

SIEMENS



Documentación didáctica SCE

Siemens Automation Cooperates with Education | 05/2017

Módulo TIA Portal 032-100
Principios básicos de la programación
de FC con SIMATIC S7-1500

Cooperates
with Education

Automation

SIEMENS

Paquetes SCE apropiados para esta documentación didáctica

Controladores SIMATIC

- **SIMATIC ET 200SP Open Controller CPU 1515SP PC F y HMI RT SW**
Referencia.: 6ES7677-2FA41-4AB1
- **SIMATIC ET 200SP Distributed Controller CPU 1512SP F-1 PN Safety**
Referencia.: 6ES7512-1SK00-4AB2
- **SIMATIC CPU 1516F PN/DP Safety**
Referencia: 6ES7516-3FN00-4AB2
- **SIMATIC S7 CPU 1516-3 PN/DP**
Referencia.: 6ES7516-3AN00-4AB3
- **SIMATIC CPU 1512C PN con software y PM 1507**
Referencia: 6ES7512-1CK00-4AB1
- **SIMATIC CPU 1512C PN con software, PM 1507 y CP 1542-5 (PROFIBUS)**
Referencia: 6ES7512-1CK00-4AB2
- **SIMATIC CPU 1512C PN con software**
Referencia: 6ES7512-1CK00-4AB6
- **SIMATIC CPU 1512C PN con software y CP 1542-5 (PROFIBUS)**
Referencia: 6ES7512-1CK00-4AB7

SIMATIC STEP 7 Software for Training

- **SIMATIC STEP 7 Professional V14 SP1 - licencia individual**
Referencia: 6ES7822-1AA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - paq. 6, licencia de aula**
Referencia: 6ES7822-1BA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - paq. 6, licencia de actualización**
Referencia: 6ES7822-1AA04-4YE5
- **SIMATIC STEP 7 Professional V14 SP1 - paq. 20, licencia de estudiante**
Referencia: 6ES7822-1AC04-4YA5

Tenga en cuenta que estos paquetes SCE pueden sufrir cambios y ser sustituidos por paquetes actualizados

Encontrará una relación de los paquetes SCE actualmente disponibles en la página:

www.siemens.com/sce/tp y www.siemens.es/sce

Cursos avanzados

Para los cursos avanzados regionales de Siemens SCE, póngase en contacto con el partner SCE de su región www.siemens.com/sce/contact

Más información en torno a SCE

www.siemens.com/sce y www.siemens.es/sce

Nota sobre el uso

La documentación formativa SCE para la solución de automatización homogénea Totally Integrated Automation (TIA) ha sido elaborada para el programa "Siemens Automation Cooperates with Education (SCE)" exclusivamente con fines formativos para centros públicos de formación e I + D. Siemens AG declina toda responsabilidad en lo que respecta a su contenido.

No está permitido utilizar este documento más que para la iniciación a los productos o sistemas de Siemens. Es decir, está permitida su copia total o parcial y posterior entrega a los alumnos para que lo utilicen en el marco de su formación. La transmisión y reproducción de este documento y la comunicación de su contenido solo están permitidas dentro de centros de formación básica y avanzada para fines didácticos.

Las excepciones requieren autorización expresa por parte del siguiente contacto de Siemens AG:
Sr. Roland Scheuerer roland.scheuerer@siemens.com.

Los infractores quedan obligados a la indemnización por daños y perjuicios. Se reservan todos los derechos, incluidos los de traducción, especialmente para el caso de concesión de patentes o registro como modelo de utilidad.

No está permitido su uso para cursillos destinados a clientes del sector Industria. No aprobamos el uso comercial de los documentos.

Queremos expresar nuestro agradecimiento a la TU Dresde, en especial al catedrático Leon Urbas, así como a la empresa Michael Dziallas Engineering y a las demás personas que nos han prestado su apoyo para elaborar este documento didáctico de SCE.

Índice de contenido

1	Objetivos	5
2	Requisitos	5
3	Hardware y software necesarios	6
4	Teoría.....	7
4.1	Sistema operativo y programa de usuario	7
4.2	Bloques de organización	8
4.3	Imagen de proceso y ejecución cíclica del programa	9
4.4	Funciones	11
4.5	Bloques de función y bloques de datos de instancia	12
4.6	Bloques de datos globales	13
4.7	Bloques lógicos aptos para librería	14
4.8	Lenguajes de programación.....	15
5	Tarea planteada.....	16
6	Planificación.....	16
6.1	PARADA DE EMERGENCIA	16
6.2	Operación manual: motor de la cinta en modo Jog	16
7	Instrucciones paso a paso estructuradas	17
7.1	Desarchivación de un proyecto existente	17
7.2	Creación de una nueva tabla de variables.....	18
7.3	Creación de nuevas variables dentro de una tabla de variables	20
7.4	Importación de la "Tag table_sorting station (Tabla_variables_planta_clasificación)"	21
7.5	Creación de la función FC1 "MOTOR_MANUAL" para el motor de cinta en modo Jog.....	25
7.6	Definición de la interfaz de la función FC1 "MOTOR_MANUAL"	27
7.7	Programación del FC1: MOTOR_MANUAL.....	30
7.8	Programación del bloque de organización OB1: control de la marcha de la cinta hacia delante en modo manual.....	37
7.9	Programación del bloque de organización OB1: control de la marcha de la cinta hacia atrás en modo manual.....	42
7.10	Almacenamiento y compilación del programa	44
7.11	Carga del programa	45
7.12	Visualización de los bloques de programa.....	46
7.13	Archivación del proyecto	48
8	Lista de comprobación.....	49
9	Ejercicio	50
9.1	Tarea planteada: ejercicio	50
9.2	Planificación	50
9.3	Lista de comprobación: ejercicio	51
10	Información adicional.....	52

PRINCIPIOS BÁSICOS DE LA PROGRAMACIÓN DE FC

1 Objetivos

En este capítulo se familiarizará con los elementos básicos de un programa de control: los **bloques de organización (OB)**, las **funciones (FC)**, los **bloques de función (FB)** y los **bloques de datos (DB)**. Además se introduce la programación de funciones y de bloques de función **apta para librería**. Aprenderá el lenguaje de programación **Diagrama de funciones (FUP)** y lo utilizará para programar una función FC1 y un bloque de organización OB1.

Pueden utilizarse los controladores SIMATIC S7 indicados en el capítulo 3.

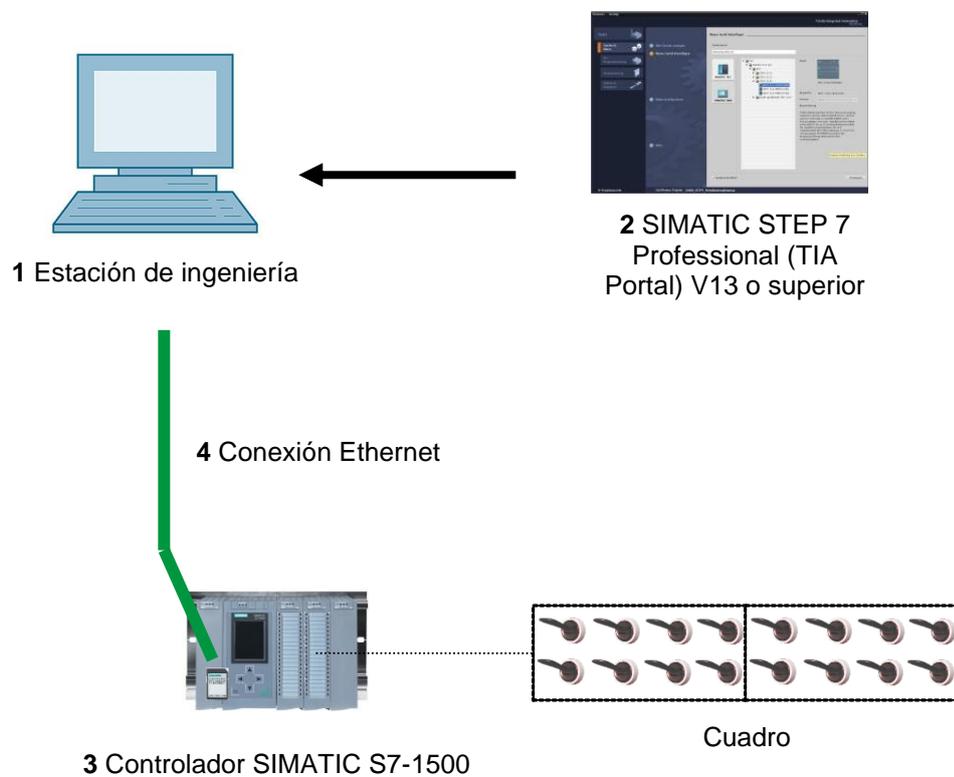
2 Requisitos

Este capítulo tiene como punto de partida la configuración hardware de SIMATIC S7 CPU1516F-3 PN/DP, pero también puede trabajarse con otras configuraciones hardware que incluyan tarjetas digitales de entrada y salida. Para poner en práctica este capítulo puede recurrir, p. ej., al siguiente proyecto:

SCE_ES_012_101 Hardware configuration CPU1516F.zap13

3 Hardware y software necesarios

- 1 Estación de ingeniería: Se requieren el hardware y el sistema operativo
(Para más información, ver Readme/Léame en los DVD de instalación del TIA portal)
- 2 SIMATIC Software STEP 7 Professional en el TIA Portal – V13 o superior
- 3 Controlador SIMATIC S7-1500/S7-1200/S7-300, p. ej., CPU 1516F-3 PN/DP –
firmware V1.6 o superior con Memory Card, 16 DI/16 DO y 2 AI/1 AO
Nota: Las entradas digitales deberían estar conectadas en un cuadro.
- 4 Conexión Ethernet entre la estación de ingeniería y el controlador



4 Teoría

4.1 Sistema operativo y programa de usuario

El **sistema operativo**, que forma parte de todo controlador (CPU), sirve para organizar todas las funciones y procesos de la CPU no relacionados con una tarea de control específica.

Algunas de las tareas del sistema operativo son p. ej.:

- Ejecutar un rearranque (en caliente)
- Actualización de las imágenes de proceso de las entradas y salidas
- Llamada cíclica del programa de usuario
- Registro de alarmas y llamada de los OBs de alarma
- Detección y tratamiento de errores
- Administración de áreas de memoria

El sistema operativo forma parte de la CPU y ya está contenido en ella en el momento de suministro.

