un OB desconocido por la CPU o que ha sido inhibido
- la CPU detecta tras la conexión de la tensión de red que no todos los módulos listados en la tabla de configuración de STEP 7 están realmente insertados (diferencia no admisible entre la parametrización teórica y la real)
- ocurren errores al evaluar los parámetros de los módulos.

Un arranque no se ejecuta, o se interrumpe, si

- la CPU ha sido borrada totalmente (tras el borrado total sólo es posible el re arranque completo (en caliente))
- se ha rebasado el límite de tiempo de interrupción (el tiempo de interrupción es aquél que transcurre desde que se sale de RUN hasta que se ejecuta el OB de arranque, incluido el ciclo residual)
- se ha modificado la configuración de módulos (p. ej. cambio de un módulo)
- por parametrización sólo es admisible re arranque completo (en caliente)
- se han cargado, borrado o modificado módulos en STOP.

Transcurso

La figura siguiente muestra las actividades de la CPU en los estados operativos ARRANQUE y RUN.



Leyenda de la figura "Actividades de la CPU en los estados operativos ARRANQUE y RUN":

1. Los módulos de periferia conmutan a estado seguro (valor por defecto = 0) todas las salidas de periferia. Este hecho es independiente de que en el programa de usuario se utilicen dentro o fuera del área de imagen del proceso.
2. Si se utilizan módulos de señales aptos para valor de sustitución, el comportamiento de las salidas puede parametrizarse, p. ej. mantener el último valor.
3. Es necesario para ejecutar el ciclo residual.
4. Al llamar por primera vez una imagen actual del proceso de las entradas, se dispone también de los Obs de alarma.
5. El estado de las salidas de periferia centralizadas y descentralizadas en el primer ciclo del programa de usuario puede definirse con las siguientes medidas:
 - Utilizando módulos de salida parametrizables, para poder emitir valores de sustitución o para mantener el valor.
 - En el re arranque: activar el parámetro de arranque de CPU "Inicializar salidas al efectuar un arranque normal" para dar salida a un "0" (equivale al ajuste predeterminado).
 - Inicializar salidas en el OB de arranque (OB 100, OB 101, OB 102).
6. En sistemas S7-300 sin respaldo de memoria sólo se mantienen las áreas de DB configuradas como remanentes.

A.1.4 Estado operativo RUN

En el estado operativo "RUN" tiene lugar la ejecución del programa cíclica, la controlada por tiempo y la controlada por alarmas:

- se lee la imagen de proceso de las entradas
- se ejecuta el programa de usuario
- se emite la imagen de proceso de las salidas.

El intercambio activo de datos entre las CPUs a través de la comunicación de datos globales (tabla de datos globales) y de SFBs para comunicaciones para enlaces configurados y a través de SFCs para comunicaciones para enlaces no configurados sólo es posible en el estado RUN.

La tabla siguiente ejemplifica cuándo es posible el intercambio de datos en diferentes estados operativos:

Tipo de comunicación	Estado operativo de la CPU 1	Sentido del intercambio de datos	Estado operativo de la CPU 2
Comunicación por datos globales	RUN	↔	RUN
	RUN	→	STOP/PARADA
	STOP	←	RUN
	STOP	X	STOP
	PARADA	X	STOP/PARADA
Comunicación unilateral a través de SFBs para comunicaciones	RUN	→	RUN
	RUN	→	STOP/PARADA
Comunicación bilateral a través de SFBs para comunicaciones	RUN	↔	RUN
Comunicación unilateral a través de SFCs para comunicaciones	RUN	→	RUN
	RUN	→	STOP/PARADA
Comunicación bilateral a través de SFCs para comunicaciones	RUN	↔	RUN
↔ significa que el intercambio de datos es posible en ambos sentidos			
→ significa que el intercambio de datos es posible en un solo sentido			
X significa que el intercambio de datos no es posible			

A.1.5 Estado operativo PARADA

El estado operativo PARADA (HALT) representa un estado especial. Se utiliza sólo para fines de prueba durante el arranque o RUN. En el estado PARADA:

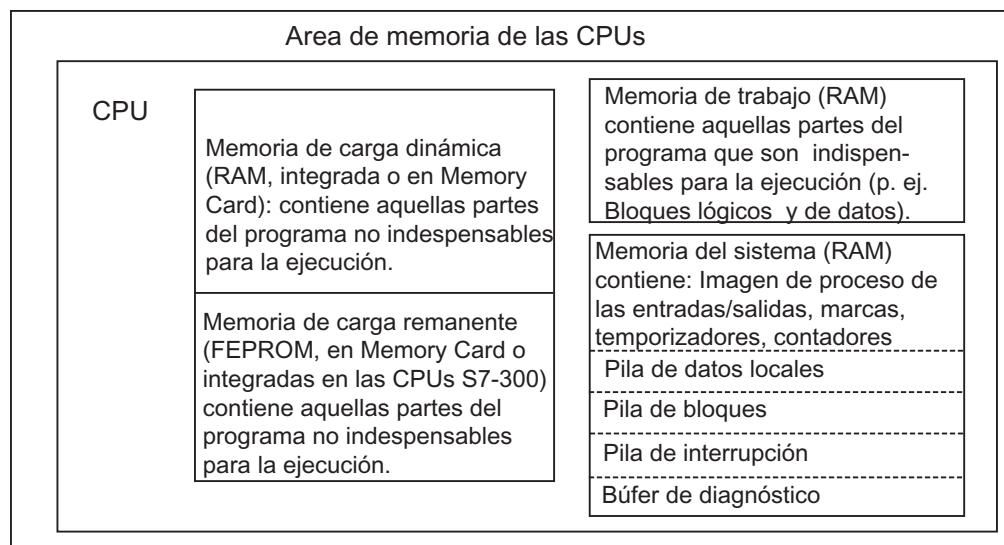
- se congelan todos los temporizadores: no se procesan los temporizadores ni los contadores de horas de funcionamiento, se detienen los tiempos de vigilancia y los ciclos básicos de los niveles con ejecución controlada por tiempo.
- corre el reloj de tiempo real
- no se habilitan las salidas, pero pueden habilitarse para fines de prueba
- se pueden forzar entradas y salidas
- en caso de corte y retorno de la tensión en el estado PARADA, las CPUs respaldadas por pila pasan a STOP y no provocan ningún re arranque o re arranque completo (en caliente) automático. Las CPUs no respaldadas provocan, al retornar la tensión, un re arranque (en caliente) automático no respaldado.
- se pueden recibir también datos globales, se puede efectuar comunicación unilateral pasiva a través de SFBS para comunicaciones para enlaces configurados y a través de SFCs para comunicaciones para enlaces no configurados (v. también tabla en Estado operativo RUN).

A.2 Áreas de memoria de las CPUs S7

A.2.1 Subdivisión de la memoria en áreas

La memoria de las CPUs S7 se subdivide en tres áreas (v. siguiente figura):

- La memoria de carga permite almacenar el programa de usuario sin asignación simbólica de operandos o comentarios (éstos permanecen en la memoria de la PG). La memoria de carga puede ser RAM o EPROM.
- Los bloques caracterizados como no relevantes para la ejecución se memorizan exclusivamente en la memoria de carga.
- La memoria de trabajo (RAM integrada) contiene la partes del programa S7 relevantes para la ejecución del programa. La ejecución del programa tiene lugar exclusivamente en el área correspondiente a las memorias de trabajo y del sistema.
- La memoria del sistema (RAM) contiene los elementos de memoria que cada CPU pone a disposición del programa de usuario, tales como: la imagen de proceso de las entradas y salidas, marcas, temporizadores y contadores. Contiene además las pilas de bloques y de interrupción.
- La memoria del sistema de la CPU ofrece además una memoria temporal (pila de datos locales) asignada al programa para los datos locales del bloque llamado. Estos datos sólo tienen vigencia mientras esté activo el bloque correspondiente.



A.2.2 Memorias de carga y de trabajo

Si el programa de usuario se carga en la CPU desde la unidad de programación, se cargan solamente los bloques lógicos y de datos en las memorias de carga y de trabajo de la CPU.

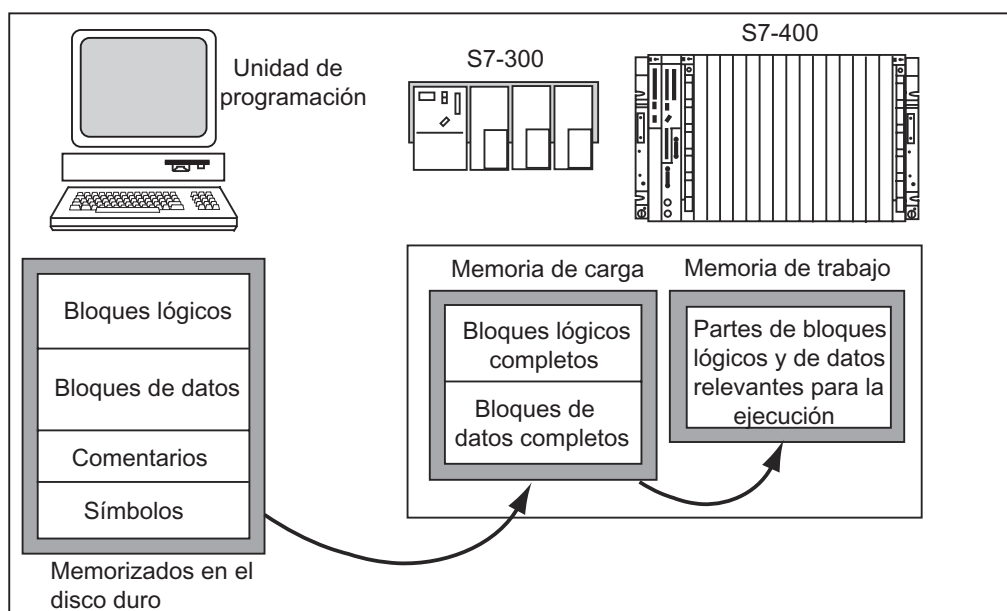
La asignación simbólica de operandos (tabla de símbolos) y los comentarios de los bloques permanecen en la memoria de la PG.

Estructura del programa de usuario

Para garantizar una rápida ejecución del programa de usuario y no sobrecargar innecesariamente la memoria de trabajo ampliable, en dicha memoria se cargan únicamente aquellas partes de los bloques que son relevantes para la ejecución del programa.

Las partes de los bloques que no son necesarias para ejecutar el programa (p. ej., encabezamientos de bloques) permanecen en la memoria de carga.

La figura siguiente muestra cómo se carga el programa en la memoria de la CPU.



Nota

La CPU guarda completamente en la memoria de trabajo los bloques de datos que se hayan creado mediante funciones del sistema (p. ej. SFC 22 CREAT_DB) en el programa de usuario.

Algunas CPUs disponen en la memoria de trabajo de áreas gestionadas por separado para el código y los datos. En dichas CPUs, el tamaño y la ocupación de las áreas se visualizan en la ficha "Memoria" de la información del módulo.

Caracterizar los bloques de datos como "no relevantes para la ejecución"

Los bloques de datos programados en un archivo fuente como parte de un programa AWL pueden caracterizarse como "no relevantes para la ejecución" (clave UNLINKED). Es decir, al efectuar la carga en la CPU estos DBs se despositan solamente en la memoria de carga. Su contenido se puede copiar, en caso necesario, en la memoria de trabajo a través de la SFC 20 BLKMOV.

Así se ahorra espacio en la memoria de trabajo. La memoria de carga ampliable sirve de memoria intermedia (p. ej. para recetas: en la memoria de trabajo se carga solamente la próxima receta a ejecutar).

Estructura de la memoria de carga

La memoria de carga se puede ampliar utilizando Memory Cards. En los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" o "Sistemas de automatización S7-400, M7-400, Datos de los módulos" se indica el tamaño máximo de la memoria de carga.

En el caso de las CPUs S7-300 la memoria de carga puede integrar una parte de memoria RAM y una parte de memoria EPROM. Mediante la parametrización con STEP 7 se pueden declarar como remanentes determinadas áreas en los bloques de datos (consulte también Areas remanentes de la memoria en CPUs S7-300).

En el caso de las CPUs S7-400 para poder ampliar la memoria de usuario es imprescindible utilizar una Memory Card (RAM o EPROM). La memoria de carga integrada es una memoria RAM, utilizada esencialmente para recargar y corregir bloques. En el caso de las CPUs S7-400 nuevas se puede enchufar también una memoria de trabajo adicional.

Comportamiento de la memoria de carga en las áreas RAM y EPROM

El comportamiento de la memoria de carga durante las operaciones de carga, recarga o borrado total puede ser diferente dependiendo de la forma de ampliación elegida: Memory Card RAM o EPROM.

La tabla siguiente muestra las posibilidades de carga:

Tipo de memoria	Posibilidades de carga	Tipo de carga
RAM	Cargar y borrar bloques individuales	Enlace PG-CPU
	Cargar y borrar un programa S7 completo	Enlace PG-CPU
	Recargar bloques individuales	Enlace PG-CPU
EPROM integrada (sólo en el S7-300) o enchufable	Cargar programas S7 completos	Enlace PG-CPU
EPROM enchufable	Cargar programas S7 completos	Cargar la EPROM en la PG e insertar la Memory Card en la CPU Cargar la EPROM en la CPU

Los programas memorizados en RAM se pierden si se borra totalmente la CPU (MRES) o si se extrae la CPU o la Memory Card RAM.

Los programas memorizados en Memory Cards EPROM no se pierden al borrar totalmente y permanecen incluso sin respaldo por pila (transporte, copias de seguridad).

A.2.3 Memoria de sistema

A.2.3.1 Uso de las áreas de memoria del sistema

La memoria de sistema de las CPUs S7 está subdividida en áreas de operandos (v. siguiente tabla). El uso de las operaciones correspondientes permite direccionar los datos en el programa directamente en las diferentes áreas de operandos.

Area de operandos	Acceso a través de unidades del siguiente tamaño:	Notación S7	Descripción
Imagen del proceso de las entradas	Entrada (bit)	E	Al comienzo de cada ciclo, la CPU lee las entradas de los módulos de entradas y memoriza los valores en la imagen de proceso de las entradas.
	Byte de entrada	EB	
	Palabra de entrada	EW	
	Palabra doble de entrada	ED	
Imagen de proceso de las salidas	Salida (bit)	A	Durante el ciclo, el programa calcula los valores para las salidas y los deposita en la imagen de proceso de las salidas. Al final del ciclo, la CPU escribe los valores de salida calculados en los módulos de salidas.
	Byte de salida	AB	
	Palabra de salida	AW	
	Palabra doble de salida	AD	
Marcas	Marca (bit)	M	Esta área ofrece capacidad de memoria para los resultados intermedios calculados en el programa.
	Byte de marcas	MB	
	Palabra de marcas	MW	
	Palabra doble de marcas	MD	
Temporizadores	Temporizador (T)	T	Esta área contiene los temporizadores disponibles.
Contadores	Contador (Z)	Z	Esta área contiene los contadores disponibles.
Bloque de datos	Bloque de datos, abierto con "AUF DB":	DB	Los bloques de datos memorizan informaciones para el programa. Pueden estar definidos de tal manera que todos los bloques de datos puedan acceder a ellos (DBs globales) o pueden estar asignados a un determinado FB o SFB (DB de instancia).
	Bit de datos	DBX	
	Byte de datos	DBB	
	Palabra de datos	DBW	
	Palabra doble de datos	DBD	
	Bloque de datos, abierto con "AUF DI":	DI	
	Bit de datos	DIX	
	Byte de datos	DIB	

Area de operandos	Acceso a través de unidades del siguiente tamaño:	Notación S7	Descripción
	Palabra de datos	DIW	
	Palabra doble de datos	DID	
Datos locales	Bit de datos locales	L	Esta área de memoria contiene los datos temporales de un bloque durante la ejecución de dicho bloque. La pila L ofrece también memoria para la transferencia de parámetros de bloques y para memorizar los resultados intermedios de segmentos KOP.
	Byte de datos locales	LB	
	Palabra de datos locales	LW	
	Palabra doble de datos locales	LD	
Area de periferia: entradas	Byte de entrada de periferia	PEB	Las áreas de periferia de las entradas y salidas permiten el acceso directo a módulos de entrada y salida centralizados y descentralizados.
	Palabra de entrada de periferia	PEW	
	Palabra doble de entrada de periferia	PED	
Area de periferia: salidas	Byte de salida de periferia	PAB	
	Palabra de salida de periferia	PAW	
	Palabra doble de salida de periferia	PAD	

Las áreas de direcciones disponibles para las diferentes CPUs se indican en las descripciones de CPUs o en las listas de operaciones :

- Manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU".
- Manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos"
- Lista de operaciones "Sistema de automatización S7-300".
- Lista de operaciones "Sistema de automatización S7-400".

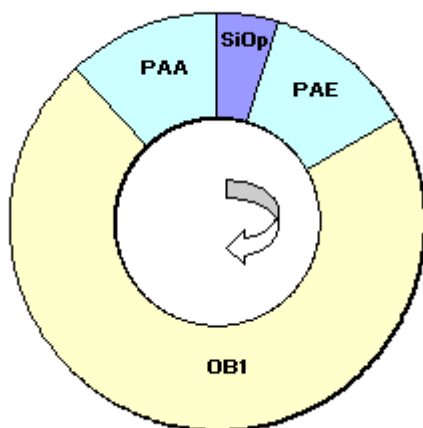
A.2.3.2 Imagen del proceso de las entradas y salidas (E/S)

Si en el programa de usuario se accede a las áreas de operandos: entradas (E) y salidas (A), no se consultan los estados de señal en los módulos de señales digitales, sino los presentes en un área de la memoria del sistema de la CPU y de la periferia descentralizada. Esta área de memoria se designa como imagen del proceso.

Actualizar la imagen de proceso

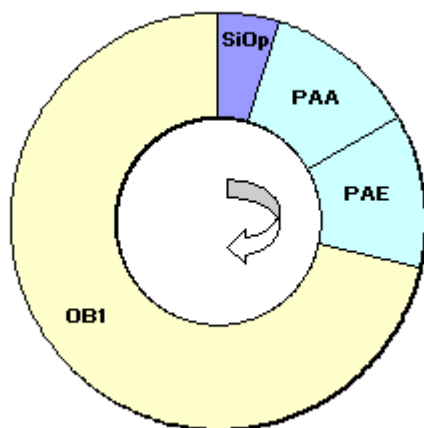
La figura siguiente muestra los pasos de procesamiento dentro de un ciclo.

Ejecución cíclica del programa (CPUs hasta 10/98)



Una vez ejecutadas las tareas internas del sistema operativo (SiOp) se lee el estado de las entradas en la imagen del proceso de las entradas (PAE). A continuación tiene lugar se ejecuta el programa de usuario con todos los bloques llamados. El ciclo cierra los módulos en las salidas al escribir la imagen del proceso de las salidas (PAA). La lectura de la imagen del proceso de las entradas y la escritura de la imagen del proceso de las salidas en las salidas de los módulos se gestionan de forma independiente desde el sistema operativo.

Ejecución cíclica del programa (CPUs a partir de 10/98)



Tras las funciones internas del sistema operativo (BESY), la imagen de proceso de las salidas (PAA) se escribe en las salidas de los módulos y el estado de las entradas se lee en la imagen de proceso de las entradas (PAE). A continuación el programa de usuario ejecuta un proceso con todos los bloques llamados. La escritura de la imagen de proceso de las salidas en las salidas de los módulos y la lectura de la imagen de proceso de las entradas las regula el sistema operativo de forma independiente.

Ventajas de la imagen del proceso

En comparación con el acceso directo a los módulos de entrada/salida, el acceso a la imagen de proceso ofrece la ventaja de que la CPU dispone de una imagen coherente de las señales del proceso durante la ejecución cíclica del programa. Si durante la ejecución del programa varía un estado de señal en un módulo de entrada, dicho estado de señal se conserva en la imagen de proceso hasta que ésta sea actualizada en el próximo ciclo. Si se consulta varias veces una señal de entrada dentro de un programa de usuario, se garantiza que la información de entrada sea siempre coherente.

Además, el acceso a la imagen de proceso requiere mucho menos tiempo que el acceso directo a los módulos de señales, ya que la imagen de proceso se encuentra en la memoria interna de la CPU.

Imágenes parciales del proceso

Además de la imagen del proceso (PAE y PAA) actualizada automáticamente por el sistema operativo, en S7-400 puede parametrizar hasta 15 imágenes parciales del proceso para una CPU (específica de CPU, N° 1 hasta máx. N° 15, consulte "Sistemas de automatización S7-400, M7-400, Datos de los módulos"). Esto permite actualizar, en caso necesario, áreas parciales de la imagen de proceso, independientemente de la actualización cíclica de dicha imagen.

Cada dirección de entrada/salida que ha asignado con STEP 7 a una imagen parcial del proceso ya no pertenece a la imagen del proceso de las entradas/salidas en el OB 1. Las direcciones de entrada/salida sólo se pueden predefinir una vez por medio de la imagen del proceso OB 1 y todas las imágenes parciales.

La imagen parcial del proceso se define con STEP 7 al asignar direcciones (qué direcciones de entrada/salida de los módulos se llevan a qué imagen parcial del proceso). La actualización de la imagen parcial del proceso se realiza a través de las SFCs por parte del usuario o por parte del sistema mediante acoplamiento automático a un OB. Excepción: El sistema no actualiza las imágenes parciales del proceso de los OB de alarmas sincrónicas de impulsos, aunque se encuentran acopladas a un OB (OB 61 a OB 64).

Nota

En las CPUs S7-300, las entradas y salidas de imágenes de proceso no asignadas se pueden utilizar como áreas de marcas adicionales. Los programas que utilizan dicha posibilidad sólo pueden procesarse en CPUs S7-400 más antiguas (es decir, anteriores a 4/99) bajo la siguiente condición:

En dichas CPUs S7-400

- las imágenes del proceso utilizadas como marcas deben estar fuera del "Tamaño de la imagen del proceso" parametrizado, o
 - deben estar en una imagen parcial del proceso que no sea actualizada ni por el sistema ni mediante la SFC 26/SFC 27.
-

Actualizar con SFCs las imágenes parciales del proceso

Con ayuda de SFCs puede actualizar la imagen del proceso completa o imágenes parciales del proceso desde el programa de usuario.

Requisito: la correspondiente imagen parcial del proceso no es actualizada por el sistema

- SFC 26 UPDAT_PI: Actualizar la imagen del proceso de las entradas
- SFC 27 UPDAT_PO: Actualizar la imagen del proceso de las salidas.

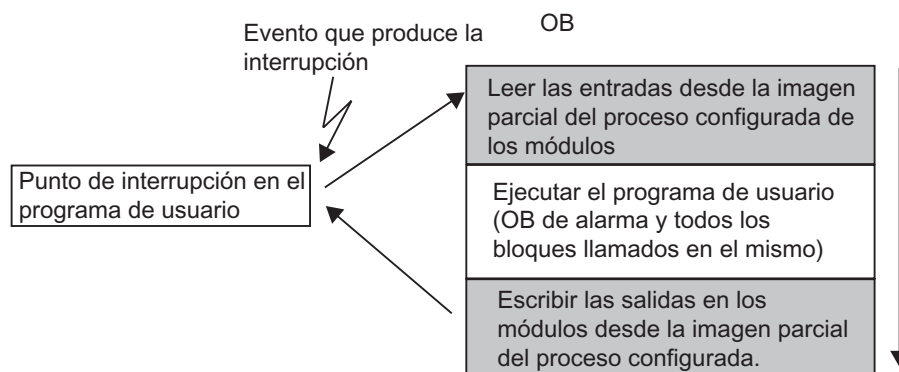
Actualizar las imágenes parciales del proceso mediante el sistema

También puede hacer que el sistema operativo actualice automáticamente las imágenes parciales del proceso (de forma similar a la imagen (global) del proceso) llamando a un OB que se actualice antes o después de la ejecución del OB 1. Esta función sólo puede parametrizarse para determinadas CPUs.

Durante el funcionamiento, la imagen parcial del proceso asignada se actualizará automáticamente:

- antes de la ejecución del OB, la imagen parcial de las entradas del proceso
- después de la ejecución del OB, la imagen parcial de las salidas del proceso

Junto con la prioridad del OB, para la CPU se parametriza qué imagen parcial del proceso se asigna a cada OB.



Error de acceso a la periferia (PZF) al actualizar la imagen del proceso

Cuando se produce un error durante la actualización de la imagen del proceso, la respuesta preajustada de las familias de CPUs (S7-300 y S7-400) es diferente:

- S7-300: No hay registro en el búfer de diagnóstico, no hay llamada a OB, los bytes de entrada correspondientes se ponen a 0 y permanecen en 0 hasta que el fallo desaparece.
- S7-400: Registro en el búfer de diagnóstico y arranque del OB 85 con cada acceso a la periferia en cada actualización de la imagen correspondiente del proceso. Los bytes de entrada defectuosos se ponen a "0" en cada acceso a la imagen del proceso.

En CPUs nuevas (a partir de 4/99) puede reparametrizar la respuesta en caso de que se produzcan errores de acceso a la periferia, de tal manera que la CPU

- genere un registro en el búfer de diagnóstico y arranque el OB 85 sólo con PZF entrante y saliente (antes de la llamada del OB 85 los bytes de entrada defectuosos se ponen a "0" y después el sistema operativo no los sobrescribe más hasta PZF saliente) o
- muestre el comportamiento preajustado para la S7-300 (sin llamada a OB-85 los bytes de entrada correspondientes se ponen a "0" y el sistema operativo no los vuelve a sobrescribir hasta que el fallo se elimina) , o
- muestre el comportamiento preajustado para la S7-400 (llamada a OB-85 con cada acceso concreto, los bytes de entrada defectuosos se ponen a "0" en cada acceso a la imagen del proceso)

¿Con qué frecuencia arranca el OB 85?

Además de la respuesta a PZF parametrizada (PZF entrante/saliente o con cada acceso a la periferia), el espacio de direcciones de un módulo también influye en la frecuencia con la que arranca el OB-85:

En un módulo con un espacio de direcciones hasta de palabra doble, el OB 85 arranca una vez, p. ej. en un módulo digital con un máximo de 32 entradas o salidas, o en un módulo analógico con 2 canales.

En módulos con espacio de direcciones mayor, el BO85 arranca tantas veces como deba accederse a él con comandos de palabra doble, p. ej. dos veces en un módulo analógico de 4 canales.

A.2.3.3 Pila de datos locales

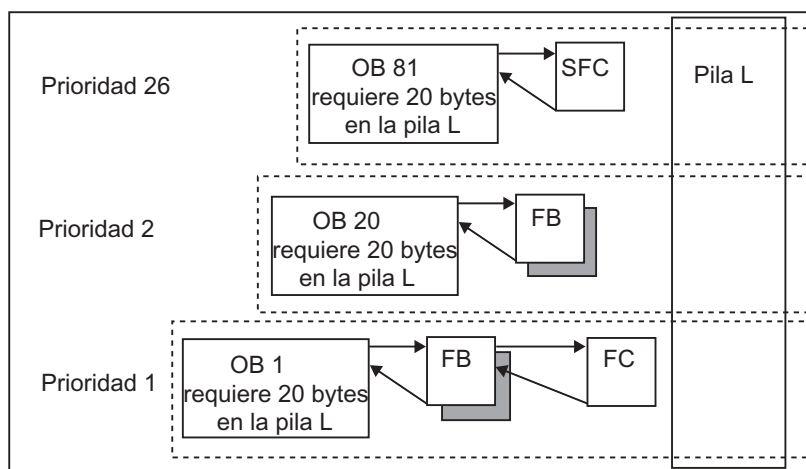
En la pila LSTACK se almacenan:

- las variables temporales de los datos locales de bloques
- la información de arranque de los bloques de organización
- informaciones para la transferencia de parámetros
- resultados intermedios de la lógica en programas escritos en Esquema de contactos.

Para crear bloques de organización se pueden declarar variables temporales (TEMP) que sólo están disponibles durante la ejecución del bloque y se pueden sobrescribir luego. Antes de efectuarse el primer acceso es preciso inicializar los datos locales. Además, cada bloque de organización requiere para su información de arranque 20 bytes de datos locales.

La CPU dispone de una memoria limitada para las variables temporales (datos locales) de los bloques en ejecución. El tamaño de dicha área de la memoria (es decir, de la pila de datos locales) depende del tipo de CPU. La pila de datos locales se divide en partes iguales entre las prioridades (ajuste estándar). Ello significa que cada prioridad dispone de su propia área de datos locales. Así se garantiza que también las prioridades altas, así como sus OBs asignados, dispongan de espacio para sus datos locales.

La figura siguiente muestra la asignación de datos locales a las prioridades. En este ejemplo, la pila LSTACK del OB 1 es interrumpida por el OB 10 que, a su vez, es interrumpido por el OB 81.



Cuidado

Todas las variables temporales (TEMP) de un OB, así como sus bloques subordinados, se memorizan en la pila L (LSTACK). Si se utilizan muchos niveles de anidado para la ejecución de bloques, se podría desbordar la pila L.

Las CPUs S7 pasan al estado operativo STOP cuando se sobrepasa el tamaño admisible de la pila L para un programa.

Compruebe en el programa el estado de la pila L (las variables temporales).

Tenga en cuenta el requerimiento de datos locales de los OBs de errores síncronos.

Asignación de datos locales a las prioridades

No todas las prioridades requieren la misma cantidad de memoria en la pila de datos locales. Las CPUs S7-400 y la permiten parametrizar con STEP 7 distintos tamaños del área de datos locales para las diferentes prioridades. Las prioridades no utilizadas se pueden deseleccionar, permitiendo así, en las CPUs S7-400 y en la CPU 318 ampliar el área de memoria para las otras prioridades. Los OBs deseleccionados no se toman en cuenta en la ejecución del programa, lo que ahorra tiempo de cálculo.

En las otras CPUs S7-300, cada prioridad tiene asignada una cantidad fija de datos locales (256 bytes), que no se puede modificar.

A.2.3.4 Pila de interrupción (USTACK)

Si la ejecución del programa es interrumpida por un OB de mayor prioridad, el sistema operativo memoriza los contenidos actuales de los acumuladores y los registros de direcciones, así como el número y tamaño de los bloques de datos abiertos en la pila de interrupción (pila U o USTACK).

Finalizada la ejecución del nuevo OB, el sistema operativo carga las informaciones desde la pila U y continúa la ejecución del bloque interrumpido en el punto donde ocurrió dicha interrupción.

En el estado operativo STOP se puede leer con STEP 7 en la PG el contenido de la pila U. Así se puede detectar con mayor facilidad por qué la CPU ha pasado a "STOP".

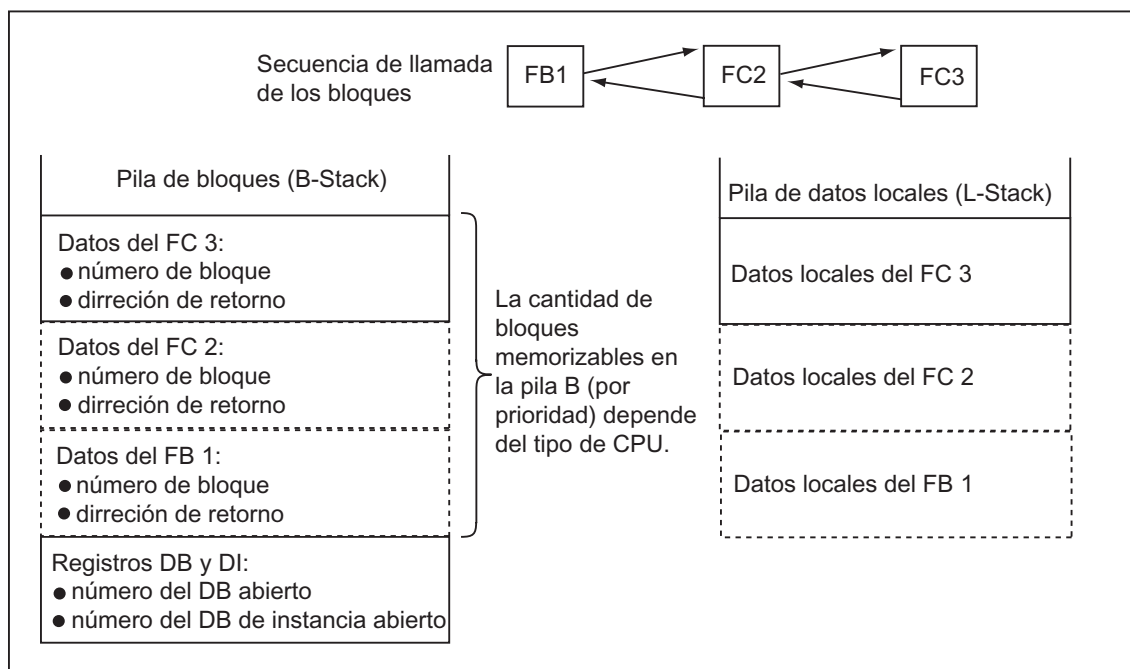
A.2.3.5 Pila de bloques (BSTACK)

Si el tratamiento de un bloque es interrumpido por la llamada de otro bloque o por una prioridad superior (tratamiento de una alarma o de un error), la pila BSTACK memoriza los datos siguientes:

- número, tipo (OB, FB, FC, SFB, SFC) y dirección de retorno del bloque que ha sido interrumpido.
- número de los bloques de datos (tomados de los registros DB y DI) que estaban abiertos en el momento de la interrupción.

Tras la interrupción, el programa de usuario puede continuar con los datos memorizados.

Si la CPU se encuentra en el estado operativo "STOP", es posible visualizar la pila B con STEP 7 en la PG. La pila B lista todos los bloques cuyo tratamiento no había sido concluido al momento en que la CPU fue llevada al estado operativo "STOP". Los bloques se visualizan en la secuencia en que se inició su tratamiento (v. fig. siguiente).



Registros de bloques de datos

Se dispone de dos registros de bloques de datos que contienen los números de los bloques de datos abiertos, a saber:

- el registro DB contiene el número del bloque de datos globales abierto
- el registro DI contiene el número del bloque de datos de instancia abierto.

A.2.3.6 Búfer de diagnóstico

En el búfer de diagnóstico de la CPU se visualizan los eventos de diagnóstico en la secuencia de su aparición. La primera entrada contiene el evento más reciente. El número de entradas del búfer de diagnóstico depende del módulo en cuestión y de su estado operativo actual.

Los eventos de diagnóstico pueden ser:

- errores en un módulo
- errores en el cableado del proceso,
- errores de sistema en la CPU
- cambios de estado operativo de la CPU,
- error en el programa de usuario
- Eventos de diagnóstico de usuario (a través de la función del sistema SFC 52).

A.2.3.7 Evaluación del búfer de diagnóstico

El búfer de diagnóstico es una parte de la lista de estado del sistema. En él se registran informaciones más detalladas acerca de los eventos de diagnóstico del sistema y los eventos definidos por el usuario en la secuencia de su aparición. La información que se inscribe en el búfer de diagnóstico es idéntica a la información de arranque que se transfiere al correspondiente bloque de organización.

Las entradas en el búfer de diagnóstico no se pueden borrar. El contenido del búfer de diagnóstico permanece aún tras el borrado total.

El búfer de diagnóstico ofrece la posibilidad de:

- en caso de parada de la instalación: evaluar los últimos eventos antes del STOP y localizar la causa del STOP
- detectar las causas de los errores con más rapidez y aumentar así la disponibilidad de la instalación
- evaluar y optimizar el comportamiento dinámico de la instalación.

Organización del búfer de diagnóstico

El búfer de diagnóstico está concebido como búfer anular para un número máximo de entradas dependiente del módulo. Si se ha alcanzado dicho número y se presenta un nuevo evento de diagnóstico, se borrará la entrada más antigua. Entonces se desplazan todas las demás entradas. Por lo tanto, el evento de diagnóstico más reciente aparece siempre en primer lugar. En el caso de la CPU 314 S7-300, se trata p.ej. de 100 entradas:

El número de entradas visualizadas en el búfer de diagnóstico depende del módulo y de su estado operativo actual. En determinadas CPUs se puede parametrizar la longitud del búfer de diagnóstico.

Contenido del búfer de diagnóstico

El cuadro **superior** contiene una lista de todos los eventos de diagnóstico que se hayan presentado, incluyendo las siguientes informaciones:

- Número correlativo de la entrada (el evento más reciente es el número 1)
- Hora y fecha del evento de diagnóstico Se visualizan la hora y la fecha del módulo, si éste dispone de un reloj. Para poder aprovechar estos datos de forma apropiada es importante ajustar la hora y la fecha del módulo, así como comprobar los ajustes de vez en cuando;
- Texto del evento (descripción breve).

En el cuadro de texto **inferior** se muestran informaciones adicionales del evento seleccionado en la ventana superior. Dichas informaciones comprenden p.ej.:

- Número del evento
- Denominación del evento
- Cambio de estado operativo debido al evento de diagnóstico
- Indicación de la posición del error en un bloque (tipo y número de bloque, dirección relativa) que ocasionó la entrada del evento
- Evento entrante o saliente
- Informaciones adicionales específicas del evento.

Haciendo clic en el botón "Ayuda del evento" se obtienen informaciones adicionales acerca del evento seleccionado en el cuadro de lista.

Encontrará explicaciones sobre los ID de eventos en la ayuda de referencia de los bloques y funciones de sistema (Saltos a descripciones de lenguajes, ayuda de bloques y atributos de sistema).

Guardar el contenido en un archivo de texto

Para guardar el contenido del búfer de diagnóstico en un archivo de texto ASCII, pulse el botón "Guardar como" en la ficha "Búfer de diagnóstico" del cuadro de diálogo "Información del módulo".

Leer el búfer de diagnóstico

El contenido del búfer de diagnóstico se puede visualizar mediante en la ficha "Búfer de diagnóstico" del cuadro de diálogo "Información del módulo", o bien en un programa a través de la función de sistema SFC 51 RDSYSST.

Ultima entrada antes de STOP

Se puede definir que la última entrada del búfer de diagnóstico antes del cambio de RUN a STOP se emita automáticamente a un visualizador registrado (p.ej. PG, OP, TD) para garantizar que la causa del cambio de estado operativo a STOP se pueda detectar y corregir más rápidamente.

