Conceptos de programación, convenciones y funciones

5

El S7-200 ejecuta continuamente el programa para controlar una tarea o un proceso. El programa se crea con STEP 7-Micro/WIN y se carga en el S7-200. STEP 7-Micro/WIN ofrece diversas herramientas y funciones para crear, implementar y comprobar el programa de usuario.

Índice del capítulo

Crear una solución de automatización con un Micro-PLC

Existen diversos métodos para crear una solución de automatización con un Micro-PLC. Las reglas generales siguientes se pueden aplicar a numerosos proyectos. No obstante, también deberá tener en cuenta las reglas de su empresa y su propia experiencia.

Estructurar el proceso o la máquina

Divida el proceso o la máquina en secciones independientes. Estas secciones determinan los límites entre los diversos sistemas de automatización e influyen en las descripciones de las áreas de funciones y en la asignación de recursos.

Especificar las unidades funcionales

Describa las funciones de cada sección del proceso o de la máquina. Considere los siguientes aspectos: entradas y salidas, descripción de la operación, estados que deben alcanzarse antes de ejecutar funciones con cada uno de los actuadores (electroválvulas, motores, accionamientos, etc.), descripción de la interfaz de operador y de las interfaces con otras secciones del proceso o de la máquina.

Diseñar los circuitos de seguridad

Determine qué aparatos requieren un cableado permanente por motivos de seguridad. Si fallan los sistemas de automatización, puede ocurrir un arranque inesperado o un cambio en el funcionamiento de las máquinas. En este caso, pueden producirse lesiones graves o daños materiales. Por tanto, es preciso utilizar dispositivos de protección contra sobrecargas electromecánicas que funcionen independientemente del S7-200, evitando así las condiciones inseguras. Para diseñar los circuitos de seguridad:

- ❏ Defina el funcionamiento erróneo o inesperado de los actuadores que pudieran causar peligros.
- ❏ Defina las condiciones que garanticen un funcionamiento seguro y determine cómo detectar esas condiciones, independientemente del S7-200.
- ❏ Defina cómo el S7-200 y los módulos de ampliación deben influir en el proceso cuando se conecte y desconecte la alimentación, así como al detectarse errores. Esta información se debe utilizar sólo para diseñar el funcionamiento normal y el funcionamiento anormal esperado, sin poderse aplicar para fines de seguridad.
- ❏ Prevea dispositivos de parada de emergencia manual o de protección contra sobrecargas electromagnéticas que impidan un funcionamiento peligroso, independientemente del S7-200.
- ❏ Desde los circuitos independientes, provea información de estado apropiada al S7-200 para que el programa y las interfaces de operador dispongan de los datos necesarios.
- ❏ Defina otros requisitos adicionales de seguridad para que el proceso se lleve a cabo de forma segura y fiable.

Definir las estaciones de operador

Conforme a las funciones exigidas, cree planos de las estaciones de operador considerando los aspectos siguientes:

- ❏ Panorámica de la ubicación de todas las estaciones de operador con respecto al proceso o máquina
- ❏ Disposición mecánica de los componentes (pantalla, interruptores y lámparas) de la estación de operador
- ❏ Esquemas eléctricos con las correspondientes E/S de la CPU S7-200 o del módulo de ampliación

Crear los planos de configuración

Conforme a las funciones exigidas, cree planos de configuración del sistema de automatización considerando los aspectos siguientes:

- ❏ Panorámica de la ubicación de todos los S7-200 con respecto al proceso o máquina
- ❏ Disposición mecánica de los S7-200 y módulos de ampliación (incluyendo armarios, etc.)
- ❏ Esquemas eléctricos de todos los S7-200 y módulos de ampliación (incluyendo los números de referencia, las direcciones de comunicación y las direcciones de E/S).

Crear una lista de nombres simbólicos (opcional)

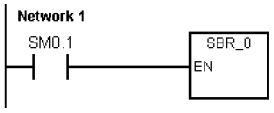
Si desea utilizar nombres simbólicos para el direccionamiento, elabore una lista de nombres simbólicos para las direcciones absolutas. Incluya no sólo las E/S físicas, sino también todos los demás elementos a utilizar en el programa.

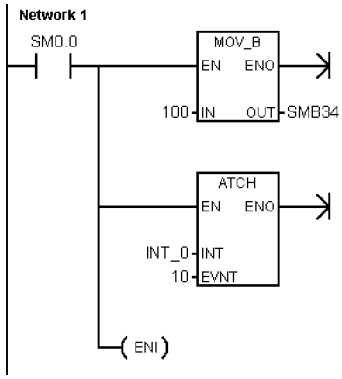
Elementos básicos de un programa

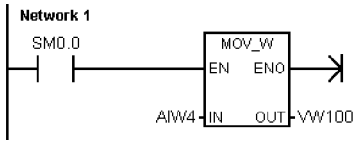
Un bloque de programa se compone del código ejecutable y los comentarios.
El código ejecutable comprende el programa principal, así como subrutinas y/o rutinas de interrupción (opcionales). El código se compila y se carga en el S7-200, a excepción de los comentarios del programa. Las unidades de organización (programa principal, subrutinas y rutinas de interrupción) sirven para estructurar el programa de control.

El programa de ejemplo siguiente incluye una subrutina y una rutina de interrupción.
Este programa utiliza una interrupción temporizada para leer el valor de una entrada analógica cada 100 ms.

Elementos básicos de un programa de control

PRINCIPAL		Network 1 //Llamar a la subrutina 0 en el //primer ciclo. LD SM0.1 CALL SBR_0

SBR0		Network 1 //Ajustar a 100 ms el intervalo
		//de la interrupción temporizada.
		//Habilitar la interrupción 0.
		LD SM0.0
		MOVB 100, SMB34
		ATCH INT_0, 10
		ENI

INT0		Network 1 //Muestrear la entrada analógica 4. LD SM0.0 MOVW AIW4,VW100

Programa principal

Esta parte del programa contiene las operaciones que controlan la aplicación. El S7-200 ejecuta estas operaciones en orden secuencial una vez por ciclo. El programa principal se denomina también OB1.