El **programa de usuario** contiene todas las funciones necesarias para ejecutar la tarea de automatización específica. Algunas de las tareas del programa de usuario son:

- Comprobación de los requisitos previos para un rearranque completo (en caliente) con ayuda de OBs de arranque
- Procesamiento de datos de proceso, es decir, control de las señales de salida en función de los estados de las señales de entrada
- Reacción a alarmas y entradas de alarma
- Tratamiento de anomalías durante la ejecución normal del programa

4.2 Bloques de organización

Los bloques de organización (OB) constituyen la interfaz entre el sistema operativo del controlador (CPU) y el programa de usuario. Estos bloques son llamados por el sistema operativo y controlan los procesos siguientes:

- Ejecución cíclica (p. ej., OB1)
- Comportamiento en arranque del controlador
- Ejecución del programa controlada por alarmas
- Tratamiento de errores

En un proyecto debe existir por lo menos **un bloque de organización para la ejecución cíclica del programa**. Para llamar un OB se necesita un **evento de arranque**, como se muestra en la Figura 1. Los distintos OB tienen prioridades definidas, p. ej., para que un OB82 pueda interrumpir el OB1 cíclico a fin de resolver un error.

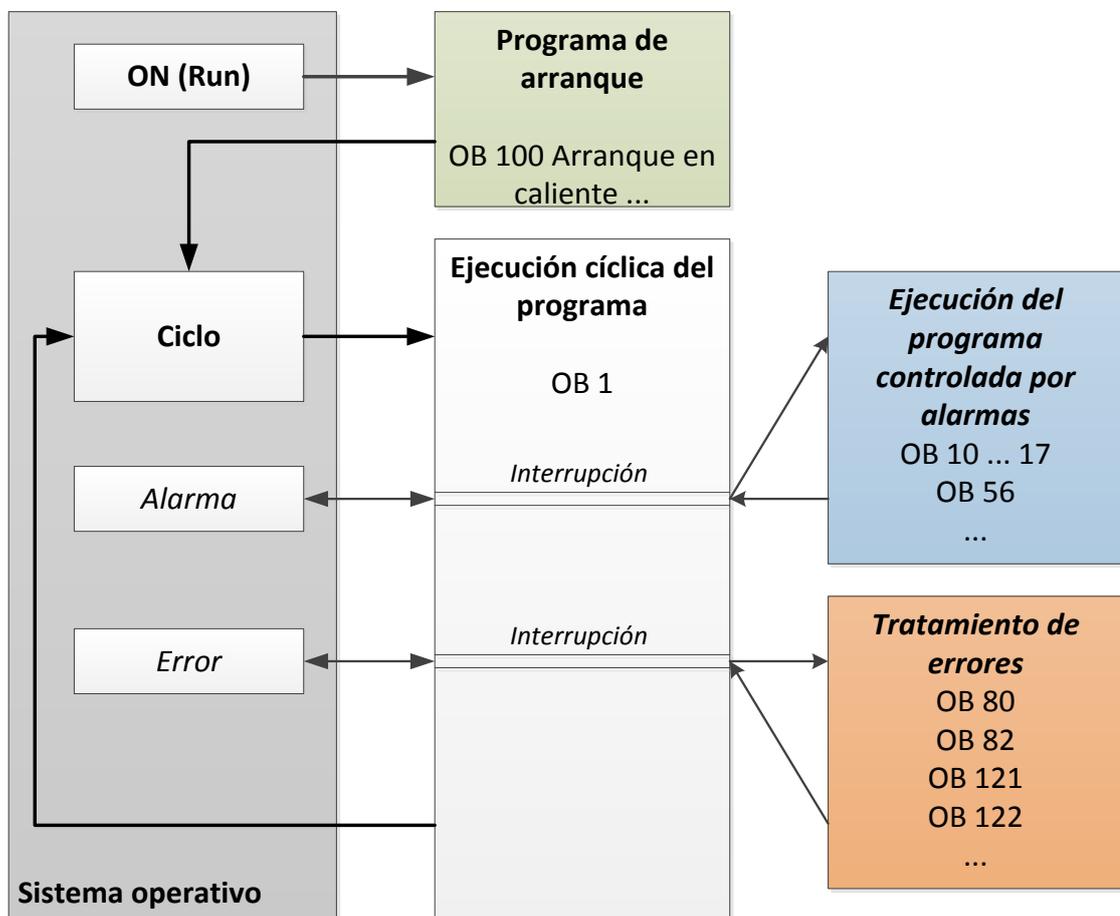


Figura 1: Eventos de arranque en el sistema operativo y llamadas de OB

Tras la aparición de un evento de arranque son posibles las siguientes reacciones:

- Si se ha sido asignado un OB al evento, este dispara la ejecución del OB asignado. Si la prioridad del OB asignado es superior a la prioridad del OB que acaba de ejecutarse, este se ejecuta de inmediato (interrupción). De lo contrario, se espera inicialmente hasta que se haya podido ejecutar el OB con la mayor prioridad.
- Si no se ha asignado ningún OB al evento, se ejecuta la reacción del sistema predeterminada.

La Tabla 1 ofrece algunos ejemplos de eventos de arranque para un SIMATIC S7-1500, sus posibles números de OB y la reacción predeterminada del sistema en caso de que el bloque de organización no se encuentre en el controlador.

Evento de arranque	Números de OB posibles	Reacción del sistema predeterminada
Arranque	100, ≥ 123	Ignorar
Programa cíclico	1, ≥ 123	Ignorar
Alarma horaria	10 a 17, ≥ 123	-
Alarma de actualización	56	Ignorar
Se ha excedido una vez el tiempo de vigilancia del ciclo	80	STOP
Alarma de diagnóstico	82	Ignorar
Error de programación	121	STOP
Error de acceso a la periferia	122	Ignorar

Tabla 1: Números de OB para distintos eventos de arranque

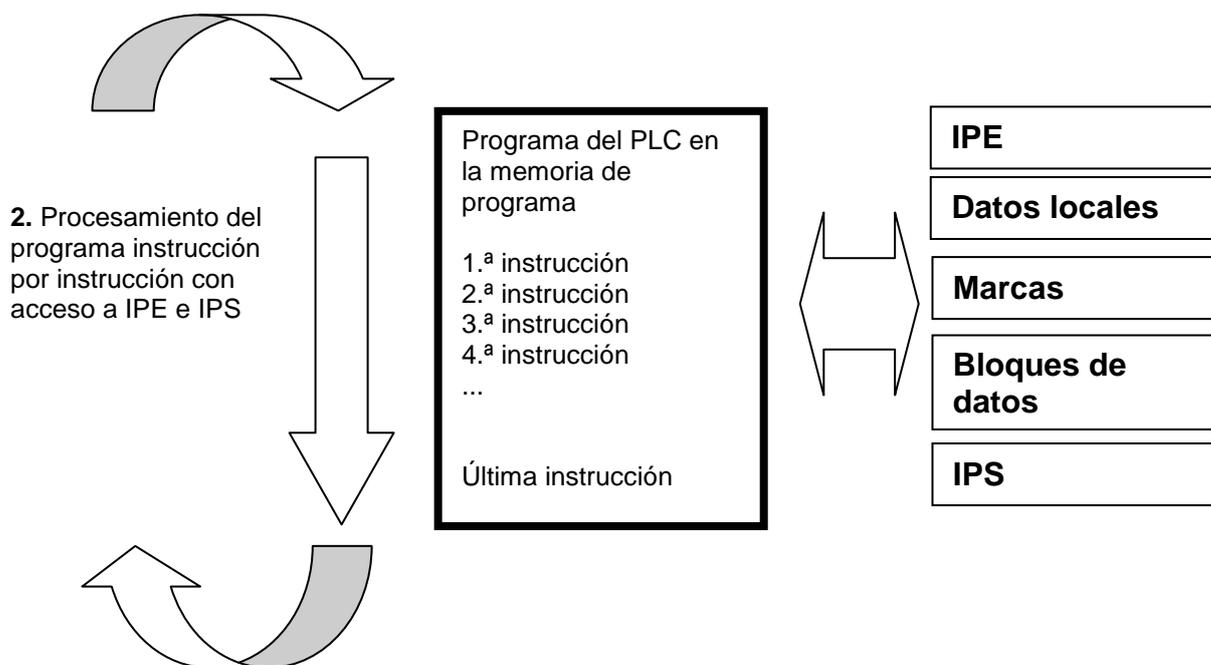
4.3 Imagen de proceso y ejecución cíclica del programa

Si en el programa de usuario cíclico se accede a las entradas (E) y salidas (S), normalmente no se consultan los estados de señal en los módulos digitales de entrada y salida, sino que se accede a un área de la memoria de sistema de la CPU. Esta área de memoria, que contiene una imagen de los estados de señal, se denomina **imagen de proceso**.

La ejecución cíclica del programa se realiza en el siguiente orden:

1. Al principio del programa cíclico se consulta si las distintas entradas están bajo tensión o no. Este estado de las entradas se guarda en la **imagen de proceso de las entradas (IPE)**. Para las entradas con tensión se guarda la información 1 o "Alta", y para las que no tienen tensión, la información 0 o "Baja".
2. Ahora el procesador ejecutará el programa almacenado en el bloque de organización cíclico. Para la información de entrada necesaria se accede a la **imagen de proceso de las entradas (IPE)** leída previamente, y el resultado lógico se escribe en la denominada **imagen de proceso de las salidas (IPS)**.
3. Al final del ciclo, la **imagen de proceso de las salidas (IPS)** se transfiere como estado lógico a los módulos de salida y estos se conectan o desconectan. A continuación se continúa de nuevo con el punto 1.

1. Guardar el estado de las entradas en la IPE.



3. Transmitir el estado de la IPS a las salidas.

Figura 2: Ejecución cíclica

Nota: El tiempo que requiere el procesador para esta secuencia se denomina tiempo de ciclo. Este tiempo varía en función del número y tipo de instrucciones, así como de la potencia del procesador del controlador.

4.4 Funciones

Las funciones (FC) son bloques lógicos sin memoria. **No poseen una memoria de datos** que permita almacenar valores de parámetros de bloque. Por este motivo, al llamar una función deben conectarse todos los parámetros de interfaz. Para guardar datos de forma permanente, deben crearse previamente bloques de datos globales.

Una función contiene un programa que se ejecuta cada vez que la función es llamada por otro bloque lógico.

Las funciones se pueden utilizar, p. ej., para los siguientes fines:

- Funciones matemáticas, que devuelven un resultado en función de los valores de entrada.
- Funciones tecnológicas, p. ej., controles individuales con operaciones lógicas binarias.

Una función también se puede llamar varias veces en diferentes puntos de un programa.