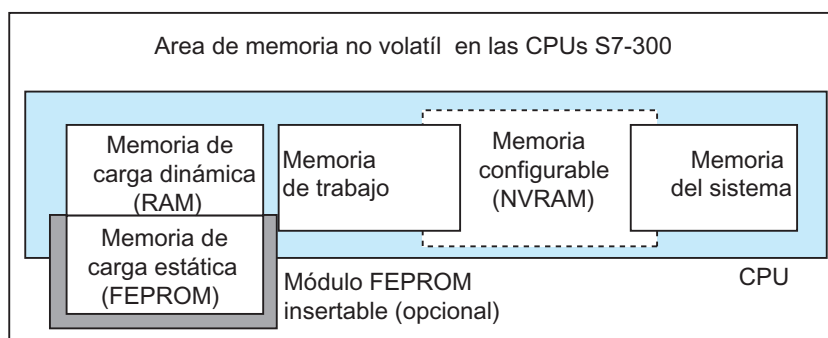
A.2.3.8 Áreas remanentes de la memoria en CPUs S7-300

En caso de corte de la alimentación o de borrado total (MRES), la memoria de la CPU S7-300 (memoria de carga dinámica (RAM), memoria de trabajo y memoria del sistema) es borrada, perdiéndose todos los datos memorizados en dichas áreas. Las CPUs S7-300 ofrecen las siguientes posibilidades para salvaguardar el programa y los datos:

- Los datos memorizados en las memorias de carga, de trabajo y en secciones de la memoria del sistema se pueden respaldar con una pila tampón.
- El programa se puede guardar en la EPROM (en una Memory Card o integrada en la CPU, v. el manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU").
- Una determinada cantidad de datos dependientes de la CPU se puede salvaguardar en una área de la NVRAM no volátil.

Uso de la RAM no volátil

La CPU S7-300 ofrece un área en NVRAM (RAM no volátil) (v. fig. siguiente). Si el programa ha sido depositado en la EPROM de la memoria de carga, algunos de los datos (en caso de corte de alimentación o al pasar la CPU de STOP a RUN) se pueden memorizar también si se prevé la configuración correspondiente.



A tal efecto, la CPU se ha de ajustar de tal manera que los datos siguientes sean memorizados en la RAM no volátil:

- datos memorizados en un DB (esto sólo resulta ventajoso si el programa ha sido depositado en una EPROM de la memoria de carga)
- valores de temporizadores y contadores
- datos memorizados en marcas

En cada CPU se puede salvaguardar una cantidad determinada de temporizadores, contadores y marcas. Además, se ofrece una cantidad específica de bytes para memorizar los datos depositados en DB.

La dirección MPI de la CPU está depositada en la NVRAM. Esto garantiza la comunicabilidad de la CPU incluso tras un corte de la alimentación o tras el borrado total.

Utilizar el respaldo por pila para almacenar los datos

El respaldo por pila permite conservar de forma remanente el contenido de las memorias de carga y de trabajo al producirse un corte de alimentación. Independientemente de dicho respaldo, si en la configuración se memorizaron los temporizadores, contadores y marcas en la RAM no volátil (NVRAM), entonces tampoco se perderán estas informaciones.

Configurar los datos de la NVRAM

Al configurar la CPU con STEP 7 se puede definir qué áreas de memoria deben ser remanentes, es decir, no volátiles.

La cantidad de memoria configurable en la NVRAM depende del tipo de CPU. No es posible salvaguardar más datos a los indicados para la correspondiente CPU.

A.2.3.9 Áreas remanentes de la memoria de las CPUs S7-400

Modo sin respaldo

En caso de falta de corriente o de borrado total (MRES) en modo sin respaldo se borra la memoria de la CPU S7-400 (memoria de carga dinámica (RAM), memoria de trabajo y memoria del sistema), perdiéndose todos los datos memorizados en dichas áreas.

En modo sin respaldo sólo es posible el re arranque completo (en caliente) y no existen áreas de memoria remanentes. Tras un corte de tensión sólo se conservan los parámetros MPI (p. ej. la dirección MPI de la CPU). Esto garantiza la comunicabilidad de la CPU tras una interrupción de corriente o un borrado total.

Modo respaldado

En modo respaldado

- se conservan completamente todas las áreas RAM al re arrancar tras un corte de tensión.
- se borran en el re arranque completo (en caliente) las áreas de operandos correspondientes a las marcas, temporizadores y contadores. Se conservan los contenidos de los bloques de datos.
- se conserva el contenido de la memoria de trabajo RAM, exceptuando las marcas, temporizadores y contadores parametrizados como no remanentes.

Configurar áreas de datos no remanentes

Conforme al tipo de CPU, se pueden configurar como remanentes una determinada cantidad de marcas, temporizadores y contadores. Así, estos datos se conservan en caso de re arranque completo (en caliente) en modo respaldado.

Al parametrizar con STEP 7 se define qué marcas, temporizadores y contadores deben ser remanentes en caso de re arranque completo (en caliente). No es posible respaldar más datos que la cantidad admisible para el tipo de CPU correspondiente.

Para obtener informaciones más detalladas acerca de la parametrización de las áreas de memoria remanentes, consulte el manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos".

A.2.3.10 Objetos configurables en la memoria de trabajo

En el caso de algunas CPUs se puede ajustar mediante HW Config el tamaño de objetos tales como los datos locales o el búfer de diagnóstico. Si, por ejemplo, reduce los valores predeterminados, se dispondrá de una mayor parte de la memoria de trabajo para otros fines. Los ajustes de dichas CPUs se pueden visualizar en la ficha "Memoria" de la información del módulo (botón "Detalles").

Tras cambiar la configuración de la memoria y cargar los datos en el sistema de destino, es preciso efectuar un arranque en frío de éste último para que los cambios tengan efecto.

A.3 Tipos de datos y parámetros

A.3.1 Introducción a los tipos de datos y de parámetros

Todos los datos utilizados en el programa de usuario se deben identificar con un tipo de datos. Se distingue entre:

- tipos de datos simples que ofrece STEP 7,
- tipos de datos compuestos generados combinando tipos de datos simples y
- tipos para definir los parámetros a transferir a los FBs o las FCs.

Informaciones generales

Las operaciones AWL, FUP o KOP utilizan objetos de datos de un tamaño determinado. Como su nombre indica, las operaciones lógicas de combinación de bits utilizan bits. Las operaciones de carga y transferencia (AWL), así como las operaciones de transferencia (FUP y KOP) utilizan bytes, palabras y palabras dobles.

Un bit es una cifra binaria ("0" o "1"). Un byte comprende 8 bits, una palabra se compone de 16 bits y una palabra doble de 32 bits.

Las operaciones aritméticas utilizan también bytes, palabras o palabras dobles. En estos operandos de bytes, palabras o palabras dobles se pueden codificar números de diversos formatos tales como enteros y números en coma flotante.

Si utiliza el direccionamiento simbólico, deberá definir los símbolos e introducir un tipo de datos para los mismos (v. la tabla siguiente). Los diversos tipos de datos tienen diferentes opciones de formato y representaciones numéricas.

En el presente capítulo sólo se describen posibles notaciones para números y constantes. En la tabla siguiente figuran formatos de números y de constantes que no se explican detalladamente.

Formato	Tamaño en bits	Representación numérica
Hexadecimal	8, 16 y 32	B#16#, W#16# y DW#16#
Binario	8, 16 y 32	2#
Fecha IEC	16	D#
Tiempo IEC	32	T#
Hora	32	TOD#
BYTE	8	'A'

A.3.2 Tipos de datos simples

Cada tipo de dato simple tiene una longitud definida. La tabla siguiente muestra los tipos de datos simples.

Tipo y descripción	Tamaño en bits	Opciones de formato	Márgenes y representación numérica (del valor inferior hasta el valor superior)	Ejemplo
BOOL (bit)	1	Texto booleano	TRUE/FALSE	TRUE
BYTE (byte)	8	Número hexadecimal	B#16#0 hasta B#16#FF	L B#16#10 L byte#16#10
WORD (palabra)	16	Número binario Número hexadecimal BCD Número decimal sin signo	2#0 hasta 2#1111_1111_1111_1111 W#16#0 hasta W#16#FFFF C#0 hasta C#999 B#(0,0) hasta B#(255,255)	L 2#0001_0000_0000_0000 L W#16#1000 L word#16#1000 L C#998 L B#(10,20) L byte#(10,20)
DWORD (palabra doble)	32	Número binario Número hexadecimal Número decimal sin signo	2#0 hasta 2#1111_1111_1111_1111_ 1111_1111_1111_1111 DW#16#0000_0000 hasta DW#16#FFFF_FFFF B#(0,0,0,0) hasta B#(255,255,255,255)	2#1000_0001_0001_1000_ 1011_1011_0111_1111 L DW#16#00A2_1234 L dword#16#00A2_1234 L B#(1, 14, 100, 120) L byte#(1,14,100,120)
INT (número entero)	16	Número decimal con signo	-32768 hasta 32767	L 1
DINT (entero de 32 bits)	32	Número decimal con signo	L#-2147483648 hasta L#2147483647	L L#1
REAL (número en coma flotante)	32	IEEE Número en coma flotante	Límite superior: $\pm 3.402823e+38$ Límite inferior: $\pm 1.175495e-38$	L 1.234567e+13
S5TIME (tiempo SIMATIC)	16	Tiempo S7 en pasos de 10 ms (valor estándar)	S5T#0H_0M_0S_10MS hasta S5T#2H_46M_30S_0MS y S5T#0H_0M_0S_0MS	L S5T#0H_1M_0S_0MS L S5TIME#0H_1H_1M_0S_0MS
TIME (tiempo IEC)	32	Tiempo IEC en intervalos de 1 ms, número entero con signo	-T#24D_20H_31M_23S_648MS hasta T#24D_20H_31M_23S_647MS	L T#0D_1H_1M_0S_0MS L TIME#0D_1H_1M_0S_0MS
DATE (fecha IEC)	16	Fecha IEC en pasos de 1 día	D#1990-1-1 hasta D#2168-12-31	L D#1994-3-15 L DATE#1994-3-15
TIME_OF_DAY (hora)	32	Hora en intervalos de 1 ms	TOD#0:0:0.0 hasta TOD#23:59:59.999	L TOD#1:10:3.3 L TIME_OF_DAY#1:10:3.3

A.3.2.3 Formato del tipo de datos REAL (números en coma flotante)

Los números en coma flotante se representan de la manera habitual "Número = $m * b$ elevado a E". La base "b" y el exponente "E" son números enteros, la mantisa "m" un número racional.

Esta representación ofrece la ventaja de poder representar valores muy grandes y muy pequeños en un espacio muy limitado. Con el número limitado de bits para mantisa y exponente se cubre otro margen numérico.

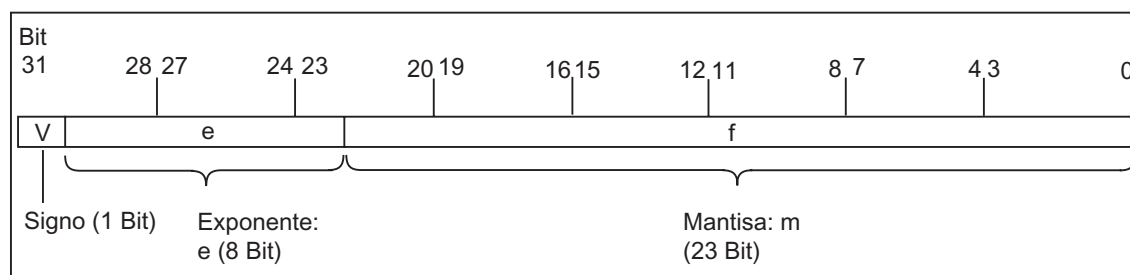
La desventaja radica en la limitada exactitud del cálculo: por ejemplo al formar la suma de dos números hay que igualar los exponentes (suma de las mantisas de dos números de igual exponente) desplazando la mantisa (coma flotante).

Coma flotante en STEP 7

Los números en coma flotante corresponden en STEP 7 al formato básico de ancho simple como se describe en la norma ANSI/IEEE Standard 754-1985, *IEEE Standard for Binary Floating-Point Arithmetic*. Están formados por los siguientes componentes:

- el signo S
- el exponente $e = E + \text{Bias}$ elevado a una constante (Bias = +127)
- la fracción de la mantisa m.
La parte entera de la mantisa no se guarda, ya que siempre es 1 dentro del margen permitido

Los tres componentes ocupan en total una palabra doble (32 bits):



La tabla siguiente muestra el valor de los distintos bits en formato de coma flotante.

Componente del número en coma flotante	Bit	Valor
Signo S	31	
Exponente e	30	2 elevado a 7
...
Exponente e	24	2 elevado a 1
Exponente e	23	2 elevado a 0
Mantisa m	22	2 elevado a -1
...
Mantisa m	1	2 elevado a -22
Mantisa m	0	2 elevado a -23

Los tres componentes **S**, **e** y **m** permiten definir el valor de un número representado en este formato aplicando la fórmula:

$$\text{Número} = 1, m * 2 \text{ elevado } (e - \text{Bias})$$

Considerando que:

- e: $1 \leq e \leq 254$
- Bias: Bias = 127. Así no hace falta un signo adicional para el exponente.
- S: si el número es positivo, S = 0 y si es negativo, S = 1.

Margen numérico de los números en coma flotante

Del formato de coma flotante que acabamos de describir resulta:

- Número en coma flotante más bajo = $1,0 * 2 \text{ elevado a } (1 - 127) = 1,0 * 2 \text{ elevado a } (-126) = 1,175 495E-38$
- Número en coma flotante más alto = $2 - 2 \text{ elevado a } (-23) * 2 \text{ elevado a } (254 - 127) = 2 - 2 \text{ elevado a } (-23) * 2 \text{ elevado a } (+127) = 3,402 823E+38$

El número 0 se representa con e = m = 0; e = 255 y m = 0 significan "infinito".

Formato	Margen ¹⁾
Números en coma flotante según ANSI/IEEE Standard	-3,402 823E+38 hasta -1,175 495E-38 y 0 y +1,175 495E-38 hasta +3,402 823E+38

La tabla siguiente muestra el estado de señal de los bits de la palabra de estado cuando los resultados de las operaciones con números en coma flotante no se encuentran dentro del margen permitido.

Margen no permitido para un resultado	A1	A0	OV	OS
-1,175494E-38 < resultado < -1,401298E-45 (número negativo) rebase por defecto	0	0	1	1
+1,401298E-45 < resultado < +1,175494E-38 (número positivo) rebase por defecto	0	0	1	1
Resultado < -3,402823E+38 (número negativo) rebase por exceso	0	1	1	1
Resultado > 3,402823E+38 (número positivo) rebase por exceso	1	0	1	1
Número u operación no válida (valor de entrada fuera del margen permitido)	1	1	1	1

Atención con las operaciones matemáticas:

El resultado "Número en coma flotante no válido" se obtiene, por ejemplo, cuando se intenta extraer la raíz cuadrada de -2. Por consiguiente, cuando trabaje con operaciones matemáticas evalúe primero los bits de estado antes de seguir calculando con el resultado.

Atención al "Forzar variables":

Si deposita los valores de las operaciones en coma flotante p.ej. en una palabra doble de marcas, estos valores se pueden modificar con cualquier configuración binaria. Sin embargo, no todas representan un número válido.

Precisión en cálculos con números en coma flotante



Cuidado

En las operaciones de cálculo con números que abarquen p. ej. varias potencias de 10, el resultado puede no ser exacto.

Los números en coma flotante se representan en STEP 7 con una precisión de 6 dígitos detrás de la coma, por lo que al indicar constantes en coma flotante no podrá indicar más de 6.

Nota

La precisión de 6 dígitos detrás de la coma significa p. ej., que la suma de número1 + número2 = número1, si el número1 es mayor que el número2 * 10 elevado a y, siendo y>6:

$$100\ 000\ 000 + 1 = 100\ 000\ 000.$$

Ejemplos de formatos de números en coma flotante

La figura siguiente muestra el formato de números en coma flotante para los siguientes valores decimales:

- 10,0
- Pi (3,141593)
- Raíz cuadrada de 2 (1,414214)

En el primer ejemplo, el número **10,0** resulta del formato en coma flotante (representación hexadecimal: 4120 0000) según la siguiente fórmula:

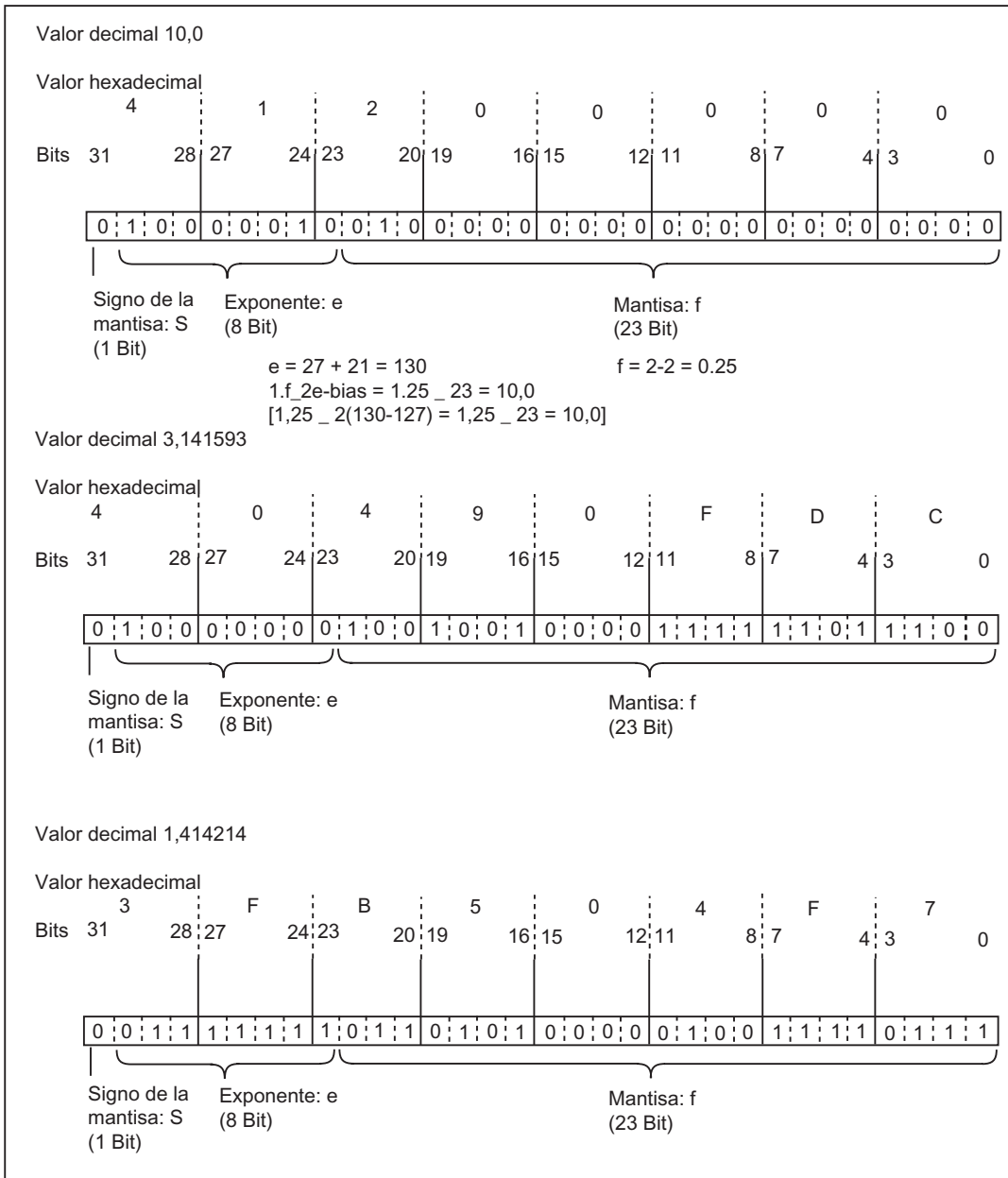
$$e = 2 \text{ elevado a } 7 + 2 \text{ elevado a } 1 = 130$$

$$m = 2 \text{ elevado a } (-2) = 0,25$$

De lo que resulta:

$$(1 + m) * 2 \text{ elevado a } (e - \text{Bias}) = 1,25 * 2 \text{ elevado a } 3 = 10,0.$$

$$[1,25 * 2 \text{ elevado a } (130-127) = 1,25 * 2 \text{ elevado a } 3 = 10,0]$$



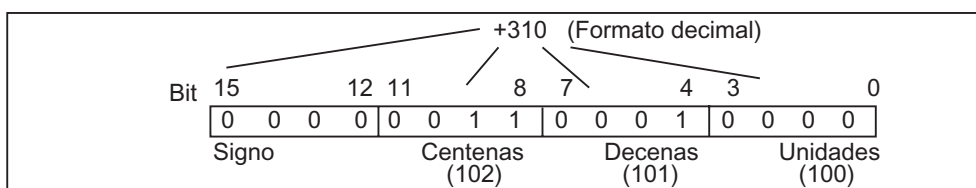
A.3.2.4 Formato de los tipos de datos WORD y DWORD en los números decimales codificados en binario

La notación codificada en binario (BCD) representa un número decimal en grupos de cifras binarias (bits). Un grupo de 4 bits representa una cifra de un número decimal o bien el signo de dicho número. Los grupos de 4 bits constituyen una palabra (16 bits) o una palabra doble (32 bits). Los cuatro bits más significativos indican el signo del número ("1111" significa "negativo" y "0000" significa "positivo"). Las instrucciones con operandos BCD sólo evalúan el bit más significativo (15 en el caso del formato de palabra o 31 en el de palabra doble). La tabla siguiente muestra el formato y el margen de los dos tipos de números BCD.

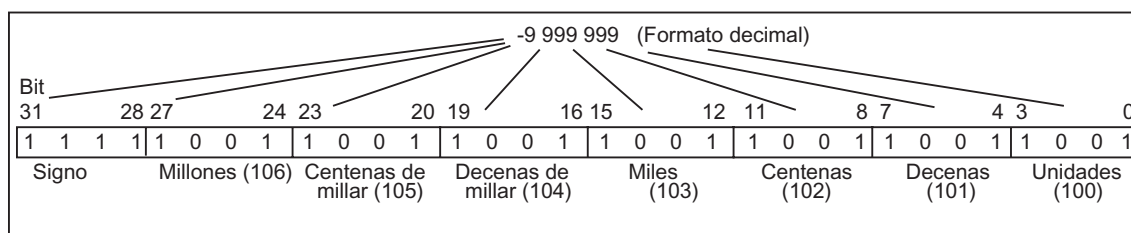
Formato	Grupo
Palabra (16 bits, número BCD de 3 cifras con signo)	-999 hasta +999
Palabra doble (32 bits, número BCD de 7 cifras con signo)	-9 999 999 hasta +9 999 999

Las figuras siguientes muestran ejemplos de un número decimal codificado en binario en los siguientes formatos:

- Formato de palabra

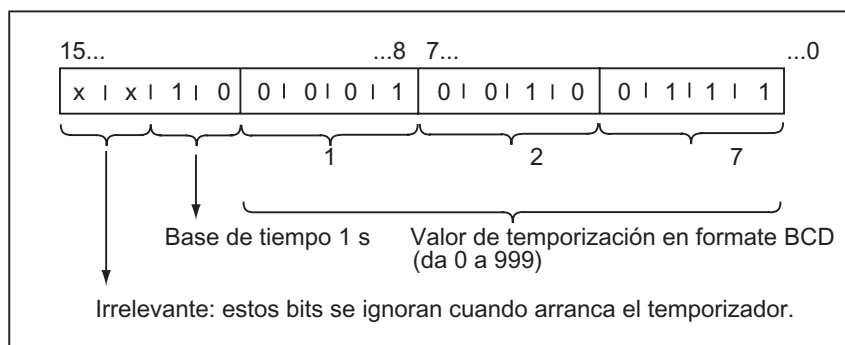


- Formato de palabra doble



A.3.2.5 Formato del tipo de datos S5TIME (intervalo)

Si introduce el intervalo con el tipo de datos S5TIME, sus entradas se guardarán en formato BCD. La figura siguiente muestra el contenido del operando de tiempo, donde el valor de tiempo es 127 y la base de tiempo es 1 s.



Al trabajar con S5TIME deberá introducir un valor de tiempo comprendido entre 0 y 999, así como determinar una base de tiempo (v. la tabla siguiente). La base de tiempo indica el intervalo en el que un temporizador reduce el valor de tiempo en una unidad hasta alcanzar "0".

Base de tiempo para S5TIME

Base de tiempo	Código binario para la base de tiempo
10 ms	00
100 ms	01
1 s	10
10 s	11

Con la siguiente sintaxis se puede cargar un valor de tiempo predefinido:

- L¹⁾ W#16#wxyz
 - con: w = base de tiempo (es decir, intervalo de tiempo o resolución)
 - xyz = valor de tiempo en formato BCD
- L¹⁾ S5T#aH_bbM_ccS_dddMS
 - con: a = horas, bb = minutos, cc = segundos y ddd = milisegundos.
 - La base de tiempo se selecciona automáticamente y el valor se redondea al número inferior siguiente que tenga dicha base de tiempo.

Se puede introducir un valor de tiempo de máx. 9 990 segundos (ó 2H_46M_30S).

¹⁾ = indicar L sólo en la programación AWL

A.3.3 Tipos de datos compuestos

Los tipos de datos compuestos se definen como grupos de datos superiores a 32 bits o grupos de datos compuestos de varios tipos de datos. STEP 7 admite los siguientes tipos de datos compuestos:

- DATE_AND_TIME
- STRING
- ARRAY (campo o matriz)
- STRUCT (estructura)
- UDT (tipos de datos de usuario)
- FB y SFB

La tabla siguiente muestra los tipos de datos compuestos. Las estructuras y los campos se definen bien en la declaración de variables del bloque lógico o bien en un bloque de datos.

Tipo de datos	Descripción
DATE_AND_TIME DT	Define un área de 64 bits (8 bytes). Este tipo de datos memoriza en formato decimal codificado en binario:
STRING	Define un grupo de un máximo de 254 caracteres (tipo de datos CHAR). El área estándar reservada para una cadena de caracteres consta de 256 bytes. Este es el espacio requerido para memorizar 254 caracteres y un encabezamiento de 2 bytes. La capacidad de memoria requerida para una cadena de caracteres se puede reducir definiendo también la cantidad de caracteres a memorizar en dicha cadena (p.ej.: string[9] 'Siemens').
ARRAY	Define un agrupamiento multidimensional, similar a una matriz, de un tipo de datos (simple o compuesto). Por ejemplo: "ARRAY [1..2,1..3] OF INT" define un campo en formato de 2 x 3 números enteros. A los datos memorizados en un campo se accede a través del índice ("[2,2]"). En un campo se pueden definir hasta un máximo de 6 dimensiones. El índice puede ser un número entero discrecional (-32768 a 32767).
STRUCT	Define un agrupamiento de tipos de datos combinados discrecionalmente. Por ejemplo, se puede definir un campo compuesto de estructuras o una estructura compuesta de estructuras y campos.
UDT	Permite estructurar grandes cantidades de datos, simplificando así la entrada de tipos de datos al crear bloques de datos o al declarar las variables en la declaración correspondiente. STEP 7 permite combinar tipos de datos compuestos y simples, creando así un tipo de datos propio "de usuario" (UDT). UDTs tienen un nombre propio y, por consiguiente, pueden utilizarse varias veces.
FB, SFB	Determinan la estructura del bloque de datos de instancia asignado y permiten la transferencia de datos de instancia para varias llamadas de FB en un DB de instancia.

Los tipos de datos estructurados se depositan alineados palabra por palabra (WORD aligned).

A.3.3.1 Formato del tipo de datos DATE_AND_TIME (fecha y hora)

Si se introduce la fecha y la hora utilizando el tipo de datos DATE_AND_TIME (DT), sus entradas se almacenarán en formato BCD en 8 bytes. El tipo de datos DATE_AND_TIME abarca el siguiente margen:

- DT#1990-1-1-0:0:0.0 a DT#2089-12-31-23:59:59.999

Los siguientes ejemplos muestran la sintaxis para introducir la fecha y la hora del jueves 25 de diciembre de 1993, 8:12 horas y 34,567 segundos. Son posibles los siguientes formatos:

- DATE_AND_TIME#1993-12-25-8:12:34.567
- DT#1993-12-25-8:12:34.567

Se dispone de las siguientes funciones estándar IEC (International Electrotechnical Commission) para trabajar con el tipo de datos DATE_AND_TIME:

- Convertir la fecha y la hora al formato DATE_AND_TIME
FC3: D_TOD_DT
- Extraer la fecha del formato DATE_AND_TIME
FC6: DT_DATE
- Extraer el día de la semana del formato DATE_AND_TIME
FC7: DT_DAY
- Extraer la hora del formato DATE_AND_TIME
FC8: DT_TOD

La tabla siguiente muestra el contenido de los bytes que comprenden las informaciones sobre la fecha y la hora. El ejemplo muestra la fecha y la hora del jueves 25 de diciembre de 1993, 8:12 horas y 34,567 segundos.

Byte	Contenido	Ejemplo
0	Año	B#16#93
1	Mes	B#16#12
2	Día	B#16#25
3	Hora	B#16#08
4	Minuto	B#16#12
5	Segundo	B#16#34#
6	Las dos cifras más significativas de MSEC	B#16#56
7 (4MSB)	Las cifra menos significativa de MSEC	B#16#7_
7 (4LSB)	Día de la semana 1 = Domingo 2 = Lunes ... 7 = Sábado	B#16#_5

El margen admisible para el tipo de datos "DATE_AND_TIME" es:

- mín.: DT#1990-1-1-0:0:0.0
- máx.: DT#2089-12-31-23:59:59.999

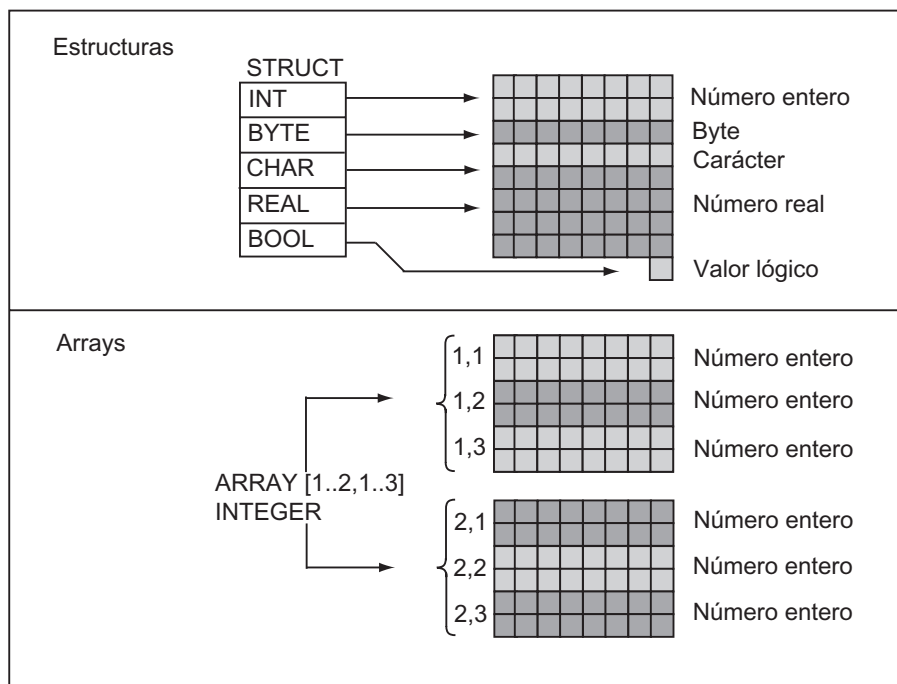
	Margen posible de valores	Código BCD
Año	1990 - 1999	90 - 99
	2000 - 2089	00 - 89
Mes	1 - 12	01 - 12
Día	1 - 31	01 - 31
Hora	00 - 23	00 - 23
Minuto	00 - 59	00 - 59
Segundo	00 - 59	00 - 59
Milisegundo	0 - 999	000 - 999
Día de la semana	Domingo - Sábado	1 - 7

A.3.3.2 Uso de tipos de datos compuestos

Es posible crear nuevos tipos de datos combinando los datos simples y compuestos con los siguientes tipos de datos compuestos:

- Array (tipo de datos ARRAY): Un array o matriz combina un grupo de datos de un tipo formando una unidad.
- Estructura (tipo de datos STRUCT): Una estructura combina datos de diferente tipo formando una unidad.
- Cadena de caracteres (tipo de datos STRING): Una cadena de caracteres define un array unidimensional con un máximo de 254 caracteres (tipo de datos CHAR). Una cadena de caracteres se puede transferir únicamente como unidad. La longitud de la cadena de caracteres debe coincidir en los parámetros formal y actual del bloque.
- Fecha y hora (tipo de datos DATE_AND_TIME): La fecha y la hora memorizan año, mes, día, horas, minutos, segundos, milisegundos y día de la semana.

La figura siguiente muestra cómo estructurar los arrays y las estructuras de tipos de datos en un área para memorizar informaciones. Un campo o una estructura se puede definir en un DB o en una declaración de variables de un FB, OB o de una FC.



A.3.3.3 Uso de arrays (matrices) para acceder a los datos

Arrays (matrices)

Un array o matriz combina un grupo de datos de un tipo (simple o compuesto) formando una unidad. Un array no se puede formar a partir de otros arrays. Los arrays se definen de la siguiente manera:

- Indicar el nombre del array.
- Declarar un array con la clave ARRAY.
- Indicar el tamaño del array a través de un índice. Se ha de indicar el primer y el último número de las diferentes dimensiones (máximo 6) en el array. El índice se indica en corchetes, separando cada dimensión por coma mientras que el primer y el último número de cada dimensión se separan por dos puntos suspensivos. El índice siguiente define, p. ej., un array tridimensional:

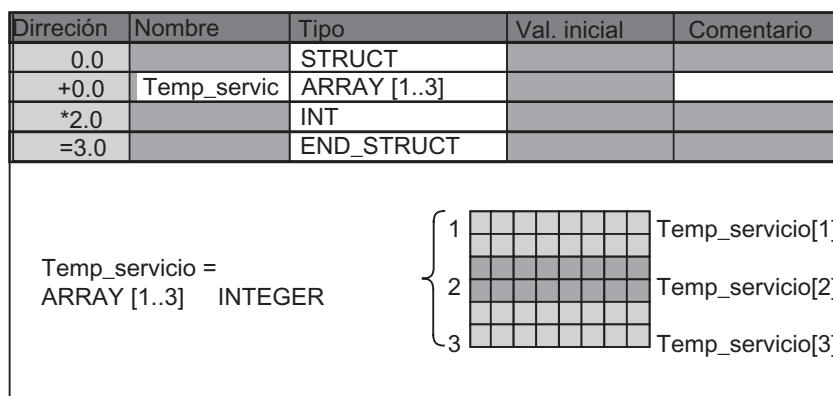
[1..5,-2..3,30..32]

- Indicar el tipo de los datos que se han de memorizar en el array.

Ejemplo 1

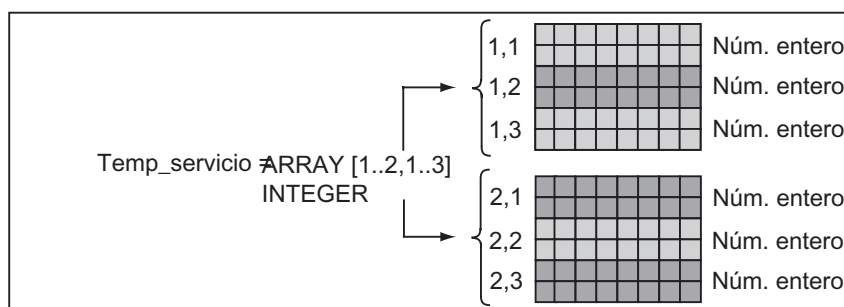
La figura siguiente muestra un array compuesto de tres números enteros. A través del índice se accede a los datos memorizados en un array. El índice consiste en el número escrito entre corchetes. El índice del segundo entero es p. ej. Temperatura_servicio[2].

El índice puede ser un entero cualquiera (-32768 a 32767), incluidos valores negativos. El array de la figura siguiente podría ser definido también como ARRAY [-1..1]. El índice del primer entero sería entonces Temp_servicio[-1], el segundo Temp_servicio[0] y el tercer entero Temp_servicio[1].



Ejemplo 2

Un array puede definir también un grupo multidimensional de tipos de datos. La figura siguiente muestra un array bidimensional compuesto de números enteros.



A los datos de un array multidimensional se accede a través del índice. En el ejemplo, el primer entero es Temp_servicio[1,1], el tercero el Temp_servicio[1,3], el cuarto el Temp_servicio[2,1] y el sexto es Temp_servicio[2,3].

Para un array se puede definir hasta un máximo de seis dimensiones (seis índices). Por ejemplo, la variable Temp_servicio se puede definir como array hexadimensional:

ARRAY [1..3,1..2,1..3,1..4,1..3,1..4]

El índice del primer elemento en este array es Temp_servicio[1,1,1,1,1,1]. El índice del último elemento es Temp_servicio[3,2,3,4,3,4].

Crear arrays

Los arrays se definen declarando los datos en un DB o en la declaración de variables. Al declarar un array se ha de indicar la palabra clave (ARRAY) y luego el tamaño entre corchetes:

[límite inferior..límite superior]

En un array multidimensional se indican los límites superior e inferior adicionales y se separan las diferentes dimensiones por coma. La figura siguiente muestra la declaración para crear un array en formato 2 x 3.

Dirección	Nombre	Tipo	Val. inicial	Comentario
0.0		STRUCT		
+0.0	Calor_2X3	ARRAY[1..2,1..3]		
*2.0		INT		
=6.0		END_STRUCT		

Introducir valores iniciales para un array

Al crear un array se puede asignar un valor inicial a cada elemento del array. STEP 7 ofrece dos posibilidades para entrar los valores iniciales:

- Entrada de valores individuales: para cada elemento del array se indica un valor válido (para el tipo de datos del campo). Indicar los valores en la secuencia de los elementos: [1,1]. Tenga en cuenta que los diferentes elementos se han de separar por coma.
- Indicar un factor de repetición: para elementos secuenciales que disponen del mismo valor inicial, se puede indicar la cantidad de elementos (el factor de repetición) y el valor inicial para dichos elementos. El formato para indicar el factor de repetición es $x(y)$, siendo x el factor de repetición e y el valor que ha de ser repetido.