Subrutinas

Estos elementos opcionales del programa se ejecutan sólo cuando se llaman desde el programa principal, desde una rutina de interrupción, o bien desde otra subrutina. Las subrutinas son elementos opcionales del programa, adecuándose para funciones que se deban ejecutar repetidamente. Así, en vez de tener que escribir la lógica en cada posición del programa principal donde se deba ejecutar una función, basta con escribirla sólo una vez en una subrutina y llamar a la subrutina desde el programa principal cada vez que sea necesario. Las subrutinas tienen varias ventajas:

- La utilización de subrutinas permite reducir el tamaño total del programa.
- La utilización de subrutinas acorta el tiempo de ciclo, puesto que el código se ha extraído del programa principal. El S7-200 evalúa el código del programa principal en cada ciclo, sin importar si el código se ejecuta o no. Sin embargo, el S7-200 evalúa el código en la subrutina sólo si se llama a ésta. En cambio, no lo evalúa en los ciclos en los que no se llame a la subrutina.
- La utilización de subrutinas crea códigos portátiles. Es posible aislar el código de una función en una subrutina y copiar ésta a otros programas sin necesidad de efectuar cambios o con sólo pocas modificaciones.

Consejo

La utilización de direcciones de la memoria V limita la portabilidad de las subrutinas, ya que la asignación de direcciones de un programa en la memoria V puede estar en conflicto con la asignación en un programa diferente. En cambio, las subrutinas que utilizan la tabla de variables locales (memoria L) para todas las asignaciones de direcciones se pueden transportar muy fácilmente, puesto que no presentan el riesgo de conflictos de direcciones entre la subrutina y otra parte del programa.

Rutinas de interrupción

Estos elementos opcionales del programa reaccionan a determinados eventos de interrupción. Las rutinas de interrupción se pueden programar para gestionar eventos de interrupción predefinidos. El S7-200 ejecuta una rutina de interrupción cuando ocurre el evento asociado.

El programa principal no llama a las rutinas de interrupción. Una rutina de interrupción se asocia a un evento de interrupción y el S7-200 ejecuta las operaciones contenidas en esa rutina sólo cada vez que ocurra el evento en cuestión.

Consejo

Puesto que no es posible saber con anterioridad cuándo el S7-200 generará una interrupción, es recomendable limitar el número de variables utilizadas tanto por la rutina de interrupción como en otra parte del programa.

Utilice la tabla de variables locales de la rutina de interrupción para garantizar que ésta utilice únicamente la memoria temporal, de manera que no se sobrescriban los datos utilizados en ninguna otra parte del programa.

Hay diversas técnicas de programación que se pueden utilizar para garantizar que el programa principal y las rutinas de interrupción compartan los datos correctamente. Estas técnicas se describen en el capítulo 6 en relación con las operaciones de interrupción.

Otros elementos del programa

Hay otros bloques que contienen información para el S7-200. A la hora de cargar el programa en la CPU, es posible indicar qué bloques se deben cargar asimismo.

Bloque de sistema

Bloque de sistema

El bloque de sistema permite configurar diversas opciones de hardware para el S7-200.

Bloque de datos

Bloque de datos

En el bloque de datos se almacenan los valores de las diferentes variables (memoria V) utilizadas en el programa. Este bloque se puede usar para introducir los valores iniciales de los datos.

Utilizar STEP 7-Micro/WIN para crear programas

Para iniciar STEP 7-Micro/WIN, haga doble clic en el icono de STEP 7-Micro/WIN o elija los comandos **Inicio** > **SIMATIC** > **STEP 7-Micro/WIN V4.0**. Como muestra la figura 5-1, STEP 7-Micro/WIN ofrece una interfaz de usuario cómoda para crear el programa de control.

Las barras de herramientas contienen botones de método abreviado para los comandos de menú de uso frecuente. Estas barras se pueden mostrar u ocultar.

La barra de navegación comprende iconos que permiten acceder a las diversas funciones de programación de STEP 7-Micro/WIN.

En el árbol de operaciones se visualizan todos los objetos del proyecto y las operaciones para crear el programa de control. Para insertar operaciones en el programa, puede utilizar el método de "arrastrar y soltar" desde el árbol de operaciones, o bien hacer doble clic en una operación con objeto de insertarla en la posición actual del cursor en el editor de programas.

El editor de programas contiene el programa y una tabla de variables locales donde se pueden asignar nombres simbólicos a las variables locales temporales. Las subrutinas y las rutinas de interrupción se visualizan en forma de fichas en el borde inferior del editor de programas. Para acceder a las subrutinas, a las rutinas de interrupción o al programa principal, haga clic en la ficha en cuestión.

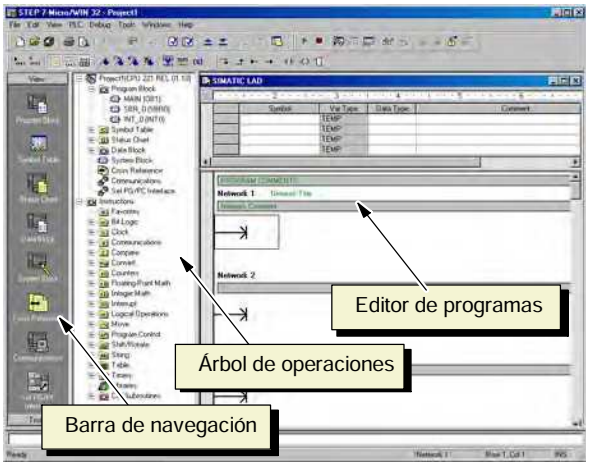


Figura 5-1 STEP 7-Micro/WIN


Editor de programas

STEP 7-Micro/WIN ofrece tres editores para crear programas: Esquema de contactos (KOP), Lista de instrucciones (AWL) y Diagrama de funciones (FUP). Con algunas restricciones, los programas creados con uno de estos editores se pueden visualizar y editar con los demás.