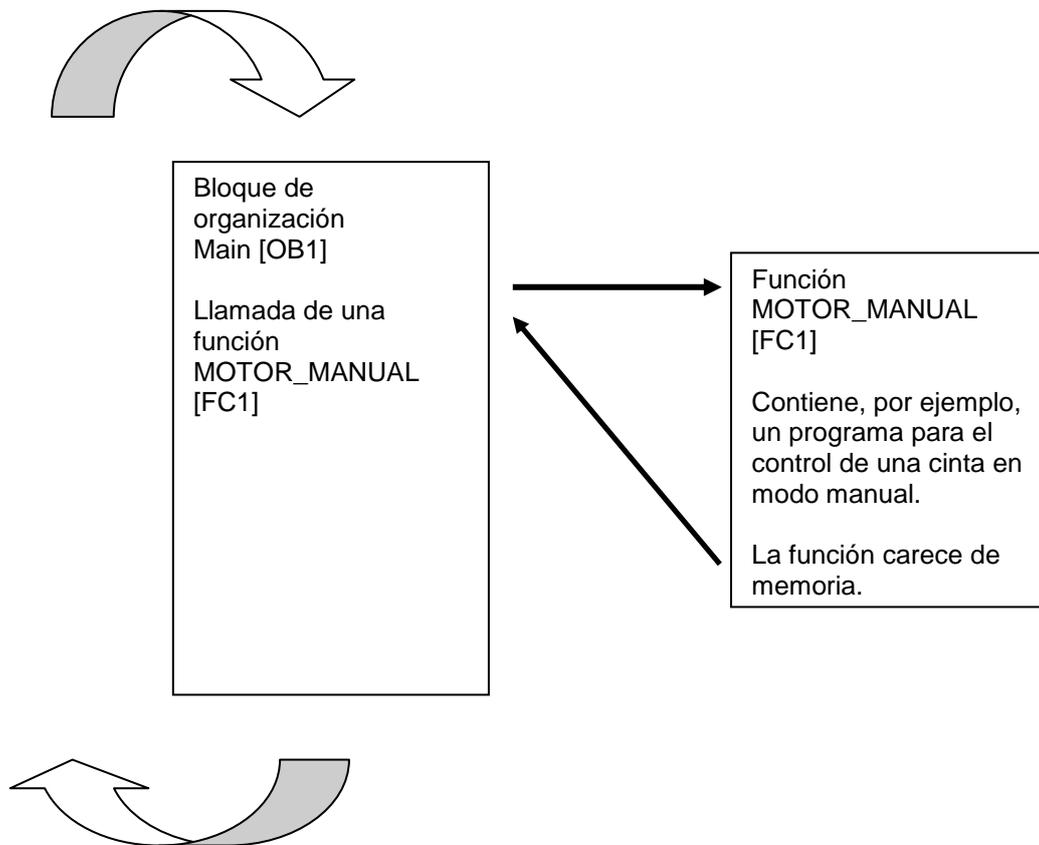


Figura 3: Función con llamada desde el bloque de organización Main[OB1]

4.5 Bloques de función y bloques de datos de instancia

Los bloques de función son bloques lógicos que depositan sus variables de entrada, salida y de entrada/salida, así como las variables estáticas, de forma permanente en bloques de datos de instancia, de modo que **continúan disponibles tras la ejecución del bloque**. Por este motivo, se conocen también como bloques con "memoria".

Los bloques de función también pueden funcionar con variables temporales. No obstante, las variables temporales no se almacenan en el DB instancia, sino que únicamente permanecen disponibles durante un ciclo.

Los bloques de función se utilizan en tareas que no se pueden realizar con funciones:

- Cuando son necesarios temporizadores y contadores en los bloques.
- Cuando hay que almacenar información en el programa. Por ejemplo, una preselección del modo de operación con un pulsador.

Los bloques de función se ejecutan cada vez que un bloque de función es llamado por otro bloque lógico. Un bloque de función también se puede llamar varias veces en diferentes puntos de un programa. Esto facilita la programación de funciones complejas que se repiten con frecuencia.

La llamada de un bloque de función se denomina instancia. A cada instancia de un bloque de función se le asigna un área de memoria que contiene los datos que utiliza el bloque de función. Esta memoria es proporcionada por bloques de datos que son creados automáticamente por el software.

La memoria también puede estar disponible para varias instancias como **multiinstancia** en un bloque de datos. El tamaño máximo de los bloques de datos de instancia varía en función de la CPU. Las variables declaradas en el bloque de función determinan la estructura del bloque de datos de instancia.

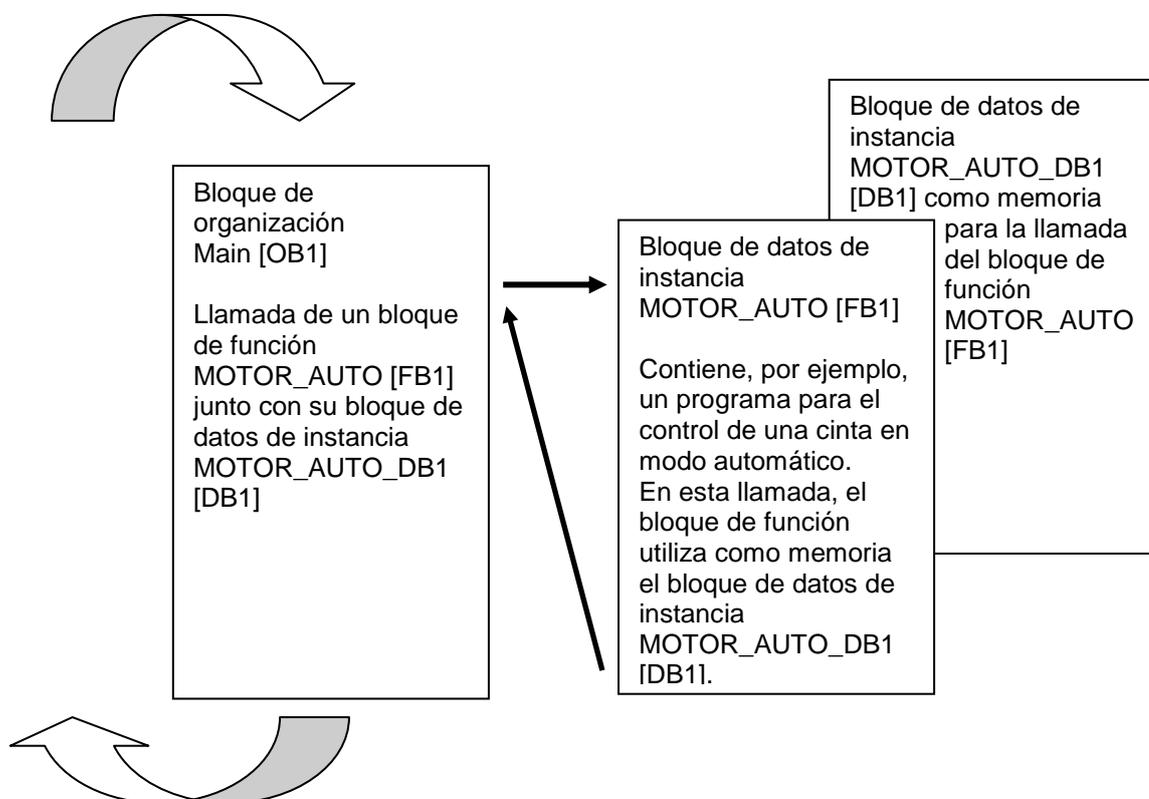


Figura 4: Bloque de función e instancia con llamada desde el bloque de organización Main[OB1]

4.6 Bloques de datos globales

Al contrario que los bloques lógicos, los bloques de datos no contienen instrucciones, sino que sirven para almacenar datos de usuario.

Así, los bloques de datos contienen datos variables con los que trabaja el programa de usuario. La estructura de bloques de datos globales puede definirse a discreción.

Los bloques de datos globales almacenan datos que pueden ser utilizados **por los demás bloques** (ver la Figura 5). Solo debe acceder a los bloques de datos de instancia el correspondiente bloque de función. El tamaño máximo de los bloques de datos varía en función de la CPU.

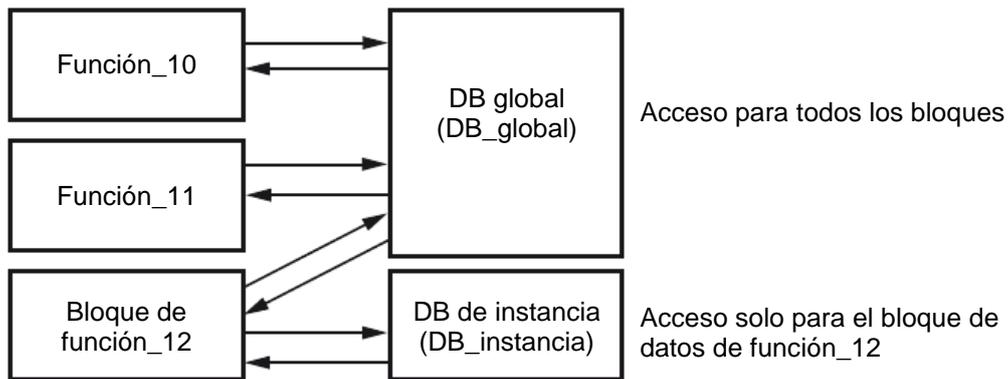


Figura 5: Diferencia entre DB global y DB de instancia.

Ejemplos de uso para **bloques de datos globales**:

- Guardar la información en un sistema de almacén: "¿Qué producto está en cada lugar?"
- Guardar recetas de determinados productos.

4.7 Bloques lógicos aptos para librería

Un programa de usuario puede crearse de modo lineal o estructurado. La **programación lineal** escribe el programa de usuario completo en el OB de ciclo, pero solo es adecuada para programas muy sencillos, para los que actualmente se utilizan otros sistemas de control más económicos, como, p. ej., LOGO!

Para programas más complejos se recomienda siempre una **programación estructurada**. Esta modalidad permite dividir la tarea de automatización en tareas parciales más pequeñas, a fin de ejecutarlas en funciones y bloques de función.

Se recomienda crear siempre bloques lógicos aptos para librería. Esto significa que los parámetros de entrada y salida de una función o bloque de función se definen de manera general y no se les asignan variables globales actuales (entradas/salidas) hasta el momento de su utilización.