Si se utiliza el array declarado en la figura anterior, el valor inicial para los seis elementos se puede indicar de la siguiente manera: 17, 23, -45, 556, 3342, 0. El valor inicial de todos los seis elementos también se puede poner a 10, indicando para ello 6(10). Para los primeros dos elementos pueden indicarse valores determinados y luego los cuatro restantes pueden ajustarse a 0, indicando 17, 23, 4(0).

Acceder a datos en un array

A los datos memorizados en un array se accede a través del índice. El índice se utiliza con el nombre simbólico.

Ejemplo: Si el array declarado en la figura anterior empieza en el primer byte del DB20 (motor), se accede al segundo elemento en el array a través de la dirección siguiente:

Motor.Calor_2x3[1,2].

Utilizar arrays como parámetros

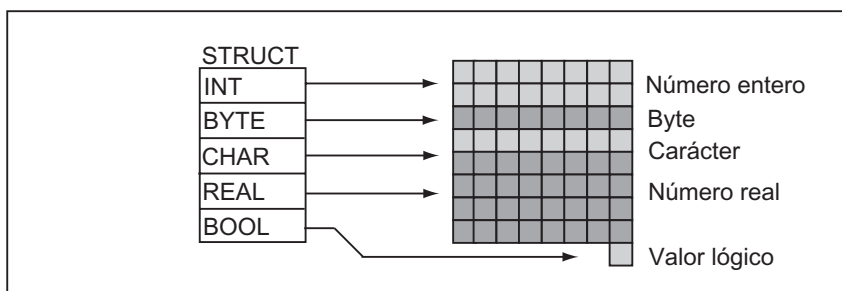
Los arrays se pueden transferir en calidad de parámetros. Si un parámetro ha sido declarado como ARRAY en la declaración de variables, entonces se ha de transferir el array completo (y no solamente elementos individuales). No obstante, es posible asignar un parámetro a un elemento del campo, llamando un bloque y siempre y cuando el elemento del array corresponda al tipo de datos del parámetro.

Si los arrays se utilizan como parámetros, los arrays no deben tener el mismo nombre (en realidad no necesitan ninguno). No obstante, ambos arrays (el parámetro formal y el actual) deben tener la misma estructura. Por ejemplo, un array en formato 2 x 3 enteros se puede transferir como parámetro solamente si el parámetro formal del bloque es un array en formato 2 x 3 enteros y también el parámetro actual, que se suministra a través de la operación de llamada, es un array en formato 2 x 3 enteros.

A.3.3.4 Uso de estructuras para acceder a los datos

Estructuras

Una estructura combina diferentes tipos de datos (datos simples y compuestos, incluidos arrays y estructuras) formando una unidad. Así, los datos se pueden agrupar conforme al control para el proceso en cuestión. Esto permite también transferir parámetros como una unidad de datos y no como elementos individuales. La figura siguiente muestra una estructura compuesta de un entero, un byte, un carácter, un número en coma flotante y un valor booleano o lógico.



Una estructura se puede anidar en 8 niveles como máximo (p. ej. una estructura de estructuras que contenga arrays).

Crear una estructura

Las estructuras se definen durante la declaración de datos dentro de un DB o en la declaración de variables de un bloque lógico.

La figura siguiente muestra la declaración de una estructura (*Lote_1*), que consta de los siguientes elementos: un entero (para memorizar la cantidad), un byte (para memorizar los datos originales), un carácter (para memorizar el código de control), un número en coma flotante (para memorizar la temperatura) y una marca booleana (para finalizar la señal).

Dirección	Nombre	Tipo	Val. inicial	Comentario
0.0	Pila_1	STRUCT		
+0.0	Cantidad	INT	100	
+2.0	Datos_originales	BYTE		
+4.0	Código_de_control	CHAR		
+6.0	Temperatura	REAL	120	
+8.0	Fin	BOOL	FALSE	
=10.0		END_STRUCT		

Asignar valores iniciales para una estructura

Si se desea asignar un valor inicial a cada elemento de una estructura, entonces se ha de indicar un valor válido tanto para el tipo de datos como para el nombre del elemento. Por ejemplo, es posible asignar (a la estructura declarada en la figura anterior) los siguientes valores iniciales:

```
Cantidad           = 100
Datos originales   = B#(0)
Código de control = 'Z'
Temperatura       = 120
Fin               = False
```

Guardar datos y acceder a los mismos en estructuras

Debe acceder a cada uno de los elementos de una estructura. Puede utilizar direcciones simbólicas (p. ej. *Lote_1.Temperatura*). Pero también se puede indicar la dirección absoluta bajo la cual se ha de memorizar el elemento (ejemplo: si *Lote_1* está memorizado en el DB20 con inicio en el byte 0, la dirección absoluta para la *cantidad* es *DB20.DBW0* y la dirección para la *temperatura* es *DB20.DBD6*).

Utilizar estructuras como parámetros

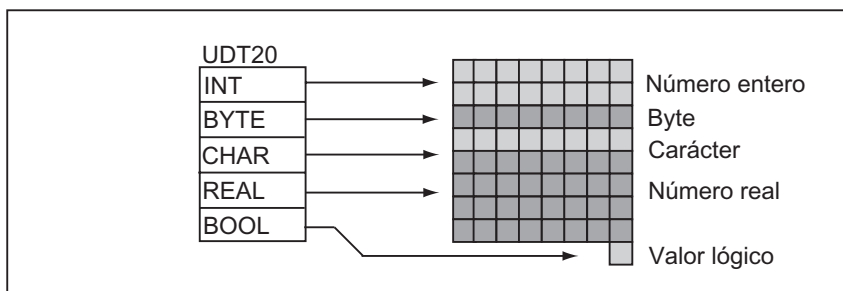
Las estructuras se pueden transferir en calidad de parámetros. Si un parámetro está declarado como STRUCT en la declaración de variables, entonces se ha de transferir una estructura con la misma configuración. No obstante, se puede asignar un elemento de una estructura a un parámetro al llamar el bloque, siempre que el elemento de la estructura corresponda al tipo de datos del parámetro.

Si se utilizan estructuras como parámetros, ambas estructuras (para el parámetro formal y para el parámetro actual) deben tener la misma configuración; es decir, los tipos de datos que sean iguales deben estar ordenados en el mismo orden.

A.3.3.5 Uso de tipos de datos de usuario para acceder a los datos

Tipos de datos de usuario

Los tipos de datos de usuario (UDT) pueden combinar tipos de datos simples y compuestos. A los UDTs se les puede dar un nombre y utilizar varias veces. La figura siguiente muestra la estructura de un UDT compuesta de un entero, un byte, un carácter, un número en coma flotante y un valor booleano o lógico.



En vez de indicar todos los tipos de datos individualmente o como estructura, sólo es necesario indicar "UDT20" como tipo de datos y STEP 7 ordena automáticamente el espacio de memoria correspondiente.

Crear un tipo de datos de usuario

Los UDTs se definen con STEP 7. La figura siguiente muestra un UDT que consta de los siguientes elementos: un entero (para memorizar la cantidad), un byte (para memorizar los datos originales), un carácter (para memorizar el código de control), un número en coma flotante (para memorizar la temperatura) y una marca booleana (para finalizar la señal). Al UDT se le puede asignar un nombre simbólico en la tabla de símbolos (p. ej., *Datos proceso*).

Dirección	Nombre	Tipo	Val. inicial	Comentario
0.0	Pila_1	STRUCT		
+0.0	Cantidad	INT	100	
+2.0	Datos_originales	BYTE		
+4.0	Código_de_contr	CHAR		
+6.0	Temperatura	REAL	120	
+8.0	Fin	BOOL	FALSE	
=10.0		END_STRUCT		

Tras haber creado un UDT puede utilizarlo al igual que un tipo de datos, p.ej. al declarar para una variable el tipo de datos *UDT200* en un DB (o en la declaración de variables de un FB).

La figura siguiente muestra un DB con la variable *Datos_proceso_1* con el tipo de datos UDT200. Debe indicar tan sólo *UDT200* y *Datos_proceso_1*. Los campos representados en cursiva se crean al compilar el DB.

Dirección	Nombre	Tipo	Val. inicial	Comentario
0.0		STRUCT		
+6.0	<i>Datos_proceso_1</i>	UDT200		
=6.0		END_STRUCT		

Asignar valores iniciales a un tipo de datos de usuario (UDT)

Si se desea asignar un valor inicial a cada elemento de un UDT, se ha de indicar un valor válido tanto para el tipo de datos como para el nombre del elemento. Por ejemplo se puede asignar (al UDT declarado en la figura anterior) los siguientes valores iniciales:

```

Cantidad          =      100
Datos originales  =      B#16#0
Código de control =      'Z'
Temperatura       =      1.200000e+002
Fin               =      False
    
```

Si se declara una variable como UDT, entonces los valores iniciales de la variable son aquello que fueron indicados al crear el UDT.

Guardar datos y acceder a los mismos en un tipo de datos de usuario (UDT)

Para acceder a cada uno de los elementos de un UDT puede utilizar direcciones simbólicas (p. ej. *Lote_1.Temperatura*) o puede indicar la dirección absoluta en la cual se memorizará el elemento (ejemplo: si *Lote_1* está memorizado en DB20 con inicio en el byte 0, entonces la dirección absoluta para la *cantidad* es *DB20.DBW0* y la dirección para la *temperatura* es *DB20.DBD6*).

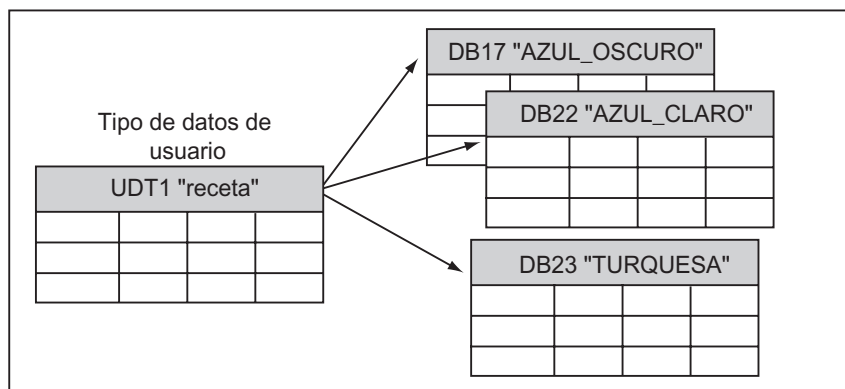
Utilizar tipos de datos de usuario en calidad de parámetros

Las variables del tipo de datos UDT se pueden transferir en calidad de parámetros. Si el parámetro está declarado como UDT en la declaración de variables, entonces se ha de transferir un UDT que tenga la misma estructura que los elementos de datos. Sin embargo, un elemento de un UDT se puede asignar a un parámetro al llamar un bloque, siempre que el elemento del UDT coincida con el tipo de datos del parámetro.

Ventaja de los DBs con UDT asociado

Con los UDTs ya creados se puede generar una gran cantidad de bloques de datos que tengan la misma estructura. Estos bloques de datos se pueden adaptar exactamente a la tarea correspondiente; para ello basta indicar valores actuales diferentes.

Si, p. ej., se estructura un UDT como receta (p. ej., para mezclar colores), entonces se puede asignar este UDT a varios DBs que contengan diferentes cantidades.



La estructura del bloque de datos queda definida por el UDT asociado.

A.3.4 Tipos de parámetros

Además de los tipos de datos simples y compuestos, es posible definir tipos de parámetros para los parámetros formales que se transfieren entre los bloques. STEP 7 conoce los siguientes tipos de parámetros:

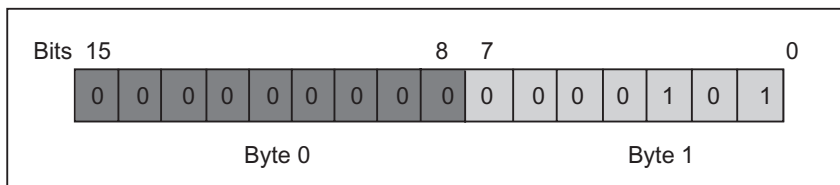
- **TIMER** o **COUNTER**: definen un determinado temporizador o contador que va a ser utilizado en la ejecución. Si se utiliza un parámetro formal del tipo **TIMER** o **COUNTER**, el parámetro actual correspondiente debe ser un temporizador o un contador, es decir, se debe indicar bien una "T" o una "Z" seguida por un número entero positivo.
- **BLOCK**: define un determinado bloque que ha de ser utilizado como entrada o como salida. La declaración del parámetro determina el tipo de bloque (FB, FC, DB etc.) que debe ser utilizado. Si se utiliza un parámetro formal del tipo **BLOCK**, se debe indicar la dirección del bloque como parámetro actual. Ejemplo: "FC101" (para direccionamiento absoluto) o "Válvula" (para direccionamiento simbólico).
- **POINTER**: accede a la dirección de una variable. Un puntero contiene una dirección en lugar de un valor. Si se utiliza un parámetro formal del tipo **POINTER**, se debe indicar la dirección como parámetro actual. STEP 7 permite indicar un puntero en formato Pointer o simplemente como dirección (p. ej., M 50.0). Ejemplo de un formato Pointer para direccionamiento de los datos que comienzan en M 50.0: P#M50.0
- **ANY**: se utiliza cuando el tipo de datos del parámetro actual no se conoce o cuando se puede utilizar cualquier tipo de datos. Para obtener informaciones más detalladas acerca del parámetro **ANY**, consulte los apartados "Formato del tipo de parámetro ANY" o "Uso del tipo de parámetro ANY".

Un tipo de parámetro puede ser también un tipo de datos de usuario (UDT). Para obtener informaciones más detalladas sobre los UDTs, consulte el apartado "Uso de tipos de datos de usuario para acceder a los datos".

Parámetro	Tamaño	Descripción
TIMER	2 bytes	Designa un temporizador determinado que ha de ser utilizado por el programa en el bloque lógico llamado. Formato: T1
COUNTER	2 bytes	Designa un contador determinado que ha de ser utilizado por el programa en el bloque lógico llamado. Formato: Z10
BLOCK_FB BLOCK_FC BLOCK_DB BLOCK_SDB	2 bytes	Designa un bloque determinado que ha de ser utilizado por el programa en el bloque lógico llamado. Formato: FC101 DB42
POINTER	6 bytes	Designa la dirección. Formato: P#M50.0
ANY	10 bytes	Se utiliza cuando el tipo de datos del parámetro actual no se conoce. Formato: P#M50.0 BYTE 10 Formato ANY en tipos de datos P#M100.0 WORD 5 L#1COUNTER 10 Formato ANY en tipos de parámetros

A.3.4.1 Formato de los tipos de parámetros BLOCK, COUNTER, TIMER

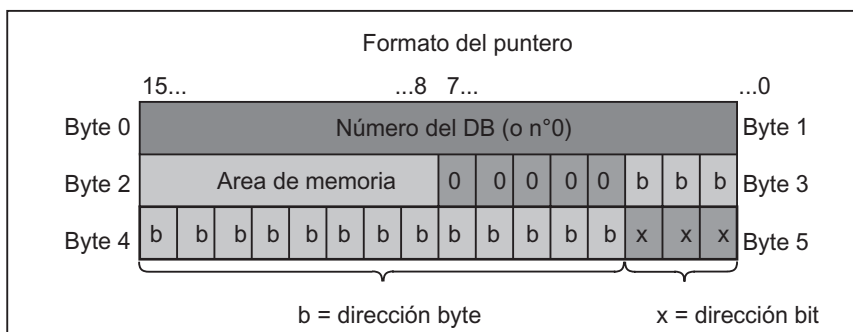
STEP 7 guarda los tipos de parámetros BLOCK, COUNTER y TIMER en forma de números binarios en una palabra (32 bits). La figura siguiente muestra el formato de dichos tipos de parámetros.



El número admisible de bloques, temporizadores y contadores depende del modelo de CPU S7. Para obtener informaciones más detalladas acerca del número admisible de temporizadores y contadores y el número máximo de bloques disponibles, consulte las hojas de datos de su CPU en los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" o "Sistemas de automatización S7-400, M7-400, Configuración e instalación".

A.3.4.2 Formato del tipo de parámetro POINTER

STEP 7 guarda el tipo de parámetro POINTER en 6 bytes (48 bits). La figura siguiente muestra el tipo de datos que se guardan en cada byte.



El tipo de parámetro POINTER almacena las informaciones siguientes:

- Número de DB (0 en caso de que los datos no se guarden en un DB)
- Área de la memoria en la CPU (la tabla siguiente muestra los códigos hexadecimales de las áreas de memoria para el tipo de parámetro POINTER)

Código hexadecimal	Área de memoria	Descripción
b#16#81	E	Área de entradas
b#16#82	A	Área de salidas
b#16#83	M	Área de marcas
b#16#84	DB	Bloque de datos
b#16#85	DI	Bloque de datos de instancia
b#16#86	L	Pila de datos locales (LSTACK)
b#16#87	V	Datos locales precedentes

- Dirección de los datos (en formato byte.bit)
- STEP 7 ofrece el formato de puntero: p#área de memoria byte.bit_dirección. (Si el parámetro formal ha sido declarado como tipo POINTER, sólo se deberá indicar el área de la memoria y la dirección. STEP 7 convertirá automáticamente la entrada al formato de puntero.) Los ejemplos siguientes muestran cómo introducir el tipo de parámetro POINTER para los datos que comiencen en M50.0:
- P#M50.0
 - M50.0 (si el tipo de parámetro se ha declarado como POINTER)

A.3.4.3 Uso del tipo de parámetro POINTER

Los punteros se utilizan para señalar operandos. La ventaja de este tipo de direccionamiento es que el operando de la instrucción se puede modificar dinámicamente durante la ejecución del programa.

Puntero para el direccionamiento indirecto por memoria

Las instrucciones del programa que utilizan el direccionamiento indirecto por memoria se componen de una operación, un identificador del operando y un desplazamiento (el desplazamiento se debe indicar entre corchetes).

Ejemplo de un puntero en formato de palabra doble:

L	P#8.7	Cargar el valor del puntero en ACU1.
T	MD2	Transferir el puntero a MD2.
U	E [MD2]	Consultar el estado de señal en la entrada E 8.7,
=	A [MD2]	y asignar el estado de señal a la salida A 8.7.

Puntero para los direccionamientos intraárea e interárea

Las instrucciones de programación que utilizan estos tipos de direccionamiento se componen de una operación y las siguientes partes: identificador del operando, identificador del registro de direcciones, desplazamiento.

El registro de direcciones (AR1/2) y el desplazamiento se deben indicar juntos entre corchetes.

Ejemplo del direccionamiento intraárea

El puntero no contiene una indicación del área de memoria:

L	P#8.7	Cargar el valor del puntero en ACU1.
LAR1		Cargar el puntero de ACU1 en AR1.
U	E [AR1, P#0.0]	Consultar el estado de señal en la entrada E 8.7 y
=	A [AR1, P#1.1]	asignar el estado de señal a la salida A 10.0.

El desplazamiento 0.0 no tiene efecto alguno. La salida 10.0 resulta haber sumado 8.7 (AR1) más el desplazamiento 1.1. El resultado es 10.0 y no 9.8 (véase el formato de puntero).

Ejemplo del direccionamiento interárea

En el caso del direccionamiento interárea, el área de memoria (en el ejemplo: E ó A) se indica en el puntero.

L	P# E8.7	Cargar el valor del puntero y el identificador de área en ACU1.
LAR1		Cargar el área de memoria E y la dirección 8.7 en AR1.
L	P# A8.7	Cargar el valor del puntero y el identificador de área en ACU1.
LAR2		Cargar el área de memoria A y la dirección 8.7 en AR2.
U	[AR1, P#0.0]	Consultar el estado de señal en la entrada E 8.7 y
=	[AR2, P#1.1]	asignar el estado de señal a la salida A 10.0.

El desplazamiento 0.0 no tiene efecto alguno. La salida 10.0 resulta de sumar 8.7 (AR2) y 1.1 (desplazamiento). El resultado es 10.0 y no 9.8 (véase el formato de puntero).

A.3.4.4 Bloque para modificar el puntero

Con el bloque de ejemplo FC3 "Cambio de punteros" se puede modificar la dirección de bit o de byte de un puntero. Al llamar la FC, el puntero a cambiar se transfiere a la variable "Puntero" (se pueden utilizar punteros de direccionamiento intraárea o interárea en formato de palabra doble).

Con el parámetro "Bit-Byte" se puede modificar la dirección de bit o de byte del puntero (0: dirección de bit, 1: dirección de byte). En la variable "Inc_valor" (en formato de entero) se indica el valor que se ha de sumar o restar al contenido de la dirección. También se pueden indicar números negativos para decrementar la dirección.

Al modificar la dirección de bit se efectúa un acarreo a la dirección de byte (también al decrementar); p.ej.:

- P#M 5.3, Bit_Byte = 0, Inc_valor= 6 => P#M 6.1 ó
- P#M 5.3, Bit_Byte = 0, Inc_valor= -6 => P#M 4.5.

La función no influye en la información de área del puntero.

La FC detecta si se exceden los límites superior o inferior del puntero. Entonces no se modifica el puntero y la variable de salida "RET_VAL" (los errores se pueden corregir) se pone a "1" (hasta la siguiente ejecución correcta de la FC 3). Este es el caso cuando:

1. se ha elegido la dirección de bit, siendo Inc_valor >7 ó <-7,
2. se ha elegido la dirección de bit o de byte y el cambio tendría por resultado una dirección de byte "negativa",
3. se ha elegido la dirección de bit o de byte y el cambio tendría por resultado una dirección de byte demasiado grande.

Bloque de ejemplo en AWL para modificar el puntero:

```

FUNCTION FC 3: BOOL
TITLE =Cambio de punteros
//La FC 3 se puede utilizar para cambiar punteros.
AUTHOR : AUT1CS1
FAMILY : INDADR
NAME : ADRPOINT
VERSION : 0.0

VAR_INPUT
    Bit_Byte : BOOL ; //0: dirección de bit, 1: dirección de byte
    Inc_valor : INT ; //Incrementar (si valor neg.=> decrementar/si valor pos.
                    //=> incrementar)

END_VAR

VAR_IN_OUT
    Puntero : DWORD ; //el puntero a modificar

END_VAR

VAR_TEMP
    Inc_valor1 : INT ; //Valor intermedio incremento
    Puntero1 : DWORD ; //Valor intermedio puntero
    Val_int : DWORD ; //Variable auxiliar

END_VAR

BEGIN
NETWORK
TITLE =
//El bloque rechaza automáticamente los cambios que modifiquen las
//informaciones de área del puntero o que conduzcan a punteros "negativos".
    SET    ; //Poner RLO a 1 y
    R      #RET_VAL; //Desactivar rebose del límite superior
    L      #Puntero; //Asignar el valor intermedio temporal
    T      #Puntero1; //del puntero
    L      #Inc_valor; //Asignar el valor intermedio temporal
    T      #Inc_Wert1; //del incremento
    U      #Bit_Byte; //si =1, entonces operación de dirección de byte
    SPB   Byte; //Salto al cálculo de dirección de byte
    L      7; //Si el valor del incremento > 7,
    L      #Inc_valor1;
    <I    ;
    S      #RET_VAL; //entonces activar RET_VAL y
    SPB   Final; //saltar al final

```

```

L      -7; //Si el valor del incremento < -7,
<I    ;
S      #RET_VAL; //entonces activar RET_VAL y
SPB   Final; //saltar al final
U      L      1.3; //si el bit 4 del valor = 1 (Inc_valor neg)
SPB   neg; //entonces saltar a sustracción de dirección de bit
L      #Puntero1; //Cargar información de dirección del puntero
L      #Inc_Wert1; //y sumar el incremento
+D    ;
SPA   test; //Saltar al test de resultado negativo
neg:  L      #Puntero1; //Cargar información de dirección del puntero
      L      #Inc_valor1; //Cargar el incremento
      NEGI   ; //negar el valor negativo,
      -D    ; //restar el valor
      SPA   test; //y saltar al test
Byte: L      0; //Comienzo del cambio de dirección de byte
      L      #Inc_valor1; //Si el incremento >=0, entonces
      <I    ;
      SPB   pos; //saltar a la suma; en caso contrario
      L      #Puntero1; //Cargar información de dirección del puntero
      L      #Inc_valor1; //Cargar el incremento
      NEGI   ; //negar el valor negativo,
      SLD   3; //desplazar el incremento 3 posiciones hacia la izquierda,
      -D    ; //restar el valor
      SPA   test; //y saltar al test
pos:  SLD   3; //desplazar el incremento 3 posiciones hacia la izquierda,
      L      #Puntero1; //Cargar información de dirección del puntero
      +D    ; //sumar el incremento
test: T      #Val_int; //Transferir resultado cálculos a Val_int,
      U      L      7.3; //Si la dirección de byte no es válida (demasiado grande
      S      #RET_VAL; //o negativa), activar RET_VAL
      SPB   Final; //y saltar al final
      L      #Val_int; //en caso contrario, transferir el resultado
      T      #Puntero; //al puntero
Fin:  NOP   0;
END_FUNCTION

```

A.3.4.5 Formato del tipo de parámetro ANY

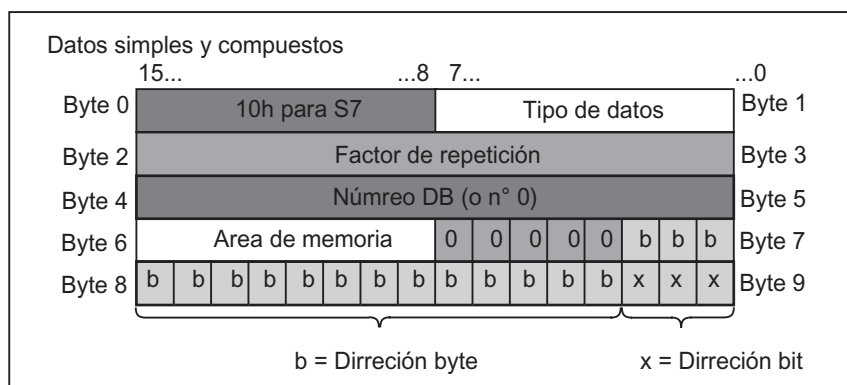
STEP 7 almacena los datos del tipo de parámetro ANY en 10 bytes. Al estructurar un parámetro del tipo ANY, los 10 bytes deben quedar ocupados, ya que el bloque llamado evalúa todo el contenido del parámetro. Si, por ejemplo, especifica un número de DB en el byte 4, deberá indicar también explícitamente el área de la memoria en el byte 6.

STEP 7 gestiona los datos de los tipos de datos simples y compuestos de forma diferente a la de los tipos de parámetros.

Formato ANY en los tipos de datos

En el caso de los tipos de datos simples y compuestos, STEP 7 almacena los siguientes datos:

- Tipos de datos
- Factor de repetición
- Número de DB
- Area de la memoria donde se almacenan las informaciones
- Dirección inicial de los datos



El factor de repetición indica una cantidad del tipo de datos identificado que el tipo de parámetro ANY debe transferir. Así se puede indicar un área de datos, así como utilizar arrays y estructuras en combinación con el tipo de parámetro ANY. STEP 7 identifica, con ayuda del factor de repetición, los arrays y las estructuras como cantidad de tipos de datos. Si, por ejemplo, se deben transferir 10 palabras, en el factor de repetición se debe indicar el valor 10, y en el tipo de datos el valor 04.

La dirección se almacena en formato byte.bit. La dirección de byte se deposita en los bits 0 a 2 del byte 7, en los bits 0 a 7 del byte 8 y en los bits 3 a 7 del byte 9, mientras que la dirección de bit se deposita en los bits 0 a 2 del byte 9.

En el puntero nulo del tipo de datos NIL, todos los bytes estarán ocupados con 0 a partir del byte 1.

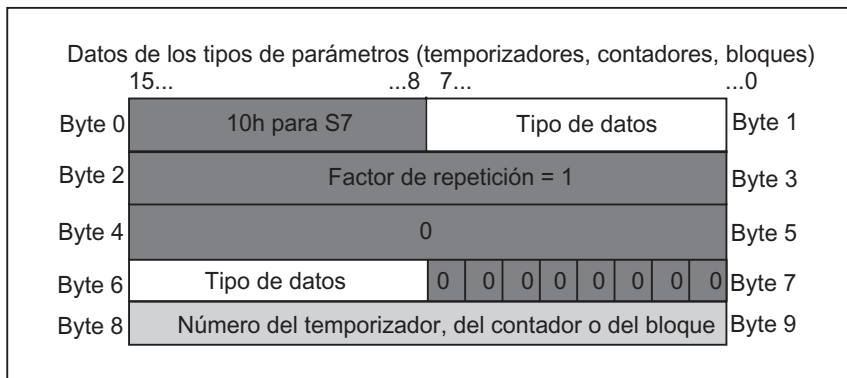
Las tablas siguientes muestran la codificación de los tipos de datos o las áreas de memoria para el tipo de parámetro ANY.

Codificación de los tipos de datos		
Código hexadecimal	Tipo de datos	Descripción
b#16#00	NIL	Puntero nulo
b#16#01	BOOL	Bits
b#16#02	BYTE	Bytes (8 bits)
b#16#03	Número entero	Carácter (8 bits)
b#16#04	WORD	Palabras (16 bits)
b#16#05	INT	Números enteros (16 bits)
B#16#06	DWORD	Palabras (32 bits)
b#16#07	DINT	Números enteros (32 bits)
b#16#08	REAL	Números reales (32 bits)
b#16#09	DATE	Fecha
b#16#0A	TIME_OF_DAY (TOD)	Hora
b#16#0B	TIME	Temporizador
b#16#0C	S5TIME	Tipo de datos S5TIME
b#16#0E	DATE_AND_TIME (DT)	Fecha y hora (64 bits)
b#16#13	STRING	Cadena de caracteres

Codificación de las áreas de memoria		
Código hexadecimal	Área	Descripción
b#16#81	E	Área de entradas
b#16#82	A	Área de salidas
b#16#83	M	Área de marcas
b#16#84	DB	Bloque de datos
b#16#85	DI	Bloque de datos de instancia
b#16#86	L	Pila de datos locales (L-Stack)
b#16#87	V	Datos locales precedentes

Formato ANY en los tipos de parámetros

En el caso de los tipos de parámetros, STEP 7 almacena el tipo de datos y la dirección de los parámetros. El factor de repetición es siempre 1. Los bytes 4, 5 y 7 son siempre 0. Los bytes 8 y 9 indican el número del temporizador, del contador o del bloque.



La tabla siguiente muestra la codificación de los tipos de datos para el formato ANY en el caso de los tipos de parámetros.

Código hexadecimal	Tipo de datos	Descripción
b#16#17	BLOCK_FB	Número de FB
b#16#18	BLOCK_FC	Número de FC
b#16#19	BLOCK_DB	Número de DB
b#16#1A	BLOCK_SDB	Número de SDB
b#16#1C	COUNTER	Número de contador
b#16#1D	TIMER	Número de temporizador

A.3.4.6 Uso del tipo de parámetro ANY

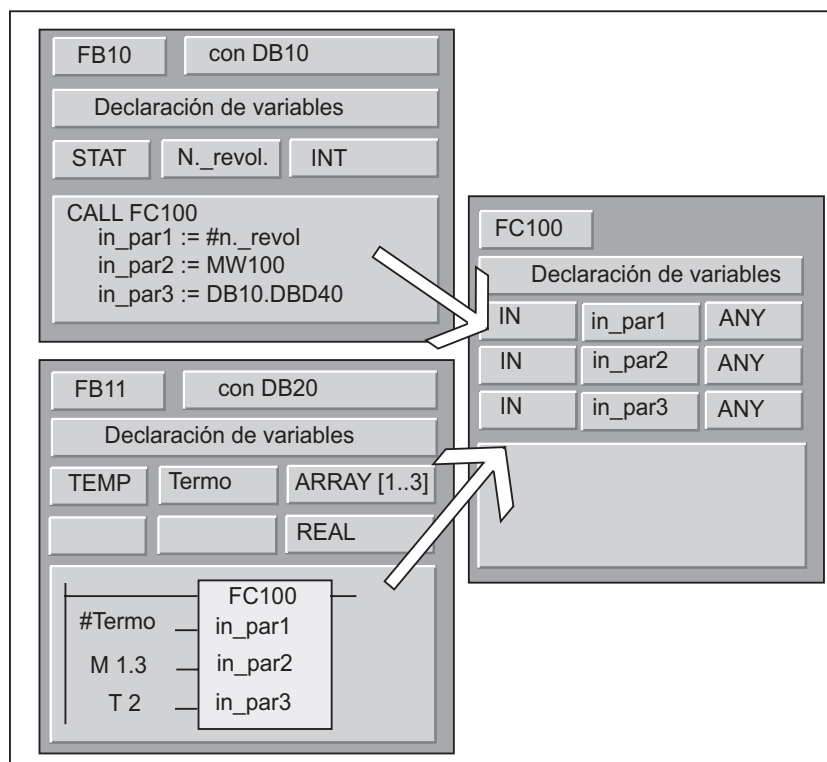
Para un bloque se pueden definir parámetros formales que sean apropiados para parámetros actuales con cualquier tipo de datos. Esto resulta sobre todo de gran utilidad cuando el tipo de datos del parámetro actual que se suministra al llamar el bloque es desconocido o puede variar (y cuando es admisible cualquier tipo de datos). En la declaración de variables del bloque se ha de declarar el parámetro con el tipo de datos ANY. En STEP 7 se puede asignar entonces un parámetro actual de un tipo de datos cualquiera.

STEP 7 asigna 80 bits de memoria a una variable con el tipo de datos ANY. Al asignar un parámetro actual a este parámetro formal, STEP 7 codifica la dirección de arranque, el tipo de datos y la longitud del parámetro actual en los 80 bits. El bloque llamado analiza los 80 bits en cuanto a datos que fueron memorizados para el parámetro ANY, obteniendo así informaciones que pueden utilizarse para el tratamiento ulterior.

Asignar un parámetro actual a un parámetro ANY

Si se declara un parámetro con el tipo de datos ANY, se podrá asignar a este parámetro formal un parámetro actual con un tipo de datos cualquiera. En STEP 7 se pueden asignar los siguientes tipos de datos como parámetros actuales:

- Tipos de datos simples: Se indica la dirección absoluta o el nombre simbólico del parámetro actual.
- Tipos de datos compuestos: Se indica el nombre simbólico de los datos con tipo de datos compuestos (p. ej., arrays y estructuras).
- Temporizadores, contadores y bloques: Introduzca el número (p. ej., T1, Z20 ó FB6).
- La figura siguiente muestra cómo se pueden transferir datos a una FC con parámetros del tipo de datos ANY.



En este ejemplo, FC100 tiene tres parámetros (*in_par1*, *in_par2* y *in_par3*) que fueron declarados con el tipo de datos ANY.

- Cuando el FB10 llama la FC100, el FB10 entrega un número entero (la variable estática 'número de revoluciones'), una palabra (MW100) y una palabra doble en DB10 (DB10.DBD40).
- Cuando el FB11 llama la FC100, el FB11 entrega un campo de números reales (la variable temporal "termo"), un valor booleano (M 1.3) y un temporizador (T2).

Indicar un área de datos para un parámetro ANY

A un parámetro ANY no sólo se le pueden asignar operandos individuales (p. ej. MW100), sino que también es posible indicar un área de datos. Si se desea asignar un área de datos como parámetro actual, entonces se debe utilizar el siguiente formato de una constante para indicar la cantidad de datos a transferir:

p# identif. de área Byte.Bit tipo de datos factor de repetición

Para el elemento *tipo de datos* en el formato para constantes se pueden indicar todos los tipos de datos simples y el tipo de datos DATE_AND_TIME. Si el tipo de datos no es BOOL, habrá que indicar la dirección de bit 0 (x.0). La tabla siguiente muestra ejemplos de formato para indicar las áreas de memoria que se desean transferir a un parámetro ANY.

Parámetro actual	Descripción
p# M 50.0 BYTE 10	Indica 10 bytes en el área de memoria Marcas: MB50 a MB59.
p# DB10.DBX5.0 S5TIME 3	Indica 3 unidades de datos del tipo S5TIME, memorizadas en el DB10: DB byte 5 a DB byte 10.
p# A 10.0 BOOL 4	Indica 4 bits en el área de memoria Salidas: A 10.0 a A 10.3.

Ejemplo de utilización del tipo de parámetro ANY

El siguiente ejemplo muestra cómo copiar un área de memoria de 10 bytes utilizando el tipo de parámetro ANY y la función de sistema SFC 20 BLKMOV.

AWL	Explicación
FUNCTION FC 10:VOID	
VAR_TEMP	
Fuente : ANY;	
Objetivo:	
ANY;	
END_VAR	
BEGIN	Cargar la dirección inicial del puntero ANY en AR1.
LAR1 P#Quelle;	
	Cargar el ID de sintaxis y transferirlo al puntero ANY.
L B#16#10;	
T LB[AR1,P#0.0];	
	Cargar el tipo de datos byte y transferirlo al puntero ANY.
L B#16#02;	
T LB[AR1,P#1.0];	
	Cargar 10 bytes y transferirlos al puntero ANY.
L 10;	
T LW[AR1,P#2.0];	
	La fuente es el DB22, DBB11
L 22;	
T LW[AR1,P#4.0];	
L P#DBX11.0;	
T LD[AR1,P#6.0];	
	Cargar la dirección inicial del puntero ANY en AR1.
LAR1 P#Ziel;	
	Cargar el ID de sintaxis y transferirlo al puntero ANY.
L B#16#10;	
T LB[AR1,P#0.0];	
	Cargar el tipo de datos byte y transferirlo al puntero ANY.
L B#16#02;	
T LB[AR1,P#1.0];	
	Cargar 10 bytes y transferirlos al puntero ANY.
L 10;	
T LW[AR1,P#2.0];	
	La meta es el DB33, DBB202
L 33;	
T LW[AR1,P#4.0];	
L P#DBX202.0;	
T LD[AR1,P#6.0];	
	Llamada de la función de sistema Blockmove
CALL SFC 20 (
SRCBLK := Quelle,	Evaluación del bit RB y de la MW 12
RET_VAL := MW 12,	
DSTBLK := Ziel	
);	
END_FUNCTION	

A.3.4.7 Asignar tipos de datos a los datos locales de bloques lógicos

STEP 7 limita los tipos de datos (tipos simples, compuestos y tipos de parámetros) que se pueden asignar en la declaración de variables a los datos locales de un bloque.