Funciones del editor AWL

El editor AWL visualiza el programa textualmente. Permite crear programas de control introduciendo la nemotécnica de las operaciones. El editor AWL sirve para crear ciertos programas que, de otra forma, no se podrían programar con los editores KOP ni FUP. Ello se debe a que AWL es el lenguaje nativo del S7-200, a diferencia de los editores gráficos, sujetos a ciertas restricciones para poder dibujar los diagramas correctamente. Como muestra la figura 5-2, esta forma textual es muy similar a la programación en lenguaje ensamblador.

El S7-200 ejecuta cada operación en el orden determinado por el programa, de arriba a abajo, reiniciando después arriba.

AWL utiliza una pila lógica para resolver la lógica de control. El usuario inserta las operaciones AWL para procesar las operaciones de pila.

```
LD    I0.0    //Leer una entrada
A     I0.1    //AND con otra entrada
=     Q1.0    //Escribir el valor en la salida 1
```

Figura 5-2 Programa de ejemplo AWL

Considere los siguientes aspectos importantes al seleccionar el editor AWL:

- El lenguaje AWL es más apropiado para los programadores expertos.
- En algunos casos, AWL permite solucionar problemas que no se podrían resolver fácilmente con los editores KOP o FUP.
- El editor AWL soporta sólo el juego de operaciones SIMATIC.
- En tanto que el editor AWL se puede utilizar siempre para ver o editar programas creados con los editores KOP o FUP, lo contrario no es posible en todos los casos. Los editores KOP o FUP no siempre permiten para visualizar un programa que se haya creado en AWL.

Funciones del editor KOP

El editor KOP visualiza el programa gráficamente, de forma similar a un esquema de circuitos. Los programas KOP hacen que el programa emule la circulación de corriente eléctrica desde una fuente de alimentación, a través de una serie de condiciones lógicas de entrada que, a su vez, habilitan condiciones lógicas de salida. Los programas KOP incluyen una barra de alimentación izquierda que está energizada. Los contactos cerrados permiten que la corriente circule por ellos hasta el siguiente elemento, en tanto que los contactos abiertos bloquean el flujo de energía.

La lógica se divide en segmentos ("networks"). El programa se ejecuta un segmento tras otro, de izquierda a derecha y luego de arriba abajo. La figura 5-3 muestra un ejemplo de un programa KOP. Las operaciones se representan mediante símbolos gráficos que incluyen tres formas básicas.

Los contactos representan condiciones lógicas de entrada, tales como interruptores, botones o condiciones internas.

Las bobinas representan condiciones lógicas de salida, tales como lámparas, arrancadores de motor, relés interpuestos o condiciones internas de salida.

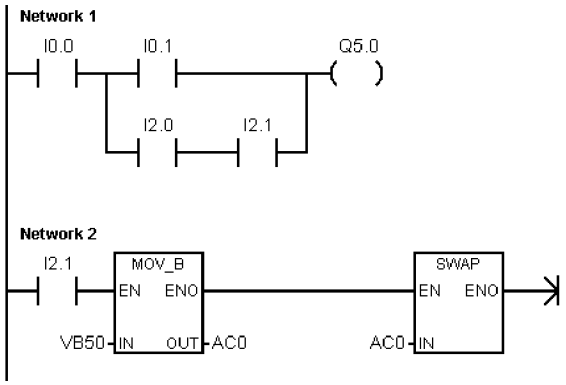


Figura 5-3 Programa de ejemplo KOP

Los cuadros representan operaciones adicionales, tales como temporizadores, contadores u operaciones aritméticas.

Considere los siguientes aspectos importantes al seleccionar el editor KOP:

- El lenguaje KOP facilita el trabajo a los programadores principiantes.
- La representación gráfica es fácil de comprender, siendo popular en el mundo entero.
- El editor KOP se puede utilizar con los juegos de operaciones SIMATIC e IEC 1131-3.
- El editor AWL se puede utilizar siempre para visualizar un programa creado con el editor SIMATIC KOP.

Funciones del editor FUP

El editor FUP visualiza el programa gráficamente, de forma similar a los circuitos de puertas lógicas. En FUP no existen contactos ni bobinas como en el editor KOP, pero sí hay operaciones equivalentes que se representan en forma de cuadros.

La figura 5-4 muestra un ejemplo de un programa FUP.

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

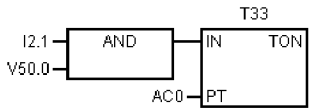


Figura 5-4 Programa de ejemplo FUP

La ruta "1" lógica por los elementos FUP se denomina circulación de corriente. El origen de una entrada de circulación de corriente y el destino de una salida de circulación de corriente se pueden asignar directamente a un operando.

La lógica del programa se deriva de las conexiones entre las operaciones de cuadro. Así pues, la salida de una operación (p. ej. un cuadro AND) se puede utilizar para habilitar otra operación (p. ej. un temporizador), con objeto de crear la lógica de control necesaria. Estas conexiones permiten solucionar numerosos problemas lógicos.

Considere los siguientes aspectos importantes al seleccionar el editor FUP:

- El estilo de representación en forma de puertas gráficas es apropiado para observar el flujo del programa.
- El editor FUP soporta los juegos de operaciones SIMATIC e IEC 1131-3.
- El editor AWL se puede utilizar siempre para visualizar un programa creado con el editor SIMATIC FUP.