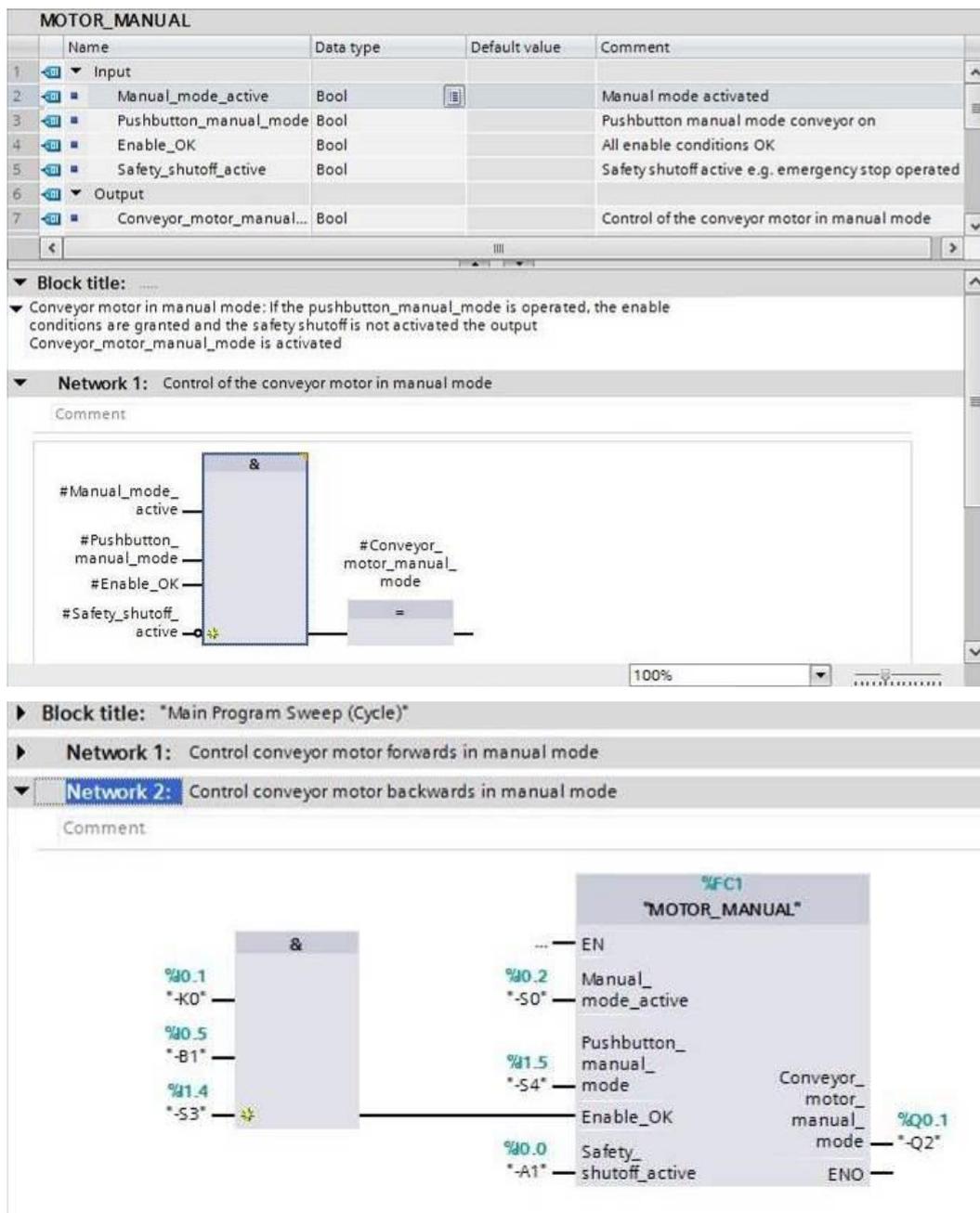


Figura 6: Función apta para librería con llamada en el OB1

4.8 Lenguajes de programación

Para programar funciones se dispone de los lenguajes de programación Diagrama de funciones (FUP), Esquema de contactos (KOP), Lista de instrucciones (AWL) y Structured Control Language (SCL). Para los bloques de función se utiliza también el lenguaje GRAPH, que permite programar cadenas secuenciales gráficas.

A continuación se presenta el lenguaje de programación **Diagrama de funciones (FUP)**.

FUP es un lenguaje de programación gráfico. Su representación es similar a los diagramas de circuitos electrónicos. El programa se mapea en segmentos. Un segmento contiene uno o varios circuitos lógicos. Las señales binarias y analógicas se combinan lógicamente mediante cuadros. Para representar la lógica binaria se utilizan los símbolos lógicos gráficos del álgebra booleana.

Las funciones binarias sirven para consultar los operandos binarios y combinar lógicamente sus estados lógicos. Las operaciones lógicas "Y", "O" y "O-exclusiva" son algunos ejemplos de funciones binarias, como se muestra en la Figura 7.

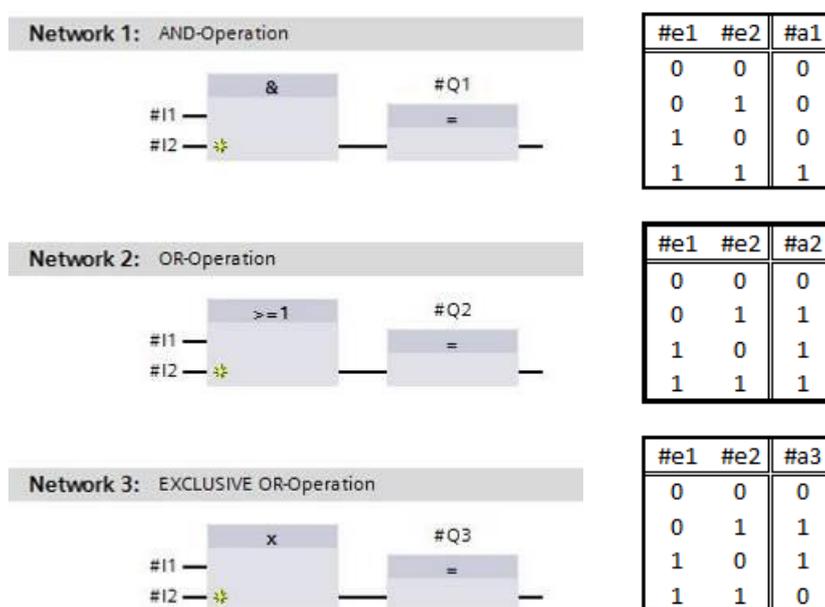


Figura 7: Funciones binarias en FUP y tabla lógica correspondiente

Las instrucciones simples sirven, p. ej., para controlar salidas binarias, evaluar flancos o ejecutar funciones de salto en el programa.

Las instrucciones complejas sirven para acceder a elementos de programa, como, p. ej., temporizadores CEI y contadores CEI.

Un cuadro vacío es un comodín en el que puede seleccionarse la instrucción deseada.

Mecanismo de entrada de habilitación EN (enable) y salida de habilitación ENO (enable output):

- Las instrucciones sin mecanismo EN/ENO se ejecutan independientemente del estado lógico de las entradas del cuadro.
- Las instrucciones con mecanismo EN/ENO se ejecutan únicamente si la entrada de habilitación "EN" tiene el estado lógico "1". Si el cuadro se ejecuta correctamente, la salida de habilitación "ENO" tendrá el estado lógico "1". En el momento en que ocurre un error durante la ejecución, se desactiva la salida de habilitación "ENO". Si la entrada de habilitación EN no está interconectada, el cuadro se ejecuta siempre.

5 Tarea planteada

En este capítulo se planificarán, programarán y probarán las siguientes funciones de la descripción de proceso para una planta de clasificación.

- Operación manual: motor de la cinta en modo Jog

6 Planificación

Para favorecer la claridad y permitir la reutilización, no se recomienda programar todas las funciones en el OB1. Por ello el código del programa se transferirá en su mayor parte a funciones (FC) y bloques de función (FB). A continuación vamos a planificar cuáles de las funciones se transferirán a FC y cuáles se ejecutarán en el OB1.

6.1 PARADA DE EMERGENCIA

La parada de emergencia no requiere una función propia. Al igual que el modo operativo, el estado actual del relé de parada de emergencia puede utilizarse directamente en los bloques.

6.2 Operación manual: motor de la cinta en modo Jog

El modo Jog del motor de la cinta se encapsulará en una función (FC) "MOTOR_MANUAL". Con ello, por un lado, se obtiene una mayor claridad en el OB1, y, por el otro, se hace posible la reutilización en caso de ampliación de la planta con una cinta transportadora adicional. Los parámetros planificados se muestran en la Tabla 2.

Input	Tipo de datos	Comentario
Modo_manual_activo	BOOL	Modo de operación Modo manual activado
Pulsador_modos_Jog	BOOL	Pulsador para conectar el motor de la cinta en modo Jog
Habilitación_OK	BOOL	Se cumplen todas las condiciones para la habilitación
Desconexión de seguridad activa	BOOL	Desconexión de seguridad activa, p. ej., parada de emergencia accionada
Output		
Motor_cinta_modos_Jog	BOOL	Manejar el motor de la cinta en modo Jog

Tabla 2: Parámetros para la FC "MOTOR_MANUAL"

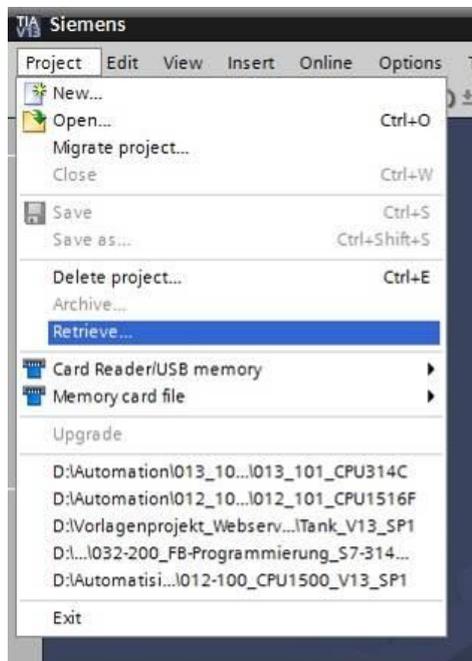
La salida Motor_cinta_modos_Jog estará en estado ON mientras se encuentre pulsado el Pulsador_modos_Jog, esté activado el modo manual, esté otorgada la habilitación y no esté activa la desconexión de seguridad.

7 Instrucciones paso a paso estructuradas

A continuación se describe cómo realizar la planificación. Si ya domina el tema, le bastará con seguir los pasos numerados. De lo contrario, límitese a seguir los pasos detallados de las presentes instrucciones.

7.1 Desarchivación de un proyecto existente

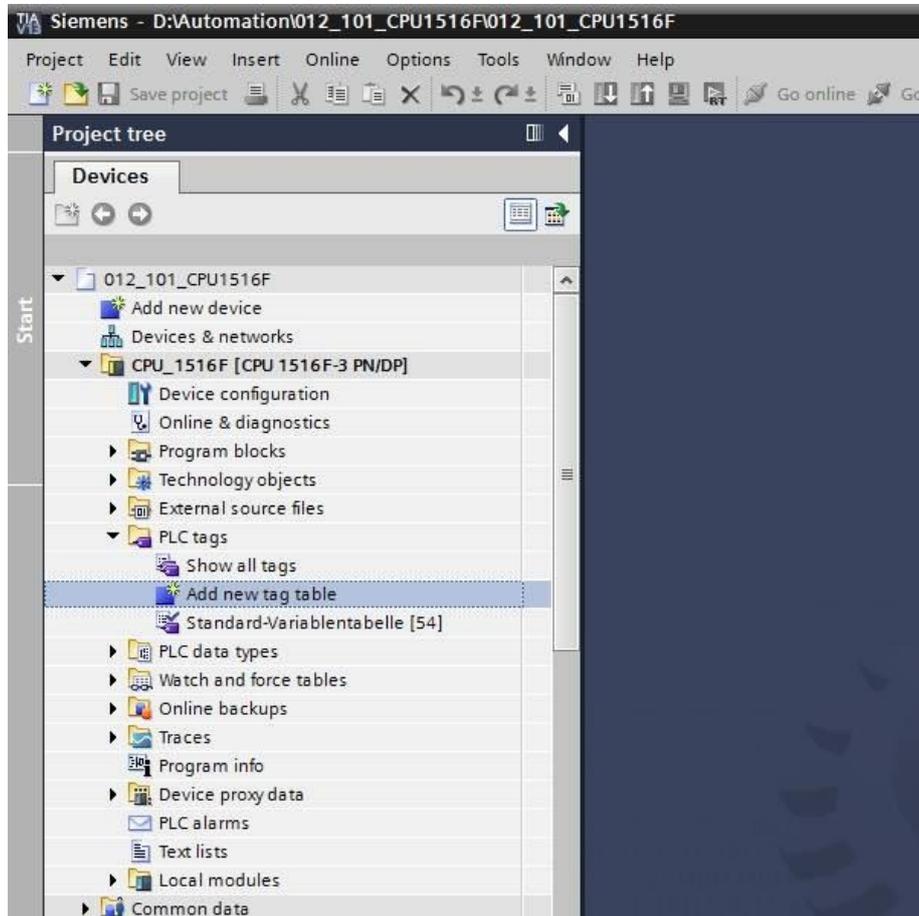
- Antes de empezar a programar la función (FC) "MOTOR_MANUAL", se necesita un proyecto con una configuración hardware (p. ej., SCE_ES_012-101 Hardware configuration S7-1516F_R1502.zap). Para desarchivar un proyecto existente desde la vista del proyecto, escoja el fichero en cuestión en → Project (Proyecto) → Retrieve (Desarchivar). A continuación, confirme la selección con Open (Abrir). (→ Project (Proyecto) → Retrieve (Desarchivar) → Seleccionar un fichero .zap → Open (Abrir))