Tipos de datos válidos para los datos locales de un OB

La tabla siguiente muestra las restricciones existentes (-) al declarar los datos locales para un OB. Como un OB no puede ser llamado, entonces tampoco puede disponer de parámetros (entrada, salida o entrada/salida). Como que el OB no tiene DB de instancia, no se pueden declarar variables estáticas para un OB. Como tipos de datos de las variables temporales de un OB se pueden utilizar datos simples o compuestos, así como ANY.

Las asignaciones válidas se indican mediante el símbolo ●.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada	—	—	—	—	—	—	—
Salida	—	—	—	—	—	—	—
Entrada/salida	—	—	—	—	—	—	—
Estáticas	—	—	—	—	—	—	—
Temporales	● ⁽¹⁾	● ⁽¹⁾	—	—	—	—	● ⁽¹⁾

⁽¹⁾ Almacenado en la pila LStack del OB.

Tipos de datos válidos para los datos locales de un FB

La tabla siguiente muestra las restricciones existentes (-) al declarar los datos locales para un FB. Como que existe el DB de instancia, habrá menos restricciones al declarar los datos locales para un FB. Para la declaración de parámetros de entrada no existen restricciones; para un parámetro de salida no se pueden indicar ningún tipo de parámetro y para parámetros de entrada/salida sólo se admiten los tipos POINTER y ANY. Las variables temporales se pueden declarar con el tipo de datos ANY. Todos los demás tipos de parámetros son ilegales.

Las asignaciones válidas se indican mediante el símbolo ●.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada	●	●	●	●	●	●	●
Salida	●	●	—	—	—	—	—
Entrada/salida	●	● ⁽¹⁾⁽³⁾	—	—	—	●	●
Estáticas	●	●	—	—	—	—	—
Temporales	● ⁽²⁾	● ⁽²⁾	—	—	—	—	● ⁽²⁾

⁽¹⁾ Como referencia (puntero de 48 bits) en el bloque de datos de instancia.
⁽²⁾ Almacenado en la pila LSTACK del FB.
⁽³⁾ Sólo pueden definirse STRINGS con la longitud estándar.

Tipos de datos válidos para los datos locales de una FC

La tabla siguiente muestra las restricciones existentes (-) al declarar los datos locales para una FC. Como la FC no tiene DB de instancia, entonces tampoco dispone de variables estáticas. Para parámetros de entrada, salida y de entrada/salida de una FC, se admiten los tipos de parámetros POINTER y ANY. También es posible declarar variables temporales con el tipo de parámetro ANY.

Las asignaciones válidas se indican mediante el símbolo ●.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos (2)	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada	●	●	●	●	●	●	●
Salida	●	●	—	—	—	●	●
Entrada/salida	●	●	—	—	—	●	●
Temporales	●(1)	●(1)	—	—	—	—	●(1)
¹ Almacenado en la pila LSTACK de la FC. ⁽²⁾ Sólo pueden definirse STRINGS con la longitud estándar.							

A.3.4.8 Tipos de datos admisibles al transferir parámetros

Reglas para la transferencia de parámetros entre bloques

Si asigna parámetros actuales a parámetros formales, puede indicar una dirección absoluta, un nombre simbólico o una constante STEP 7 limita las asignaciones válidas correspondientes para los diferentes parámetros. Por ejemplo, a los parámetros de salida y de entrada/salida no se les puede asignar un valor constante (ya que la finalidad de una salida o de una entrada/salida consiste en modificar el valor). Estas restricciones rigen sobre todo para parámetros con tipos de datos compuestos, a los cuales no se les puede asignar una dirección absoluta ni tampoco una constante.

Las tablas siguientes contienen las restricciones (-) de los tipos de datos de parámetros actuales que se asignan a los parámetros formales.

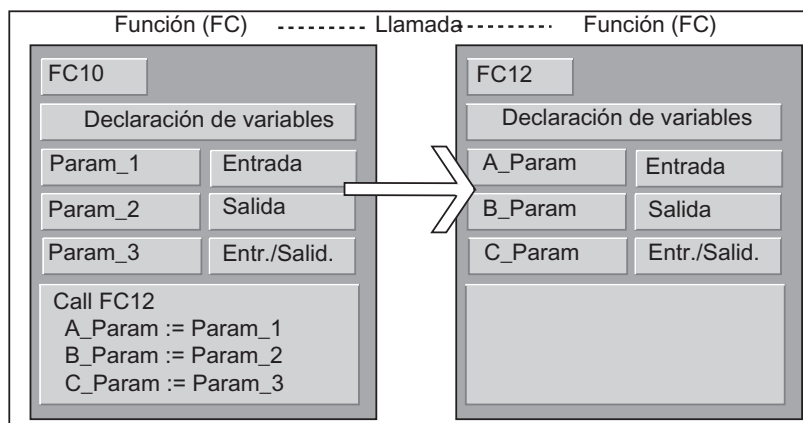
Las asignaciones válidas se indican mediante el símbolo ●.

Tipos de datos simples				
Tipo de declaración	Dirección absoluta	Nombre simbólico (en la tabla de símbolos)	Símbolo local del bloque	Constante
Entrada	●	●	●	●
Salida	●	●	●	—
Entrada/salida	●	●	●	—

Tipos de datos compuestos				
Tipo de declaración	Dirección absoluta	Nombre simbólico del elemento del DB (en la tabla de símbolos)	Símbolo local del bloque	Constante
Entrada	—	●	●	—
Salida	—	●	●	—
Entrada/salida	—	●	●	—

Tipos de datos permitidos al llamar una FC desde otra FC

A los parámetros formales de una FC llamada se pueden asignar los parámetros formales de una FC invocante. La figura siguiente muestra los parámetros formales de FC10 que se asignan en calidad de parámetros actuales a los parámetros formales de FC12.



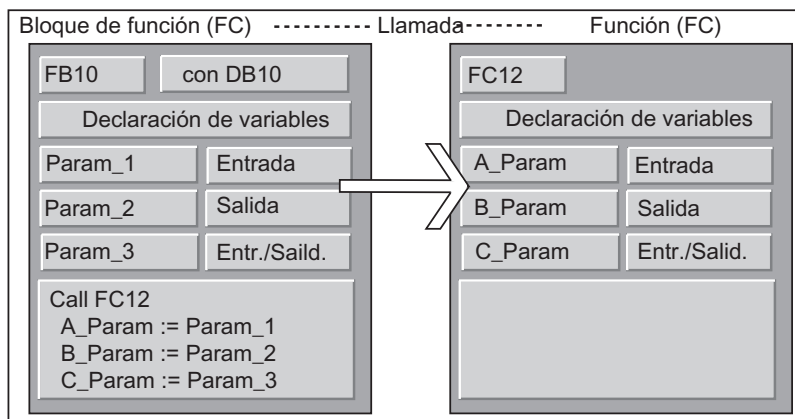
STEP 7 limita la asignación de parámetros formales - en calidad de parámetros actuales - de una FC a los parámetros formales de otra FC. Por ejemplo, no se pueden asignar como parámetros actuales aquellos parámetros con tipo de datos compuestos o tipo de parámetros.

La tabla siguiente muestra los tipos de datos admisibles (●) cuando una FC llama a otra.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada → Entrada	●	—	—	—	—	—	—
Entrada → Salida	—	—	—	—	—	—	—
Entrada → Entrada/salida	—	—	—	—	—	—	—
Salida → Entrada	—	—	—	—	—	—	—
Salida → Salida	●	—	—	—	—	—	—
Salida → Entrada/salida	—	—	—	—	—	—	—
Entrada/salida → Entrada	●	—	—	—	—	—	—
Entrada/salida → Salida	●	—	—	—	—	—	—
Entrada/salida → Entrada/salida	●	—	—	—	—	—	—

Tipos de datos permitidos al llamar una FC desde un FB

A los parámetros formales de una FC invocada se pueden asignar los parámetros formales de un FB invocante. La figura siguiente muestra los parámetros formales de FB10 que se asignan en calidad de parámetros actuales a los parámetros formales de FC12.

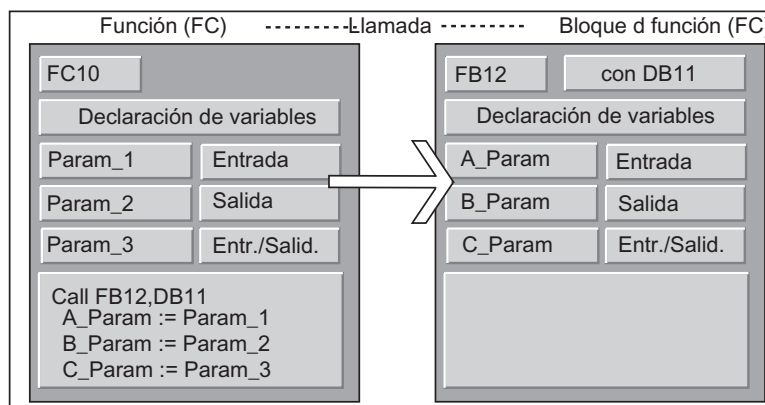


STEP 7 limita la asignación de parámetros formales de un FB a los parámetros formales de una FC. Por ejemplo, los parámetros con tipo de parámetro no se pueden asignar como parámetros actuales. La tabla siguiente muestra los tipos de datos admisibles (●) cuando un FB llama a una FC.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada → Entrada	●	●	—	—	—	—	—
Entrada → Salida	—	—	—	—	—	—	—
Entrada → Entrada/salida	—	—	—	—	—	—	—
Salida → Entrada	—	—	—	—	—	—	—
Salida → Salida	●	●	—	—	—	—	—
Salida → Entrada/salida	—	—	—	—	—	—	—
Entrada/salida → Entrada	●	—	—	—	—	—	—
Entrada/salida → Salida	●	—	—	—	—	—	—
Entrada/salida → Entrada/salida	●	—	—	—	—	—	—

Tipos de datos permitidos al llamar un FB desde una FC

A los parámetros formales de un FB llamado se pueden asignar los parámetros formales de una FC invocante. La figura siguiente muestra los parámetros formales de FC10 que se asignan en calidad de parámetros actuales a los parámetros formales de FB12.



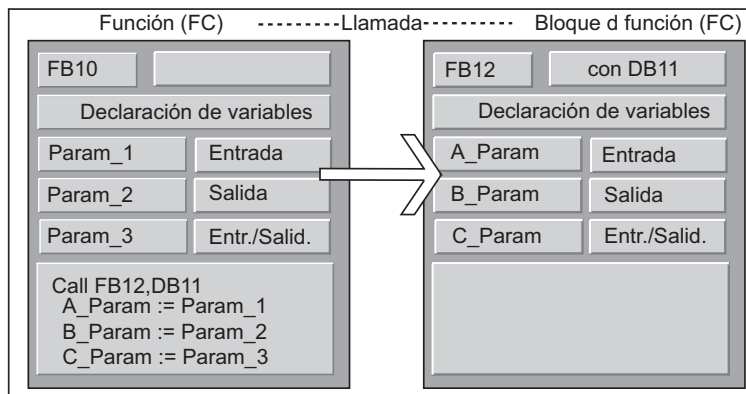
STEP 7 limita la asignación de parámetros formales de una FC a los parámetros formales de un FB. Por ejemplo, los parámetros con tipo de datos compuestos no se pueden asignar como parámetros actuales. Sin embargo, los parámetros de entrada con el tipo de parámetro TIMER, COUNTER o BLOCK se pueden asignar a los parámetros de entrada del FB invocado.

La tabla siguiente muestra los tipos de datos admisibles (●) cuando una FC llama a un FB.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro	Tipo de parámetro
			TIMER	COUNTER	BLOCK	POINTER	ANY
Entrada → Entrada	●	—	●	●	●	—	—
Entrada → Salida	—	—	—	—	—	—	—
Entrada → Entrada/salida	—	—	—	—	—	—	—
Salida → Entrada	—	—	—	—	—	—	—
Salida → Salida	●	—	—	—	—	—	—
Salida → Entrada/salida	—	—	—	—	—	—	—
Entrada/salida → Entrada	●	—	—	—	—	—	—
Entrada/salida → Salida	●	—	—	—	—	—	—
Entrada/salida → Entrada/salida	●	—	—	—	—	—	—

Tipos de datos permitidos al llamar un FB desde otro FB

A los parámetros formales de un FB llamado se pueden asignar los parámetros formales de un FB invocante. La figura siguiente muestra los parámetros formales de FB10 que se asignan en calidad de parámetros actuales a los parámetros formales de FB12.



STEP 7 limita la asignación de parámetros formales de un FB a los parámetros formales de otro FB. Por ejemplo, los parámetros de entrada y de salida con tipos de datos compuestos no se pueden asignar como parámetros actuales a los parámetros de entrada y de salida de un FB llamado. Sin embargo, los parámetros de entrada con el tipo de parámetro TIMER, COUNTER o BLOCK se pueden asignar a los parámetros de entrada del FB llamado.

La tabla siguiente muestra los tipos de datos admisibles (●) cuando un FB llama a otro.

Tipo de declaración	Tipos de datos simples	Tipos de datos compuestos	Tipo de parámetro TIMER	Tipo de parámetro COUNTER	Tipo de parámetro BLOCK	Tipo de parámetro POINTER	Tipo de parámetro ANY
Entrada → Entrada	●	●	●	●	●	—	—
Entrada → Salida	—	—	—	—	—	—	—
Entrada → Entrada/salida	—	—	—	—	—	—	—
Salida → Entrada	—	—	—	—	—	—	—
Salida → Salida	●	●	—	—	—	—	—
Salida → Entrada/salida	—	—	—	—	—	—	—
Entrada/salida → Entrada	●	—	—	—	—	—	—
Entrada/salida → Salida	●	—	—	—	—	—	—
Entrada/salida → Entrada/salida	●	—	—	—	—	—	—

A.3.4.9 Transferencia a parámetros IN_OUT de un FB

El transferir tipos de datos compuestos a parámetros IN_OUT de un bloque de función (FB) se transfiere solamente la dirección del operando de la variable (call by reference).

Al transferir tipos de datos simples a parámetros IN_OUT de un FB se copian los valores en el bloque de datos de instancia antes de arrancar el FB, y cuando se termina el FB se vuelven a copiar del bloque de datos de instancia.

Así se pueden inicializar con un valor las variables IN_OUT de un tipo de datos simple.

No obstante, en una llamada no se puede indicar una constante como parámetro actual en la posición de una variable IN_OUT, ya que no se puede escribir en una constante.

Las variables del tipo de datos STRUCT o ARRAY no se pueden inicializar, ya que en este caso hay una sola dirección en el bloque de datos de instancia.

A.4 Trabajar con proyectos de versiones anteriores

A.4.1 Convertir proyectos de la versión 1

Convertiendo los proyectos de la versión 1 en proyectos de la versión 2 podrá reutilizar proyectos que se hayan creado con la versión 1 de STEP 7.

Tras la conversión se conservan los siguientes componentes del proyecto de la versión 1:

- Estructura del proyecto y sus programas
- Bloques
- Archivos fuente AWL
- Tabla de símbolos

La configuración del hardware no se convierte. Los componentes de programa pueden copiarse a otros proyectos. Además, es posible añadir al nuevo proyecto un equipo que deberá ser configurado y parametrizado debidamente.

Tras realizar la conversión a la versión 2 podrá decidir en un cuadro de diálogo si el proyecto de la versión 2 se ha de convertir en un proyecto de la versión que Ud. dispone de STEP 7.

Nota

Las propiedades de los bloques siguen siendo las correspondientes a la versión 1. El código generado en la versión 1 no cambia, por lo que los bloques no pueden utilizarse junto con multiinstancias.

Si desea declarar multiinstancias en los bloques convertidos, utilizando la aplicación "KOP/AWL: programar bloques" genere fuentes AWL con los bloques convertidos y compílelas de nuevo en bloques.

Una nueva opción de la versión 2 de STEP 7 es la programación de multiinstancias para crear bloques de función (FBs). Si desea volver a utilizar los FBs creados con la versión 1 para las mismas funciones en proyectos de la versión 2, no necesita convertirlos.

Procedimiento

Para convertir proyectos de la versión 1:

1. Elija el comando **Archivo > Abrir proyecto versión 1**.
2. En el cuadro de diálogo que aparece a continuación, elija el proyecto de la versión 1 que desea reutilizar. Los proyectos de la versión 1 tienen la extensión *.s7a.
3. En el cuadro de diálogo siguiente, introduzca el nombre del nuevo proyecto en el que se ha de convertir el proyecto de la versión 1.

A.4.2 Convertir proyectos de la versión 2

Con el comando **Archivo > Abrir** es posible abrir proyectos creados con la versión 2 de STEP 7.

Los proyectos y las librerías de la versión 2 se pueden convertir a la versión actual de STEP 7 (migración) utilizando el comando de menú **Archivo > Guardar como** y el ajuste "reorganizando". Así, el proyecto se guarda como proyecto de la actual versión STEP 7.

Los proyectos y las librerías que se hayan creado con versiones antiguas de STEP 7 se pueden editar y guardar conservando el formato. Para ello se debe seleccionar en el cuadro de diálogo "Guardar proyecto como" el tipo de datos de la correspondiente versión antigua de STEP 7. Para elaborar objetos p. ej. con la versión 2.1 de STEP 7, elija aquí Proyecto 2.x o Librería 2.x (guardar como versión 2 ya no es posible a partir de la versión 5.1, consulte también Editar proyectos y librerías de la versión 2).

Tipos de archivo

	STEP 7 V3	a partir de STEP 7 V4
Tipo de archivo de la versión actual	Proyecto3.x Librería3.x	Proyecto Librería
Tipo de archivo de la versión antigua	Proyecto2.x Librería2.x	Proyecto2.x Librería2.x

No obstante, en ese caso dispondrá sólo de las funciones de la versión antigua de STEP 7. Sin embargo, los proyectos y las librerías se podrán seguir manteniendo con la versión antigua de STEP 7.

Nota

Al pasar de la versión 3 a la versión 4 u otra posterior se cambia solamente el nombre del tipo de archivo, el formato no se modifica. Por esta razón en STEP 7 V4 no existe ningún tipo de archivo del tipo Proyecto3.x.

Procedimiento

Para convertir los proyectos de la versión 2 al formato de la versión actual de STEP 7:

1. Ejecute la función "Guardar como" reorganizando (Menú Archivo).
2. Elija en el cuadro de diálogo "Guardar proyecto como " el tipo de archivo "Proyecto" y active el botón de comando "Guardar".

Para convertir proyectos de la versión 2 a la versión actual de STEP 7 (conservando el formato original):

1. En caso necesario efectúe el paso 1 anteriormente descrito.
2. Elija en el cuadro de diálogo "Guardar proyecto como " el tipo de archivo de la antigua versión de STEP 7 y pulse el botón de comando "Guardar".

A.4.3 Observaciones sobre los proyectos de STEP 7 V2.1 con comunicación de datos globales (GD)

- Si desea convertir un proyecto con datos globales de STEP 7 V2.1 a STEP 7 V5 deberá abrir previamente la tabla GD con STEP 7 V5.0 en el proyecto de STEP 7 V2.1. Todos los datos de comunicación configurados hasta entonces se convertirán automáticamente a la nueva estructura por comunicación GD.
- Al archivar proyectos de STEP 7 V2.1 puede suceder que los antiguos programas de compresión (ARJ, PKZIP...) generen un mensaje de error si el proyecto contiene archivos cuyos nombres excedan los 8 caracteres. Este mensaje se visualizará también si la red MPI se ha editado en el proyecto de STEP 7 V2.1 con una denominación que comprenda más de 8 caracteres. En los proyectos de STEP 7 V2.1 con datos globales, utilice un nombre para la red MPI que no exceda los 8 caracteres antes de arrancar por primera vez la configuración de la comunicación de datos globales.
- Si desea cambiar el nombre de un proyecto de STEP 7 V2.1 deberá asignar nuevamente los títulos de las columnas (CPUs) en la tabla GD, seleccionando de nuevo la CPU en cuestión. Cuando restablezca el nombre antiguo del proyecto, las asignaciones existirán nuevamente.

A.4.4 Esclavos DP con archivos GSD defectuosos o inexistentes

Si edita configuraciones antiguas de equipos con la versión 5.1 de STEP 7, puede ocurrir en algunos casos que falte o no se pueda compilar el archivo GSD de un esclavo DP (p. ej., debido a errores de sintaxis en el archivo GSD).

En este caso STEP 7 genera un esclavo "Dummy" que representa al esclavo configurado, p. ej., después de cargar un equipo en PG o de abrir y continuar editando un proyecto antiguo. Este esclavo "Dummy" sólo se puede editar de forma muy limitada; no podrá modificar la estructura del esclavo (identificación DP) ni los parámetros. Sin embargo se puede volver a cargar en el equipo; la configuración original del esclavo permanecerá intacta. También es posible borrar todo el esclavo DP.

Modificar la configuración y parametrización del esclavo DP

Si desea modificar la configuración o parametrización del esclavo DP deberá solicitar del fabricante un archivo GSD actualizado para este esclavo DP y ponerlo a disposición a través del comando de menú **Herramientas > Instalar archivos GSD**.

Después de instalar el archivo GSD correcto se utilizará para la representación del esclavo DP. El esclavo DP conserva sus datos y se puede editar de nuevo completamente.

A.5 Programas de ejemplo

A.5.1 Proyectos y programas de ejemplo

El CD de instalación de STEP 7 contiene varios proyectos de ejemplo útiles, que aparecen relacionados en la siguiente tabla. Una vez que los haya instalado, encontrará los proyectos de ejemplo en el cuadro de diálogo "Abrir" del administrador SIMATIC (ficha "Proyectos de ejemplo"). Instalando paquetes opcionales puede ampliar los proyectos de ejemplo. Consulte los detalles de estos proyectos de ejemplo en la documentación de cada paquete opcional.

Programas y proyectos de ejemplo	Contenidos en el CD	Descripción en este capítulo	Descripción IEC / internacional el OB 1
Proyectos "ZEs01_01_STEP7_**" a "ZEs01_06_STEP7_**" (Introducción y ejercicios prácticos)	•	Manual	•
Proyecto "Zes01_11_STEP7_DezP" (Ejemplo de una configuración PROFIBUS DP)	•	-	-
Proyecto "ZEs01_08_STEP7_Mezcla" (proceso de mezcla industrial)	•	•	
Proyecto "ZEs01_09_STEP7_CEBRA" (control de semáforo en un paso de peatones)	•		•
Proyecto "ZEs01_10_STEP7_COM_SFB" (intercambio de datos entre dos CPUs S7-400)	•		•
Proyecto ZXX01_14_HSystem_S7400H (proyecto de acceso para sistemas de alta disponibilidad) Proyecto ZXX01_15_HSystem_RED_IO (proyecto de acceso para sistemas de alta disponibilidad con periferia redundante)	• •	Manual propio Manual propio	• •
Proyectos "ZEs01_11_STEP7_COM_SFC1" y "ZEs01_12_STEP7_COM_SFC2" (intercambio de datos a través de SFCs de comunicación para enlaces no configurados)	•		•
Proyecto "ZEs01_13_STEP7_PID-Temp" (ejemplo de reguladores de temperatura FB 58 y FB 59)	•		•
Ejemplo de manipulación de alarmas horarias		•	
Ejemplo de manipulación de alarmas de retardo		•	
Ejemplo de enmascarar y desenmascarar eventos de errores síncronos		•	
Ejemplo de bloquear y habilitar eventos de alarma y de error asíncrono		•	
Ejemplo de procesamiento retardado de eventos de alarma y asíncronos		•	

El objetivo principal de los ejemplos no reside en proporcionar un determinado estilo de programación ni tampoco los conocimientos técnicos requeridos para controlar un determinado proceso. Estos ejemplos persiguen, más bien, reproducir los pasos requeridos para diseñar un programa.

Borrar, copiar e instalar los proyectos de ejemplo adjuntos

Los proyectos de ejemplo adjuntos se pueden borrar e instalar de nuevo en el Administrador SIMATIC. Para instalarlos se deberá arrancar el programa Setup de STEP 7. Los proyectos de ejemplo se pueden instalar posteriormente de forma selectiva. La función "Guardar como" permite guardar como proyectos de usuario copias generadas con proyectos de ejemplo y "ejemplos de proyecto" propios.

Nota

A no ser que se ordene lo contrario, los proyectos de ejemplo adjuntos se copiarán automáticamente durante la instalación de STEP 7. Si ha modificado ya dichos proyectos, éstos se sobrescribirán con los originales al instalarse nuevamente STEP 7.

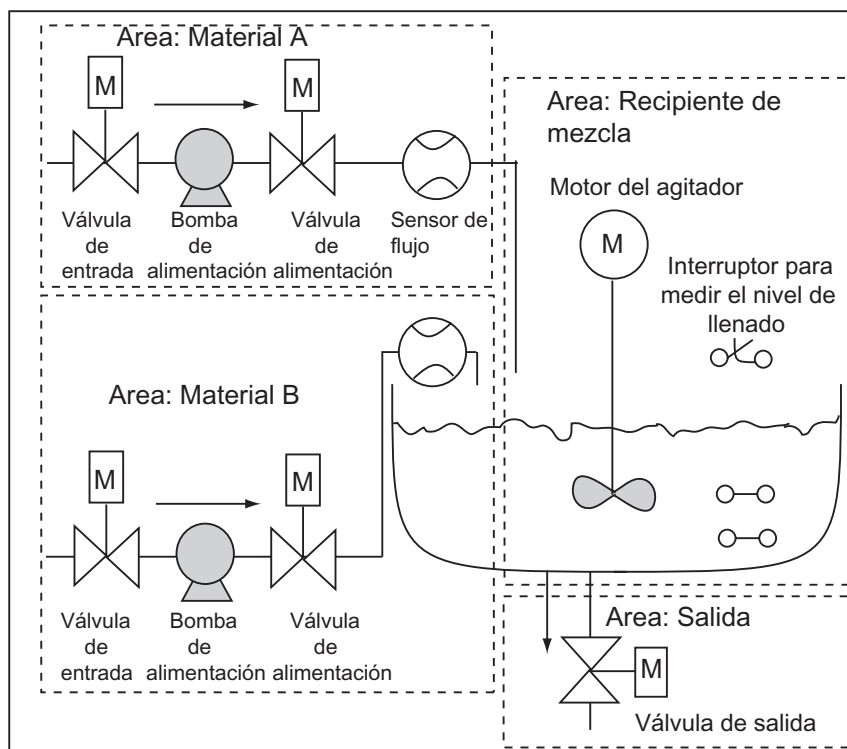
Por tanto, antes de efectuar cambios es recomendable copiar los proyectos de ejemplo adjuntos y editar sólo las copias.

A.5.2 Programa de ejemplo para un proceso de mezcla industrial

El programa de ejemplo se basa en informaciones suministradas en la 1ª parte del manual, en la que se describía el control de un proceso de mezcla industrial.

Tarea

Dos materiales (A y B) se han de mezclar en un recipiente con ayuda de un agitador. La masa se descarga del recipiente de mezcla mediante una válvula de salida. La figura siguiente muestra un diagrama del proceso de ejemplo.



Descripción de las partes del proceso

En la 1ª parte del manual se describió cómo subdividir el proceso de ejemplo en áreas funcionales y tareas individuales. A continuación se describen las diferentes áreas.

Area Materiales A y B:

- Las tuberías de entrada de material deben tener una válvula de entrada, una válvula de alimentación y una bomba de alimentación.
- Las tuberías de entrada tienen instalado un sensor de flujo.
- La conexión de las bombas de alimentación debe estar bloqueada cuando el medidor de nivel de flujo indica "recipiente lleno".
- La conexión de las bombas de alimentación debe estar bloqueada cuando la válvula de salida está abierta.
- Las válvulas de entrada y de alimentación no se pueden abrir antes de que haya transcurrido 1 segundo tras la activación de la bomba de alimentación.
- Las válvulas se deben cerrar inmediatamente tras parar las bombas de alimentación (señal del sensor de flujo) para impedir que el material salga de la bomba.
- La activación de las bombas de alimentación es temporizada, es decir, el sensor debe señalar la existencia del flujo dentro de los 7 segundos subsiguientes a la activación.
- Las bombas de alimentación se han de desconectar lo más rápido posible en el caso de que los sensores de flujo no señalicen ningún flujo durante la marcha de las bombas.
- La cantidad de arranques de las bombas de alimentación debe ser contada (define el intervalo de mantenimiento).

Area Recipiente de mezcla:

- La conexión del motor del agitador debe estar enclavada cuando el medidor de nivel de llenado señala "recipiente bajo el mínimo" o está abierta la válvula de salida.
- El motor del agitador emite una señal de respuesta (señalización en respuesta) tras alcanzar el número de revoluciones nominal. Si esta señal no se emite dentro de los 10 segundos subsiguientes al arranque del motor, entonces se ha de desconectar dicho motor.
- La cantidad de arranques del motor del agitador debe contarse (intervalo de mantenimiento).
- En el recipiente de mezcla se han de instalar tres sensores:
 - recipiente lleno: Contacto normalmente cerrado. Si se alcanza el nivel máximo de llenado, se abre el contacto.
 - nivel de llenado en el recipiente sobre el mínimo: Contacto normalmente abierto. Si se alcanza el nivel mínimo de llenado, se cierra el contacto..
 - recipiente no vacío: Contacto normalmente abierto. Si el recipiente no está vacío, el contacto está cerrado.

Area Salida:

- La salida debe controlarse a través de una válvula de solenoide.
- La válvula de solenoide es controlada por el operador, pero, en cualquier caso siempre se cierra, lo más tardar, al aparecer la señal "recipiente vacío".
- La apertura de la válvula de salida está enclavada cuando
 - el motor del agitador está en marcha
 - el recipiente está vacío

Panel de manejo

Para que el operador pueda arrancar, parar y observar el proceso, se ha de prever un panel de manejo. El panel de manejo contiene

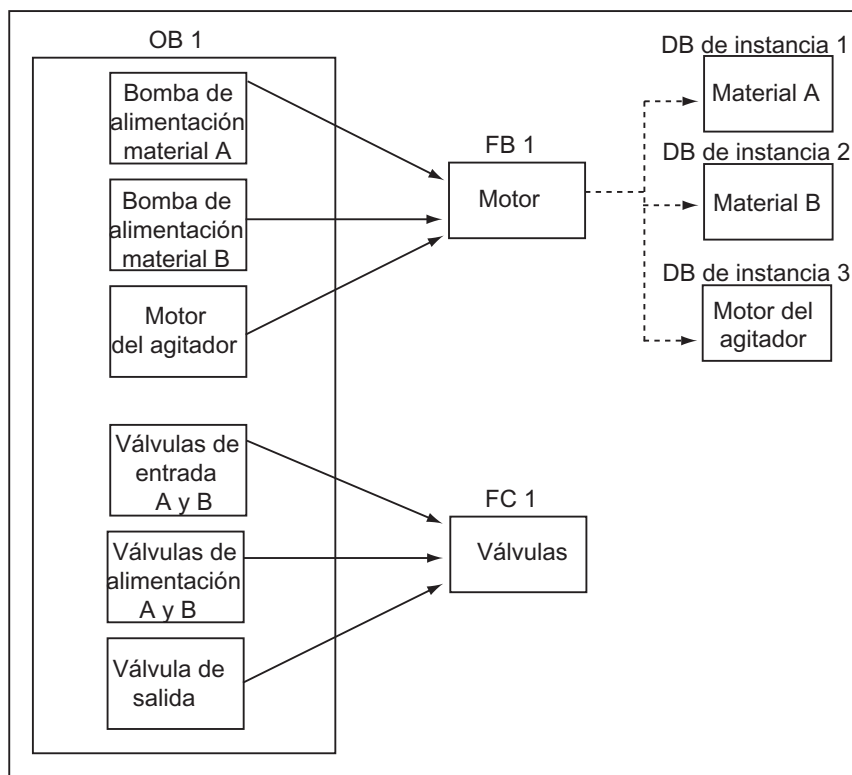
- pulsadores para controlar las operaciones principales Con el pulsador "Desactivar lámpara de señalización" se apagan las lámparas de señalización de mantenimiento del motor afectado y se pone a 0 el contador que controla los intervalos de mantenimiento.
- lámparas de señalización para leer el estado operativo
- el pulsador de PARO DE EMERGENCIA

A.5.2.1 Definir bloques lógicos

La estructura del programa se define subdividiendo el programa de usuario en diferentes bloques y jerarquizando las llamadas de bloques.

Jerarquía de las llamadas de bloques

La figura siguiente representa la jerarquía de los bloques que se pueden llamar en el programa estructurado.



- OB1: Representa el interface con el sistema operativo de la CPU y contiene el programa principal. En el OB 1 se llaman los bloques FB 1 y FC 1 y se transfieren los parámetros específicos necesarios para controlar el proceso.
- FB 1: Un sólo bloque de función puede controlar la bomba de alimentación para el material A, la bomba de alimentación para el material B y el motor del agitado, puesto que las exigencias (conectar, desconectar, contar las utilidades, etc.) son idénticas.
- DBs de instancia 1-3: Los parámetros actuales y los datos estáticos para el control del material A, del material B y del motor del agitador son diferentes, por lo que se depositan en tres DBs de instancia asociados al FB 1.
- FC 1: las válvulas de entrada y de alimentación para los materiales A y B y para la válvula de salida utilizan también un bloque lógico común. Como sólo se deben programar las funciones "Abrir" y "Cerrar", basta con utilizar una sola FC.

A.5.2.2 Asignar nombres simbólicos

Definir nombres simbólicos

En el programa ejemplificado se utilizan símbolos que se deben definir con STEP 7 en la tabla de símbolos. Las tablas siguientes muestran los nombres simbólicos y las direcciones absolutas de los elementos utilizados en el programa.

Direcciones simbólicas de la bomba de alimentación, del motor del agitador y de las válvulas de entrada:			
Nombre simbólico	Dirección	Tipo de datos	Descripción
Feed_pump_A_start	E 0.0	BOOL	Pulsador de arranque de la bomba de alimentación del material A
Feed_pump_A_stop	E 0.1	BOOL	Pulsador de parada de la bomba de alimentación del material A
Flow_A	E 0.2	BOOL	Material A fluye
Inlet_valve_A	A 4.0	BOOL	Activación de la válvula de entrada para material A
Feed_valve_A	A 4.1	BOOL	Activación de la bomba de alimentación para material A
Feed_pump_A_on	A 4.2	BOOL	Lámpara de señalización "Bomba alimentación material A en marcha"
Feed_pump_A_off	A 4.3	BOOL	Lámpara de señalización "Bomba alimentación material A no en marcha"
Feed_pump_A	A 4.4	BOOL	Activación de la bomba de alimentación para material A
Feed_pump_A_fault	A 4.5	BOOL	Lámpara de señalización "Error de la bomba de alimentación A"
Feed_pump_A_maint	A 4.6	BOOL	Lámpara de señalización "Mantenimiento bomba aliment. A"
Feed_pump_B_start	E 0.3	BOOL	Pulsador de arranque de la bomba de alimentación del material B
Feed_pump_B_stop	E 0.4	BOOL	Pulsador de parada de la bomba de alimentación del material B
Flow_B	E 0.5	BOOL	Material B fluye
Inlet_valve_B	A 5.0	BOOL	Activación de la válvula de entrada para material B
Feed_valve_B	A 5.1	BOOL	Activación de la bomba de alimentación para material B
Feed_pump_B_on	A 5.2	BOOL	Lámpara de señalización "Bomba alimentación material B en marcha"
Feed_pump_B_off	A 5.3	BOOL	Lámpara de señalización "Bomba alimentación material B no en marcha"
Feed_pump_B	A 5.4	BOOL	Activación de la bomba de alimentación para material B
Feed_pump_B_fault	A 5.5	BOOL	Lámpara de señalización "Error de la bomba de alimentación B"
Feed_pump_B_maint	A 5.6	BOOL	Lámpara de señalización "Mantenimiento bomba aliment. B"
Agitator_running	E 1.0	BOOL	Señal de respuesta del motor del agitador

Direcciones simbólicas de la bomba de alimentación, del motor del agitador y de las válvulas de entrada:			
Nombre simbólico	Dirección	Tipo de datos	Descripción
Agitator_start	E 1.1	BOOL	Pulsador de arranque del agitador
Agitator_stop	E 1.2	BOOL	Pulsador de parada del agitador
Agitator	A 8.0	BOOL	Activación del agitador
Agitator_on	A 8.1	BOOL	Lámpara de señalización "Agitador ON"
Agitator_off	A 8.2	BOOL	Lámpara de señalización "Agitador OFF"
Agitator_fault	A 8.3	BOOL	Lámpara de señalización "Fallo del motor del agitador"
Agitator_maint	A 8.4	BOOL	Lámpara de señalización "Mantenimiento del agitador"

Direcciones simbólicas de sensores e indicador de nivel del recipiente:

Nombre simbólico	Dirección	Tipo de datos	Descripción
Tank_below_max	E 1.3	BOOL	Sensor "Recipiente de mezcla no está lleno"
Tank_above_min	E 1.4	BOOL	Sensor "Recipiente de mezcla sobre el mínimo"
Tank_not_empty	E 1.5	BOOL	Sensor "Recipiente de mezcla no está vacío"
Tank_max_disp	A 9.0	BOOL	Lámpara de señalización "Recipiente de mezcla lleno"
Tank_min_disp	A 9.1	BOOL	Lámpara de señalización "Recipiente de mezcla bajo mínimo"
Tank_empty_disp	A 9.2	BOOL	Lámpara de señalización "Recipiente de mezcla vacío"

Direcciones simbólicas de la válvula de salida

Nombre simbólico	Dirección	Tipo de datos	Descripción
Drain_open	E 0.6	BOOL	Pulsador para abrir la válvula de salida
Drain_closed	E 0.7	BOOL	Pulsador para cerrar la válvula de salida
Drain	A 9.5	BOOL	Activación de la válvula de salida
Drain_open_disp	A 9.6	BOOL	Lámpara de visualización "Válvula de salida abierta"
Drain_closed_disp	A 9.7	BOOL	Lámpara de visualización "Válvula de salida cerrada"

Direcciones simbólicas de los demás elementos del programa

Nombre simbólico	Dirección	Tipo de datos	Descripción
EMER_STOP_off	E 1.6	BOOL	Pulsador de PARO DE EMERGENCIA
Reset_maint	E 1.7	BOOL	Tecla de reset de las lámparas de señalización de mantenimiento de todos los motores
Motor_block	FB 1	FB 1	FB para controlar las bombas y el motor
Valve_block	FC 1	FC 1	FC para controlar las válvulas
DB_feed_pump_A	DB 1	FB 1	DB de instancia para controlar la bomba de alimentación A
DB_feed_pump_B	DB 2	FB 1	DB de instancia para controlar la bomba de alimentación B
DB_agitator	DB 3	FB 1	DB de instancia para controlar el motor del agitador

A.5.2.3 Crear el FB para el motor

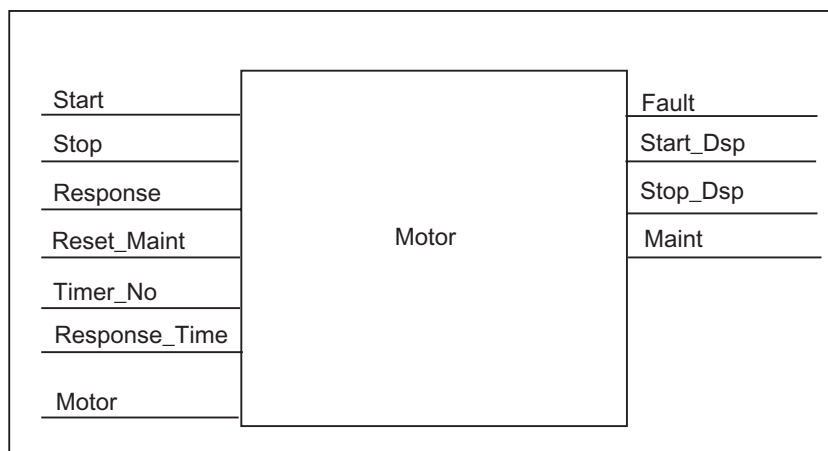
Lógica requerida en el FB

El FB para el motor contiene las siguientes funciones lógicas:

- Existe una entrada para arranque y una para parada.
- Una serie de enclavamientos permite el servicio de los equipos (bombas y motor de agitador). El estado de los enclavamientos se memoriza en los datos locales temporales (pila L) del OB 1 (p.ej., "Habilitación_motor") y se combinan lógicamente con las entradas de arranque y de parada al ejecutarse el FB para el motor.
- La respuesta de los equipos debe recibirse dentro de un determinado tiempo. En caso contrario se supone que se ha presentado un error. En tal caso, el FB detiene el motor.
- El temporizador y el intervalo para el ciclo de respuesta o de fallo deben ser definidos.
- Tras oprimir el pulsador de arranque (Marcha) y concederse la habilitación, se activa el equipo y continúa operando hasta que se oprima el pulsador de parada (Paro).
- Conectado el equipo, el temporizador comienza a correr. Si la señal de respuesta del aparato no se recibe antes de transcurrir el temporizador, entonces se detiene la marcha del equipo.