Juegos de operaciones SIMATIC e IEC 1131-3

La mayoría de los sistemas de automatización ofrecen los mismos tipos básicos de operaciones. No obstante, existen pequeñas diferencias en cuanto al aspecto, operación, etc. de los productos de los distintos fabricantes. Durante los últimos años, la Comisión Electrotécnica Internacional (CEI) o International Electrotechnical Commission (IEC) ha desarrollado una norma global dedicada a numerosos aspectos de la programación de autómatas programables (denominados "sistemas de automatización" en la terminología SIMATIC). El objetivo de esta norma es que los diferentes fabricantes de autómatas programables ofrezcan operaciones similares tanto en su aspecto como en su funcionamiento.

El S7-200 ofrece dos juegos de operaciones que permiten solucionar una gran variedad de tareas de automatización. El juego de operaciones IEC cumple con la norma IEC 11313 para la programación de autómatas programables (PLCs), en tanto que el juego de operaciones SIMATIC se ha diseñado especialmente para el S7-200.

Consejo
Si en STEP 7-Micro/WIN está ajustado el modo IEC, en el árbol de operaciones se visualiza un diamante rojo (♦) junto a las operaciones no definidas en la norma IEC 1131-3.

Existen algunas diferencias básicas entre los juegos de operaciones SIMATIC e IEC:

- El juego de operaciones IEC se limita a las operaciones estándar comunes entre los fabricantes de autómatas programables. Algunas operaciones incluidas en el juego SIMATIC no están normalizadas en la norma IEC 1131-3. Éstas se pueden utilizar en calidad de operaciones no normalizadas. No obstante, en este caso, el programa ya no será absolutamente compatible con la norma IEC 1131-3.
- Algunos cuadros IEC soportan varios formatos de datos. A menudo, ésto se denomina sobrecarga. Por ejemplo, en lugar de tener cuadros aritméticos por separado, tales como ADD_I (Sumar enteros), ADD_R (Sumar reales) etc., la operación ADD definida en la norma IEC examina el formato de los datos a sumar y selecciona automáticamente la operación correcta en el S7-200. Así se puede ahorrar tiempo al diseñar los programas.
- Si se utilizan las operaciones IEC, se comprueba automáticamente si los parámetros de la operación corresponden al formato de datos correcto (p. ej. entero con signo o entero sin signo). Por ejemplo, si ha intentado introducir un valor de entero en una operación para la que se deba utilizar un valor binario (on/off), se indicará un error. Esta función permite reducir los errores de sintaxis de programación.

Considere los siguientes aspectos a la hora de seleccionar el juego de operaciones (SIMATIC o IEC):

- Por lo general, el tiempo de ejecución de las operaciones SIMATIC es más breve. Es posible que el tiempo de ejecución de algunas operaciones IEC sea más prolongado.
- El funcionamiento de algunas operaciones IEC (p. ej. temporizadores, contadores, multiplicación y división) es diferente al de sus equivalentes en SIMATIC.
- Las operaciones SIMATIC se pueden utilizar en los tres editores de programas disponibles (KOP, AWL y FUP). Las operaciones IEC sólo se pueden utilizar en los editores KOP y FUP.
- El funcionamiento de las operaciones IEC es igual en las diferentes marcas de autómatas programables (PLCs). Los conocimientos acerca de cómo crear un programa compatible con la norma IEC se pueden nivelar a lo largo de las plataformas de PLCs.
- Aunque la norma IEC define una menor cantidad de operaciones de las disponibles en el juego de operaciones SIMATIC, en los programas IEC se pueden incluir siempre también operaciones SIMATIC.
- La norma IEC 11313 especifica que las variables se deben declarar tipificadas, soportando que el sistema verifique el tipo de datos.

Convenciones utilizadas en los editores de programas

En todos los editores de programas de STEP 7-Micro/WIN rigen las convenciones siguientes:

- Si un nombre simbólico (p. ej. #var1) va antecedido de un signo de número (#), significa que se trata de un símbolo local.
- En las operaciones IEC, el símbolo % identifica una dirección directa.
- El símbolo de operando "?.?" ó "????" indica que el operando se debe configurar.

Los programas KOP se dividen en segmentos denominados "networks". Un segmento o "network" es una red organizada, compuesta por contactos, bobinas y cuadros que se interconectan para formar un circuito completo. No se permiten los cortocircuitos, ni los circuitos abiertos, ni la circulación de corriente inversa. STEP 7-Micro/WIN ofrece la posibilidad de crear comentarios para cada uno de los segmentos del programa KOP. El lenguaje FUP utiliza el concepto de segmentos para subdividir y comentar el programa.

Los programas AWL no utilizan segmentos. Sin embargo, la palabra clave NETWORK se puede utilizar para estructurar el programa.

Convenciones específicas del editor KOP

En el editor KOP, las teclas de función F4, F6 y F9 sirven para acceder a los contactos, los cuadros y las bobinas. El editor KOP utiliza las convenciones siguientes:

- El símbolo ">>" representa un circuito abierto o una conexión necesaria para la circulación de corriente.
- El símbolo "→|" indica que la salida es una conexión opcional para la circulación de corriente en una operación que se puede disponer en cascada o conectar en serie.
- El símbolo ">>" indica que se puede utilizar la circulación de corriente.