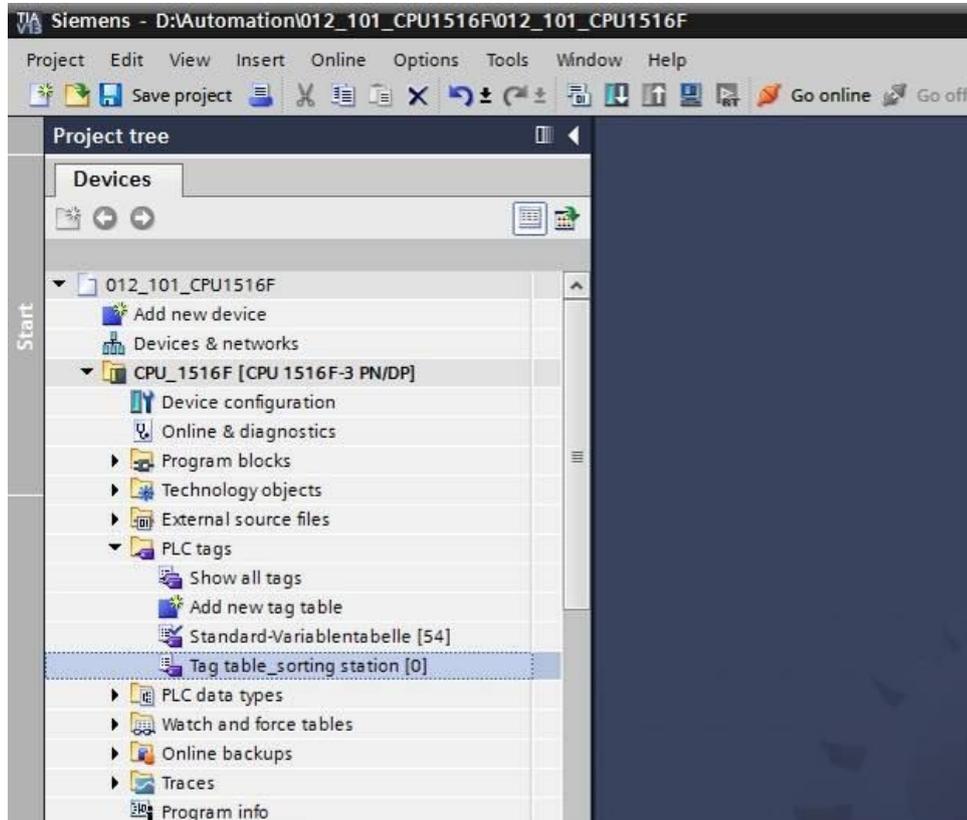
- A continuación puede seleccionarse el directorio de destino en el que se guardará el proyecto desarchivado. Confirme la selección con "OK (Aceptar)". (→ Directorio de destino → OK (Aceptar))

7.2 Creación de una nueva tabla de variables

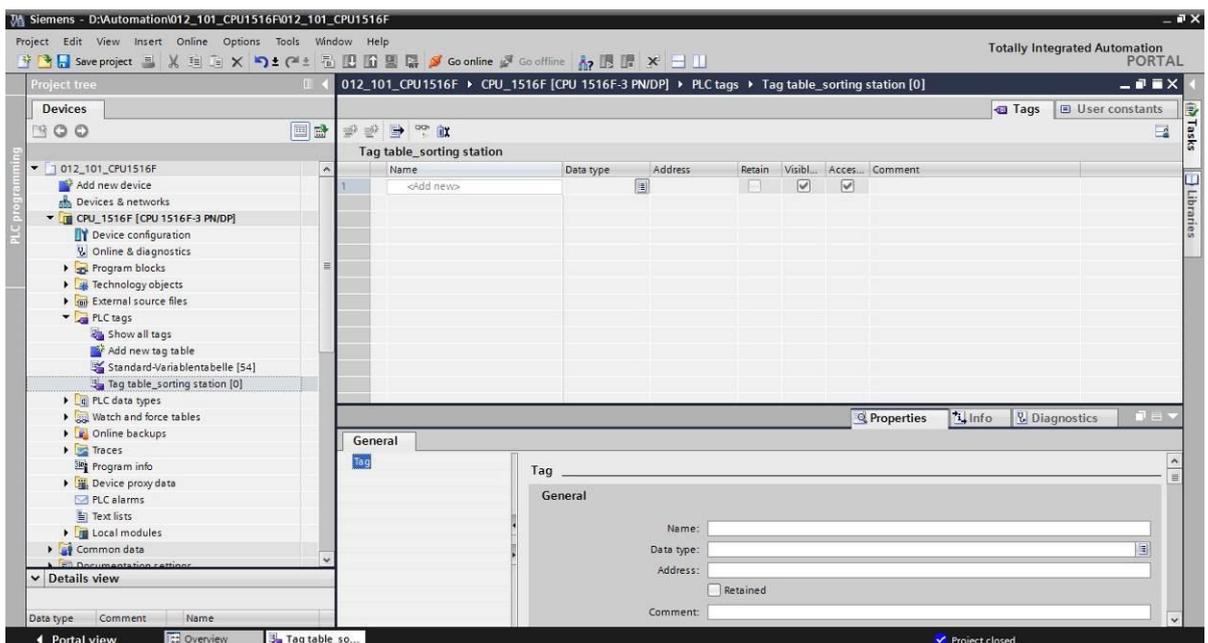
- Navegue por el árbol del proyecto hasta las → variables de PLC de su controlador y cree una nueva tabla de variables haciendo doble clic en → "Add new tag table (Agregar nueva tabla de variables)".



- Asigne el nombre "Tag table_sorting station (Tabla_variables_planta_clasificación)" a la tabla de variables que acaba de crear. (→ Haga clic con el botón derecho en "Tag table_1 (Tabla_variables_1)" → "Rename (Cambiar nombre)" → Tag table_sorting station)

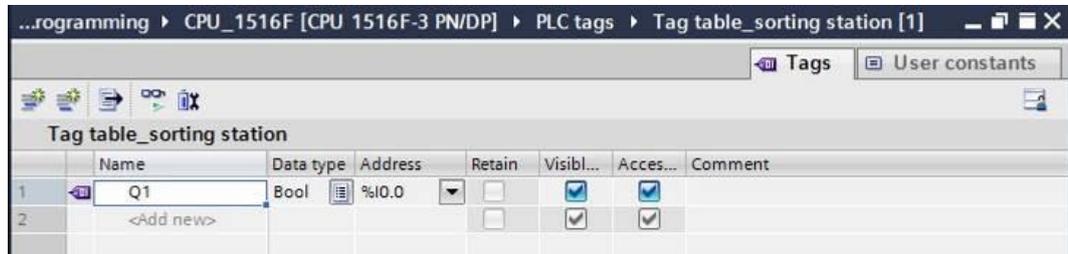


- A continuación, ábrala haciendo doble clic. (→ Tag table_sorting station (Tabla_variables_planta_clasificación))

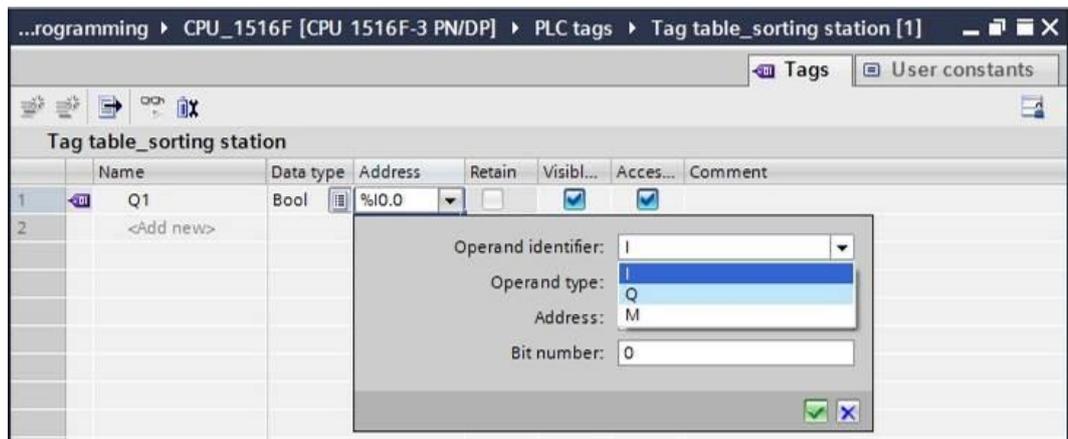


7.3 Creación de nuevas variables dentro de una tabla de variables

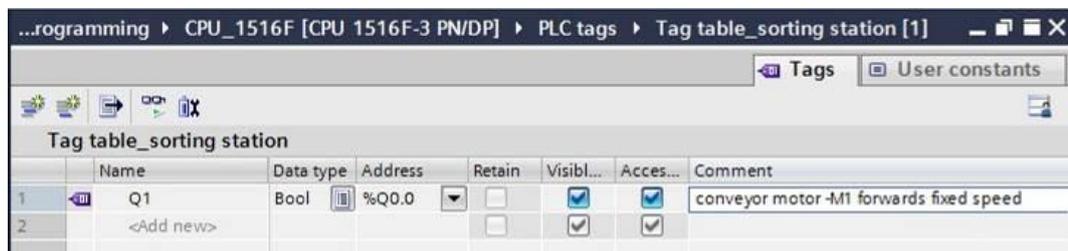
- Agregue el nombre Q1 y confirme con la tecla Intro. Si todavía no ha creado ninguna otra variable, el TIA Portal habrá asignado automáticamente el tipo de datos "Bool" y la dirección %I0.0. (→<Add new (Agregar)> → Q1 → Intro)



- Cambie la dirección a %Q0.0 introduciendo el nuevo nombre directamente o haciendo clic en la flecha de lista desplegable para abrir el menú de direccionamiento. Allí, cambie el identificador del operando a Q y confirme con Intro o haciendo clic en la marca de verificación. (→ %I0.0 → Operand identifier (Identificador de operando) → Q →)



- Asigne a la variable el comentario "conveyor motor -M1 forwards fixed speed (Motor de cinta -M1 hacia delante, velocidad fija)".