Determinar las entradas y las salidas

La figura siguiente muestra las entradas y las salidas del FB general para el motor.



Definir los parámetros para el FB

Si se desea crear un FB apto para todos los motores (para controlar ambas bombas y el motor), se han de definir nombres de parámetros generales para las entradas y las salidas.

El FB para el motor del proceso ejemplificado tiene los siguientes requerimientos:

- Necesita señales del panel de operador para parar o arrancar el motor y las bombas.
- Necesita una señal de respuesta del motor y de las bombas de que el motor está en marcha.
- Tiene que calcular el tiempo transcurrido entre la transmisión de la señal de marcha del motor y la recepción de la señal de respuesta. Si la señal de respuesta no se recibe durante un tiempo determinado, se debe desconectar el motor.
- Debe conectar o desconectar las lámparas de señalización correspondientes en el panel de operador.
- Transmite una señal para controlar el motor.

Estos requerimientos se pueden definir como entradas y salidas del FB. La tabla siguiente muestra los parámetros del FB para el motor del proceso de ejemplo.

Nombre del parámetro	Entrada	Salida	Entrada/salida
Start	n		
Stop	n		
Response	n		
Reset_Maint	n		
Timer_No	n		
Response_Time	n		
Fault		n	
Start_Dsp		n	
Stop_Dsp		n	
Maint		n	
Motor			n

Declaración de variables del FB para el motor

Es preciso declarar los parámetros de entrada, de salida y de entrada/salida del FB para el motor.

Dirección	Declaración	Designación	Tipo	Valor inicial
0.0	IN	Arranque	BOOL	FALSE
0.1	IN	Stop	BOOL	FALSE
0.2	IN	Response	BOOL	FALSE
0.3	IN	Reset_Maint	BOOL	FALSE
2.0	IN	Time_No	TIMER	
4.0	IN	Response_Time	S5TIME	S5T#0MS
6.0	OUT	Fault	BOOL	FALSE
6.1	OUT	Start_Dsp	BOOL	FALSE
6.2	OUT	Stop_Dsp	BOOL	FALSE
6.3	OUT	Maint	BOOL	FALSE
8.0	IN_OUT	Motor	BOOL	FALSE
10.0	STAT	Time_bin	WORD	W#16#0
12.0	STAT	Time_BCD	WORD	W#16#0
14.0	STAT	Starts	INT	0
16.0	STAT	Start_Edge	BOOL	FALSE

En los FBs, las variables de entradas, salidas, entrada/salida y variables estáticas se memorizan en el DB de instancia indicado en la instrucción de llamada. Las variables temporales se memorizan en la pila L.

Programar el FB para el motor

En STEP 7, cada bloque que es llamado por otro bloque debe ser creado antes del bloque que contiene la llamada. Por lo tanto, en el programa de ejemplo deberá crear el FB para el motor antes que el OB 1.

El área de instrucciones del FB 1 tiene la siguiente estructura en el lenguaje de programación AWL:

Segmento 1 Marcha/parada y autorretención

```
U(
O   #Start
O   #Motor
)
UN  #Stop
=   #Motor
```

Segmento 2 Vigilancia del arranque

```
U   #Motor
L   #Response_Time
SE  #Timer_No
UN  #Motor
R   #Timer_No
L   #Timer_No
T   #Timer_bin
LC  #Timer_No
T   #Timer_BCD
U   #Timer_No
UN  #Response
S   #Fault
R   #Motor
```

Segmento 3 Lámpara Marcha y Reset Error

```
U   #Response
=   #Start_Dsp
R   #Fault
```

Segmento 4 Lámpara Parada

```
UN  #Response
=   #Stop_Dsp
```

Segmento 5 Contaje de los arranques

```
U   #Motor
FP  #Start_Edge
SPBN lab1
L   #Starts
+   1
T   #Starts

lab1: NOP    0
```

Segmento 6 Lámpara de señalización de mantenimiento

```
L   #Starts
L   50
>=|
=   #Maint
```

Segmento 7 Reset del contador del número de arranques

```
U   #Reset_Maint
U   #Maint
SPBN END
L   0
T   #Starts
END: NOP    0
```

Crear los bloques de datos de instancia

Cree tres bloques de datos y ábralos uno tras otro. En el cuadro de diálogo "Nuevo bloque de datos", elija la opción "DB asociado a un FB". En el cuadro de lista "Asignación", elija "FB 1". Así, los bloques de datos quedarán asociados fijamente al FB 1 en calidad de DBs de instancia.

A.5.2.4 Crear la FC para las válvulas

Lógica requerida en la FC

La FC para las válvulas de entrada, alimentación y de salida contiene las siguientes funciones lógicas:

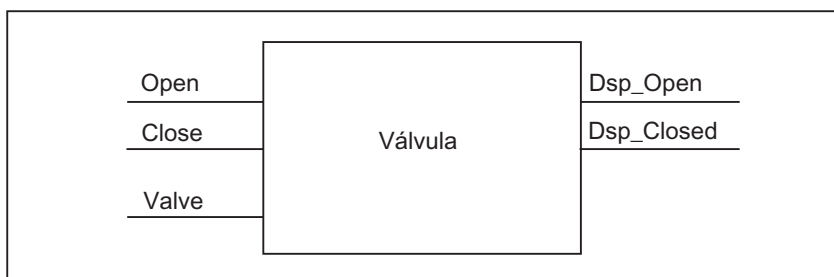
- Existe una entrada para abrir y una para cerrar las válvulas.
- Una serie de enclavamientos permiten la apertura de las válvulas. El estado de los enclavamientos se memoriza en los datos locales temporales (pila L) del OB 1 (p.ej., "Habilitación_válvula") y se combinan luego con las entradas para abrir y cerrar al ejecutarse la FC para las válvulas.

La tabla siguiente muestra los parámetros que deben transferirse a la FC.

Parámetros para las válvulas	Entrada	Salida	Entrada/salida
Open	✓		
Close	✓		
Dsp_Open		✓	
Dsp_Closed		✓	
Valve			✓

Determinar las entradas y las salidas

La figura siguiente muestra las entradas y las salidas de la FC general para las válvulas. Los aparatos que llaman el FB para el motor transfieren los parámetros de entrada. La FC para las válvulas devuelve los parámetros de salida.



Declaración de variables de la FC para las válvulas

Al igual que en el FB para el motor, en la FC para las válvulas se deben declarar los parámetros de entrada, salida y entrada/salida (v. siguiente tabla de declaración de variables).

Dirección	Declaración	Designación	Tipo	Valor inicial
0.0	IN	Open	BOOL	FALSE
0.1	IN	Close	BOOL	FALSE
2.0	OUT	Dsp_Open	BOOL	FALSE
2.1	OUT	Dsp_Closed	BOOL	FALSE
4.0	IN_OUT	Valve	BOOL	FALSE

En las FC, las variables temporales se memorizan en la pila L. Las variables de entrada, salida y de entrada/salida se memorizan como punteros en el bloque lógico que ha llamado la FC. Para estas variables se utiliza un área de memoria adicional en la pila L (tras las variables temporales).

Programar la FC para las válvulas

También la función FC 1 para las válvulas debe ser creada antes del OB 1, ya que los bloques llamados deben ser creados antes de los invocantes.

El área de instrucciones de la FC 1 en el lenguaje de programación AWL tiene la estructura siguiente:

Segmento 1 Abrir/cerrar y autorretención

```

U(
O   #Open
O   #Valve
)
UN  #Close
=   #Valve

```

Segmento 2 Señalización "válvula abierta"

```

U   #Valve
=   #Dsp_Open

```

Segmento 3 Señalización "válvula cerrada"

```

UN  #Valve
=   #Dsp_Closed

```


A.5.2.5 Crear el OB 1

El OB 1 determina la estructura del programa de ejemplo. Además, el OB 1 contiene los parámetros que se transfieren a las diferentes funciones, tales como:

- Los segmentos AWL para las bombas de alimentación y el motor suministran al FB para el motor los parámetros de entrada para arrancar ("Marcha"), parar ("Paro") y para la señalización en respuesta ("Respuesta"), así como para el reset del indicador de mantenimiento (Reset_mant). El FB para el motor se ejecuta en cada ciclo del PLC.
- Cuando se ejecuta el FB para el motor, las entradas "Timer_No" y "Response_Time" determinan qué temporizador se utilizará y el intervalo de tiempo en el que se debe recibir una respuesta.
- Las FC para las válvulas son ejecutadas automáticamente en cada ciclo de programa del PLC.

El programa utiliza el FB para el motor con diferentes DBs de instancia para cumplir las tareas de control de las bombas de alimentación y del motor del agitador.

Declarar variables para el OB 1

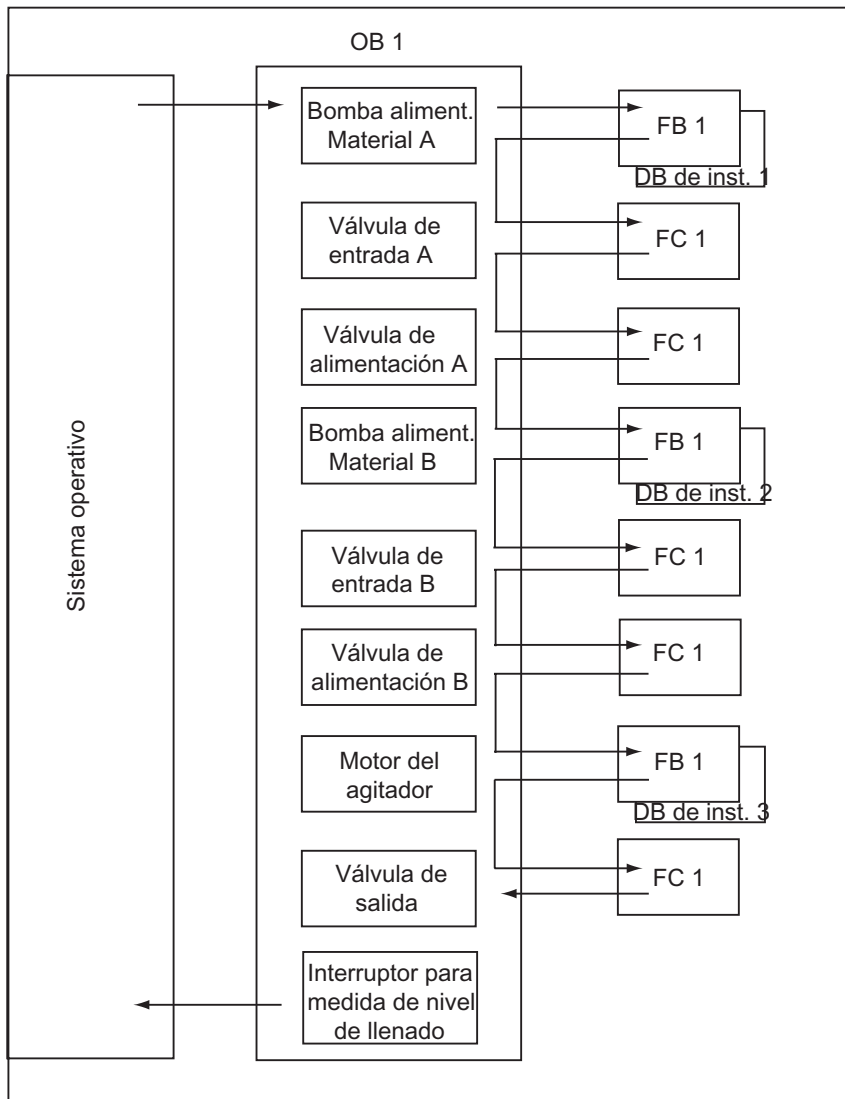
A continuación se muestra la tabla de declaración de variables del OB 1. Los primeros 20 bytes contienen la información de arranque del OB 1 y no se pueden modificar.

Dirección	Declaración	Designación	Tipo
0.0	TEMP	OB1_EV_CLASS	BYTE
1.0	TEMP	OB1_SCAN1	BYTE
2.0	TEMP	OB1_PRIORITY	BYTE
3.0	TEMP	OB1_OB_NUMBR	BYTE
4.0	TEMP	OB1_RESERVED_1	BYTE
5.0	TEMP	OB1_RESERVED_2	BYTE
6.0	TEMP	OB1_PREV_CYCLE	INT
8.0	TEMP	OB1_MIN_CYCLE	INT
10.0	TEMP	OB1_MAX_CYCLE	INT
12.0	TEMP	OB1_DATE_TIME	DATE_AND_TIME
20.0	TEMP	Enable_Motor	BOOL
20.1	TEMP	Enable_Valve	BOOL
20.2	TEMP	Start_Fulfilled	BOOL
20.3	TEMP	Stop_Fulfilled	BOOL
20.4	TEMP	Inlet_Valve_A_Open	BOOL
20.5	TEMP	Inlet_Valve_A_Closed	BOOL
20.6	TEMP	Feed_Valve_A_Open	BOOL
20.7	TEMP	Feed_Valve_A_Closed	BOOL
21.0	TEMP	Inlet_Valve_B_Open	BOOL
21.1	TEMP	Inlet_Valve_B_Closed	BOOL
21.2	TEMP	Feed_Valve_B_Open	BOOL
21.3	TEMP	Feed_Valve_B_Closed	BOOL
21.4	TEMP	Open_Drain	BOOL
21.5	TEMP	Close_Drain	BOOL
21.6	TEMP	Close_Valve_Fulfilled	BOOL

Crear el programa para el OB 1

En STEP 7, cada bloque que es llamado por otro bloque debe ser creado antes que el bloque que contiene la llamada. Por lo tanto, en el programa de ejemplo, tanto el FB para el motor como la FC para las válvulas se deben crear antes que el programa contenido en el OB 1.

Los bloques FB 1 y FC 1 se llaman varias veces en el OB 1 (el FB 1 con diversos DBs de instancia):



El área de instrucciones del OB 1 en el lenguaje de programación AWL tiene la estructura siguiente:

Segmento 1 Enclavamientos para bomba de alimentación A

```
U   "EMER_STOP_off"
U   "Tank_below_max"
UN  "Drain"
=   #Enable_Motor
```

Segmento 2 Llamada FB de motor para material A

```
U           "Feed_pump_A_start"
U           #Enable_Motor
=           #Start_Fulfilled
U(
O           "Feed_pump_A_stop"
ON  #Enable_Motor
)
=           #Stop_Fulfilled
CALL "Motor_block", "DB_feed_pump_A"
Start      :=#Start_Fulfilled
Stop       :=#Stop_Fulfilled
Response   :="Flow_A"
Reset_Maint :="Reset_maint"
Timer_No   :=T12
Reponse_Time:=S5T#7S
Fault      :="Feed_pump_A_fault"
Start_Dsp  :="Feed_pump_A_on"
Stop_Dsp   :="Feed_pump_A_off"
Maint      :="Feed_pump_A_maint"
Motor      :="Feed_pump_A"
```

Segmento 3 Retardo de la habilitación de válvula material A

```
U   "Feed_pump_A"
L   S5T#1S
SE  T    13
UN  "Feed_pump_A"
R   T    13
U   T    13
=   #Enable_Valve
```

Segmento 4 Control de válvula de entrada para material A

```
UN  "Flow_A"
UN  "Feed_pump_A"
=   #Close_Valve_Fulfilled
CALL "Valve_block"
Open   :=#Enable_Valve
Close  :=#Close_Valve_Fulfilled
Dsp_Open   :=#Inlet_Valve_A_Open
Dsp_Closed:=#Inlet_Valve_A_Closed
Valve     :="Inlet_Valve_A"
```

Segmento 5 Control de válvula de alimentación para material A

```

UN   "Flow_A"
UN   "Feed_pump_A"
=     #Close_Valve_Fulfilled
CALL "Valve_block"
      Open   :=#Enable_Valve
      Close  :=#Close_Valve_Fulfilled
      Dsp_Open   :=#Feed_Valve_A_Open
      Dsp_Closed:=#Feed_Valve_A_Closed
      Valve   :="Feed_Valve_A"

```

Segmento 6 Enclavamientos para bomba de alimentación B

```

U     "EMER_STOP_off"
U     "Tank_below_max"
UN    "Drain"
=     "Enable_Motor"

```

Segmento 7 Llamada FB de motor para material B

```

U           "Feed_pump_B_start"
U           #Enable_Motor
=           #Start_Fulfilled
U(
O           "Feed_pump_B_stop"
ON          #Enable_Motor
)
=           #Stop_Fulfilled
CALL "Motor_block", "DB_feed_pump_B"
      Start   :=#Start_Fulfilled
      Stop    :=#Stop_Fulfilled
      Response :="Flow_B"
      Reset_Maint :="Reset_maint"
      Timer_No :=T14
      Reponse_Time:=S5T#7S
      Fault    :="Feed_pump_B_fault"
      Start_Dsp :="Feed_pump_B_on"
      Stop_Dsp  :="Feed_pump_B_off"
      Maint    :="Feed_pump_B_maint"
      Motor    :="Feed_pump_B"

```

Segmento 8 Retardo de la habilitación de válvula material B

```

U     "Feed_pump_B"
L     S5T#1S
SE    T     15
UN    "Feed_pump_B"
R     T     15
U     T     15
=     #Enable_Valve

```

Segmento 9 Control de válvula de entrada para material B

```

UN   "Flow_B"
UN   "Feed_pump_B"
=     #Close_Valve_Fulfilled
CALL "Valve_block"
      Open   :=#Enable_Valve
      Close  :=#Close_Valve_Fulfilled
      Dsp_Open   :=#Inlet_Valve_B_Open
      Dsp_Closed:=#Inlet_Valve_B_Closed
      Valve   :="Inlet_Valve_B"

```

Segmento 10 Control de válvula de alimentación para material B

```

UN   "Flow_B"
UN   "Feed_pump_B"
=     #Close_Valve_Fulfilled
CALL "Valve_block"
      Open   :=#Enable_Valve
      Close  :=#Close_Valve_Fulfilled
      Dsp_Open   :=#Feed_Valve_B_Open
      Dsp_Closed:=#Feed_Valve_B_Closed
      Valve   :="Feed_Valve_B"

```

Segmento 11 Enclavamientos para agitador

```

U     "EMER_STOP_off"
U     "Tank_above_min"
UN    "Drain"
=     #Enable_Motor

```

Segmento 12 Llamada FB de motor para agitador

```

U           "Agitator_start"
U           #Enable_Motor
=           #Start_Fulfilled
U(
O           "Agitator_stop"
ON          #Enable_Motor
)
=           #Stop_Fulfilled
CALL "Motor_block", "DB_Agitator"
      Start   :=#Start_Fulfilled
      Stop    :=#Stop_Fulfilled
      Response :="Agitator_running"
      Reset_Maint :="Reset_maint"
      Timer_No :=T16
      Reponse_Time:=S5T#10S
      Fault   :="Agitator_fault"
      Start_Dsp :="Agitator_on"
      Stop_Dsp  :="Agitator_off"
      Maint    :="Agitator_maint"
      Motor    :="Agitator"

```

Segmento 13 Enclavamientos para válvula de salida

```
U    "EMER_STOP_off"
U    "Tank_not_empty"
UN   "Agitator"
=    "Enable_Valve"
```

Segmento 14 Control de válvula de salida

```
U          "Drain_open"
U          #Enable_Valve
=          #Open_Drain
U(
O          "Drain_closed"
ON        #Enable_Valve
)
=          #Close_Drain
CALL "Valve_block"
      Open   :=#Open_Drain
      Close  :=#Close_Drain
      Dsp_Open   :="Drain_open_disp"
      Dsp_Closed :="Drain_closed_disp"
      Valve    :="Drain"
```

Segmento 15 Indicación del nivel de llenado de recipiente

```
UN"Tank_below_max"
= "Tank_max_disp"
UN "Tank_above_min"
= "Tank_min_disp"
UN "Tank_not_empty"
= "Tank_empty_disp"
```

A.5.3 Ejemplo de aplicación de las alarmas horarias

Estructura de las alarmas horarias en el programa de usuario

FC 12

OB 10

OB 1 y OB 80

A.5.3.1 Estructura del programa de usuario "Alarmas horarias"

Tarea

La salida A 4.0 debe estar activada en el período comprendido entre el lunes, 5.00 horas y el viernes, 20.00 horas. En el período comprendido entre el viernes, 20.00 horas y el lunes, 5.00 horas, la salida A 4.0 debe estar desactivada.

Aplicación en el programa de usuario

La tabla siguiente muestra las tareas parciales de los bloques utilizados.

Bloque	Tarea parcial
OB 1	Llamada de la función FC 12
FC 12	Dependiendo del estado de la salida A 4.0, del estado de la alarma horaria y de las entradas E 0.0 y E 0.1 <ul style="list-style-type: none"> • Predeterminar hora de arranque • Activar la alarma horaria • Disparar la alarma horaria • CAN_TINT
OB 10	Dependiendo del día actual de la semana <ul style="list-style-type: none"> • Predeterminar la hora de arranque • Activar o desactivar la salida A 4.0 • Activar la siguiente alarma horaria • Disparar la alarma horaria
OB 80	Activar la salida A 4.1 Almacenar en el área de marcas la información de evento de arranque del OB 80

Operandos utilizados

La tabla siguiente muestra los operandos globales utilizados. Las variables temporales de los bloques se declaran en el área de declaración del bloque en cuestión.

Operando	Significado
E0.0	Entrada de habilitación de "Activar alarma horaria" y "Disparar alarma horaria"
E0.1	Entrada para cancelar una alarma horaria
A 4.0	Salida activada/desactivada por el OB de alarma horaria (OB 10)
A 4.1	Salida activada al ocurrir un error de tiempo (OB 80)
MW 16	ESTADO (STATUS) de la alarma horaria (SFC 31 "QRY_TINT")
MB 100 hasta MB 107	Memoria para la información de evento de arranque del OB 10 (sólo hora)
MB 110 hasta MB 129	Memoria para la información de evento de arranque del OB 80 (error de tiempo)
MW 200	RET_VAL de la SFC 28 "SET_TINT"
MB 202	Memoria intermedia del resultado binario (bit de estado RB) para SFCs
MW 204	RET_VAL de la SFC 30 "ACT_TINT"
MW 208	RET_VAL de la SFC 31 "QRY_TINT"

SFCs y FCs utilizadas

En el programa de ejemplo se utilizan las siguientes funciones de sistema:

- SFC 28 "SET_TINT" : Ajustar la alarma horaria
- SFC 29 "CAN_TINT" : Cancelar la alarma horaria
- SFC 30 "ACT_TINT" : Disparar la alarma horaria
- SFC 31 "QRY_TINT" : Consultar la alarma horaria
- FC 3 "D_TOD_DT" : Reunir DATE y TIME_OF_DAY para formar DT

A.5.3.2 FC 12

Area de declaración

En el área de declaración de la FC 12 se declaran las siguientes variables temporales del bloque:

Nombre de la variable	Tipo de datos	Declaración	Comentario
IN_HORA	TIME_OF_DAY	TEMP	Tiempo de arranque predeterminado
IN_FECHA	DATE	TEMP	Fecha de arranque predeterminada
OUT_HORA_FECHA	DATE_AND_TIME	TEMP	Cambio fecha/hora de arranque
OK_MARCA	BOOL	TEMP	Activar habilitación de alarma horaria

Area de instrucciones en AWL

Introduzca el siguiente programa de ejemplo AWL en el área de instrucciones de la FC 12:

AWL (FC 12)	Explicación
Segmento 1: CALL SFC 31 NUM OB := 10 RET_VAL:= MW 208 STATUS := MW 16	SFC QRY_TINT Consultar ESTADO (STATUS) de alarmas horarias.
Segmento 2: UN A 4.0 SPB lun L D#1995-1-27 T #IN_FECHA L TOD#20:0:0.0 T #IN_HORA SPA wnd1 mont: L D#1995-1-23 T #IN_FECHA L TOD#5:0:0.0 T #IN_HORA wnd1: NOP 0	Predeterminar el momento de arranque dependiendo de A 4.0 (en la variable #IN_DATUM y #IN_UHRZEIT) La fecha de arranque es un viernes. La fecha de arranque es un lunes.

AWL (FC 12)	Explicación
Segmento 3: CALL FC 3 IN1 := #IN_FECHA IN2 := #IN_HORA RET_VAL := #OUT_HORA FECHA	Cambiar formato de DATE y TIME_OF_DAY a DATE_AND_TIME (ajustar para alarma horaria)
Segmento 4: U E 0.0 UN M 17.2 U M 17.4 = #OK_MARCA	¿Cumplidos todos los requisitos para ajustar la alarma horaria? (entrada activada para habilitación y alarma horaria no activa y OB de alarma horaria cargado)
Segmento 5: U #OK_MARCA SPBNB m001 CALL SFC 28 NUM OB_ := 10 SDT := #OUT_HORA FECHA PERIOD := W#16#1201 RET_VAL := MW 200 m001 U RB = M 202.3	En caso afirmativo, ajustar la alarma horaria ...
Segmento 6: U #OK_MARCA SPBNB m002 CALL SFC 30 NUM OB := 10 RET_VAL := MW 204 m002 U RB = M 202.4	...y activar la alarma horaria.
Segmento 7: U E 0.1 SPBNB m003 CALL SFC 29 NUM OB := 10 RET_VAL := MW 210 m003 U RB = M 202.5	Cancelar la alarma horaria si está activada la entrada para cancelar alarmas horarias.

A.5.3.3 OB 10

Area de declaración

A diferencia del área de declaración predeterminada para el OB 10, se declaran las siguientes variables temporales del bloque:

- Estructura para toda la información de evento de arranque (STARTINFO)
- Dentro de la estructura STARTINFO, una estructura para la hora (T_STMP)
- Otras variables temporales del bloque WTAG, IN_DATUM, IN_UHRZEIT y OUT_UHRZEIT_DATUM

Nombre de la variable	Tipo de datos	Declaración	Comentario
STARTINFO	STRUCT	TEMP	Toda la información de evento de arranque del OB 10 declarada como estructura
E_ID	WORD	TEMP	ID del evento:
PRIORIDAD	BYTE	TEMP	Prioridad
NUM_OB	BYTE	TEMP	Nº de OB
RESERVED_1	BYTE	TEMP	Reservado
RESERVED_2	BYTE	TEMP	Reservado
PERIOD	WORD	TEMP	Periodicidad de la alarma horaria
RESERVED_3	DWORD	TEMP	Reservado
T_STMP	STRUCT	TEMP	Estructura para las indicaciones de hora
AÑO	BYTE	TEMP	
MES	BYTE	TEMP	
DIA	BYTE	TEMP	
HORA	BYTE	TEMP	
MINUTOS	BYTE	TEMP	
SEGUNDOS	BYTE	TEMP	
MSEC_DIA	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	
DIA	INT	TEMP	Día de la semana
IN_FECHA	DATE	TEMP	Variable de entrada para FC 3 (conversión del formato de tiempo)
IN_HORA	TIME_OF_DAY	TEMP	Variable de entrada para FC 3 (conversión del formato de tiempo)
OUT_HORA_FECHA	DATE_AND_TIME	TEMP	Variable de salida para FC 3 y variable de entrada para SFC 28

Area de instrucciones en AWL

Introduzca el siguiente programa de ejemplo AWL en el área de instrucciones del OB 10:

AWL (OB 10)	Explicación
Segmento 1:	
L #STARTINFO.T_STMP.MSEG_DIA	Seleccionar día de la semana
L W#16#F	
UW	
T #DIA	y memorizar.
Segmento 2:	
L #DIA	Si el día de la semana no es un lunes,
L 2	predeterminar el lunes, 5.00 horas como
<>I	siguiente fecha y hora de arranque y
SPB lun	desactivar A 4.0.
Segmento 3:	
L D#1995-1-27	
T #IN_FECHA	De otro modo, es decir, si el día de la
L TOD#20:0:0.0	semana es un lunes, predeterminar el
T #IN_HORA	viernes, 20.00 horas como siguiente fecha
SET	y hora de arranque y activar la salida A
= A 4.0	4.0.
SPA wndl	
mont: L D#1995-1-23	
T #IN_FECHA	
L TOD#5:0:0.0	
T #IN_HORA	
CLR	
= A 4.0	
wndl: NOP 0	
Segmento 4:	Concluida la predeterminación de la fecha
CALL FC 3	y hora de arranque.
IN1 := #IN_FECHA	Convertir la fecha y hora de arranque
IN2 := #IN_HORA	predeterminada al formato DATE_AND_TIME
RET_VAL := #OUT_HORA_FECHA	(para SFC 28).
Segmento 5:	
CALL SFC 28	
NUM OB := 10	Ajustar la alarma horaria.
SDT := #OUT_HORA_FECHA	
PERIOD := W#16#1201	
RET_VAL := MW 200	
U RB	
= M 202.1	
Segmento 6:	
CALL SFC 30	Activar la alarma horaria.
NUM OB := 10	
RET_VAL := MW 204	
U RB	
= M 202.2	
Segmento 7:	
CALL SFC 20	Transferencia de bloque: Salvar la
SRCBLK := #STARTINFO.T_STMP	indicación de hora de la información de
RET_VAL := MW 206	evento de arranque del OB 10 en el área de
DSTBLK := P#M 100.0 BYTE 8	marcas comprendida entre MB 100 y MB 107.

A.5.3.4 OB 1 y OB 80

Puesto que la información de evento de arranque del OB 1 (programa cíclico) no se evalúa en el presente ejemplo, aquí sólo se representa la información de evento de arranque del OB 80.

Area de instrucciones del OB 1

Introduzca el siguiente programa de ejemplo AWL en el área de instrucciones del OB 1:

AWL (OB 1)	Explicación
CALL FC 12	Llamada de la función FC 12

Area de declaración del OB 80

A diferencia del área de declaración predeterminada para el OB 80, se declaran las siguientes variables temporales del bloque:

- Estructura para toda la información de evento de arranque (STARTINFO)
- Dentro de la estructura STARTINFO, una estructura para la hora (T_STMP)

Nombre de la variable	Tipo de datos	Declaración	Comentario
STARTINFO	STRUCT	TEMP	Toda la información de evento de arranque del OB 80 declarada como estructura
E	WORD	TEMP	ID del evento
P	BYTE	TEMP	Prioridad
O	BYTE	TEMP	Nº de OB
R	BYTE	TEMP	Reservado
R	BYTE	TEMP	Reservado
Z1_INFO	WORD	TEMP	Información adicional sobre el evento que ha causado el error.
Z2_INFO	DWORD	TEMP	Informaciones adicionales sobre el ID del evento, la prioridad y el número de OB del evento de error
T	STRUCT	TEMP	Estructura para las indicaciones de hora
AÑO	BYTE	TEMP	
MES	BYTE	TEMP	
DIA	BYTE	TEMP	
HORA	BYTE	TEMP	
MINUTOS	BYTE	TEMP	
SEGUNDOS	BYTE	TEMP	
MSEG_DIA	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

Area de instrucciones del OB 80

Introduzca el siguiente programa de usuario AWL en el área de instrucciones del OB 80 que es invocado por el sistema operativo al presentarse un error de tiempo:

AWL (OB 80)	Explicación
Segmento 1:	
UN A 4.1	Activar salida A 4.1 al ocurrir un error
S A 4.1	de tiempo.
CALL SFC 20	Transferencia de bloque: Salvar toda la
SRCBLK := #STARTINFO	información de evento de arranque en el
RET_VAL := MW 210	área de marcas comprendida entre MB 110 y
DSTBLK := P#M 110.0 Byte 20	MB 129.

A.5.4 Ejemplo de aplicación de las alarmas de retardo

Estructura de las alarmas de retardo en el programa de usuario

OB 20

OB 1

A.5.4.1 Estructura del programa de usuario "Alarmas de retardo"

Tarea

Cuando se active la entrada E 0.0, la salida A 4.0 se deberá activar 10 segundos después. Cada vez que se active la entrada E 0.0 se deberá iniciar de nuevo el tiempo de retardo.

En calidad de identificador específico de la aplicación, el momento (segundos y milisegundos) en el que comience la alarma de retardo deberá aparecer en la información de evento de arranque del OB de alarma de retardo (OB 20).

Si E 0.1 se activa en esos 10 segundos, el bloque de organización OB 20 no se deberá llamar, es decir que la salida A 4.0 no se deberá activar.

Cuando se active la entrada E 0.2, la salida A 4.0 se deberá desactivar nuevamente.

Aplicación en el programa de usuario

La tabla siguiente muestra las tareas parciales de los bloques utilizados.

Bloque	Tarea parcial
OB 1	Leer la hora actual y preparar para el inicio de la alarma de retardo Dependiendo del cambio de flanco en la entrada E 0.0, iniciar la alarma de retardo Dependiendo del estado de la alarma de retardo y del cambio de flanco en la entrada E 0.1, cancelar la alarma de retardo Dependiendo del estado de la entrada E 0.2, desactivar la salida A 4.0
OB 20	Activar la salida A 4.0 Leer la hora actual y preparar Salvar la información de evento de arranque en el área de marcas

Operandos utilizados

La tabla siguiente muestra los datos globales utilizados. Las variables temporales de los bloques se declaran en el área de declaración del bloque en cuestión.