Convenciones específicas del editor FUP

En el editor FUP, las teclas de función F4, F6 y F9 sirven para acceder a las operaciones AND y OR, así como a las operaciones con cuadros. El editor FUP utiliza las convenciones siguientes:

- El símbolo ">>" en un operando EN es un indicador de circulación de corriente o de operando. También puede representar un circuito abierto o una conexión necesaria para la circulación de corriente.
- El símbolo "→|" indica que la salida es una conexión opcional para la circulación de corriente en una operación que se puede disponer en cascada o conectar en serie.
- Los símbolos "<<" y ">>" indican que se puede utilizar bien sea un valor, o bien la circulación de corriente.
- Símbolo de negación: La condición lógica NOT (la condición invertida) del operando o la corriente se representa mediante un círculo pequeño en la entrada. En la figura 5-5, Q0.0 es igual al NOT de I0.0 AND I0.1. Los símbolos de negación sólo son aplicables a las señales booleanas, que se pueden especificar en forma de parámetros o circulación de corriente.

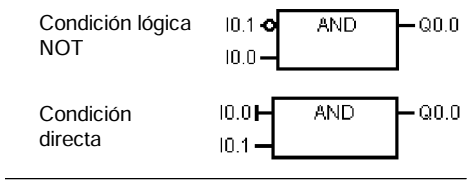


Figura 5-5 Convenciones FUP

- Indicadores directos: Como muestra la figura 5-5, el editor FUP visualiza una condición directa de un operando booleano mediante una línea vertical en la entrada de una operación FUP. El indicador directo causa un lectura directa de la entrada física indicada. Los indicadores directos sólo son aplicables a las entradas físicas.
- Cuadro sin entradas ni salidas: Un cuadro sin entradas ni salidas indica que la operación no depende de la circulación de corriente.

Consejo

La cantidad de operandos se puede incrementar hasta 32 entradas en el caso de las operaciones AND y OR. Para agregar o quitar operandos, utilice las teclas "+" y "" del teclado, respectivamente.

Convenciones generales para programar el S7-200

Definición de EN/ENO

EN (entrada de habilitación) es una entrada booleana para los cuadros KOP y FUP. Para que la operación se pueda ejecutar debe circular corriente por esta entrada. En AWL, las operaciones no tienen una entrada EN, pero el valor del nivel superior de la pila debe ser un "1" lógico para poder ejecutar la respectiva operación AWL.

ENO (salida de habilitación) es una salida booleana para los cuadros KOP y FUP. Si circula corriente por la entrada EN y el cuadro ejecuta la función sin errores, la salida ENO conducirá corriente al siguiente elemento. Si se detecta un error en la ejecución del cuadro, la circulación de corriente se detendrá en el cuadro que ha generado el error.

En AWL no existe la salida ENO, pero las operaciones AWL correspondientes a las funciones KOP y FUP con salidas ENO activan un bit ENO especial. A este bit se accede mediante la operación AND ENO (AENO), pudiendo utilizarse para generar el mismo efecto que el bit ENO de un cuadro.

Consejo

Los operandos y los tipos de datos EN/ENO no figuran en la tabla de operandos válidos de las operaciones, puesto que son idénticos para todas las operaciones KOP y FUP. La tabla 5-1 muestra los operandos y tipos de datos EN/ENO para KOP y FUP. siendo aplicables a todas las operaciones KOP y FUP descritas en el presente manual.

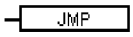
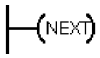
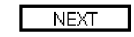
Tabla 5-1 Operandos y tipos de datos EN/ENO para KOP y FUP

Editor de programas	Entradas/salidas	Operandos	Tipos de datos
KOP	EN, ENO	Circulación de corriente	BOOL
FUP	EN, ENO	I, Q, V, M, SM, S, T, C, L	BOOL

Entradas condicionadas e incondicionadas

En KOP y FUP, un cuadro o una bobina que dependa de la circulación de corriente aparece conectado cualquier elemento a la izquierda. Una bobina o un cuadro que no dependa de la circulación de corriente se muestra con una conexión directa a la barra de alimentación izquierda. La tabla 5-2 muestra dos entradas: una condicionada y otra incondicionada.

Tabla 5-2 Representación de entradas condicionadas e incondicionadas

Circulación de corriente	KOP	FUP
Operación dependiente de la circulación de corriente (condicionada)		
Operación independiente de la circulación de corriente (incondicionada)		

Operaciones sin salidas

Los cuadros que no se puedan conectar en cascada se representan sin salidas booleanas. Estos cuadros incluyen las llamadas a subrutinas, JMP y CRET. También hay bobinas KOP que sólo se pueden disponer en la barra de alimentación izquierda, incluyendo las operaciones Definir meta, NEXT, Cargar relé de control secuencial, Fin condicionado del relé de control secuencial y Fin del relé de control secuencial. Estas operaciones se representan en FUP en forma de cuadros con entradas sin meta y sin salidas.

Operaciones de comparación

Las operaciones de comparación se ejecutan sin tener en cuenta el estado de señal. Si el estado es FALSO, el estado de señal de la salida también será FALSO. Si el estado de señal es VERDADERO, la salida se activará dependiendo del resultado de la comparación.
Las operaciones de comparación FUP (SIMATIC), LD (IEC) y FBD (IEC) se representan con cuadros, aunque la operación se ejecute en forma de contacto.

Utilizar asistentes para facilitar la creación de programas

STEP 7-Micro/WIN provee asistentes para facilitar y automatizar algunas funciones de programación. En el capítulo 6, las operaciones que tienen un asistente asociado se identifican con el icono siguiente:

Asistente de operaciones

Eliminar errores en el S7-200

En el S7-200 pueden ocurrir errores fatales o no fatales. Para visualizar los códigos generados por los errores, elija el comando de menú **CPU > Información**.

La figura 5-6 muestra el cuadro de diálogo "Información CPU". Allí se visualizan el código y la descripción del error.

El campo "Último fatal" muestra el último código de error fatal generado por el S7-200. Al desconectarse la alimentación, este valor se conserva si se respalda la RAM. El valor se borra si se efectúa un borrado total del S7-200 o si la RAM no se respalda tras un corte prolongado de la alimentación.