→ En la línea 2, agregue una nueva variable Q2. El TIA Portal ha asignado automáticamente el mismo tipo de datos que en la línea 1 y ha incrementado la dirección en 1 hasta %Q0.1. Introduzca el comentario "conveyor motor -M1 backwards fixed speed (Motor de cinta -M1 hacia atrás, velocidad fija)".

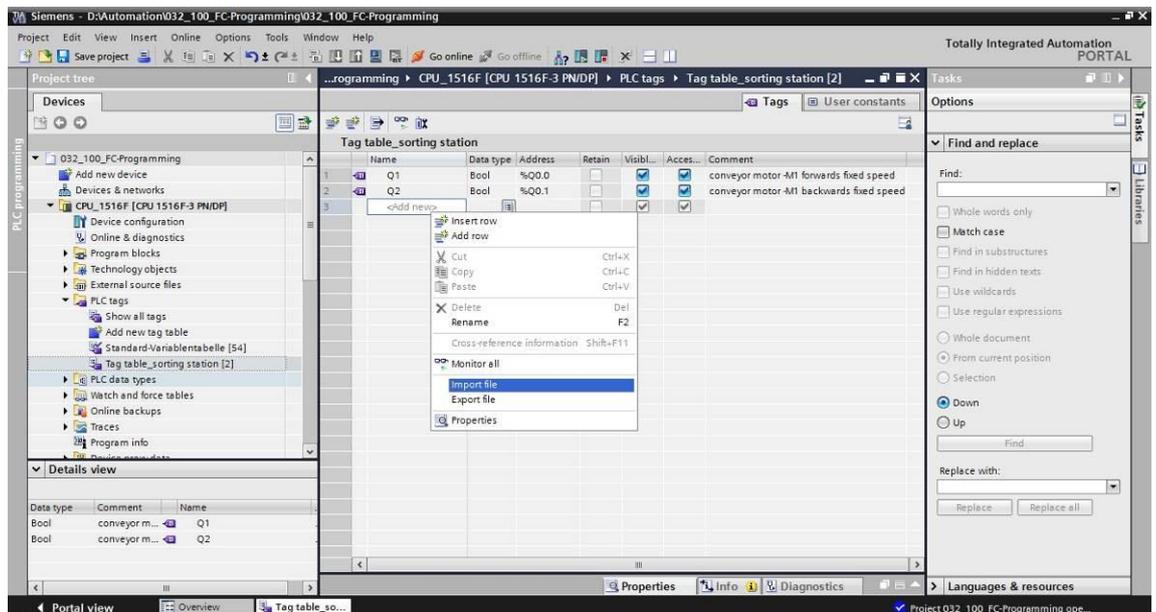
(→ <Add new (Agregar)> → Q2 → Intro → Comentario → Motor de cinta M1 hacia atrás, velocidad fija)

	Name	Data type	Address	Retain	Visibl...	Acces...	Comment
1	Q1	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor motor -M1 forwards fixed speed
2	Q2	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	conveyor motor -M1 backwards fixed speed
3	<Add new>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

7.4 Importación de la "Tag table_sorting station (Tabla_variables_planta_clasificación)"

→ Para agregar una tabla de símbolos existente, haga clic con el botón derecho del ratón en un campo vacío de la "Tag table_sorting station (Tabla_variables_planta_clasificación)". Seleccione en el menú contextual "Import file (Importar archivo)".

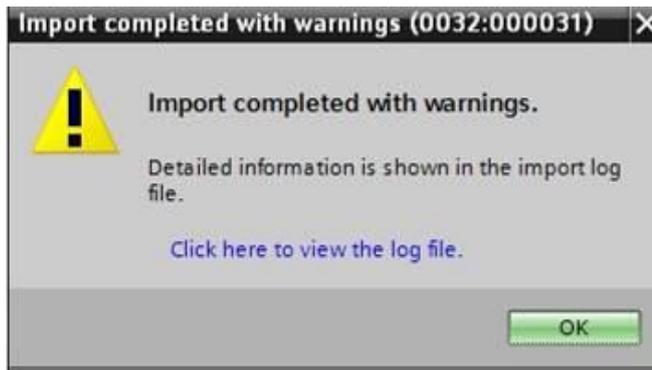
(→ Haga clic con el botón derecho en un campo vacío de la tabla de variables → Import file (Importar archivo))



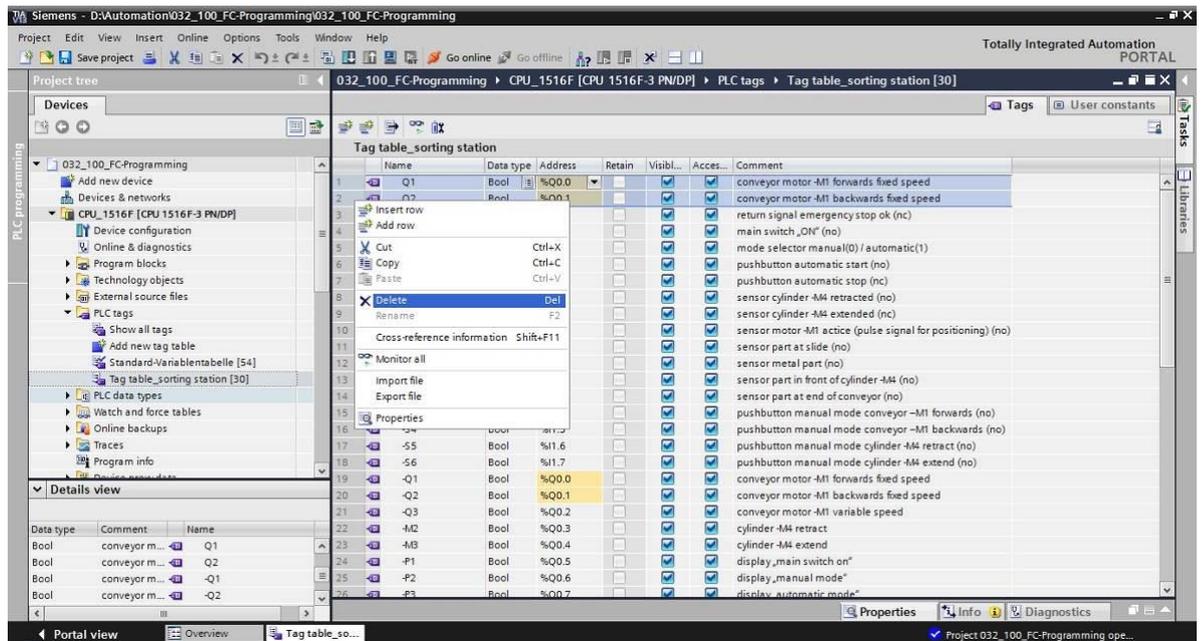
→ Seleccione la tabla de símbolos deseada (p. ej., en formato .xlsx) y confirme la selección con "Open (Abrir)".

(→ SCE_ES_020-100 Tag table_sorting station... → Open (Abrir))

→ Una vez finalizada la importación, aparecerá una ventana de confirmación con la posibilidad de ver el archivo de registro de importación. Haga clic en → OK (Aceptar).

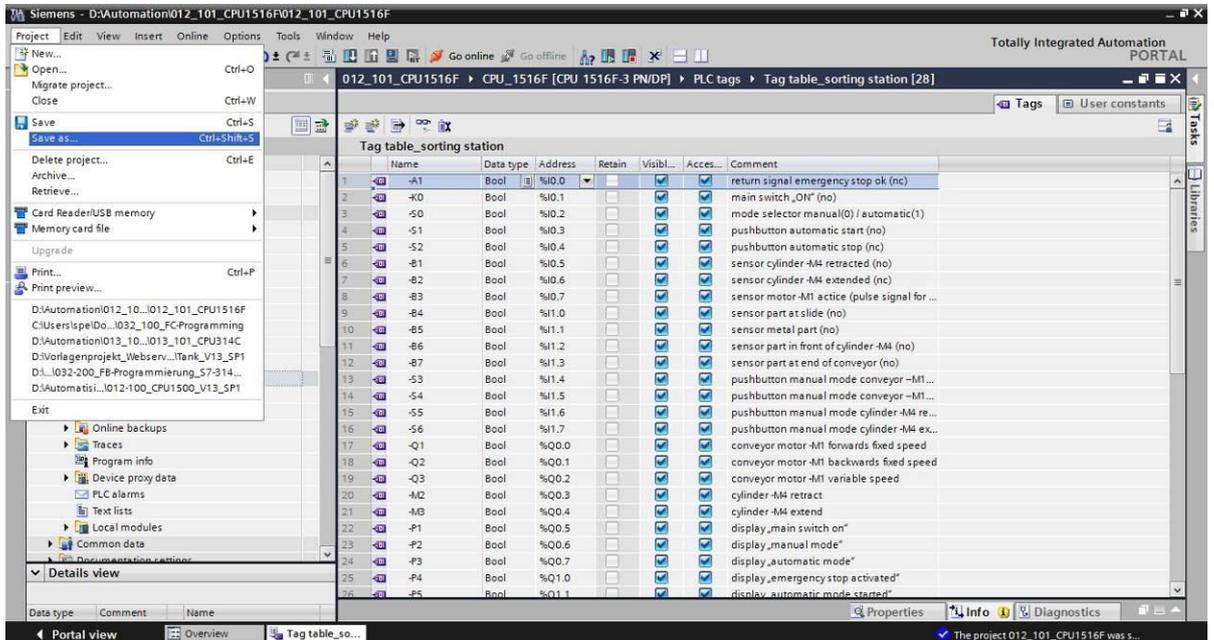


- Como verá, algunas direcciones aparecen resaltadas en naranja. Se trata de las que están duplicadas; se han numerado automáticamente las variables correspondientes a fin de evitar ambigüedades.
- Borre las variables duplicadas seleccionando las líneas correspondientes mediante la tecla Supr de su teclado o la opción "Delete (Eliminar)" del menú contextual.
- (→ Clic con el botón derecho en las variables seleccionadas → Delete (Eliminar))



→ Ahora tendrá en su pantalla una tabla de símbolos completa de las entradas y salidas digitales. Guarde el proyecto con el nombre 032-100 FC Programming.