Operando	Significado
E0.0	Entrada para la habilitación de "Arrancar alarma de retardo"
E0.1	Entrada para cancelar una alarma de retardo
E0.2	Entrada para desactivar la salida A 4.0
A 4.0	Salida activada por el OB de alarma de retardo (OB 20)
MB 1	Utilizada para la marca de flanco y la memoria intermedia del resultado binario (bit de estado RB) para SFCs
MW 4	STATUS (ESTADO) de la alarma de retardo (SFC 34 "QRY_TINT")
MD 10	Segundos y milisegundos en formato decimal codificado en binario de la información de evento de arranque del OB 1
MW 100	RET_VAL de la SFC 32 "SRT_DINT"
MW 102	RET_VAL de la SFC 34 "QRY_DINT"
MW 104	RET_VAL de la SFC 33 "CAN_DINT"
MW 106	RET_VAL de la SFC 20 "BLKMOV"
MB 120 hasta MB 139	Memoria para la información de evento de arranque del OB 20
MD 140	Segundos y milisegundos en formato decimal codificado en binario de la información de evento de arranque del OB 20
MW 144	Segundos y milisegundos en formato decimal codificado en binario de la información de evento de arranque del OB 1; conforme a la información de evento de arranque del OB 20 (identificador específico de usuario SIGN)

SFCs utilizadas

En el programa de usuario "Alarmas de retardo" se utilizan las siguientes funciones de sistema:

- SFC 32 "SRT_DINT" : Arrancar la alarma de retardo
- SFC 33 "CAN_DINT" : Cancelar la alarma de retardo
- SFC 34 "QRY_DINT" : Consultar el estado de una alarma de retardo

A.5.4.2 OB 20

Area de declaración

A diferencia del área de declaración predeterminada para el OB 20, se declaran las siguientes variables temporales del bloque:

- Estructura para toda la información de evento de arranque (STARTINFO)
- Dentro de la estructura STARTINFO, una estructura para la hora (T_STMP)

Nombre de la variable	Tipo de datos	Declaración	Comentario
STARTINFO	STRUCT	TEMP	Información de arranque del OB 20
E_ID	WORD	TEMP	ID del evento
AE_NR	BYTE	TEMP	Nivel de ejecución
OB_NR	BYTE	TEMP	Nº de OB
DK1	BYTE	TEMP	Identificador de datos 1
DK2	BYTE	TEMP	Identificador de datos 2
SIGN	WORD	TEMP	Identificador específico de usuario
DTIME	TIME	TEMP	Tiempo con el que se arranca la alarma de retardo
T_STMP	STRUCT	TEMP	Estructura para las indicaciones de hora (indicación de hora y fecha)
AÑO	BYTE	TEMP	
MES	BYTE	TEMP	
DIA	BYTE	TEMP	
HORA	BYTE	TEMP	
MINUTOS	BYTE	TEMP	
SEGUNDOS	BYTE	TEMP	
MSEC_DIA	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

Area de instrucciones

Introduzca el siguiente programa de ejemplo AWL en el área de instrucciones del OB 20:

AWL (OB 20)	Explicación
Segmento 1: SET = A 4.0	Activar incondicionalmente la salida A 4.0
Segmento 2: L AW 4 T PAW 4	Actualizar inmediatamente la palabra de salida
Segmento 3: L #STARTINFO.T_STMP.SEGUNDOS T MW 140 L #STARTINFO.T_STMP.MSEC_DIA T MW 142 L MD 140 SRD 4 T MD 140	Leer los segundos de la información de evento de arranque Leer los milisegundos y el día de la semana de la información de evento de arranque
Segmento 4: L #STARTINFO.SIGN T MW 144	Eliminar el día de la semana y regrabar los milisegundos (están ahora en formato decimal codificado en binario en MW 142).
Segmento 5: CALL SFC 20 SRCBLK := STARTINFO RET_VAL := MW 106 DSTBLK := P#M 120.0 BYTE 20	Leer el momento de arranque de la alarma de retardo (=llamada de SFC 32) de la información de evento de arranque Copiar la información de evento de arranque en el área de marcas (MB 120 hasta MB 139)

A.5.4.3 OB 1

Area de declaración

A diferencia del área de declaración predeterminada para el OB 1, se declaran las siguientes variables temporales del bloque:

- Estructura para toda la información de evento de arranque (STARTINFO)
- Dentro de la estructura STARTINFO, una estructura para la hora (T_STMP)

Nombre de la variable	Tipo de datos	Declaración	Comentario
STARTINFO	STRUCT	TEMP	Información de arranque del OB 1
E_ID	WORD	TEMP	ID del evento
AE_NR	BYTE	TEMP	Nivel de ejecución
OB_NR	BYTE	TEMP	Nº de OB
DK 1	BYTE	TEMP	Identificador de datos 1
DK 2	BYTE	TEMP	Identificador de datos 2
CIC_ACT	INT	TEMP	Tiempo de ciclo actual
CIC_MIN	INT	TEMP	Tiempo de ciclo mínimo
CIC_MAX	INT	TEMP	Tiempo de ciclo máximo
T_STMP	STRUCT	TEMP	Estructura para las indicaciones de hora (indicación de hora y fecha)
AÑO	BYTE	TEMP	
MES	BYTE	TEMP	
DIA	BYTE	TEMP	
HORA	BYTE	TEMP	
MINUTOS	BYTE	TEMP	
SEGUNDOS	BYTE	TEMP	
MSEC_DIA	WORD	TEMP	
	END_STRUCT	TEMP	
	END_STRUCT	TEMP	

Area de instrucciones

Introduzca el siguiente programa de ejemplo AWL en el área de instrucciones del OB 1:

AWL (OB 1)	Explicación
Segmento 1:	
L #STARTINFO.T_STMP.SEGUNDOS	Leer los segundos de la información del evento de arranque
T MW 10	
L #STARTINFO.T_STMP.MSEC_DIA	Leer los milisegundos y el día de la semana de la información del evento de arranque
T MW 12	
L MD 10	
SRD 4	Eliminar el día de la semana y
T MD 10	regrabar los milisegundos (están ahora en formato decimal codificado en binario en MW 12).
Segmento 2:	
U E 0.0	¿Flanco positivo en la entrada E 0.0?
FP M 1.0	
= M 1.1	
Segmento 3:	
U M 1.1	
SPBNB m001	En caso afirmativo, arrancar la alarma de retardo (momento de arranque de la alarma de retardo asignado al parámetro SIGN)
CALL SFC 32	
OB_NR := 20	
DTME := T#10S	
SIGN := MW 12	
RET_VAL:= MW 100	
m001: NOP 0	
Segmento 4:	
CALL SFC 34	Consultar el estado de la alarma de retardo (SFC QRY_DINT)
OB_NR := 20	
RET_VAL:= MW 102	
STATUS := MW 4	
Segmento 5:	
U E 0.1	¿Flanco positivo en la entrada E 0.1?
FP M 1.3	
= M 1.4	
Segmento 6:	
U M 1.4	... ¿y alarma de retardo activada (bit 2 del estado (STATUS) de la alarma de retardo)?
U M 5.2	
SPBNB m002	Cancelar entonces la alarma de retardo
CALL SFC 33	
OB_NR := 20	
RET_VAL:= MW 104	
m002: NOP 0	
U E 0.2	Desactivar la salida A 4.0 junto con la entrada E 0.2
R A 4.0	

A.5.4.4 Ejemplo de enmascarar y desenmascarar eventos de errores síncronos

El ejemplo siguiente de un programa de usuario muestra cómo enmascarar y desenmascarar eventos de errores síncronos. Con la SFC 36 "MSK_FLT" se enmascara los siguientes errores en la máscara de errores de programación:

- Errores de infracción de área al leer
- Errores de infracción de área al escribir

Al llamarse la SFC 36 "MSK_FLT" por segunda vez se enmascara además un error de acceso:

- Error de acceso a la periferia al escribir.

Con la SFC 38 "READ_ERR" se consultan los eventos de errores síncronos enmascarados. El "error de acceso a la periferia al escribir" se desenmascara nuevamente con la SFC 37 "DMSK_FLT".

Instrucción

A continuación se muestra el OB 1 programado en el ejemplo para el programa de usuario en AWL.

AWL (Segmento 1)	Explicación
UN M 255.0	Marca no remanente M 255.0 (sólo en la primera ejecución=0)
SPBNB m001	
CALL SFC 36	SFC 36 MSK_FLT (enmascarar eventos de errores síncronos)
PRGFLT_SET_MASK :=DW#16#C	Bit2=Bit3=1 (BLFL y BLFS se enmascaran) todos los bits=0 (no se enmascara ningún error de acceso)
ACCFLT_SET_MASK :=DW#16#0	Valor de retorno
RET_VAL :=MW 100	Salida de la máscara de error de programación actual en MD 10
PRGFLT_MASKED :=MD 10	Salida de la máscara de error de acceso actual en MD 14
ACCFLT_MASKED :=MD 14	
m001: U RB	Activar M255.0 cuando se haya enmascarado correctamente
S M 255.0	

AWL (Segmento 2)	Explicación
CALL SFC 36	SFC 36 MSK_FLT (enmascarar eventos de errores síncronos)
PRGFLT_SET_MASK :=DW#16#0	Todos los bits=0 (no se enmascara ningún otro error de programación)
ACCFLT_SET_MASK :=DW#16#8	Bit3=1 (se enmascaran los errores de acceso de escritura)
RET_VAL :=MW 102	Valor de retorno
PRGFLT_MASKED :=MD 20	Salida de la máscara de error de programación actual en MD 20
ACCFLT_MASKED :=MD 24	Salida de la máscara de error de acceso actual en MD 24

AWL (Segmento 3)	Explicación
UN M 27.3	Fin del bloque cuando el error de acceso de escritura (bit3 en ACCFLT_MASKED) no esté enmascarado
BEB	

AWL (Segmento 4)	Explicación
------------------	-------------

L	B#16#0	
T	PAB 16	Acceso de escritura (con valor 0) a PAB 16

AWL (Segmento 5)		Explicación
CALL	SFC 38	SFC 38 READ_ERR (consultar eventos de errores síncronos)
PRGFLT_QUERY	:=DW#16#0	Todos los bits=0 (no se enmascara ningún error de programación)
ACCFLT_QUERY	:=DW#16#8	Bit3=1 (se consulta el error de acceso de escritura)
RET_VAL	:=MW 104	Valor de retorno
PRGFLT_CLR	:=MD 30	Salida de la máscara de error de programación actual en MD 30
ACCFLT_CLR	:=MD 34	Salida de la máscara de error de acceso actual en MD 34
U	RB	No se ha presentado ningún error y se ha detectado un error de acceso de escritura
U	M 37.3	
NOT		Invertir RLO
=	M 0.0	M 0.0=1, si existe PAB 16

AWL (Segmento 6)		Explicación
L	B#16#0	
T	PAB 17	Acceso de escritura (con valor 0) a PAB 17

AWL (Segmento 7)		Explicación
CALL	SFC 38	SFC 38 READ_ERR (consultar eventos de errores síncronos)
PRGFLT_QUERY	:=DW#16#0	Todos los bits=0 (no se enmascara ningún error de programación)
ACCFLT_QUERY	:=DW#16#8	Bit3=1 (se consulta el error de acceso de escritura)
RET_VAL	:=MW 104	Valor de retorno
PRGFLT_CLR	:=MD 30	Salida de la máscara de error de programación actual en MD 30
ACCFLT_CLR	:=MD 34	Salida de la máscara de error de acceso actual en MD 34
U	RB	No se ha presentado ningún error y se ha detectado un error de acceso de escritura
U	M 37.3	
NOT		Invertir RLO
=	M 0.1	M 0.1=1, si existe PAB 17

AWL (Segmento 8)		Explicación
L	B#16#0	
T	PAB 18	Acceso de escritura (con valor 0) a PAB 18

AWL (Segmento 9)		Explicación
CALL	SFC 38	SFC 38 READ_ERR (consultar eventos de errores síncronos)
PRGFLT_QUERY	:=DW#16#0	Todos los bits=0 (no se enmascara ningún error de programación)
ACCFLT_QUERY	:=DW#16#8	Bit3=1 (se consulta el error de acceso de escritura)
RET_VAL	:=MW 104	Valor de retorno
PRGFLT_CLR	:=MD 30	Salida de la máscara de error de programación actual en MD 30
ACCFLT_CLR	:=MD 34	Salida de la máscara de error de acceso actual en MD 34
U	RB	No se ha presentado ningún error y se ha detectado un error de acceso de escritura
U	M 37.3	

NOT		Invertir RLO
=	M 0.2	M 0.2=1, si existe PAB 18
AWL (Segmento 10)		Explicación
L	B#16#0	
T	PAB 19	Acceso de escritura (con valor 0) a PAB 19
AWL (Segmento 11)		Explicación
CALL	SFC 38	SFC 38 READ_ERR (consultar eventos de errores síncronos)
	PRGFLT_QUERY :=DW#16#0	Todos los bits=0 (no se enmascara ningún error de programación)
	ACCFLT_QUERY :=DW#16#8	Bit3=1 (se consulta el error de acceso de escritura)
	RET_VAL :=MW 104	Valor de retorno
	PRGFLT_CLR :=MD 30	Salida de la máscara de error de programación actual en MD 30
	ACCFLT_CLR :=MD 34	Salida de la máscara de error de acceso actual en MD 34
U	RB	No se ha presentado ningún error y se ha detectado un error de acceso de escritura
U	M 37.3	
NOT		Invertir RLO
=	M 0.3	M 0.3=1, si existe PAB 19
AWL (Segmento 12)		Explicación
CALL	SFC 37	SFC 37 DMSK_FLT (desenmascarar eventos de errores síncronos)
	PRGFLT_RESET_MASK :=DW#16#0	todos los bits=0 (no se desenmascara ningún error de programación)
	ACCFLT_RESET_MASK :=DW#16#8	Bit3=1 (se desenmascara el error de acceso de escritura)
	RET_VAL :=MW 102	Valor de retorno
	PRGFLT_MASKED :=MD 20	Salida de la máscara de error de programación actual en MD 20
	ACCFLT_MASKED :=MD 24	Salida de la máscara de error de acceso actual en MD 24
AWL (Segmento 13)		Explicación
U	M 27.3	Fin del bloque cuando el error de acceso de escritura (bit3 en ACCFLT_MASKED) no esté desenmascarado
	BEB	
AWL (Segmento 14)		Explicación
U	M 0.0	
SPBNB	m002	
L	EB 0	Transferir EB 0 a PAB 16, si existe
T	PAB 16	
m002:	NOP 0	
AWL (Segmento 15)		Explicación
U	M 0.1	
SPBNB	m003	
L	EB 1	Transferir EB 1 a PAB 17, si existe
T	PAB 17	
m003:	NOP 0	
AWL (Segmento 16)		Explicación
U	M 0.2	
SPBNB	m004	
L	EB 2	Transferir EB 2 a PAB 18, si existe
T	PAB 18	

m004: NOP 0

AWL (Segmento 17)	Explicación
U M 0.3	
SPBNB m005	
L EB 3	Transferir EB 3 a PAB 19, si existe
T PAB 19	
m005: NOP 0	

A.5.4.5 Ejemplo de bloquear y habilitar eventos de alarma y de error asíncrono (SFC 39 y 40)

En el siguiente ejemplo de programa de usuario se utiliza una parte del programa que no puede ser interrumpida por alarmas. Para esta parte del programa se inhiben con la SFC 39 "DIS_IRT" las llamadas del OB 35 (alarma horaria) que se habilitan nuevamente con la SFC 40 "EN_IRT".

En el OB 1 se llaman la SFC 39 y la SFC 40:

AWL (OB 1)	Explicación
U M 0.0	Parte del programa que se puede
S M 90.1	interrumpir sin problema alguno:
U M 0.1	
S M 90.0	
:	
:	Parte del programa que no se puede
	interrumpir:
CALL SFC 39	Bloquear y rechazar alarmas
MODE :=B#16#2	Modo 2: bloquear OBs de alarma
OB_NR :=35	individuales
RET_VAL :=MW 100	Bloquear el OB35
:	
:	
L PEW 100	
T MW 200	
L MW 90	
T MW 92	
:	
:	
CALL SFC 40	Habilitar alarmas
MODE :=B#16#2	Modo 2: habilitar OBs de alarma
OB_NR :=35	individuales
RET_VAL :=MW 102	Habilitar el OB35
U M 10.0	Parte del programa que se puede
S M 190.1	interrumpir sin problema alguno:
U M 10.1	
S M 190.0	
:	
:	

A.5.4.6 Ejemplo del tratamiento retardado de eventos de alarma y de error asíncrono (SFC 41 y 42)

En el siguiente ejemplo de programa de usuario se utiliza una parte del programa que no puede ser interrumpida por alarmas. En esta parte del programa, las alarmas se retardan con la SFC 41 "DIS_AIRT", habilitándose luego nuevamente con la SFC 42 "EN_AIRT".

En el OB 1 se llaman la SFC 41 y la SFC 42:

AWL (OB 1)	Explicación
U M 0.0	Parte del programa que se puede interrumpir sin problema alguno:
S M 90.1	
U M 0.1	
S M 90.0	
:	Parte del programa que no se puede interrumpir:
:	
CALL SFC 41	Bloquear y retardar alarmas
RET_VAL :=MW 100	
L PEW 100	
T MW 200	
L MW 90	
T MW 92	
:	
:	
:	
CALL SFC 42	Habilitar alarma
RET_VAL :=MW 102	
L MW 100	En el valor de retorno se indica el número de bloqueos de alarmas introducidos
DEC 1	
L MW 102	En el valor de retorno se indica el número de bloqueos de alarmas introducidos
<>I	Tras habilitarse la alarmas, el número deberá tener el mismo valor
SPB err	que antes de bloquearse las mismas (aquí "0")
U M 10.0	Parte del programa que se puede interrumpir sin problema alguno:
S M 190.1	
U M 10.1	
S M 190.0	
:	
:	
BEA	
err: L MW 102	Se indica el número de bloqueos de alarmas introducidos
T AW 12	

A.6 Acceso a áreas de datos del proceso y de la periferia

A.6.1 Acceso al área de datos de proceso

La CPU puede acceder, bien indirectamente a través de la imagen del proceso o bien directamente a través del bus posterior/de periferia, a las entradas y salidas de los módulos de entradas/salidas digitales centralizados y descentralizados.

La CPU accede directamente a las entradas y salidas de módulos de entradas/salidas analógicas centralizados y descentralizados a través del bus posterior/de periferia. Sin embargo, también tiene la posibilidad de depositar las direcciones de los módulos analógicos en el área de la imagen del proceso.

Direccionamiento de módulos

La asignación entre direcciones que se utilizan en el programa de usuario y los módulos se lleva a cabo mediante la configuración de los módulos con STEP 7

- En el caso de la periferia centralizada: la disposición del bastidor y la asignación de los módulos a slots tiene lugar en la tabla de configuración
- En el caso de la periferia descentralizada (PROFIBUS DP o PROFINET IO): disposición de los esclavos DP o dispositivos IO otorgando la dirección PROFIBUS o el nombre de dispositivo y asignando los módulos a slots.

La configuración de los módulos sustituye el ajuste de direcciones de los diferentes módulos a través de interruptores. Como resultado de la configuración, la CPU recibe de la PG los datos que identifican a los módulos asignados.

Direccionamiento de periferia

Las entradas y las salidas disponen de una área propia de direcciones. Por consiguiente, la dirección de una área de periferia debe contener, además de la indicación de byte o palabra, el identificador E para las entradas y A para las salidas.

La tabla siguiente muestra las áreas de direcciones de periferia disponibles.

Area de operandos	Acceso a través de unidades del tamaño siguiente:	Notación S7
Area de periferia: entradas	Byte de entrada de periferia Palabra de entrada de periferia Palabra doble de entrada de periferia	PEB PEW PED
Area de periferia: salidas	Byte de salida de periferia Palabra de salida de periferia Palabra doble de salida de periferia	PAB PAW PAD

Los manuales siguientes indican qué áreas de direcciones son posibles en cada módulo:

- CPU 31xC y CPU 31x, Datos técnicos
- Sistema de automatización S7-400, Datos de las CPU

Dirección inicial de módulos

La dirección inicial de un módulo es la menor dirección de byte de dicho módulo. Representa la dirección inicial del área de datos útiles del módulo y se utiliza en muchos casos como representante de todo el módulo.

Se introduce en la información de arranque del correspondiente bloque de organización, p. ej., en el caso de alarmas de proceso, alarmas de diagnóstico, alarmas de presencia de módulo y errores de alimentación, identificando así al módulo que ha indicado la alarma.

A.6.2 Acceso al área de datos de periferia

El área de datos de periferia se subdivide en:

- datos útiles y
- datos de diagnóstico y de parámetros.

Ambas disponen de una área de entrada (sólo acceso en lectura) y una área de salida (sólo acceso en escritura).

Datos útiles

A los datos útiles se accede a través de la dirección de byte (para módulos de señales digitales) o de la dirección de palabra (para módulos de señales analógicas) del área de entrada o salida. A los datos útiles se puede acceder a través de instrucciones de carga y transferencia, funciones de comunicaciones (accesos M+V) o a través de transferencia de imágenes de proceso. Datos útiles pueden ser:

- señales de entrada/salida digitales y analógicas de módulos de señales,
- informaciones de control y estado de módulos de función e
- informaciones para conexiones punto a punto y por bus vía módulos de comunicaciones (sólo S7-300).

Al transferirse datos útiles, se puede alcanzar una coherencia de datos de 4 bytes como máximo (con excepción de los esclavos estándar DP, consulte también "Ajustar el comportamiento de servicio"). Si se utiliza la instrucción "transferir palabra doble", se transmiten los 4 bytes correspondientes y sin modificar (coherentes). Si se utilizan cuatro instrucciones individuales "transferir byte de entrada", en un límite de instrucción podría arrancarse un OB de alarma de proceso, el cual transmite los datos a la misma dirección y, por consiguiente, modifica el contenido de los cuatro bytes originales.

Datos de diagnóstico y de parámetros

Los datos de diagnóstico y de parámetros de un módulo no se pueden direccionar individualmente, sino reunidos en registros completos. Se transfieren normalmente de forma coherente.

Los datos de diagnóstico y de parámetros se direccionan a través de la dirección inicial del módulo correspondiente y del número de registro. Los registros se subdividen en registros de entrada y de salida. Los registros de entrada sólo se pueden leer; en los de salida sólo se puede escribir. A los registros se puede acceder a través de las funciones del sistema o de funciones de comunicaciones (manejo y visualización, M+V). La tabla siguiente muestra la asignación de registros a los datos de diagnóstico y de parámetros.

Datos	Descripción
Datos de diagnóstico	En módulos diagnosticables, al leer los registros 0 y 1 se obtienen los datos de diagnóstico del módulo.
Datos de parámetros	En módulos parametrizables, al escribir en los registros 0 y 1 se transfieren los parámetros del módulo.

Acceso a los registros

Las informaciones contenidas en los registros de un módulo se pueden utilizar para reparametrizar módulos parametrizables y leer las informaciones de diagnóstico de módulos diagnosticables.

La tabla siguiente muestra con qué funciones del sistema se puede acceder a los registros.

SFC	Aplicación
Parametrizar módulos	
SFC 55 WR_PARM	Transmitir los parámetros modificables (registro 1) al módulo de señales direccionado.
SFC 56 WR_DPARM	Transmitir los parámetros desde los SDB 100 a 129 al módulo de señales direccionado.
SFC 57 PARM_MOD	Transmitir los parámetros desde los SDB 100 a 129 al módulo de señales direccionado.
SFC 58 WR_REC	Transmitir un registro cualquiera al módulo de señales direccionado.
Leer informaciones de diagnóstico	
SFC 59 RD_REC	Leer los datos de diagnóstico.

Nota

Si un esclavo DPV1 está configurado mediante un archivo GSD (GSD a partir de rev. 3) y la interfaz DP del maestro está ajustada a "**Compatible S7**", los módulos E/A no podrán leer ni escribir registros de datos en el programa de usuario con SFC 58/59 ó SFB 53/52. En ese caso el maestro DP direcciona el slot incorrecto (slot configurado +3).

Remedio: Cambiar el ajuste de la interfaz del maestro DP a "DPV1".

Direccionamiento de módulos S5

Tiene la posibilidad de:

- acoplar aparatos de ampliación de SIMATIC S5 a un S7-400 a través del módulo de interconexión (interfase) IM 463-2 e
- insertar algunos módulos S5, metidos en cápsulas de adaptación, en los bastidores centrales del S7-400.

Para obtener informaciones acerca de cómo direccionar los módulos S5 en SIMATIC S7, consulte el manual de instalación "Sistemas de automatización S7-400, M7-400, Configuración e instalación" o la descripción adjunta de la cápsula de adaptación.

A.7 Ajustar el comportamiento del sistema

Este capítulo explica cómo se puede actuar sobre aquellas propiedades de los sistemas de automatización S7-300 y S7-400 que no están predeterminadas ajustando los parámetros del sistema o utilizando las funciones del sistema SFC.

Para información más detallada sobre los parámetros de los módulos, consultar la Ayuda en pantalla STEP 7, así como los siguientes manuales:

- Manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU"
- Manual de referencia "Sistemas de automatización S7-300, M7-300, Datos de los módulos"
- Manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos"

El manual de referencia "Software de sistema para S7-300/400, funciones estándar y funciones de sistema" contiene descripciones detalladas de las SFCs.

Direccionamiento de esclavos DP estándar

Si en esclavos DP estándar se requiere transferir o recibir datos mayores de 4 bytes, es necesario utilizar SFCs especiales para la transferencia de dichos datos.

En el caso de CPUs que admiten el intercambio de datos coherentes (> 4 bytes) en el área de periferia los SFC 14/15 no son necesarios (consulte Leer y escribir datos coherentes de forma descentralizada).

SFC	Aplicación
Parametrizar módulos	
SFC 15 DPWR_DAT	Transmitir un dato cualquiera al módulo de señales direccionado.
Leer informaciones de diagnóstico	
SFC 13 DPNRM_DG	Leer los datos de diagnóstico (operación de lectura asíncrona).
SFC 14 DPRD_DAT	Leer datos coherentes (longitud 3 o mayor de 4 bytes).

Cuando llega un telegrama de diagnóstico DP se envía a la CPU una alarma de diagnóstico con 4 bytes de datos de diagnóstico. Estos 4 bytes se pueden leer con la SFC 13 DPNRM_DG.

A.7.1 Modificar el comportamiento y las propiedades de los módulos

Ajustes por defecto

- Todos los módulos parametrizables del sistema de automatización S7 se suministran con ajustes por defecto apropiados para aplicaciones estándar. Estos valores por defecto permiten utilizar los módulos sin necesidad de ajustes adicionales. Los valores estándar se indican en las descripciones de los módulos contenidas en los siguientes manuales:
- Manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU"
- Manual de referencia "Sistemas de automatización S7-300, M7-300, Datos de los módulos"
- Manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos"

¿Qué módulos pueden parametrizarse?

También es posible parametrizar el comportamiento y las propiedades de los módulos para ajustarlos a los requerimientos y las particularidades de la instalación. Módulos parametrizables son CPUs, FMs y CPs, así como módulos de entradas/salidas analógicas y módulos de entradas digitales.

Se dispone de módulos parametrizables con y sin respaldo de memoria.

Los módulos sin respaldo se deben abastecer con los datos correspondientes tras cada corte de tensión. Los parámetros de dichos módulos están memorizados en el área de memoria remanente de la CPU (parametrización indirecta a través de CPU).

Ajustar y cargar los parámetros

Los parámetros de los módulos se ajustan con STEP 7. Al guardar los parámetros, STEP 7 crea el objeto "Bloques de datos del sistema" que se carga con el programa de usuario en la CPU, transfiriéndose desde allí durante el arranque a los módulos correspondientes.

¿Qué se puede parametrizar?

Los parámetros de módulos se subdividen en bloques de parámetros. En los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos" se indican los bloques de parámetros disponibles en cada una de las CPUs.

Ejemplos de bloques de parámetros:

- comportamiento en arranque
- ciclo
- MPI
- diagnóstico
- remanencia
- marcas de ciclo
- tratamiento de alarmas
- periferia integrada (sólo para S7-300)
- nivel de protección
- datos locales
- reloj de tiempo real
- error asíncrono

Parametrizar con SFCs

Además de la parametrización con STEP 7, también es posible modificar desde el programa S7 los parámetros de los módulos a través de las funciones del sistema. La tabla siguiente muestra con qué SFCs se pueden modificar qué parámetros de los módulos.

SFC	Aplicación
SFC 55 WR_PARM	Transmitir los parámetros modificables (registro 1) al módulo de señales direccionado.
SFC 56 WR_DPARAM	Transmitir los parámetros desde los SDB correspondientes al módulo de señales direccionado.
SFC 57 PARM_MOD	Transmitir los parámetros desde los SDB correspondientes al módulo de señales direccionado.
SFC 58 WR_REC	Transmitir un registro cualquiera al módulo de señales direccionado.

Las funciones de sistema se describen detalladamente en el manual de referencia "Software de sistema para S7-300/400, funciones estándar y funciones de sistema".

Los manuales siguientes indican qué parámetros de módulos se pueden modificar dinámicamente.

- Manual "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU"
- Manual de referencia "Sistemas de automatización S7-300, M7-300, Datos de los módulos"
- Manual de referencia "Sistemas de automatización S7-400, M7-400, Datos de los módulos"

A.7.2 Actualización offline del firmware (del sistema operativo) de módulos y tarjetas

A continuación se describe cómo transferir una versión nueva de firmware (= nueva versión del sistema operativo) mediante una Memory Card a un módulo, como p. ej. una CPU.

Para la actualización es necesario ejecutar dos pasos:

1. Crear una "Memory Card de actualización" (transferir los archivos de actualización a una Memory Card) con la PG o el PC y un dispositivo de programación de tarjetas externo).
2. Efectuar la actualización del sistema operativo sirviéndose de la "Memory Card de actualización" en la CPU.

Requisitos

- La Memory Card debe disponer de suficiente capacidad de memoria. Para más información al respecto, consulte las páginas de descargas del Customer-Support. Aquí encontrará asimismo los archivos de actualización.
- PG o PC con dispositivo para programar Memory Cards.

Transferir los archivos de actualización a una Memory Card

1. Cree un directorio nuevo en el Explorador de Windows.
2. Transfiera el archivo de actualización deseado a este directorio y descomprímalo. En este directorio se encontrará a continuación el archivo UPD.
3. Inserte la Memory Card S7 en la PG o en el dispositivo de programación.
4. Borre la Memory Card (comando de menú **Memory Card S7 > Borrar** en el Administrador SIMATIC)
5. Elija en el Administrador SIMATIC el comando de menú **Sistema de destino > Actualizar sistema operativo**.
6. Elija en el cuadro de diálogo que aparece a continuación el directorio que contiene el archivo UDP.
7. Haga doble clic en el archivo UPD.
De este modo se iniciará el proceso de programación. Cuando finalice dicho proceso aparecerá el mensaje: La versión actualizada del firmware para el módulo ... ha sido transferida con éxito a la Memory Card S7".

Enchufe la Memory Card en el sistema de destino.

Procedimiento para actualizar el sistema operativo:

1. Inserte la Memory Card con la actualización en el sistema de destino (es decir, en la CPU)
2. Desconecte la fuente de alimentación (PS) de la CPU
3. Inserte en la CPU la Memory Card preparada con la actualización.
4. Conecte la fuente de alimentación de la CPU.
El sistema operativo se transfiere de la Memory Card S7 a la FLASH-EPROM interna. Durante la transferencia se iluminan todos los indicadores LED de la CPU.
5. La actualización termina aprox. en 2 minutos, y se reconoce porque en la CPU parpadea el LED "STOP" a baja frecuencia (petición del sistema de borrado total)
6. Desconecte la fuente de alimentación e inserte, si la hubiere, la Memory Card prevista para el funcionamiento.
7. Vuelva a conectar la fuente de alimentación. La CPU ejecutará automáticamente un borrado total y estará lista para funcionar.

A.7.3 Aplicación de las funciones de reloj

Todas las CPUs S7-300 y S7-400 están dotadas con un reloj (reloj de tiempo real o reloj software). En el sistema de automatización el reloj puede operar tanto como maestro como de esclavo con sincronización externa. Permite utilizar alarmas horarias y contadores de horas de funcionamiento.

Formato de hora

El reloj muestra siempre la hora (resolución mínima 1 s) y la fecha con el día de la semana. En algunas CPUs también se pueden indicar los milisegundos (consulte los manuales "Sistema de automatización S7-300, Configuración, instalación y datos de las CPU" y "Sistemas de automatización S7-400, M7-400, Datos de los módulos").

Ajustar y leer la hora

La hora y la fecha del reloj de la CPU puede ser ajustada bien llamando la SFC 0 SET_CLK desde el programa de usuario o bien desde la PG mediante el comando de menú, arrancándose así el reloj. Con la SFC 1 READ_CLK o por comando de menú desde la PG se puede leer la fecha y hora actual de la CPU.

Nota

Para evitar diferencias en la indicación de la hora en sistemas HMI, ajuste el **horario de invierno** en la CPU.

Parametrizar el reloj

Si una red dispone de más de un módulo con reloj, es necesario parametrizar con STEP 7 qué CPU debe funcionar como maestra para la sincronización de la hora y cuál como esclavo. Con la parametrización se define también si la sincronización debe tener lugar a través del bus de comunicación o a través del puerto MPI y con qué intervalos debe efectuarse la sincronización automática.

Sincronizar la hora

Para garantizar que coincida la hora de todos los módulos existentes en la red, los relojes esclavos son sincronizados por el programa del sistema en intervalos regulares (parametrizables). Con la función del sistema SFC 48 SFC_RTCB se puede transferir la fecha y la hora del reloj maestro a los relojes esclavos.

Utilizar un contador de horas de funcionamiento

Un contador de horas de funcionamiento totaliza los períodos de conexión de un equipo o la duración de servicio de la CPU como suma de las horas de funcionamiento.

El contador de horas de funcionamiento se para durante el estado operativo STOP. Su valor se conserva incluso tras el borrado total. En caso de re arranque completo (en caliente), el contador de horas de funcionamiento debe ser arrancado nuevamente por el programa de usuario y, en caso de re arranque, continúa operando si ha sido arrancado previamente.

La SFC 2 SET_RTM permite ajustar el contador de horas de funcionamiento a un valor inicial. Con la SFC 3 CTRL_RTM se puede arrancar o parar el contador de horas de funcionamiento. La SFC 4 READ_RTM permite leer la cantidad actual de horas de funcionamiento y el estado del contador ("parado" o "contando").

Una CPU puede tener hasta 8 contadores de horas de funcionamiento. La numeración comienza a partir de 0.

A.7.4 Uso de marcas de ciclo y temporizadores

Marcas de ciclo

Una marca de ciclo es una marca que modifica su estado binario periódicamente con un ciclo de trabajo de 1:1. Parametrizando la marca de ciclo con STEP 7 se puede definir qué byte de marcas de la CPU se utiliza como byte de marcas de ciclo.

Utilidad

Las marcas de ciclo se pueden utilizar en el programa de usuario, por ejemplo, para controlar avisadores luminosos con luz intermitente o para iniciar procesos que se repitan periódicamente (como la captación de un valor real).

Frecuencias posibles

Cada bit del byte de marcas de ciclo tiene asignada una frecuencia. La tabla siguiente muestra la asignación:

Bit del byte de la marca de ciclo	7	6	5	4	3	2	1	0
Duración del período (s)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
Frecuencia (Hz)	0,5	0,625	1	1,25	2	2,5	5	10

Nota

Las marcas de ciclo corren de forma asíncrona al ciclo de la CPU, es decir, en los ciclos largos puede cambiar varias veces el estado de la marca de ciclo.

Temporizadores

Los temporizadores representan una área de memoria de la memoria del sistema. La función de un temporizador es determinada por el programa de usuario (p. ej., retardo a la conexión), mientras que la cantidad de temporizadores disponibles depende del tipo de CPU.

Nota

- Si en el programa de usuario se utilizan más temporizadores de los que admite la CPU, se señala un error síncrono y se arranca el OB 121.
- En el S7-300 (a excepción de la CPU 318), los temporizadores sólo se pueden arrancar y actualizar simultáneamente en el OB 1 y en el OB 100; en todos los demás OBs solamente se pueden arrancar los temporizadores.