El campo "Total fatales" muestra el contaje total de errores fatales generados por el S7-200 desde la última vez que se efectuó un borrado total de la memoria. Al desconectarse la alimentación, este valor se conserva si se respalda la RAM. El valor se borra si se efectúa un borrado total del S7-200 o si la RAM no se respalda tras un corte prolongado de la alimentación.

En el anexo C figuran los códigos de error del S7-200 y en el anexo D se describen las marcas especiales (SM) que pueden utilizarse para detectar errores.

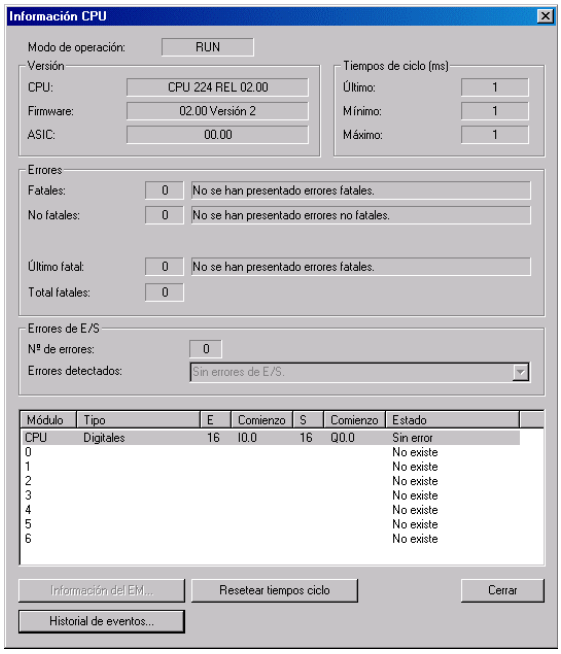


Figura 5-6 Cuadro de diálogo "Información CPU"

Errores no fatales

Los errores no fatales indican problemas relativos a la estructura del programa de usuario, la ejecución de una operación en el programa de usuario o los módulos de ampliación.
STEP 7-Micro/WIN permite visualizar los códigos generados por los errores no fatales. Hay tres categorías básicas de errores no fatales.

Errores de compilación del programa

Al cargar un programa en el S7-200, éste lo compila. Si el S7-200 detecta una violación de las reglas de compilación, el proceso de carga se suspenderá, generándose entonces un código de error. (Si ya se ha cargado un programa en el S7-200, seguirá existiendo en la EEPROM, por lo que no se perderá). Una vez corregido el programa, se podrá cargar de nuevo. El anexo C contiene una lista de violaciones de las reglas de compilación.

Errores de E/S

Al arrancar, el S7-200 lee la configuración de E/S de todos los módulos. Durante el funcionamiento normal, el S7-200 comprueba periódicamente el estado de todos los módulos y lo compara con la configuración obtenida durante el arranque. Si el S7-200 detecta una diferencia, activará el bit de error de configuración en el registro de errores del módulo. El S7-200 no lee datos de las entradas ni escribe datos en las salidas de ese módulo hasta que la configuración concuerde de nuevo con la obtenida durante el arranque.

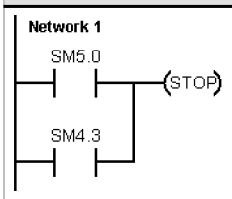
La información de estado del módulo se guarda en marcas especiales (SM). El programa puede supervisar y evaluar estas marcas. En el anexo D encontrará más información acerca de las marcas especiales utilizadas para indicar los errores de E/S. SM5.0 es la marca global de errores de E/S, permaneciendo activada mientras exista una condición de error en un módulo de ampliación.

Errores de programación en el tiempo de ejecución

El programa puede crear condiciones de error mientras se está ejecutando. Estos errores pueden ocurrir debido al uso incorrecto de una operación, o bien si una operación procesa datos no válidos. Por ejemplo, un puntero de direccionamiento indirecto que era válido cuando se compiló el programa puede haber cambiado durante la ejecución del programa, señalando entonces a una dirección fuera de área. Este es un ejemplo de un error de programación en el tiempo de ejecución. La marca especial SM4.3 se activa al ocurrir este error y permanece activada mientras que el S7-200 se encuentre en modo RUN. (El anexo C incluye una lista de los errores de programación en el tiempo de ejecución). La información de los errores de ejecución del programa se guarda en marcas especiales (SM). El programa puede supervisar y evaluar estas marcas. En el anexo D encontrará más información acerca de las marcas especiales utilizadas para indicar los errores de ejecución del programa.

El S7-200 no cambia a modo STOP cuando detecta un error no fatal. Tan sólo deposita el evento en la marca especial en cuestión y continúa ejecutando el programa. No obstante, es posible programar que el S7200 cambie a modo STOP cuando se detecte un error no fatal. El siguiente programa de ejemplo muestra un segmento de un programa que supervisa las dos marcas globales de errores no fatales, cambiando el S7-200 a STOP cuando se active una de esas marcas.

Programa de ejemplo: Lógica para detectar una condición de error no fatal




```
Network 1       //Si ocurre un error de E/S o durante el tiempo
                //de ejecución, pasar a modo STOP.
LD      SM5.0
O       SM4.3
STOP
```

Errores fatales

Cuando ocurre un error fatal, el S7-200 detiene la ejecución del programa. Según la gravedad del error, es posible que el S7-200 no pueda ejecutar todas las funciones, o incluso ninguna de ellas. El objetivo del tratamiento de errores fatales es conducir al S7-200 a un estado seguro, en el que se puedan analizar y eliminar las condiciones que hayan causado el error. Cuando se detecta un error fatal, el S7-200 cambia a modo STOP, enciende los LEDs "SF/DIAG" (rojo) y "STOP", omite la tabla de salidas y desactiva las salidas. El S7-200 permanece en ese estado hasta que se haya eliminado la causa del error fatal.