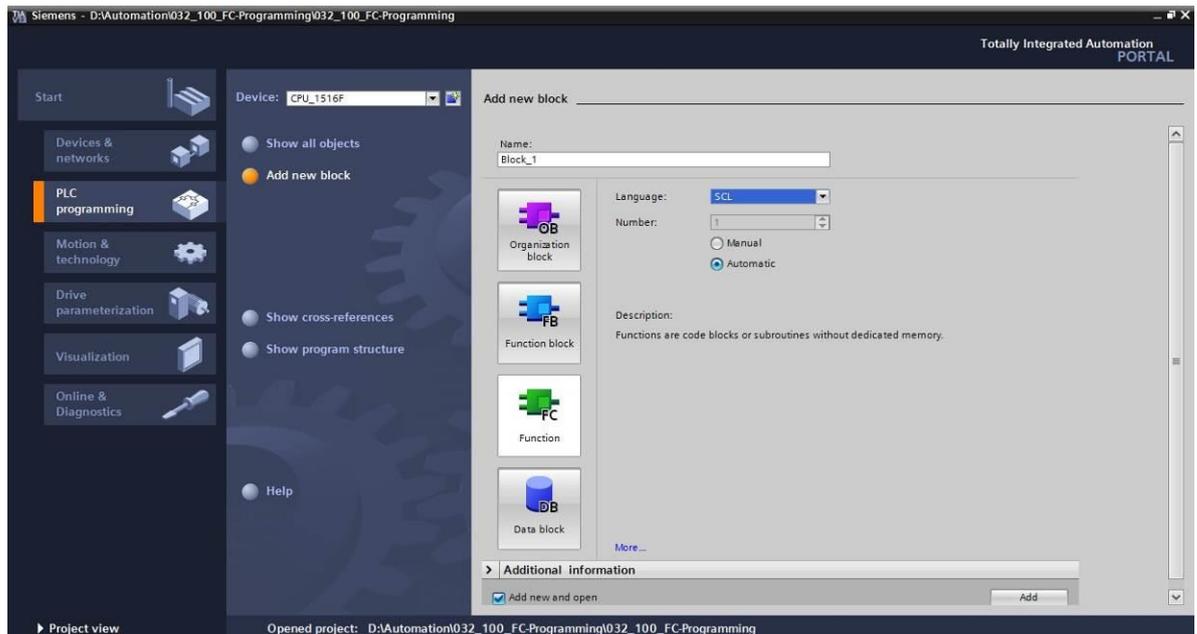
(→ Project (Proyecto) → Save as... (Guardar como...) → 032-100 FC Programming → Save (Guardar))



7.5 Creación de la función FC1 "MOTOR_MANUAL" para el motor de cinta en modo Jog

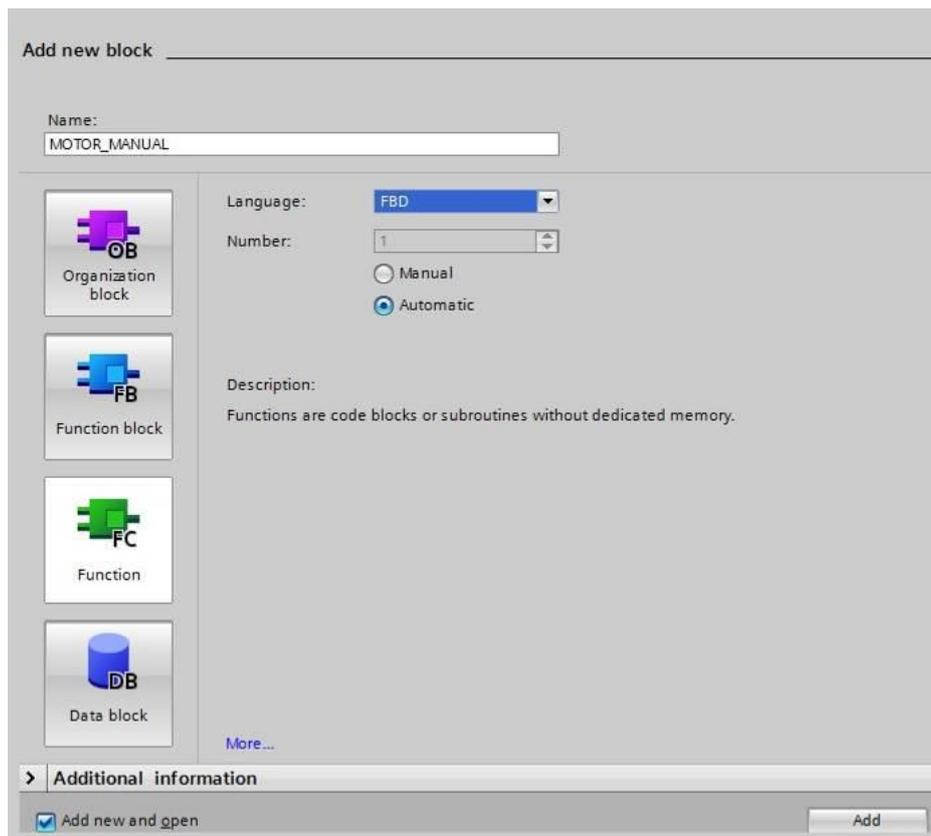
→ En la vista del portal, haga clic en el apartado "PLC programming (Programación de PLC)", opción "Add new block (Agregar nuevo bloque)" para crear una nueva función.

(→ PLC programming (Programación de PLC) → Add new block (Agregar nuevo bloque) → )



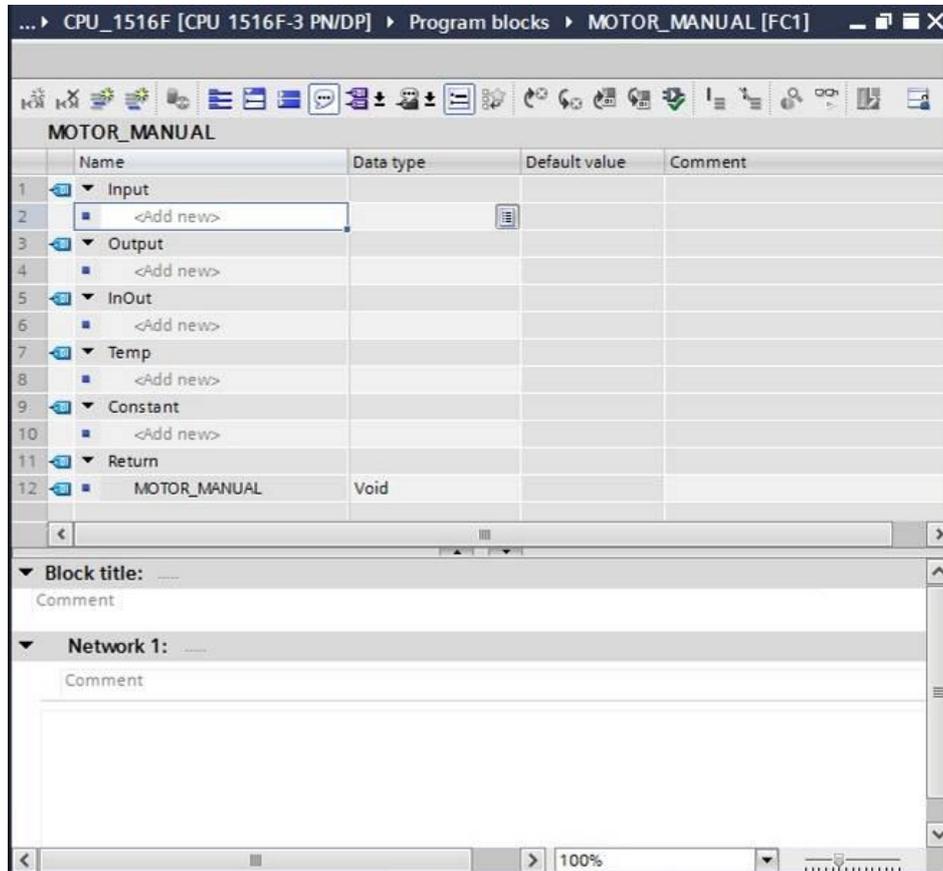
→ Asigne al nuevo bloque el nombre: "MOTOR_MANUAL", seleccione el lenguaje FUP y deje que el programa asigne el número automáticamente. Si activa la marca de verificación "Add new and open (Agregar y abrir)", pasará automáticamente, en la vista del proyecto, al bloque de función que acaba de crear. Haga clic ahora en "Add (Agregar)".

(→ Name (Nombre): MOTOR_MANUAL → Language (Lenguaje): FDB (FUP) → Number (Número): Automatic (Automático) → Add new and open (Agregar y abrir) → Add (Agregar))

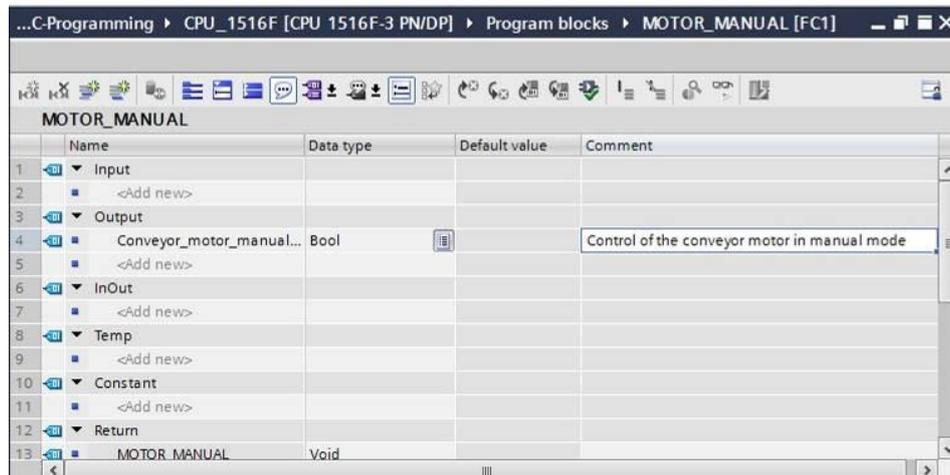


7.6 Definición de la interfaz de la función FC1 "MOTOR_MANUAL"

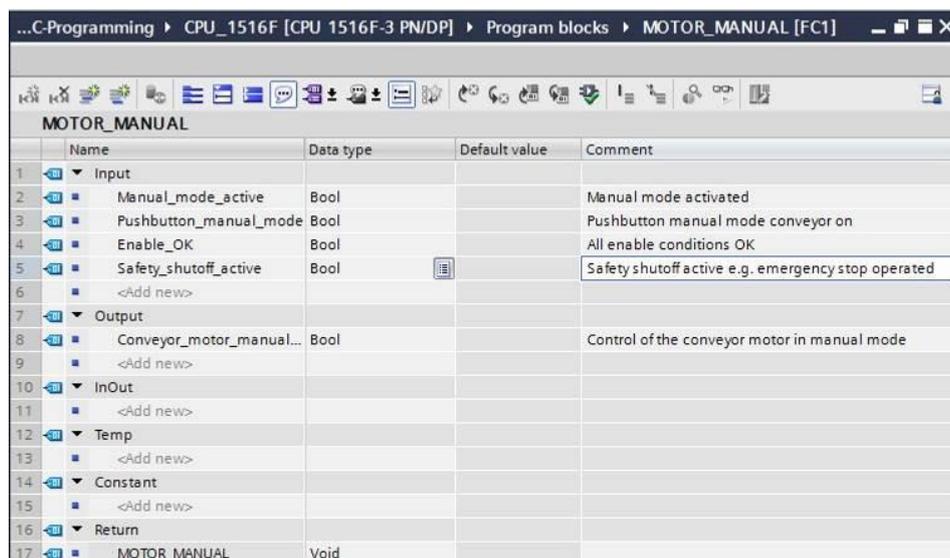
- Al hacer clic en "Add new and open (Agregar y abrir)", se abre la vista del proyecto con una ventana para editar el bloque que se acaba de crear.
- En la parte superior de la vista de programación encontrará la descripción de interfaces de la función.