Índice alfabético

¿

- ¿Qué bloques de notificación existen? 16-6
- ¿Qué procedimientos existen para crear avisos? 16-1

A

Abrir

- tabla de variables 20-3
- tablas de símbolos 8-14

Acceder a la vista rápida 23-5

Acceso

- información del módulo desde la vista del proyecto (online) 23-6

Acceso al área de datos de la periferia A-116

Acceso al área de datos de periferia A-117

Acceso online a sistemas de destino desde el multiproyecto 18-4

Accionamiento del contacto 26-4

ACT_TINT 4-28, 4-29

Activar

- visualización de símbolos en el bloque 8-13

Actualización del firmware A-123

Actualización del sistema operativo (consulte Actualizar el firmware de módulos y bloques online) 18-10

Actualización offline del sistema operativo de módulos y tarjetas A-123

Actualización online del firmware de módulos y submódulos 18-10

Actualizar 18-10, 18-11, 18-12, A-123, A-124

- Firmware (sistema operativo) de módulos y submódulos - online 18-10

- firmware (sistema operativo) de módulos y tarjetas - offline A-123

- imagen del proceso A-20, A-21, A-22, A-23

- Imagen del proceso 4-11

- llamadas a bloques 10-26

Actualizar firmware 18-12

Administrador SIMATIC 5-1, 9-18, 9-20

bloques

- visualizar longitudes de bloques 9-17

Advertencia A-24

- sobreescribir la pila L A-24

Ajustar

- comportamiento del sistema A-120

- condiciones de disparo

- para forzar variables 20-18

- el disparo para observar variables 20-15

- interface PG/PC 2-11

- memoria de trabajo virtual 26-5

Ajustar el idioma de Windows 6-8

Ajustar la preferencia de operandos (simbólica/absoluta) 8-5

Ajuste de la hora 18-9

Ajuste del idioma de Windows 6-5, 6-7

Ajustes

- editor AWL 10-4

- para el lenguaje de programación AWL 10-25

Ajustes para el lenguaje de programación FUP 10-22

Ajustes para el lenguaje de programación KOP 10-18

Ajustes para notificar errores del sistema 16-48

Alarma cíclica 4-31, 4-32

- arrancar 4-31

- Regulación 4-31

Alarma de diagnóstico (OB 82) 23-38

Alarma de presencia de módulo (OB 83) 23-39

Alarma de proceso

- arrancar 4-33

- prioridad 4-33

- reglas 4-33

Alarma de retardo A-105

- aplicación A-105

- arrancar 4-30

- estructura A-105

- prioridad 4-30

- reglas 4-30

Alarma horaria 4-28, 4-29, 18-10

- arrancar 4-28

- cambiar la hora 4-29

- consultar 4-28

- desactivar 4-29

- prioridad 4-29

- regular 4-28

Alarmas cíclicas 4-31

Alarmas de proceso 4-33

Alarmas horarias 4-28, 4-29, A-97

- estructura A-97

- uso A-97

Ampliar esclavos DP (creados con versiones anteriores de STEP 7) 7-1

Ancho del campo de operando 10-18, 10-22

ANY A-51, A-58, A-59, A-60

- parámetro

- descripción y utilización A-61

Aparatos de campo PA 23-14

- Aparatos PA PROFIBUS 23-14
- Aplicación de las funciones de reloj A-124
- Archivar
 - Casos de aplicación 24-5
 - procedimiento 24-6
 - proyectos de STEP 7 V2.1 con comunicación de datos globales A-74
 - proyectos y librerías 24-4
 - requisitos 24-6
- Archivo *.awl 6-26
- Archivo *.k7e 6-26
- Archivo *.k7p 6-26
- Archivo *.sdf 6-26
- Archivo de exportación S7 6-26
- Archivo de tipo 7-1
- archivo GSD A-74
- Archivo GSD 7-1, 7-2
- Archivo Memory Card 6-25
- Archivos fuente en S7-GRAPH 9-8
- Área de instrucciones 10-3, 10-7
 - editar 10-11
 - en KOP 10-6
 - estructura 10-11
 - función de búsqueda de errores 10-17
- Área EPROM A-16
- Área RAM A-16
- Área remanente
 - en CPUs S7-300 A-29
- Áreas de memoria A-15
 - Áreas de direccionamiento A-18
 - memoria de carga A-15
 - memoria de sistema A-15
 - memoria de trabajo A-15
- Áreas de memoria
 - memoria remanente A-29
 - particularidades en S7-300 A-16
 - particularidades en S7-400 A-16
- Áreas remanentes de la memoria de las CPUs S7-400 A-30
- Áreas remanentes de la memoria en CPUs S7-300 A-29
- Áreas y tareas
 - subdividir el proceso en 3-2
- Arquitectura del sistema
 - ciclo 4-11
 - estados operativos de la CPU A-1
- Arrancar 4-28
 - alarma cíclica 4-31
 - alarma de proceso 4-33
 - alarma de retardo 4-30
 - alarma horaria 4-28
- Arrancar STEP 7 con los parámetros de arranque estándar 5-2
- ARRANQUE A-5, A-7, A-8, A-9, A-10, A-11, A-12
 - actividades de la CPU A-5
 - estado operativo de la CPU A-1
 - interrumpir A-5
- Arranque en frío A-1
- ARRAY A-41
- Array (tipo de datos ARRAY)
 - descripción A-44
 - número de niveles anidados A-43
- Arrays
 - crear A-46
 - uso de arrays para acceder a los datos A-44
- Asignación de números de aviso 16-11
- Asignar
 - nombres simbólicos A-81
- Asignar memoria A-24
 - en la pila L A-24
- Asignar números a los avisos 16-10
- Asignar tipos de datos a los datos locales de bloques lógicos A-64
- Asignar y editar avisos de símbolos (para toda la CPU) 16-26
- Asignar y editar avisos de símbolos (para todo el proyecto) 16-18
- Asistente para crear un proyecto 6-9
- Atributos de bloques y parámetros 9-21
- Atributos de sistema 16-8
 - en la tabla de símbolos 8-9
 - para la configuración de avisos 16-8
 - para la configuración de avisos PCS7 (para todo el proyecto) 16-17
- Atributos de sistema para parámetros 10-6
- Atributos del sistema para la configuración de avisos PCS 7 (para toda la CPU) 16-24
- Atributos M+V 17-1
 - configurar con AWL FUP KOP 17-3
 - configurar desde e la tabla de símbolos 17-4
 - modificar con CFC 17-5
- Automation License Manager 2-1
- Autorización de utilización con el Automation License Manager 2-1
- Aviso
 - Aviso de diagnóstico personalizado 16-19
 - ejemplo 16-6
 - partes 16-5
- Aviso (para toda la CPU) de símbolos 16-26
- Aviso (para todo el proyecto) de símbolo 16-18
- Aviso de diagnóstico 23-23
 - enviar a estaciones 23-23
 - escribir uno personalizado 23-23
- Aviso SCAN
 - vea aviso de símbolo (para todo el proyecto) 16-18
- Aviso SCAN (para toda la CPU)
 - Consulte Aviso de símbolos 16-26
- Avisos 16-1, 16-30
 - el sistema y su definición 16-1
 - Integrar textos de librerías 16-30

- Avisos de bloques
 - asignar y editar 16-12
- Avisos de bloques (para toda la CPU)
 - crear 16-21, 16-23
 - editar 16-24
- Avisos de bloques (para todo el proyecto)
 - crear 16-13, 16-15
 - editar 16-16
- Avisos de CPU
 - configurar 16-42
 - tamaño del fichero 16-39
 - visualizar 16-39
- Avisos de diagnóstico definidos por el usuario
 - visualizar 16-39
- Avisos de diagnóstico personalizados
 - crear y editar 16-19
- Avisos de símbolos (para toda la CPU)
 - Asignación a la tabla de símbolos 16-26
 - Señales permitidas 16-26
- Avisos de símbolos (para todo el proyecto)
 - Aignación a la tabla de símbolos 16-18
 - Señales permitidas 16-18
- AWL 9-2, 9-3, 9-6
 - ajustes 10-25
 - introducir bloques 10-12
 - visualizar información sobre el bloque 14-8
- Ayuda (en pantalla)
 - llamar 5-4
 - Temas 5-4
- Ayuda contextual 5-4
- Ayuda en pantalla
 - cambiar el tamaño de letra 5-4
 - llamar 5-4
 - temas 5-4

- B**
- Barra de estado
 - ejemplo 5-21
- Barra de herramientas
 - botones 5-21
- Barra de menús 5-21
- Barra de puntos de parada 21-5, 21-7
- Barra de título 5-21
- Base de tiempo (ver hora del módulo) 18-9
- Base de tiempo para S5 TIME A-40
- BCD A-39
- BLKMOV A-16
- BLOCK
 - tipo de parámetro A-51
- BLOCK_DB A-51
- BLOCK_FB A-51
- BLOCK_FC A-51
- BLOCK_SDB A-51
- Bloque
 - para modificar el puntero A-55
- Bloque apto para notificación 16-13, 16-21
- Bloque de datos (DB) 4-2
 - bloques de datos de instancia 4-18, 4-21
 - estructura 4-24
 - global 4-24
 - remanente A-29
- Bloque de datos de instancia 4-21, 4-22, 4-23
 - crear varias instancias para un FB 4-18
 - remanente A-29
- Bloque de organización (OB)
 - OB de tarea no prioritaria (OB 90) 4-3, 4-36
- Bloque de organización de tarea no prioritaria (OB 90) 4-36
- Bloque de organización para la ejecución cíclica del programa (OB1) 4-11
- Bloquear eventos de alarma y de error
 - asíncrono
 - ejemplo A-114
- Bloques 4-2, 15-1, 15-2
 - atributos 9-21
 - borrar en el sistema de destino 19-18
 - cargar desde una CPU S7 19-14
 - crear con S7-GRAPH 9-8
 - del programa de usuario 4-2
 - derechos de acceso 10-4
 - guardar 10-28
 - introducir en AWL 10-12
 - recablear 9-21
- Bloques (cargados)
 - editar en la PG/en el PC 19-15
 - guardar en la EPROM integrada 19-6
- Bloques de datos 11-1, 12-1
 - cambiar valores en la vista "Datos" 11-8
 - guardar 11-9
 - inicializar los valores de datos 11-8
 - introducir/visualizar la estructura de datos con FB asociado (DBs de instancia) 11-5
 - nociones básicas 11-1
 - parametrizar 12-1
 - ver datos 11-3
 - ver declaración 11-2
- Bloques de datos de instancia 4-21
 - indicación de fecha y hora 15-5
- Bloques de datos globales 15-5
 - indicación de fecha y hora 15-5
 - introducir la estructura 11-4
- Bloques de datos globales (DB) 4-24
- Bloques de función (FB) 4-18
- Bloques de función (FBs) 4-2, 4-18
 - campo de aplicación 4-18
 - crear
 - FB para el ejemplo de un proceso de mezcla industrial A-84
 - parámetros actuales 4-18, 4-19, 4-20
- Bloques de función del sistema 4-2
- Bloques de funciones del sistema 4-25
 - tipos 4-25
- Bloques de funciones del sistema (SFB) y funciones del sistema (SFC) 4-25
- Bloques de notificación
 - Sinopsis 16-6
- Bloques de organización 4-2

- crear un OB para el ejemplo de un proceso de mezcla industrial A-91
 - definición 4-3
 - Detección de errores
 - OB122
 - valores de sustitución 23-32
 - información de arranque 4-5
 - prioridades 4-3, 4-5
 - reaccionar a errores 4-38
 - Bloques de organización de alarma cíclica (OB 30 a OB 38) 4-31
 - Bloques de organización de alarma de proceso (OB 40 a OB 47) 4-33
 - Bloques de organización de alarma de retardo (OB 20 a OB 23) 4-30
 - Bloques de organización de alarma horaria (OB 10 a OB 17) 4-28
 - Bloques de organización de arranque (OB100/OB101/OB102) 4-34
 - Bloques de organización de tratamiento de errores (OB70aOB87 / OB 121 a OB 122) 4-38
 - Bloques de organización para la alarma horaria (OB 10 a OB 17) 4-28
 - Bloques de organización para la ejecución controlada por alarmas 4-27
 - Bloques de organización y estructura del programa 4-3
 - Bloques lógicos 15-4
 - crear 10-3
 - definir
 - ejemplo A-80
 - en el editor incremental 10-3
 - estructura 10-3
 - guardar 10-28
 - indicación de fecha y hora 15-4
 - Bloques modificados 19-6
 - cargar en el sistema de destino 19-6
 - Bloques preprogramados 4-25
 - Bobinas
 - emplazar 10-19
 - BOOL
 - como tipo de datos A-32
 - margen A-33
 - Borrado total A-4
 - de la CPU 19-17
 - Borrar 5-23, 5-27
 - bloques S7 en el sistema de destino 19-18
 - memorias de carga y de trabajo 19-17
 - objetos STEP 7 5-23
 - Botones
 - barra de herramientas 5-21
 - BSTACK
 - datos guardados en la pila BSTACK A-26
 - llamadas anidadas A-26
 - Búfer anular (búfer de diagnóstico A-27
 - Búfer de diagnóstico A-27, A-28
 - contenido 23-24, A-27, A-28
 - definición A-27
 - evaluar A-27
 - leer 23-19, A-28
 - Buscar errores en fuentes AWL 13-18
 - Búsqueda de errores
 - en bloques 10-17
 - Búsqueda de errores y fallos 23-1
 - Byte
 - como tipo de datos A-32
 - margen A-33
- C**
- Cambiar 4-28
 - estado operativo 18-8
 - la hora de una alarma horaria 4-28
 - valores en la vista "Datos" de bloques de datos 11-8
 - Cambiar de estado operativo A-1
 - Cambiar de una ventana a otra 5-36
 - Cambiar el nombre 5-25
 - proyectos 5-23, 5-25
 - Cambiar interfaces 10-27
 - Cambiar la organización de las ventanas 5-29
 - Cambios de estado A-2, A-3
 - CAN_TINT 4-29
 - Carácter (CHAR)
 - margen A-33
 - Carga del ciclo por comunicaciones 4-15
 - Carga por comunicaciones 4-15
 - Cargar 19-8, 19-9
 - bloques desde una CPU S7 19-14
 - con gestión de proyectos 19-5
 - en la PG desde el sistema de destino 19-13
 - la configuración actual y todos los bloques en la PG 19-13
 - mediante Memory Cards EPROM 19-7
 - programa de usuario A-16
 - programas de usuario en el sistema de destino 19-3
 - requisitos 19-1
 - sin gestión de proyectos 19-5
 - varios objetos 19-8
 - Cargar objetos 19-10
 - Carpeta
 - bloques 9-12
 - Carpeta de bloques 5-14, 9-12
 - Carpeta de fuentes 5-18
 - Casos de aplicación para Guardar / Archivar 24-5
 - Certificate of License 2-1, 2-3
 - CFC 9-2
 - Ciclo 4-3, 4-4, 4-6, 4-11, 4-12, 4-13, 4-14, 4-15
 - Ciclo residual A-6, A-9, A-12
 - Claves de licencia 2-5
 - Código fuente
 - insertar de bloques existentes en fuentes AWL 13-15

- Coherencia en fuentes AWL
 - comprobar 13-18
- Combinaciones de teclas para acceder a la Ayuda en pantalla 5-35
- Combinaciones de teclas para cambiar de una ventana a otra 5-36
- Combinaciones de teclas para comandos de menú 5-31
- Combinaciones de teclas para mover el cursor 5-33
- Combinaciones de teclas para seleccionar textos 5-35
- Comentarios
 - de bloques 10-14
 - de segmentos 10-14, 10-15
- Comentarios de bloques 10-14, 10-15
 - introducir 10-15
- Comentarios de segmentos 10-14
 - introducir 10-15
- Cómo iniciar STEP 7 5-1
- Comparación de bloques 9-18, 9-19, 9-20
- Comparación de estado teórico/real A-5
- Comparar bloques 9-18
- Comparar detalles 9-18
- Comparar interlocutores On-/Offline 9-18
- Compatibilidad A-74
- Compatibilidad (comunicación directa) 7-3
- Compatibilidad (Enlace online a través de interface PROFIBUS DP) 7-3
- Compatibilidad (esclavos DP) 7-1, A-74
- Compatibilidad (proyectos y librerías V2) 7-1
- Compatibilidad con versiones anteriores 7-3
- Compilar
 - fuentes AWL 13-19
- Compilar objetos 19-10
- Compilar objetos y cargar 19-8, 19-10
- Compilar y cargar 19-8
- Componentes compatibles y su funcionalidad 16-45
- Componentes de proyectos 24-1
 - imprimir 24-1
- Componentes SIMATIC para la configuración de avisos 16-5
- Comprimir
 - el contenido de la memoria de una CPU S7 19-20
- Comprimir la memoria de usuario 19-19
- Comprobar
 - coherencia en fuentes AWL 13-18
 - datos de referencia 14-10
- Comprobar coherencia de bloques 15-1
- Comprobar en modo Etapa individual 21-5, 21-7
- Comprobar la asignación de direcciones 2-11
- Comprobar la asignación de interrupciones 2-12
- Comprobar los paquetes de software utilizados en el proyecto 6-15
- Comunicación de datos globales A-74
- Comunicación directa 7-3
- Comunicación GD A-74
- Condición de disparo 20-15
- Configuración
 - Establecer un enlace online desde la ventana "Estaciones accesibles" 18-2
- Configuración de avisos
 - componentes SIMATIC 16-5
 - transferir datos a WinCC 16-38
- Configuración de avisos PCS 7 (para toda la CPU) 16-24
- Configuración de avisos PCS 7 (para todo el proyecto) 16-17
- Configurar atributos M+V en AWL
 - KOP y FUP 17-3
- Configurar atributos M+V desde la tabla de símbolos 17-4
- Configurar avisos con errores del sistema 16-43
- Configurar avisos de CPU 16-42
- Configurar hardware 26-1
- Configurar la notificación de errores del sistema 16-43
- Configurar variables para manejo y visualización 17-1
- Conflictos de fecha y hora 15-3
- Consejos y 26-2
- Consejos y trucos 26-1, 26-3, 26-5
- Consignas de seguridad A-24
 - desbordar la pila L A-24
- Consultar la alarma horaria 4-28
- Contador de horas de funcionamiento A-125
- Contadores 14-5, 14-6
 - área de memoria remanente A-29
 - Plano de ocupación 14-5, 14-6
 - valores máximos de introducción 20-9
- Contenido de las pilas en estado operativo STOP 23-16
- Contraseña 18-6
- Contraseña para acceder a los sistemas de destino 18-6
- Control de los tiempos de ciclo para evitar errores de tiempo 23-18
- Control de variables 20-17
 - introducción 20-17
- Control por alarma
 - ejecución del programa 4-7
- Controlar el tiempo de ciclo para evitar errores de tiempo 23-18
- Convenciones para nombres
 - para datos de configuración 17-1
- Convertir A-74
 - proyecto con comunicación de datos globales A-74
 - proyectos de STEP 7 V2.1 con comunicación de datos globales A-74
- Convertir proyectos de la versión 1 A-72

Convertir proyectos de la versión 2 A-73
Copiar/mover tablas de variables 20-3
Corregir
 interfaces en una FC
 un FB o un UDT 15-6
Cortocircuito 10-21
COUNTER A-51
 tipo de parámetro A-51
CPU
 borrado total 19-17
 simular 22-1
CPU (Central Processing Unit)
 estados operativos A-1, A-2, A-3
CPU 31xC 6-24, 6-25, 6-26
Creación de huecos en la memoria de usuario
 (RAM) 19-19
Crear 5-23
 arrays A-44, A-46, A-47
 datos de referencia 14-10
 diagrama de E/S para las válvulas 3-7
 diagrama de E/S para los motores 3-7
 esquema de configuración 3-10
 estructura A-47, A-48
 FB para el motor A-84, A-85, A-86, A-87
 FC para las válvulas A-89, A-90
 fuentes AWL 13-14
 OB 1 para el ejemplo de un proceso de
 mezcla industrial A-91
 objetos 5-23
 tabla de variables 20-3
 un tipo de dato de usuario A-49
Crear avisos de bloque
 (para todo el proyecto) 16-13
Crear bloque de datos en la memoria
 de carga 6-24
Crear controles secuenciales 9-8
 con S7-GRAPH 9-8
Crear librerías de textos de usuario 16-34
Crear objeto 5-23
Crear programas de usuario 10-3
Crear textos de aviso en otros idiomas en
 'Notificar errores de sistema' 16-51
Crear un diagrama de entradas para las
 válvulas 3-7
Crear un diagrama de entradas para los
 motores 3-7
Crear un diagrama de salidas
 para las válvulas 3-7
Crear un diagrama de salidas
 para los motores 3-7
Crear un esquema de configuración a la vista
 del ejemplo de un proceso de mezcla
 industrial 3-10
Crear un proyecto 6-9
Crear una protección de acceso 6-3
Crear y manejar objetos 5-23
CREAT_DB A-16
CRST/WRST A-5, A-6, A-7
CTRL_RTM A-125

Cuadro combinado
 definición 5-22
Cuadro de diálogo de selección 5-28
Cuadro de lista 5-22
Cuadros
 emplazar 10-18, 10-24
 retirar
 cambiar 10-23
Cuadros de diálogo 5-22
Cuadros de diálogo con fichas 5-22

D

DATE_AND_TIME A-41
Datos
 tipos de datos de usuario 9-13
Datos de configuración 17-1, 17-2
 requisitos para la transferencia 17-6
 requisitos para transferir 16-38
 transferir 16-38, 17-6
Datos de diagnóstico en los módulos 23-22
Datos de estado de diagnóstico 23-21
Datos de periferia A-117
Datos de referencia 14-1
 aplicación 14-1
 crear 14-10
 sinopsis 14-1
 visualizar 14-9, 14-10
Datos de sistema 23-20
Datos útiles A-117
DB 4-24
 formato 13-13
DBs en fuentes AWL
 ejemplos 13-26
Declaración de datos locales A-64
Declaración de variables en fuentes AWL
 ejemplos 13-20
Declaración de variables locales
 OB para el ejemplo de un proceso de
 mezcla industrial A-91
Declarar parámetros
 FC para el ejemplo de un proceso de
 mezcla industrial A-89
Declarar variables locales
 FB para el ejemplo de un proceso de mezcla
 industrial A-84
Defectuosa
 estado operativo de la CPU A-1
Definir
 bloques lógicos A-80
 modo de funcionamiento para el test 21-10
 requerimientos de seguridad 3-8
 símbolos al introducir el programa 8-13
 visualización del estado del programa 21-9
Definir el diseño del texto fuente 13-15
Derechos de acceso a bloques y fuentes 10-4
Desactivar 4-28
 alarma horaria 4-28

- Desarchivar
 - procedimiento 24-6
 - Desbordar la pila L A-24
 - Describir
 - elementos necesarios para manejo y visualización 3-9
 - requerimientos de seguridad para el ejemplo de un proceso de mezcla industrial 3-8
 - Describir el panel de manejo a la vista del ejemplo de un proceso de mezcla industrial 3-9
 - Describir las tareas y áreas individuales
 - ejemplo de un proceso de mezcla industrial 3-4
 - Describir los elementos de manejo a la vista del ejemplo de un proceso de mezcla industrial 3-9
 - Descripción del panel de manejo 3-9
 - Desenmascarar
 - eventos de arranque 4-38
 - Desenmascarar eventos de errores síncronos ejemplo A-111
 - Desfase 4-31
 - Desinstalar
 - la autorización de utilización 2-5
 - STEP 7 2-13
 - Detección de errores
 - Programas de ejemplo
 - valores de sustitución 23-32
 - utilizar OBs de error para reaccionar a errores 4-38
 - Detección e errores
 - tipos de OBs
 - OB81 23-27
 - Determinar estaciones en la subred 18-2
 - Diagnosticar hardware 23-1
 - Diagnóstico de sistema
 - ampliar 23-23
 - Diagnóstico del hardware 23-1
 - Diagrama de funciones 9-5
 - Diálogo de control de procesos
 - consulte Configuración de avisos PCS 7 (para toda la CPU) 16-24
 - vea Configuración de avisos PCS 7 (para todo el proyecto) 16-17
 - Diferencia entre guardar y cargar bloques 19-2
 - Diferencias entre asignación de números de aviso para todo el proyecto y para toda la CPU 16-11
 - Diferencias entre el forzado normal y el forzado permanente 20-23
 - DINT
 - tipo de datos A-34
 - Dirección inicial A-117
 - Dirección inicial de los módulos A-116
 - Direccionamiento
 - absoluto 8-1
 - de esclavos DP estándar A-120
 - de módulos S5 A-117
 - indirecto por memoria A-53
 - interárea A-54
 - intraárea A-53
 - simbólico 8-1
 - Direccionamiento absoluto y simbólico 8-1
 - Direccionamiento de módulos A-116
 - Direccionamiento simbólico 8-4
 - en el programa de ejemplo A-81
 - Direcciones y tipos de datos admisibles en la tabla de símbolos 8-10
 - DIS_AIRT 4-38
 - DIS_IRT 4-38
 - Diseño de página FUP 10-22
 - Diseño de página KOP 10-18
 - Disparo
 - ajustar para observar variables 20-15
 - Disponer
 - cuadros 10-23, 10-24
 - Distribución de las áreas de memoria A-15
 - DMSK_FLT 4-38
 - DOCPRO 24-1
 - Documentación 1-1, 1-4
 - de componentes de un proyecto
 - imprimir 24-1
 - de un proyecto entero
 - imprimir 24-1
 - Documentación del proyecto
 - imprimir 24-1
 - DPNRM_DG A-120
 - DPRD_DAT A-120
 - DPWR_DAT A-120
 - DWORD
 - tipo de datos A-39
- ## E
- Editar 16-35
 - bloques cargados la PG o el PC no contiene el programa de usuario 19-16
 - bloques cargados si la PG o el PC contiene el programa de usuario 19-16
 - en la tabla de símbolos 8-13
 - fuentes S7 13-14
 - librerías de textos de sistema 16-35
 - los bloques cargados en la PG/ en el PC 19-15
 - Editar áreas en tablas de símbolos 8-20
 - Editar configuraciones actuales con versiones anteriores de STEP 7 7-3
 - Editar símbolos en varios segmentos 26-2
 - Editor
 - ajustes para AWL 10-4
 - Editor de lenguaje
 - arrancar 9-2
 - Editor de programas 9-18, 9-19, 9-20, 10-1
 - Ejecución cíclica del programa 4-3, 4-7

- Ejecución del programa 4-27
 - cíclica 4-3, 4-6, 4-7
 - controlada por alarma 4-3
 - controlada por alarmas 4-27
- Ejemplo A-97
 - aplicación de alarmas de retardo A-105
 - bloquear y habilitar eventos de alarma y de error asíncrono (SFC 39 y 40) A-114
 - de aplicación de alarmas horarias A-97
 - enmascarar y desenmascar eventos de errores síncronos A-111
 - FBs en fuentes AWL 13-24
 - FCs en fuentes AWL 13-22
 - formato de número en coma flotante A-35
 - introducción de operandos en tablas de variables 20-10
 - OBs en fuentes AWL 13-21
 - tratamiento retardado de eventos de alarma y de error asíncrono (SFC 41 y 42) A-115
 - UDTs en fuentes AWL 13-27
- Ejemplo de introducción de un área de operandos conexos 20-11
- Ejemplo para trabajar con puntos de aplicación 14-12
- Ejemplos
 - DBs en fuentes AWL 13-26
 - declaración de variables en fuentes AWL 13-20
 - introducción de valores de forzado normal y de forzado permanente 20-12
- Ejemplos de programa
 - FC para el ejemplo de un proceso de mezcla industrial A-89
 - OB para el ejemplo de un proceso de mezcla industrial A-91
- Ejemplos de programas
 - ejemplo de un proceso de mezcla industrial crear un esquema de configuración 3-10
 - Describir los requerimientos de seguridad 3-8
 - descripción de las distintas tareas y áreas 3-4
 - Descripción de las tareas y áreas individuales
 - Crear un diagrama de E/S 3-6
 - Panel de manejo
 - describir 3-9
 - Subdividir un proceso en áreas de tareas 3-2
 - FB para el ejemplo de un proceso de mezcla industrial A-84
- El software estándar STEP 7 1-6
- Elaborar 6-14
 - proyecto 6-14
- Elegir el lenguaje de programación 9-2
- Elegir objetos en cuadros de diálogo 5-28
- Elementos de los cuadros de diálogo 5-22
- Elementos de programa
 - insertar 10-5
- Elementos de visualización
 - describir 3-9
- Elementos FUP
 - representación 10-22
- Elementos KOP
 - representación 10-18
- Eliminación de errores
 - programas de ejemplo 23-27
- Eliminar la protección de acceso 6-3
- Eliminar un cuello de botella en la memoria 19-19
- Emplazar
 - cuadros 10-23
- EN / ENO
 - conexión 10-23
- EN_AIRT 4-38
- EN_IRT 4-38
- Encabezados y pies de página 24-3
- Enlace
 - establecer con la CPU 20-14
- Enlace DP/PA (IM 157) 23-14
- Enlace online 18-3
 - establecer desde la ventana "Estaciones accesibles" 18-2
 - establecerlo en la ventana online 18-3
- Enlace online a través de interface DP 7-3
- Enlace Y 23-14
- Enlaces online
 - establecer 18-1
- Enmascarar
 - eventos de arranque 4-38
- Enmascarar eventos de errores síncronos
 - ejemplo A-111
- Entero (16 bits)
 - Formato A-34
- Entero (16 bits) (INT)
 - margen A-33
- Entero (32 bits)
 - Formato A-34
- Entero (32 bits) (DINT)
 - margen A-33
- Entradas
 - imagen del proceso A-20
 - listar 3-6
 - Plano de ocupación 14-5
- Entradas en cuadros de diálogo 5-22
- Entradas/salidas
 - listar 3-6
- Enviar 23-23
 - avisos de diagnóstico personalizados 23-23
- Enviar avisos de diagnóstico personalizados 23-23
- EPROM A-29
- EPROM integrada
 - guardar los bloques cargados 19-6
- Equipo 5-9
 - cargar en PG 19-13

- Equipo PC 7-4, 7-5
- Equipos
 - insertar 6-11
- Error de acceso a la periferia (OB 122) 23-44
- Error de comunicación (OB 87) 23-43
- Error de ejecución del programa (OB 85) 23-41
- Error de programación (OB 121) 23-43
- Error de redundancia en CPU (OB 72) 23-35
- Error de redundancia en periferia (OB 70) 23-34
- Error de tiempo (OB 80) 23-36
- Errores
 - durante la instalación 2-9
 - evitarlos al llamar bloques 15-7
- Errores asíncronos
 - OB81 23-27
 - utilizar OBs para reaccionar a errores 4-38
- Errores de acceso a la periferia (PZF) al actualizar la imagen del proceso A-20
- Errores de sistema 23-24
 - notificar 16-45
- Errores del sistema 16-49
 - Ajustes para la notificación 16-48
- Errores detectables 23-27
- Errores síncronos
 - utilizar OBs para reaccionar a errores 4-38
- esclavo Dummy A-74
- Esclavos con archivos GSD defectuosos o inexistentes A-74
- Esclavos DP 7-1, 7-2
- Esclavos DP con archivos GSD defectuosos o inexistentes A-74
- Esclavos DP estándar A-120
- Escribir en un bloque de datos de la memoria de carga 6-24
- Esquema de configuración
 - crear 3-10
- Esquema de contactos 9-4
- Establecer un enlace con la CPU 20-14
- Establecer un enlace online desde la ventana online del proyecto 18-3
- Establecimiento
 - enlaces online 18-1
- Estaciones PROFInet 18-2
- Estado de programas
 - visualización 21-3
- Estado del programa 21-1
 - definir la visualización 21-9
- Estado operativo
 - ARRANQUE A-1, A-3, A-5
 - PARADA A-3
 - RUN A-3
 - STOP A-2, A-3, A-4
 - contenido de las pilas 23-16
 - visualizar y cambiar 18-8
- Estado operativo "RUN" A-13
- Estado operativo PARADA A-14
- Estados operativos
 - cambiar A-3
 - de la CPU A-1, A-3
 - prioridad A-3
- Estados operativos y cambios de estado A-1
- Estructura
 - área de instrucciones 10-11
 - crear A-48
 - de bloques en fuentes AWL 13-8
 - de bloques lógicos en fuentes AWL 13-8
 - del programa de usuario
 - "Alarmas horarias" A-97
 - lista de referencias cruzadas 14-2
 - memoria de carga A-16, A-17
 - UDT 9-13
 - ventana 5-21
 - ventana de declaración de variables 10-8
- Estructura (STRUCT)
 - descripción A-47
- Estructura (tipo de datos STRUCT)
 - número de niveles anidados A-43
- Estructura de bloques de datos en fuentes AWL 13-9
- Estructura de la ventana del editor de programas 10-1
- Estructura de los proyectos 6-1
- Estructura de proyectos 6-2
- Estructura de tipos de datos de usuario en fuentes AWL 13-9
- Estructura del archivo de exportación 6-18
- Estructura del programa 14-3, 14-4
 - visualizar 14-9
- Estructura del programa de usuario "Alarmas de retardo" A-105
- Estructura en árbol 14-3
- Estructura jerárquica de las librerías 9-23
- Estructura y componentes de la tabla de símbolos 8-8
- Evaluación del búfer de diagnóstico A-27
- Evaluar
 - parámetro de salida RET_VAL 23-26
- Evento 4-11
- Evento de diagnóstico 23-24, A-27
- Eventos
 - asíncronos 4-11
- Eventos asíncronos 4-15
- Eventos de alarma
 - bloquear y habilitar A-114
 - tratamiento retardado A-115
- Eventos de arranque
 - enmascarar 4-39
 - OBs de arranque 4-34
 - retardar 4-39
- Eventos de error asíncrono
 - bloquear y habilitar A-114
 - tratamiento retardado A-115
- Eventos de errores síncronos
 - enmascarar y desenmascarar A-111
- Evitar errores al llamar bloques 15-7

Exportar
 fuentes 13-17
 tabla de símbolos 8-17

F

Factor de corrección 18-9
 Fallo de alimentación (OB 81) 23-37
 Fallo de CPU (OB 84) 23-40
 Fallo del bastidor (OB 86) 23-42
 FB 4-18, 4-19, 4-20, A-41
 corregir el interface 15-6
 DB generados 16-49
 formato 13-11
 FBs en fuentes AWL
 ejemplo 13-24
 FC 4-16, 4-17
 corregir el interface 15-6
 formato 13-12
 FC 12 A-99
 FCs en fuentes AWL
 ejemplo 13-22, 13-23
 Fecha y hora 15-3
 propiedad del bloque 15-3
 Fecha y hora como propiedad del bloque y
 conflictos de fecha y hora 15-3
 FEPRM A-29
 Fichero
 avisos de CPU 16-39, 16-41
 Filosofía de manejo 5-20
 Filtrar
 símbolos 8-14
 Flash File System 2-8
 Flujo de corriente 10-21
 FORCE-LED parpadea 18-2
 Formato
 BLOCK A-52
 COUNTER A-52
 TIMER A-52
 BLOCK
 COUNTER A-52
 tipo de datos DATE_AND_TIME
 (fechar y hora) A-42
 tipo de datos DINT
 (enteros de 32 bits) A-34
 tipo de datos INT (enteros de 16 bits) A-34
 tipo de datos REAL (números en coma
 flotante) A-35
 tipo de datos S5TIME (intervalo) A-40
 tipo de parámetro ANY A-58
 tipo de parámetro POINTER A-52, A-53
 tipos de datos WORD y DWORD en
 números decimales codificados
 en binario A-39
 tipos de parámetros BLOCK
 COUNTER
 TIMER A-52
 Formato BCD A-40
 Formato de la hora A-124

Formato de las FCs 13-12
 Formato de los DBs 13-13
 Formato de los FBs 13-11
 Formato de los OBs 13-10
 Formato de página
 ajustar 24-3
 Formato de puntero A-51, A-53
 Formatos de archivos para importar/exportar
 una tabla de símbolos 8-17
 Formatos de bloques en fuentes AWL 13-10
 Forzado permanente de variables
 reglas de seguridad 20-20
 Forzar variables 20-22
 ajustar las condiciones de disparo 20-18
 introducción 20-21
 procedimiento básico 20-2
 Forzar variables desde el editor de programas
 26-4
 Fuente AWL
 formatos de bloques 13-10
 insertar el contenido de
 otras fuentes AWL 13-15
 Fuentes
 derechos de acceso 10-4
 exportar 13-17
 externas 6-12
 Generar fuentes AWL de bloques 13-16
 Guardar fuentes AWL 13-18
 importar 13-17
 Insertar fuentes externas 13-16
 Reglas para declarar variables en fuentes
 AWL 13-3
 Reglas para definir atributos de sistema en
 fuentes AWL 13-4
 Reglas para definir las propiedades de
 bloques en fuentes AWL 13-5
 Reglas para introducir instrucciones en
 fuentes AWL 13-2
 Reglas sobre el orden de los bloques en
 fuentesAWL 13-4
 Fuentes AWL
 buscar errores 13-18
 compilar 13-19
 comprobar coherencia 13-18
 crear 13-14
 ejemplo de FBs 13-24
 ejemplo de FCs 13-22
 ejemplo de OBs 13-21
 ejemplo de UDTs 13-27
 ejemplos de DBs 13-26
 ejemplos de declaración de variables 13-20
 estructura de bloques 13-8
 estructura de bloques de datos 13-9
 estructura de los bloques lógicos 13-8
 estructura de tipos de datos de usuario 13-9
 generar de bloques 13-16
 guardar 13-18
 insertar el código de
 bloques existentes 13-15

Insertar fuentes externas 13-16
 insertar plantillas de bloques 13-15
 Reglas para definir atributos de sistema 13-4
 Reglas para definir las propiedades de bloques 13-5
 Reglas sobre el orden de los bloques 13-4
 sintaxis de bloques 13-10
 Fuentes S7
 editar 13-14
 Fuentes AWL
 Reglas para declarar variables 13-3
 Función (FC) 4-16
 campo de aplicación 4-16
 crear FC para el ejemplo de un proceso de mezcla industrial A-89
 Función de búsqueda de errores en el área de instrucciones 10-17
 Funcionalidad de "Notificar errores de sistema" 16-45
 Funciones 24-2
 Funciones (FC) 4-2, 4-16
 Funciones de diagnóstico 23-24
 Funciones de información 23-12
 Funciones de información de la vista de diagnóstico 23-8
 Funciones de información de la vista rápida 23-5
 Funciones de la información del módulo 23-10
 Funciones de reloj A-124
 Funciones del sistema 4-2, 4-25
 tipos 4-25
 FUP 9-5
 visualizar información sobre el bloque 14-8
 FW Update 18-10

G

Generar
 Fuentes AWL de bloques 13-16
 Generar bloques para la notificación de errores del sistema 16-49
 Gestionar textos de usuario cuya fuente no está instalada 6-20
 Gestionar textos en varios idiomas 6-15
 Grafo de estado 9-9
 Grandes proyectos 26-1
 GRAPH 9-2
 Guardar
 bloques 10-28
 bloques de datos 11-9
 bloques lógicos 10-28
 Casos de aplicación 24-5
 en la EPROM integrada los bloques cargados 19-6
 Fuentes AWL 13-18
 organización de las ventanas 5-30
 tabla de variables 20-3, 20-4
 Guardar como 6-26

Guardar datos de proyecto en la Micro Memory Card (MMC) 6-26
 Guía de orientación de STEP 7 1-1

H

Habilitar eventos de alarma y de error asíncrono
 ejemplo A-114
 Herramientas de ingeniería 1-13
 HiGraph 9-2
 Hora 4-28, 4-29
 ajustar A-124, A-125
 cambiar 4-29
 Hora (TIME OF DAY)
 margen A-33
 Hora de módulo 18-9
 Hora local 18-9
 Horario de verano/invierno
 Horario de invierno
 Estado horario 18-9, 18-10

I

Iconos para objetos en el Administrador SIMATIC 5-4
 Identificación de símbolos 8-4
 Identificar estaciones conectadas directamente a la PG 18-2
 Idioma 16-32
 Idioma del proyecto 6-5, 6-7
 IM 157 (enlace DP/PA) 23-14
 Imagen del proceso 4-11, A-20
 actualizar 4-14
 borrar 4-35
 entradas y salidas A-21
 Imagen del proceso OB 1 A-21
 Imagen parcial A-21, A-22
 actualizar con SFCs A-22
 actualizar por parte del sistema A-20
 Importar
 fuente externa 6-12
 fuentes 13-17
 Importar la tabla de símbolos 8-17
 Impresora
 instalar 24-2
 Imprimir
 bloques 24-1
 componentes del proyecto 24-1
 contenido del búfer de diagnóstico 24-1
 datos de referencia 24-1
 documentación del proyecto 24-1
 tabla de configuración 24-1
 tabla de datos globales 24-1
 tabla de símbolos 24-1
 tabla de variables 24-1
 IN (declaración de variables) A-64
 IN_OUT (declaración de variables) A-64

- Incompatibilidad A-74
- Indicación de fecha y hora 15-3, 15-4, 18-9
 - de bloques de datos de instancia 15-5
 - de bloques de datos globales 15-5
 - de bloques lógicos 15-4
 - de UDTs y DBs derivados de UDTs 15-6
- Índice 16-30
- Índice de librerías de textos 16-30
- Información del módulo 23-1, 23-6, 23-12, 23-14
 - acceder desde la vista del proyecto (online) 23-6
 - funciones 23-10
 - llamar 23-9
 - visualizar 23-1
 - visualizar la de aparatos de campo PA 23-14
 - visualizar la de esclavos DP a través de un enlace Y 23-14
- Información del símbolo 8-2
- Información importante para comprobar en modo Etapa individual / Puntos de parada 21-5
- Información importante para comprobar en modo Etapa individual / Puntos de parada 21-7
- Información importante sobre la protección de acceso 6-3
- Información importante sobre Micro Memory Cards (MMC) 6-24
- Informe de cambios 6-3, 6-4
- Inicializar valores de datos 11-8
- Iniciar instalación de STEP 7 2-9
- Insertar
 - contenido de otras fuentes AWL 13-15
 - en fuentes AWL el código de bloques existentes 13-15
 - Fuentes externas 13-16
 - operandos o símbolos en una tabla de variables 20-4
 - plantillas de bloques en fuentes AWL 13-15
 - valores de forzado 20-7
 - valores de sustitución al detectar errores 23-32
- Insertar equipos 6-11
- Insertar fuentes externas 13-16
- Insertar un área de operandos conexos en una tabla de variables 20-7
- Instalación
 - Flash File System 2-10
 - introducir nº de identificación 2-8
 - parametrización de Memory Card 2-8
- Instalar
 - STEP 7 2-6, 2-7
- Instalar el Automation Licence Manager 2-4
- Instalar STEP 7 2-6
- Instancia 4-21, 4-22, 4-23
- Instrucciones
 - introducir
 - Procedimiento 10-12
- Instrucciones AWL
 - reglas de introducción 10-25
 - Reglas de introducción 13-2
- Instrucciones de la vista general de elementos de programa 10-5
- INT
 - tipo de datos A-34
- Interacción entre la vista detallada de variables y el área de instrucciones 10-7
- Intercambio de datos
 - en diversos estados operativos A-13
- Interconexiones no admisibles en KOP 10-21
- Interface de usuario 5-21
- Interface PG/PC 2-11
 - parametrizar 2-11
- Interfaces hombre-máquina 1-17
- Interfaz MPI 2-6
- Introducción a los tipos de datos y de parámetros A-32
- Introducción al forzado de variables 20-17
- Introducción al forzado permanente de variables 20-21
- Introducción al test con tablas de variables 20-1
- Introducir
 - estructura de bloques de datos con FB asociado (DBs de instancia) 11-5
 - estructura de bloques de datos con UDT asociado 11-7
 - estructura de bloques de datos globales 11-4
 - estructura de tipos de datos de usuario (UDT) 11-6
 - Multiinstancia en la ventana de declaración de variables 10-10
 - símbolos 8-14
 - símbolos globales en un cuadro de diálogo 8-13
 - símbolos globales en un programa 10-13
- Introducir comentarios de bloques/segmentos 10-15
- Introducir líneas de comentario 20-9
- Introducir nº de identificación 2-8
- Introducir símbolos globales en la tabla de símbolos 8-14