Tras remediar las condiciones que causaron el error fatal, rearranque el S7-200 utilizando uno de los métodos siguientes:

- Desconecte la alimentación y vuelva a conectarla luego.
- Cambie el selector de modo de RUN o TERM a STOP.
- Elija el comando de menú **CPU > Reset al arrancar** en STEP 7-Micro/WIN para rearrancar el S7-200. Ello obliga al S7-200 a efectuar un rearranque y borrar todos los errores fatales.

Al rearrancar el S7-200 se borra la condición de error fatal y se ejecuta un diagnóstico de arranque para verificar si se ha corregido el error. En caso de detectarse otro error fatal, el S7-200 enciende de nuevo el LED de fallo. De lo contrario, el S7-200 comienza a funcionar con normalidad.

Algunas condiciones de error incapacitan al S7-200 para la comunicación. Estos errores indican fallos de hardware, por lo que es necesario reparar el S7-200. No se pueden solucionar modificando el programa ni borrando la memoria del S7-200.

Asignar direcciones y valores iniciales en el editor de bloque de datos

Bloque de datos

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

El editor de bloques de datos es un editor de texto de libre formato. Por tanto, no hay campos específicos definidos para un tipo determinado de información. Tras introducir una línea, pulse la tecla INTRO. El editor formatea la línea (alinea las columnas de direcciones, los datos y los comentarios; pone las direcciones de la memoria V en mayúsculas) y la visualiza de nuevo. Si pulsa CTRLINTRO, tras completar una línea de asignación, la dirección se incrementará automáticamente a la siguiente dirección disponible.

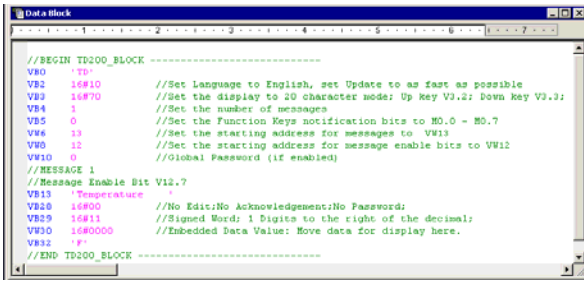


Figura 5-7 Editor de bloques de datos

El editor asigna una cantidad suficiente de la memoria V, en función de las direcciones que se hayan asignado previamente, así como del tamaño (byte, palabra o palabra doble) del (de los) valor(es) de datos.

La primera línea del bloque de datos debe contener una asignación de dirección explícita. Las líneas siguientes pueden contener asignaciones de direcciones explícitas o implícitas. El editor asigna una dirección implícita si se introducen varios valores de datos tras asignarse una sola dirección o si se introduce una línea que contenga únicamente valores de datos.

En el editor de bloques de datos pueden utilizarse mayúsculas y minúsculas. Además, es posible introducir comas, tabuladores y espacios que sirven de separadores entre las direcciones y los valores de datos.

Utilizar la tabla de símbolos para el direccionamiento simbólico de variables


Tabla de símbolos

En la tabla de símbolos es posible definir y editar los símbolos a los que pueden acceder los nombres simbólicos en cualquier parte del programa. Es posible crear varias tablas de símbolos. La tabla de símbolos incluye también una ficha que contiene los símbolos definidos por el sistema utilizables en el programa de usuario. La tabla de símbolos se denomina también tabla de variables globales.

A los operandos de las operaciones se pueden asignar direcciones absolutas o simbólicas. Una dirección absoluta utiliza el área de memoria y un bit o un byte para identificar la dirección. Una dirección simbólica utiliza una combinación de caracteres alfanuméricos para identificar la dirección.

En los programas SIMATIC, los símbolos globales se asignan utilizando la tabla de símbolos. En los programas IEC, los símbolos globales se asignan utilizando la tabla de variables globales.

			Symbol	Address	Comment
1			AlwaysOn	SM0.0	Always on contact
2			Pump1	Q2.3	Pump 1 on/off
3			Pump1Limit	I1.1	Pump 1 pressure limit switch
4			Pump1Pressure	VD100	Pump 1 current pressure (real)
5			Pump1Rpm	VW200	Pump1 PRMs (integer)
6					

Figura 5-8 Tabla de símbolos

Para asignar un símbolo a una dirección:

1. Haga clic en el icono "Tabla de símbolos" en la barra de navegación para abrir la tabla de símbolos.
2. En la columna "Nombre simbólico", teclee el nombre del símbolo (p. ej. "Entrada1"). Un nombre simbólico puede comprender 23 caracteres como máximo.
3. En la columna "Dirección", teclee la dirección (p. ej. I0.0).
4. Si está utilizando la tabla de variables globales (IEC), introduzca un valor en la columna "Tipo de datos" o seleccione uno del cuadro de lista.

Es posible crear varias tablas de símbolos. No obstante, una misma cadena no se puede utilizar más de una vez como símbolo global, ni en una misma tabla ni en tablas diferentes.

Utilizar variables locales

La tabla de variables locales del editor de programas se puede utilizar para asignar variables que existan únicamente en una subrutina o en una rutina de interrupción individual (v. fig. 5-9).

Las variables locales se pueden usar como parámetros que se transfieren a una subrutina, lo que permite incrementar la portabilidad y reutilización de la subrutina.