J

- Jerarquía 5-4
- Jerarquía de llamada en el programa de usuario 4-9
- Jerarquía de objetos 5-24
 - crear 5-23, 5-24, 5-25

K

- k7e 6-26
- k7p 6-26
- KOP 9-2, 9-3, 9-4
 - interconexiones no admisibles 10-21
 - Visualizar información sobre el bloque 14-8

L

- Lectura y ajuste de la hora y del estado horario 18-9
- Leer del bloque de datos en la memoria de carga 6-24
- Lenguaje de programación
 - definir 9-2
 - elegir 9-2
 - FUP (diagrama de funciones) 9-5
 - KOP (esquema de contactos) 9-4
 - S7-Graph (control secuencial) 9-8
 - S7-SCL 9-7
- Lenguaje de programación AWL (lista de instrucciones) 9-6
- Lenguaje de programación S7-CFC 9-11
- Lenguaje de programación S7-HiGraph (grafo de estado) 9-9
- Lenguajes de programación 1-6
- Librería 5-8
- Librería de textos 16-30
 - Integrar textos en avisos 16-30
- Librería estándar 6-12
- Librerías 6-13
 - archivar 24-4
 - estructura jerárquica 9-23
 - trabajar con 9-22
- Librerías (versión 2)
 - editar 7-1
- Librerías de texto de usuario 16-34
- Librerías de textos 16-36
- Librerías de textos 16-35
- Librerías de textos de sistema 16-35
- Librerías de textos de sistema 16-35
- Librerías estándar
 - disponibles 9-24
- Licencias 2-1
- License Key 2-1
- License Manager 2-1, 2-2
- Línea de comentario 20-4
- Líneas de comentario
 - introducir 20-9

- Lista de estado del sistema 23-20, 23-22
 - contenido 23-20
 - leer 23-20
 - Lista de estado del sistema (SZL) 23-20
 - Lista de instrucciones 9-6
 - Lista de referencias cruzadas 14-2
 - Listado de entradas y salidas 3-6
 - Listas de textos
 - v. Listas de textos de usuario 16-32
 - Listas de textos de usuario 16-32
 - Llamadas a bloques
 - actualizar 10-26
 - Llamadas anidadas de bloques lógicos
 - efectos en las pilas
 - BSTACK y LSTACK A-26
 - Llamadas de bloques 4-9
 - Llamar las funciones de ayuda 5-4
 - Localización de fallor 23-1
 - Localizar los fallos 23-1
 - Longitudes de bloques
 - visualizar 9-17
 - LSTACK
 - edición de datos en llamadas anidadas A-26
- M**
- M7-300/400 25-1, 25-2
 - Make (consulte Compilar objetos y cargar) 19-10
 - Make (v. Comprobar coherencia del bloque) 15-1
 - Make (véase Compilar objetos y cargar) 19-8
 - Manejar 5-23
 - objetos 5-23, 5-24, 5-25, 5-26, 5-27
 - Manejo de las teclas 5-31
 - Manejo y visualización de variables 17-1
 - Manipulación de grandes proyectos 26-1
 - Manuales de circuitos 24-1
 - imprimir 24-1
 - Marcas
 - área de memoria remanente A-29
 - Plano de ocupación 14-5
 - Marcas de ciclo A-126
 - Matrices A-44
 - Mayúsculas y minúsculas
 - en los símbolos 8-15
 - Medidas en el programa
 - para tratar fallos 23-25
 - Memoria A-31
 - configurable A-31
 - Memoria de carga 19-3, 19-4, A-15, A-16
 - guardar DBs no relevantes para la ejecución A-16
 - Memoria de sesión 5-29
 - Memoria de sistema A-15
 - Memoria de trabajo 19-3, 19-4, A-15, A-16, A-17
 - Memoria de trabajo virtual 26-5

Memoria de usuario
 comprimir 19-19

Memoria remanente
 de CPUs S7-400 A-30

Memorias de carga y de trabajo A-16
 borrar 19-17

Memorias de carga y de trabajo
 de la CPU 19-3

Memory Card A-17
 parametrizar 2-9, 2-10

Método de creación de programas
 seleccionar 9-1

Métodos de diseño
 diseñar un programa estructurado A-80

Métodos para acceder a la información del
 módulo 23-9

Métodos para introducir
 símbolos globales 8-12

Micro Memory Card (MMC) 6-25, 6-26

Micro Memory Cards (MMC) 6-24

MMC 6-24, 6-25, 6-26

Modificar atributos M+V con CFC 17-5

Modificar atributos para manejo y visualización
 con CFC 17-5

Modificar el comportamiento y las propiedades
 de los módulos A-121

Modo "sobreescribir" 10-17

Modo con respaldo
 áreas remanentes de la memoria A-30

Modo de funcionamiento
 definir para el test 21-10

Modo sin respaldo
 áreas remanentes de la memoria A-30

Módulo
 simular 22-1

Módulo de señales
 simular 22-1

Módulo sustituto 7-6, 7-7

Módulos 26-1
 intercambiar 26-1
 parametrizar A-121, A-122

Módulos de señales aptos para alarmas de
 proceso
 parametrizar 4-33

Módulos parametrizables A-121

Mostrar
 operandos sin símbolo 14-9

Mostrar estaciones accesibles 18-2

Motores
 crear un diagrama de E/S 3-7

Mover 5-26
 objeto 5-23, 5-24, 5-25, 5-27

MSK_FLT 4-38

Multiinstancia 4-21
 Introducir en la ventana de declaración de
 variables 10-10

Multiinstancias 4-18
 reglas 10-10
 utilización 10-9

Multiproyectos con protección de acceso 6-3

N

Nemotécnica
 ajustar 10-25

Nociones básicas para crear bloques de datos
 11-1

Nociones básicas para programar en fuentes
 AWL 13-1

Nombres simbólicos
 asignar A-81

Non-Retain 9-16

Nota respecto a la actualización del contenido
 de ventanas 18-7

Notificación 16-44

Notificar errores de sistema
 componentes compatibles 16-45

Notificar errores del sistema 16-43, 16-48, 16-
 49
 Generar bloques 16-49

Novedades de la versión 5.4 de STEP 7 1-10

Número en coma flotante
 campos de componentes A-35
 ejemplo A-35, A-36, A-37
 elementos básicos A-35
 parámetros A-35

Números de aviso 16-11

Números de avisos
 asignar 16-10

Números decimales codificados en binario
 (BCD) A-39

Números en coma flotante
 Formato A-35, A-36, A-37

O

OB 4-3, 4-4, 4-5, 4-6, 4-7, 4-8
 Formato 13-10

OB 1 A-109

OB 1 y OB 80 A-103

OB 10 A-101, A-102

OB 121 23-43

OB 122 23-44

OB 20 A-107

OB 72 23-35

OB 80 23-36

OB 81 23-37

OB 82 23-38

OB 83 23-39

OB 84 23-40

OB 85 23-41, A-23

OB 86
 fallo del bastidor 23-42

OB 87 23-43

OB de alarma de diagnóstico 23-38, 23-40

- OB de alarma de presencia de módulo 23-39
 - OB de error 23-27, 23-28, 23-29
 - tipos de OBs
 - OB121 y OB122 4-38
 - OB70 y OB72 4-38
 - OB80 a OB87 4-38
 - utilizar OBs de error para reaccionar a eventos 4-38
 - OB de error de acceso a la periferia 23-44
 - OB de error de comunicación 23-43
 - OB de error de ejecución del programa 23-41
 - OB de error de programación 23-43
 - OB de error de redundancia
 - en periferia 23-34, 23-35
 - OB de error de tiempo 23-36
 - OB de fallo de alimentación 23-37
 - OB de fallo de CPU 23-40
 - OB de tarea no prioritaria
 - prioridad 4-36, 4-37
 - programar 4-37
 - Objeto
 - abrir 5-23, 5-24
 - borrar 5-23
 - cambiar el nombre 5-23
 - cortar
 - copiar
 - pegar 5-25
 - elegir 5-28
 - manejar 5-23
 - propiedades 5-24
 - Objeto "Carpeta de bloques" 5-14
 - Objeto "Carpeta de fuentes" 5-18
 - Objeto "Equipo" 5-9
 - Objeto "Librería" 5-8
 - Objeto "Módulo programable" 5-10
 - Objeto "Programa S7/M7" 5-12
 - Objeto "Proyecto" 5-6
 - Objetos 5-4, 5-5
 - como carpetas 5-4
 - como portadores de funciones 5-5
 - como portadores de propiedades 5-4
 - Objetos configurables en la memoria de trabajo A-31
 - Objetos y su jerarquía 5-4
 - OBs de alarma 4-27
 - aplicación 4-27
 - deseleccionar 4-5
 - parametrizar 4-5, 4-28
 - OBs de arranque 4-34
 - eventos de arranque 4-34
 - vigilancia de la configuración teórica/real de los módulos 4-34
 - OBs de error para reaccionar a la detección de un error 23-27
 - OBs en fuentes AWL
 - ejemplo 13-21
 - Observaciones
 - sobre proyectos de STEP 7 V2.1 con comunicación de datos globales (GD) A-74
 - Observaciones generales
 - introducir símbolos 8-12
 - Observar el estado de bloques de datos 21-8
 - Observar variables 20-2
 - ajustar el disparo 20-15
 - introducción 20-15
 - procedimiento básico 20-2
 - Operaciones FUP
 - reglas de introducción 10-23
 - Operaciones KOP
 - reglas de introducción 10-18
 - Operandos
 - insertar en una tabla de variables 20-4
 - recablear 9-21
 - sin símbolo 14-9
 - Operandos no utilizados 14-9
 - visualizar 14-9
 - Operandos sin símbolo 14-8
 - visualizar 14-9
 - Optimización de la plantilla para la traducción 6-22
 - Optimización del proceso de traducción 6-23
 - Ordenar
 - la lista de referencias cruzadas 14-2
 - símbolos 8-14
 - Organización de las ventanas 5-29
 - cambiar 5-29
 - guardar 5-30
 - restablecer 5-30
 - OUT (declaración de variables) A-64
- P**
- PAA (imagen del proceso de las salidas) A-20
 - PAE (imagen del proceso de las entradas) A-20
 - Palabra (WORD)
 - margen A-33
 - tipo de datos A-32
 - Palabra doble (DWORD)
 - margen A-33
 - tipo de datos A-32
 - Paquetes de software 6-15
 - para toda la CPU 16-11
 - para todo el proyecto 16-11
 - PARADA A-14
 - estado operativo de la CPU A-1
 - Parametrización indirecta A-121
 - Parametrizar
 - con SFCs A-121
 - con STEP 7 A-121, A-122
 - indirectamente A-121
 - módulos de señales aptos para alarmas de proceso 4-33
 - reloj A-124
 - Parametrizar bloques de datos 12-1

- Parametrizar el interface PG/PC 2-11
- Parametrizar funciones tecnológicas 12-2
- Parámetro de salida
 - evaluar RET_VAL 23-26
- Parámetros
 - atributos 9-21
- Parámetros actuales 4-16
- Parámetros de CPU "Carga del ciclo por comunicaciones" 4-11
- Parámetros de entrada A-64, A-65
- Parámetros de entrada/salida A-64
- Parámetros de salida A-64
- Parámetros del módulo A-121
 - transmitir con SFCs A-121
 - transmitir con STEP 7 A-121
- Parámetros del sistema A-120
- Parámetros formales 16-8
 - atributos de sistema y bloques de notificación 16-8
- Parámetros IN_OUT de un bloque de función A-71
- PARAM_MOD A-117, A-121
- Partes de un aviso 16-5
- Particularidades al imprimir el árbol de objetos 24-3
- Pasos
 - para comprimir el contenido de la memoria de una CPU S7 19-20
 - para visualizar y cambiar el estado operativo 18-8
- Pasos de programación
 - S7 1-1
- PC SIMATIC - Utilizar configuraciones de versiones anteriores 7-4
- Periferia
 - áreas de direccionamiento A-116
- Periferia descentralizada 7-1, 7-3
- Permiso de acceso 18-6
- PG/PC
 - editar los bloques cargados 19-15
- Pila de bloques (BSTACK) A-26
- Pila de datos locales A-24, A-25
- Pila de interrupción (USTACK) A-25
- Pila de interrupciones A-15
- Pila L A-24
 - sobreescribir la pila L A-24
- Pila LSTACK
 - almacenamiento de variables temporales 4-18
- Planificar una solución de automatización 3-1
 - crear un diagrama de E/S
 - para las válvulas 3-7
 - crear un diagrama de E/S
 - para los motores 3-7
 - crear un esquema de configuración 3-10
 - definir los requerimientos de seguridad 3-8
 - describir los elementos necesarios de manejo y visualización 3-9
 - listado de entradas y salidas 3-6
- Procedimiento básico 3-1
 - Tareas y áreas
 - subdividir el proceso en 3-2
- Plantilla 16-9, 16-10
- Plantilla y avisos 16-9
- Plantillas de bloques
 - insertar en fuentes AWL 13-15
- POINTER A-51
 - tipo de parámetro A-51
- Posibilidades de ampliar el software estándar
 - STEP 1-12
- Posibilidades de carga A-17
- Posibilidades de visualización
 - para avisos de CPU y avisos de diagnóstico definidos por el usuario 16-39
- Posicionamiento rápido en los puntos de aplicación del programa 14-11
- Preajustes para el editor de programas
 - KOP/FUP/AWL 10-4
- Preferencia (operando simbólico/absoluto) 8-5
- Preferencia de operandos
 - (simbólica/absoluta) 8-5
- Prevenir daños a personas 20-21
- Prevenir daños materiales 20-21
- Prioridad 4-28
 - alarma de proceso 4-33
 - alarma de retardo 4-30
 - alarma horaria 4-28
 - cambiar 4-5
 - OB de tarea no prioritaria 4-36
- Procedimiento 25-1
 - para archivar y desarchivar 24-6
 - para introducir instrucciones 10-12
 - para los sistemas M7 25-1
- Procedimiento básico
 - para averiguar la causa de un STOP 23-16
 - para crear bloques lógicos 10-3
 - para imprimir 24-2
 - para observar y forzar variables 20-2
- Procedimiento básico para planificar una solución de automatización 3-1
- Procedimiento de instalación 2-8
- Procedimiento de notificación
 - selección 16-3
- Procedimiento de notificación
 - por bits 16-1, 16-2
- Procedimiento para sistemas M7 25-1
- Procedimiento por números de avisos 16-2
- Procedimientos de carga en función de la memoria de carga 19-4
- Procedimientos para cambiar la asignación de números de aviso de un proyecto 16-11
- Proceso 4-11, 4-13, 4-14, 4-15, 4-16
 - subdividir 3-3, A-78
 - subdividir en áreas de tareas a la vista de un proceso de mezcla industrial 3-2
- Procesos de comunicaciones 4-11
- PROFIBUS DP 7-3
- PROFIBUS-DP 7-1

- Profundidad de anidamiento 4-9
 - Programa CFC 25-1
 - Programa de arranque 4-34
 - Programa de ejemplo
 - proceso de mezcla industrial A-77
 - Programa de simulación 22-1
 - Programa de usuario
 - cargar A-17
 - Elementos 4-2
 - en la memoria de la CPU A-16
 - tareas 4-1
 - Programa estructurado
 - diseñar A-80
 - Ventajas 4-2
 - Programa M7
 - insertar 6-12, 6-13
 - Programa S7
 - insertar 6-12
 - Programa S7/M7 sin equipo ni CPU 5-19
 - Programación
 - diseñar un programa estructurado A-80
 - transferencia de parámetros 4-18
 - utilización de bloques de datos 4-18
 - Programación estructurada 4-3
 - Programación lineal 4-8
 - Programar
 - FB
 - ejemplo A-85, A-87
 - OB de tarea no prioritaria 4-36
 - un OB 1
 - ejemplo A-91, A-92
 - una FC
 - ejemplo A-89
 - Programas de ejemplo A-75
 - insertar valores de sustitución 23-32
 - reaccionar a fallos de la pila 23-27
 - valores de sustitución 23-32
 - Programas de una CPU 4-1
 - Programas de usuario
 - cargar en el sistema de destino 19-4
 - Propiedad del bloque 15-3
 - fecha y hora 15-3
 - Propiedades de bloques
 - visualizar longitudes de bloques 9-17
 - Propiedades de la carpeta de bloques
 - visualizar longitudes de bloques 9-17
 - Propiedades del bloque 9-16, 10-3
 - Propiedades válidas para los diferentes tipos de bloques 13-7
 - Protección de acceso 6-3
 - Protección por contraseña para acceder a sistemas de destino 18-6
 - Proyecto 5-6
 - abrir 6-14
 - borrar 6-14
 - con comunicación de datos globales
 - archivar
 - convertir
 - cambiar de nombre A-74
 - copiar 6-14
 - crear con el Asistente 6-9
 - crear manualmente 6-9
 - imprimir la documentación 24-1
 - Proyectos 5-23
 - archivar 24-4
 - cambiar el nombre 5-23
 - orden de elaboración 6-9
 - reorganizar 26-2
 - Proyectos (versión 2) editar 7-1
 - Proyectos antiguos
 - utilizar A-72, A-73
 - Proyectos con protección de acceso 6-3
 - Proyectos con un gran número de equipos en red 26-1
 - Proyectos de ejemplo A-75, A-76
 - Proyectos de la versión 1
 - convertir A-72
 - Proyectos de la versión 2
 - convertir A-73
 - Prueba de testigos
 - (ver test de parpadeo) 18-2
 - Puntero A-53, A-54, A-55, A-56
 - Puntero nulo A-58
 - PZF (error de acceso a la periferia) A-20
- ## Q
- QRY_TINT 4-28
- ## R
- RAM A-15, A-29, A-30
 - RAM no volátil A-29
 - RAM NV (no volátil) A-29
 - Rama T 10-23
 - RDSYSST 23-19, 23-20, A-28
 - READ_CLK A-124
 - READ_RTM A-125
 - Real
 - margen A-33
 - tipo de datos A-33, A-35
 - Rearranque A-1, A-2, A-5, A-6, A-7, A-8, A-9, A-10
 - automático A-6, A-7
 - interrumpir A-5
 - manual A-5, A-6
 - Rearranque completo A-1, A-5, A-6, A-7, A-9, A-10
 - automático A-5
 - sin respaldo por pila A-5
 - interrumpir A-10
 - manual A-5
 - Recablear
 - bloques 9-21
 - operandos 9-21
 - Recargar bloques en el sistema de destino 19-6

- Registro
 - acceder a A-117, A-118, A-121
 - escribir A-118
 - leer A-118
 - Registros de bloques de datos A-26
 - Reglas
 - alarma de proceso 4-33
 - alarma de retardo 4-30
 - para declarar multiinstancias 10-10
 - para exportar tablas de símbolos 8-17
 - para FUP 10-23
 - para importar tablas de símbolos 8-17
 - para introducir instrucciones AWL 10-25
 - para introducir operaciones FUP 10-23
 - para introducir operaciones KOP 10-18
 - para KOP 10-18
 - para la declarar variables en fuentes AWL 13-3
 - para la introducción de instrucciones en fuentes AWL 13-2
 - para la manipulación de claves de licencia 2-5
 - Reglas de seguridad para el forzado permanente de variables 20-20
 - Reglas para definir las propiedades de bloques en fuentes AWL 13-5
 - Reglas para definir los atributos de sistema en fuentes AWL 13-4
 - Reglas para la manipulación de claves de licencia 2-5
 - Reglas sobre el orden de los bloques en fuentes AWL 13-4
 - Regulación 4-28
 - alarma cíclica 4-31
 - alarma horaria 4-28
 - Relación
 - entre los estados operativos de la CPU A-1
 - Reloj A-124, A-125
 - leer A-124, A-125
 - parametrizar A-124
 - sincronizar A-125
 - Relojes de la CPU con ajuste de zonas horarias 18-9
 - Remanencia
 - tras corte de alimentación A-5
 - Reorganizar proyectos y librerías 26-2
 - Representación
 - elementos FUP 10-22
 - elementos KOP 10-18
 - símbolos globales y locales 8-4
 - Representación de módulos desconocidos 7-6
 - Representación numérica
 - bit A-32
 - byte A-32
 - entero (16 bits) A-34
 - entero (32 bits) A-34
 - fecha y hora (DATE AND TIME) A-42
 - números decimales codificados en binario (BCD) A-39
 - números en coma flotante A-35, A-36, A-37
 - palabra A-32
 - palabra doble A-32
 - S5 TIME A-40
 - requerimiento de datos locales 14-3, 14-4
 - Requerimientos de seguridad
 - descripción para el ejemplo de un proceso de mezcla industrial 3-8
 - Requisitos
 - para cargar 19-1
 - para poder archivar 24-6
 - Requisitos para la instalación 2-6
 - Requisitos y observaciones para cargar 19-8
 - Restablecer
 - organización de las ventanas 5-30
 - Retardar
 - eventos de arranque 4-39
 - RPL_VAL 23-32
 - RUN A-13
 - actividades de la CPU A-11
 - estado operativo de la CPU A-1
- S**
- S5 TIME
 - base de tiempo A-40
 - formato A-40
 - margen A-33
 - S5TIME
 - tipo de datos A-40
 - S7-CFC 9-11
 - S7-GRAPH 9-8
 - S7-HiGraph 9-9
 - S7-SCL 9-7
 - Salidas
 - imagen del proceso A-20
 - listar 3-6
 - Plano de ocupación 14-5
 - SCL 9-2
 - sdf 6-26
 - Segmentos 9-5
 - finalización en KOP 10-18
 - Selección del procedimiento de notificación 16-3
 - Seleccionar
 - método de creación de programas 9-1
 - SET_CLK 4-29, A-124
 - SET_CLKSK 18-9
 - SET_RTM A-125
 - SET_TINT 4-28, 4-29
 - SFB 4-25, A-41
 - SFB 20 STOP 4-11
 - SFB 33 16-7
 - SFB 34 16-7
 - SFB 35 16-6, 16-7
 - SFB 36 16-6, 16-7
 - SFB 3767 16-6

- SFC 4-25
 - utilizar A-21
- SFC 0 SET_CLK 4-28, A-124
- SFC 1 READ_CLK A-124
- SFC 2 SET_RTM A-124
- SFC 3 CTRL_RTM A-124
- SFC 4 READ_RTM A-124
- SFC 13 DPNRM_DG A-120
- SFC 14 DPRD_DAT A-120
- SFC 15 DPWR_DAT A-120
- SFC 17/18 16-6
- SFC 20 BLKMOV A-17
- SFC 22 CREAT_DB A-16
- SFC 26 UPDAT_PI 4-11, A-20
- SFC 27 UPDAT_PO 4-14, A-20
- SFC 28 SET_TINT 4-28
 - ejemplo en AWL A-97
- SFC 29 CAN_TINT 4-28
 - ejemplo en AWL A-97
- SFC 30 ACT_TINT 4-28
 - ejemplo en AWL A-97
- SFC 31 QRY_TINT 4-28
 - ejemplo en AWL A-97
- SFC 32 SRT_DINT
 - ejemplo en AWL A-105
- SFC 33 CAN_DINT
 - ejemplo en AWL A-105
- SFC 34 QRY_DINT
 - ejemplo en AWL A-105
- SFC 36 MSK_FLT 4-38
 - ejemplo en AWL A-111
 - ejemplo en KOP A-111
- SFC 37 DMSK_FLT 4-38
 - ejemplo en AWL A-111
 - ejemplo en KOP A-111
- SFC 38 READ_ERR
 - ejemplo en AWL A-111
 - ejemplo en KOP A-111
- SFC 39 DIS_IRT 4-38
 - ejemplo en AWL A-114
- SFC 40 EN_IRT 4-38
 - ejemplo en AWL A-114
- SFC 41 DIS_AIRT 4-38
 - ejemplo en AWL A-115
- SFC 42 EN_AIRT 4-38
 - ejemplo en AWL A-115
- SFC 44 RPL_VAL 23-32
- SFC 46 STP 4-12
- SFC 48 SNC_RTCB A-124
- SFC 51 RDSYSST 23-19, 23-20, A-27
- SFC 52 WR_USMSG 23-23
- SFC 55 WR_PARM A-117, A-121
- SFC 56 WR_DPARM A-117, A-121
- SFC 57 PARM_MOD A-117, A-121
- SFC 82 6-24
- SFC 83 6-24
- SFC 84 6-24
- Signo de comentario 20-4
- SIMATIC
 - Equipo PC 7-4
- Símbolo para módulos desconocidos 7-6
- Símbolos 8-4, 8-15, 8-16
 - definir al introducir el programa 8-13
 - en la estructura del programa 14-3
 - filtrar 8-14
 - globales 8-3
 - insertar en una tabla de variables 20-4
 - introducir 8-14
 - locales 8-3
 - mayúsculas y minúsculas 8-15
 - no utilizados 14-7
 - ordenar 8-14
- Símbolos de diagnóstico
 - en la vista online 23-3
- Símbolos de objetos en el Administrador SIMATIC 5-4
- Símbolos globales
 - introducir en la tabla de símbolos 8-14
 - introducir en un cuadro de diálogo 8-13
 - introducir en un programa 10-13
- Símbolos globales y locales 8-3
- Símbolos incompletos y ambiguos en la tabla de símbolos 8-11
- Símbolos no utilizados 14-7
- Simulación de una CPU o de un módulo de señales 22-1
- Sin tensión
 - estado operativo de la CPU A-1
- Sincronización horaria
 - SFC 100 'SET_CLKS' 18-9
- Sincronizar A-124
 - reloj A-124
- Sinopsis
 - Bloques de notificación 16-6
 - de los datos de referencia 14-1
- Sintaxis de bloques en fuentes AWL 13-10
- Sistema de destino 19-6
 - cargar bloques modificados 19-6
- Sistema de notificación 16-1
- Sistema operativo
 - tareas 4-1
- Sistema operativo de la CPU 4-15, A-123
- Sistemas operativos para M7-300/400 25-5
- SlotPLC 6-25
- SNC_RTCB A-124
- Software estándar 1-6
- software opcional 22-1
- Software opcional 25-3, 25-4
- Software opcional para la programación M7 25-3
- Software PLC 6-25
- Software Runtime 1-15

- Solución de automatización
 - planificar
 - crear un diagrama de E/S para las válvulas 3-7
 - crear un diagrama de E/S para los motores 3-7
 - crear un esquema de configuración 3-10
 - definir los requerimientos de seguridad 3-8
 - planificar
 - describir las áreas de funcionamiento 3-4
 - elementos de manejo y visualización describir 3-9
 - listar entradas y salidas 3-6
 - proceso
 - subdividir en áreas y tareas 3-2
- Soporte de datos 6-25
- SRT_DINT 4-30
- STAT (declaración de variables) A-64
- STEP 7 1-6
 - desinstalar 2-13
 - errores durante la instalación 2-10
 - iniciar el software 5-1
 - Instalación 2-6, 2-7
 - interface de usuario 5-21
 - lenguajes de programación 1-6, 1-8
 - OBs de error
 - reaccionar a errores 4-38
 - software estándar 1-6, 1-8
- STOP A-4
 - averiguar la causa 23-16
 - estado operativo de la CPU A-1
- STRING A-41
- STRUCT A-41
- Subdividir el proceso en tareas y áreas 3-2
- Subdividir un proceso en áreas de tareas a la vista de un proceso de mezcla industrial 3-2
- Subdivisión de la memoria en áreas A-15
- Sustituir módulos 26-1
- Sustituir módulos en la tabla de configuración 26-1
- SZL 23-20, 23-21

- T**
- Tabla de declaración de variables 10-3, 10-6
 - atributos de sistema para parámetros 10-6
 - FB para el ejemplo de un proceso de mezcla industrial A-84
 - FC para el ejemplo de un proceso de mezcla industrial A-89
 - OB para el ejemplo de un proceso de mezcla industrial A-91
 - tarea 10-6
- Tabla de declaración de variables para el OB81 23-27
- Tabla de símbolos 8-4
 - abrir 8-14
 - configurar atributos M+V 17-4
 - direcciones admisibles 8-10
 - estructura y componentes 8-8
 - formatos de archivos
 - para importar/exportar 8-17
 - para símbolos globales 8-8
 - tipos de datos admisibles 8-10
- Tabla de variables 20-3
 - comprobación de sintaxis 20-4
 - copiar/mover 20-3
 - editar 20-6
 - ejemplo 20-4
 - ejemplo de introducción de operandos 20-10
 - guardar 20-1, 20-3, 20-4
 - insertar operandos o símbolos 20-4
 - insertar un área de operandos conexos 20-7
 - tamaño máximo 20-6
 - uso 20-1
- Tarjeta MPI en la PG/el PC 2-11
- Tarjeta MPI-ISA (Auto) 2-11
- TEMP (declaración de variables) A-64
- Temporizadores 14-5, 14-6
 - Plano de ocupación 14-5
 - valores máximos de introducción 20-8
- Temporizadores (T) A-126
 - área de memoria remanente A-29
- Test 21-1, 21-2
 - con el estado del programa 21-1
 - con el programa de simulación (software opcional) 22-1
 - definir el modo de funcionamiento 21-10
- Test con el estado del programa 21-1
- Test con la tabla de variables 26-3
- Test de parpadeo 18-2
- Test de programas con la tabla de variables 20-1
- Texto fuente 13-15
- Textos para operarios
 - exportar/importar 16-32
 - requisitos 16-32
 - v. Textos de usuario 16-32
- Tiempo de ciclo 4-11
- Tiempo de ciclo de OB 1 4-11
- Tiempo de ciclo máximo 4-11
- Tiempo de interrupción A-5, A-10
- Tiempo de vigilancia de ciclo 4-11
- Tiempo mínimo de ciclo 4-11
- Tiempos de vigilancia 4-35
- TIMER A-51, A-52
 - tipo de parámetro A-51

- Tipo de datos
 - DATE AND TIME: fecha y hora A-42
 - DWORD A-39
 - REAL:número en coma flotante A-36
 - S5 TIME A-40
 - UDT 9-13
 - WORD A-39
- Tipo de declaración
 - modificar 10-8
- tipo de parámetro
 - TIMER A-51
- Tipo de parámetro
 - ANY A-51
 - BLOCK_DB A-51
 - BLOCK_FB A-51
 - BLOCK_FC A-51
 - BLOCK_SDB A-51
 - COUNTER A-51
 - POINTER A-51
- Tipos datos
 - STRUCT A-41
- Tipos de alarmas 4-3
- Tipos de datos
 - ANY A-58
 - ARRAY A-41
 - BOOL A-32, A-33
 - byte A-32
 - BYTE A-33
 - carácter (CHAR) A-33
 - compuestos A-41
 - DATE_AND_TIME A-41
 - definidos por le usuario A-41
 - descripción A-33
 - DINT:entero (32 bits) A-34
 - entero (16 bits) (INT) A-33
 - entero (32 bits) (DINT) A-33
 - FB
 - SFB 4-18, A-41
 - fecha A-33
 - hora (TIME OF DAY) A-33
 - INT:entero (16 bits) A-34
 - introducción A-32
 - palabra A-32
 - palabra (WORD) A-33
 - palabra doble A-32
 - palabra doble (DWORD) A-33
 - real (REAL) A-33
 - S5 TIME A-33
 - simples A-33
 - STRING A-41
 - tiempo (TIME) A-33
 - tipos de parámetros
 - ANY
 - parámetro A-61, A-62, A-63
 - UDT A-41
- Tipos de datos admisibles al transferir
 - parámetros A-66
- Tipos de datos compuestos A-41
- Tipos de datos de usuario
 - crear A-49, A-50
 - descripción A-49
 - introducir la estructura 11-6
- Tipos de datos de usuario (UDT) 9-13
- Tipos de datos estructurados A-41, A-43
 - array A-44, A-45, A-46, A-47
 - estructuras de anidado y arrays A-43
 - estructura A-47
 - estructuras de anidado y arrays A-43
- Tipos de datos simples A-33
- Tipos de licencia 2-3
 - Enterprise License 2-1
 - Floating License 2-3
 - Rental License 2-3
 - Trial License 2-3
 - Upgrade License 2-3
- Tipos de parámetros A-51
 - ANY A-61, A-62, A-63
 - COUNTER A-51
 - introducción A-32
- Tipos de textos gestionados
 - en varios idiomas 6-17
- Títulos
 - de bloques 10-14
 - de segmentos 10-14
- Títulos de bloques 10-14
- Títulos de segmentos 10-14
- TPA (véase Imagen del proceso) A-20
- Trabajar
 - con librerías 9-22
- Trabajar con plantillas de segmentos 10-16
- Trabajar con proyectos y librerías V2 7-1
- Traducir librerías de textos 16-36
- Traducir y editar
 - textos de usuario 16-32
- Traducir y editar textos de usuario 16-32
- Transferencia a parámetros IN_OUT
 - de un FB A-71
- Transferencia de parámetros
 - almacenamiento de los
 - valores transferidos 4-18
- Transferir los datos de configuración al
 - sistema de destino 16-38
- Transferir los datos de configuración al
 - sistema de destino M+V 17-6
- Transferir parámetros
 - diseñar parámetros para un programa
 - estructurado A-84
 - FB para el ejemplo de un proceso de mezcla
 - industrial A-84
 - tipos de parámetros A-51
- Transmisión de informaciones de diagnóstico
 - 23-19
- Tratamiento de fallos 23-25
- Tratamiento retardado de eventos de alarma y
 - de error asíncrono
 - ejemplo A-115

U

- UDT 9-13, A-41
 - corregir el interface 15-6
- UDTs
 - introducir la estructura 11-6
- UDTs en fuentes AWL
 - ejemplo 13-27
- UPDAT_PI 4-14, A-20
- UPDAT_PO 4-11, A-20
- Update (sistema operativo de la CPU) A-123
- Uso
 - arrays para acceder a los datos A-44
 - estructuras para acceder a los datos A-47
 - tipo de parámetro ANY A-61
 - tipo de parámetro POINTER A-53
 - tipos de datos compuestos A-43
 - tipos de datos de usuario para acceder a los datos A-49
- Uso de las áreas de memoria del sistema A-18
- Uso del tipo de parámetro POINTER A-53
- USTACK
 - descripción A-25
 - utilización por la memoria del sistema A-25
- Utilidad de las funciones de reloj A-124
- Utilización de la declaración de variables en los bloques lógicos 10-6
- Utilización de multiinstancias 10-9
- Utilizar
 - marcas de ciclo y temporizadores A-126
 - proyectos antiguos A-73
 - SFC A-21, A-22
- Utilizar juegos de caracteres de otros idiomas 6-5
- Utilizar la Micro Memory Card como soporte de datos 6-25
- Utilizar proyectos antiguos A-72

V

- Valor asociado
 - borrar 16-31
 - insertar avisos 16-28
- Valores de datos
 - cambiar en la vista "Datos" de los bloques de datos 11-8
 - inicializar 11-8
- Valores de forzado
 - insertar 20-7
- Valores de forzado normal
 - ejemplos de introducción 20-12
- Valores de forzado permanente
 - ejemplos de introducción 20-12
- Valores de sustitución
 - utilización de SFC44 (RPL_VAL) 23-32
- Valores máximos para introducir contadores 20-9

- Valores máximos para introducir temporizadores 20-8
- Válvulas
 - crear un diagrama de E/S 3-7
- Variables 20-17
 - forzar 20-17
 - manejo y visualización 17-1
 - observar 20-15
- Variables temporales A-64, A-65
- Ventana "Estaciones accesibles" 18-2
- Ventana de declaración de variables
 - Introducir multiinstancia 10-10
- Ventana del proyecto 6-1, 6-2
- Ventanas
 - cambiar de una a otra 5-36
- Ver datos de los bloques de datos 11-3
- Ver declaración de bloques de datos 11-2
- Versión 2
 - modificar proyectos y librerías a posteriori 7-1
- Vigilancia de la configuración teórica/real de los módulos
 - OBs de arranque 4-34
- Vigilancia del proceso 20-2
- Vista del proyecto 6-2
- Vista detallada de variables 10-7
 - Estructura 10-8
- Vista online
 - símbolos de diagnóstico 23-3, 23-4
- Visualizar avisos de CPU y avisos de diagnóstico definidos por el usuario 16-39
- Visualización
 - activar los símbolos en el bloque 8-13
 - bloque borrado 14-3
 - definir para el estado del programa 21-9
 - estructura en árbol 14-3
 - estructura por padres e hijos 14-3
 - requerimiento máximo de datos locales en la estructura en árbol 14-3
- Visualización del estado de programas 21-3
- Visualizar
 - datos de referencia 14-9, 14-10
 - estado operativo 18-8
 - estructura de bloques de datos con FB asociado (DBs de instancia) 11-5
 - estructura de bloques de datos con UDT asociado 11-7
 - estructura del programa 14-9
 - información del bloque en KOP
 - FUP
 - AWL 14-8
 - información del módulo 23-1
 - listas en ventanas de trabajo adicionales 14-9
 - longitudes de bloques 9-17
 - operandos no utilizados 14-9
 - operandos sin símbolo 14-9

Visualizar el informe de cambios 6-4
Visualizar longitudes de bloques 9-17
Visualizar los avisos de
 CPU guardados 16-42
Volumen de información del módulo en función
 del tipo de módulo 23-12

W

WinAC 6-25

WinLC 6-25
WORD
 tipo de datos A-39
WR_DPARM A-117, A-121
WR_PARM A-117, A-121
WR_USMSG 23-23

Z

Zonas horarias 18-9