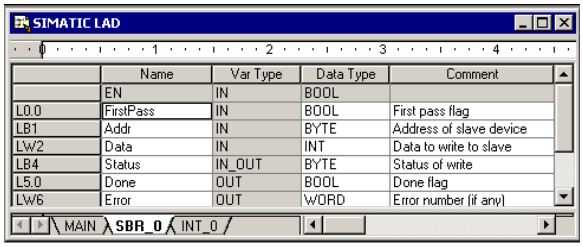


Figura 5-9 Tabla de variables locales

Utilizar la tabla de estado para supervisar el programa


Tabla de estado

La tabla de estado sirve para supervisar o modificar los valores de las variables del proceso a medida que el S7-200 ejecuta el programa de control. Es posible supervisar el estado de las entradas, salidas o variables del programa, visualizando para ello los valores actuales. La tabla de estado también permite forzar o modificar los valores de las variables del proceso.

Es posible crear varias tablas de estado para visualizar elementos de diferentes partes del programa.

Para acceder a la tabla de estado, elija el comando de menú **Ver > Componente > Tabla de estado** o haga clic en el icono "Tabla de estado" en la barra de navegación.

Al crear una tabla de estado es preciso introducir las direcciones de las variables del proceso a supervisar. No es posible visualizar el estado de las constantes, ni de los acumuladores, ni tampoco de las variables locales. Los valores de los temporizadores y contadores pueden visualizarse en formato de bit o palabra. En formato de bit, se visualiza el estado del bit del temporizador o del contador. En formato de palabra, se visualiza el valor del temporizador o del contador.

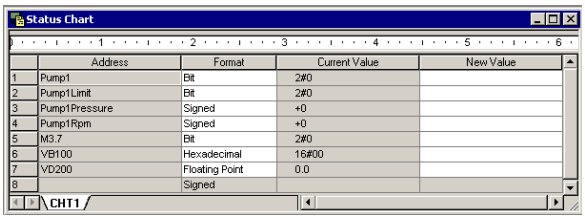


Figura 5-10 Tabla de estado

Para crear una tabla de estado y supervisar las variables:

1. En el campo "Dirección", introduzca la dirección de cada valor deseado.
2. En la columna "Formato", seleccione el tipo de datos.
3. Para visualizar el estado de las variables del proceso en el S7-200, elija el comando de menú **Test > Tabla de estado**.
4. Para muestrear continuamente los valores o para efectuar una sola lectura del estado, haga clic en el botón correspondiente en la barra de herramientas. La tabla de estado también permite modificar o forzar los valores de las variables del proceso.

Para insertar filas adicionales en la tabla de estado, elija los comandos de menú **Edición > Insertar > Fila**.

Consejo

Es posible crear varias tablas de estado para estructurar las variables en grupos lógicos, de manera que cada grupo se pueda visualizar por separado en una tabla de estado más pequeña.

Crear una librería de operaciones

STEP 7-Micro/WIN permite crear librerías de operaciones personalizadas, o bien utilizar una librería creada por otro usuario (v. fig. 5-11).

Para crear una librería de operaciones, es preciso generar subrutinas y rutinas de interrupción estándar en STEP 7-Micro/WIN y agruparlas luego. El código de estas rutinas se puede ocultar para que no sea modificado involuntariamente, o bien para proteger el knowhow del autor.

Para crear una librería de operaciones, proceda del siguiente modo:

1. Introduzca el programa en forma de proyecto estándar en STEP 7-Micro/WIN y deposite en una subrutina o en una rutina de interrupción la operación que desea incluir en la librería.
2. Asigne nombres simbólicos a todas las direcciones de la memoria V contenidas en las subrutinas o rutinas de interrupción. Para reducir la cantidad de memoria V que necesita la librería, utilice direcciones consecutivas de la memoria V.
3. Cambie los nombres de las subrutinas o de las rutinas de interrupción, indicando cómo deben aparecer en la librería de operaciones.
4. Elija el comando de menú **Archivo > Crear librería** para compilar la nueva librería de operaciones.

En la Ayuda en pantalla de STEP 7-Micro/WIN encontrará más información acerca de cómo crear librerías.

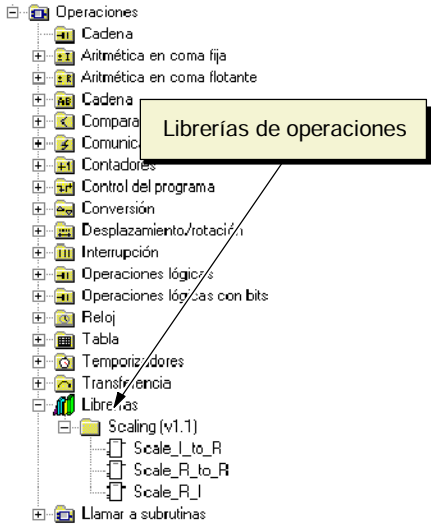


Figura 5-11 Árbol de operaciones con librerías

Para acceder a una operación contenida en una librería, proceda del siguiente modo:

1. Para agregar el directorio "Librerías" al árbol de operaciones, elija el comando de menú **Archivo > Agregar librerías**.
2. Seleccione la operación deseada e insértela en el programa (de igual manera que al insertar una operación estándar).

   Si la rutina de la librería necesita memoria V, una vez compilado el proyecto STEP 7-Micro/WIN le indicará que debe asignar un bloque de memoria. Utilice el cuadro de diálogo "Asignar memoria a librería" para asignar bloques de memoria.

Funciones para comprobar el programa

STEP 7-Micro/WIN provee las siguientes funciones para comprobar el programa:

- ❏ Los marcadores sirven para desplazarse fácilmente (hacia arriba y hacia abajo) por un programa extenso.
- ❏ La tabla de referencias cruzadas permite comprobar las referencias utilizadas en el programa.
- ❏ La edición en modo RUN se utiliza para efectuar cambios pequeños en el programa sin afectar demasiado a los equipos controlados. El bloque de programa también se puede cargar en la CPU durante la edición en modo RUN.

Para más información sobre cómo comprobar el programa, consulte el capítulo 8.