- Para controlar el motor de cinta se requiere una señal de salida binaria. Por ello debe crearse previamente la variable Output local #Conveyor_motor_manual_mode (Motor_cinta_modos_Jog), del tipo "Bool". Agregue al parámetro el comentario "Control of the conveyor motor in manual mode (Control del motor de la cinta en modo Jog)".
(→ Output (Salida): Conveyor_motor_manual_mode (Motor_cinta_modos_Jog) → Bool → Control of the conveyor motor in manual mode (Control del motor de la cinta en modo Jog))



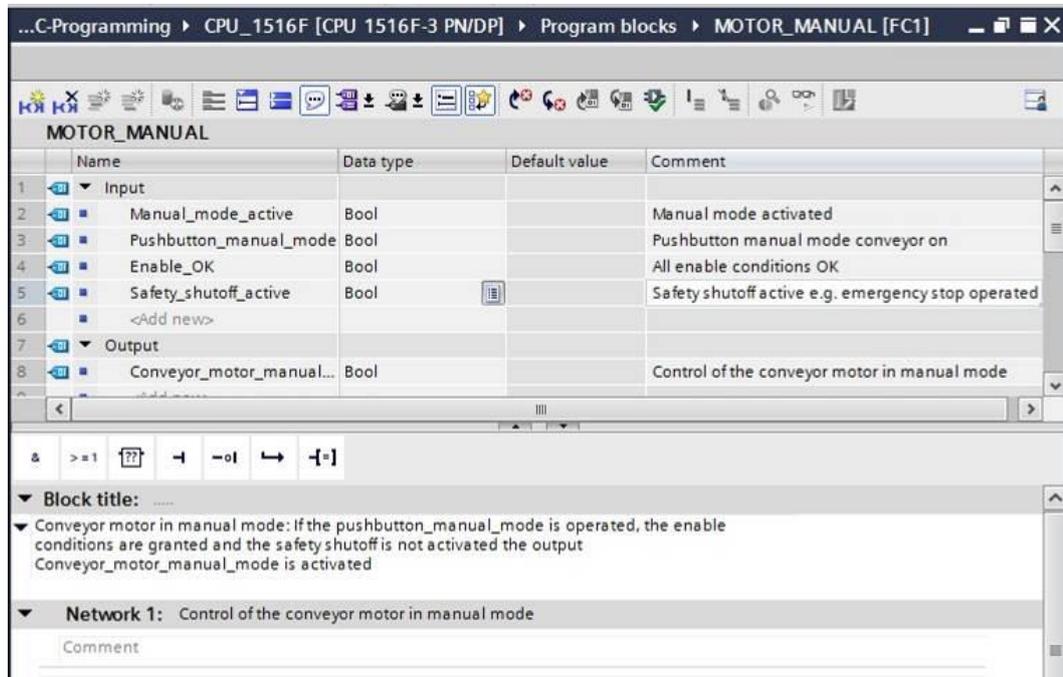
- En primer lugar, agregue como interfaz de entrada en Input (Entrada) el parámetro #Manual_mode_active (Modo_manual_activo) y confirme con la tecla Intro o saliendo del campo de entrada. Se asignará automáticamente el tipo de datos "Bool". El tipo de datos se mantendrá. A continuación introduzca el comentario "Modo de operación manual activado".
(→ Manual_mode_active (Modo_manual_activo) → Intro → Bool → Modo de operación manual activado)
- A continuación, introduzca en Input (Entrada), como parámetros de entrada binarios, #Pushbutton_manual_mode (Pulsador_modos_Jog), #Enable_OK (Habilitación_OK) y #Safety_shutoff_active (Desconexión de seguridad activa), y compruebe los tipos de datos de dichos parámetros. Añada comentarios descriptivos.



→ Para documentar el programa, defina el título del bloque, un comentario de bloque y un título de segmento explicativo para el segmento 1.

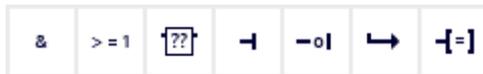
(→ Título del bloque: Motor control in manual mode (Control del motor en modo manual)

→ Network 1 (Segmento 1): Control of the conveyor motor in manual mode (Control del motor de la cinta en modo Jog))



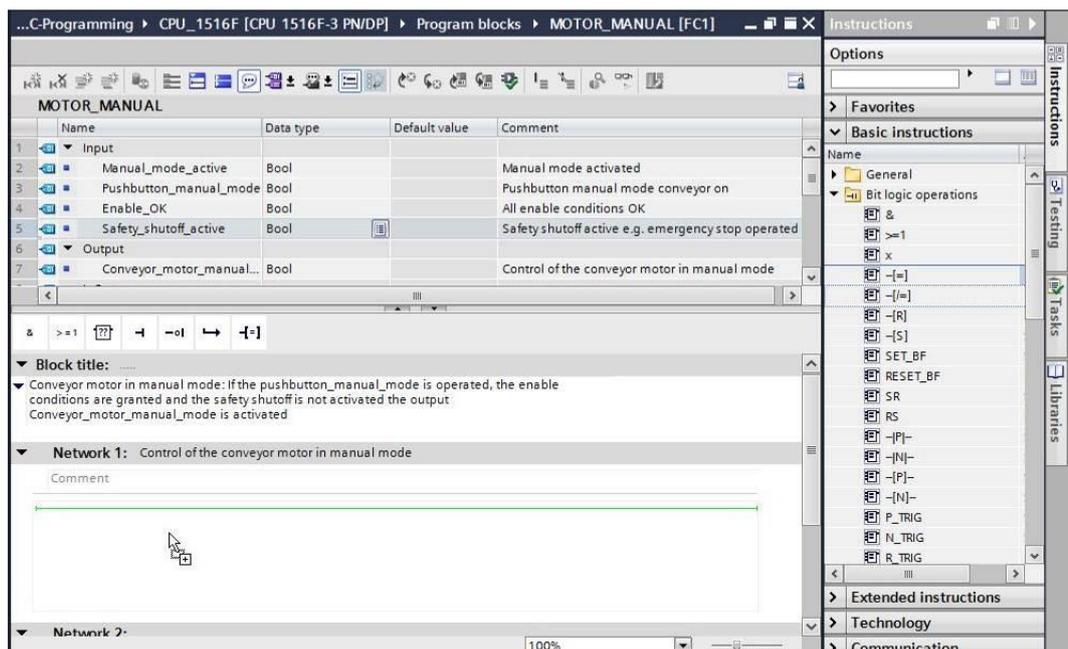
7.7 Programación del FC1: MOTOR_MANUAL

→ En la ventana de programación, debajo de la descripción de interfaces, encontrará una barra de herramientas con varias funciones lógicas, y debajo de ella un área con segmentos. Hemos definido ya en ella el título del bloque y el título del primer segmento. Dentro de los segmentos, la programación se realiza utilizando distintos bloques lógicos. Para mayor claridad, se recomienda crear varios segmentos. A continuación le presentamos las distintas posibilidades para insertar bloques lógicos.



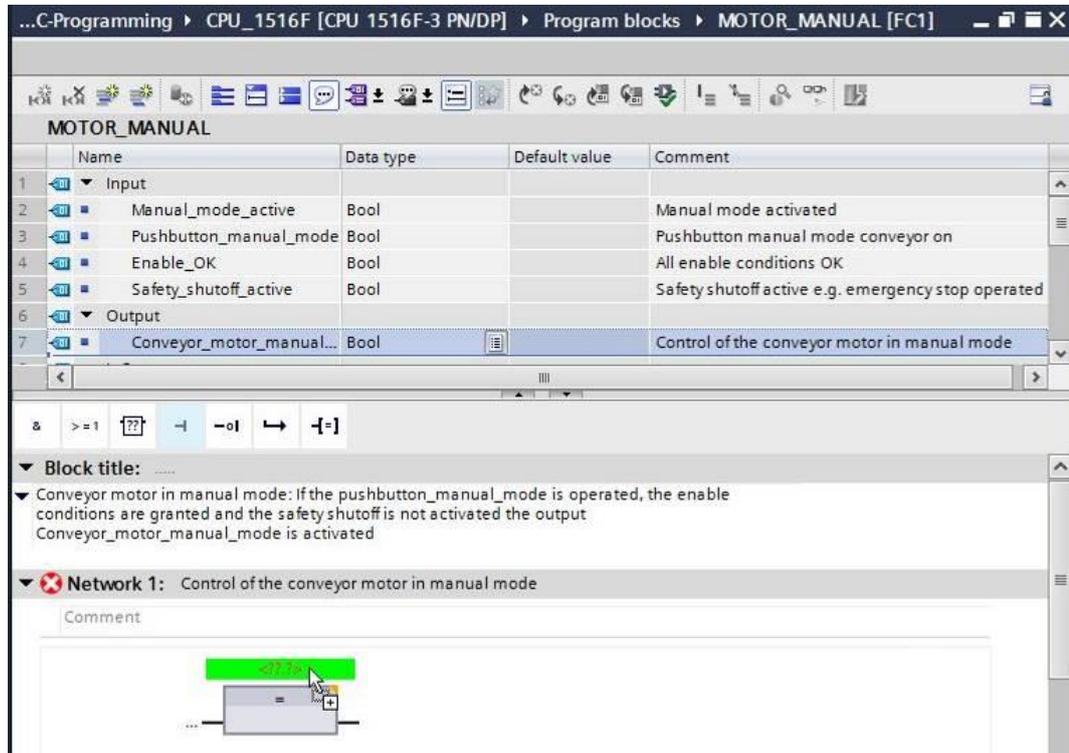
→ En el lado derecho de la ventana de programación hay una lista de instrucciones que pueden utilizarse en el programa. Busque en → Basic instructions (Instrucciones básicas) → Bit logic operations (Operaciones lógicas con bits) la función –[=] (asignación) y cópiela mediante "arrastrar y soltar" a su Network 1 (Segmento 1) (aparecerá una línea verde y el puntero del ratón mostrará el símbolo +).

(→ Instructions (Instrucciones) → Basic instructions (Instrucciones básicas) → Bit logic operations (Operaciones lógicas con bits) → –[=])



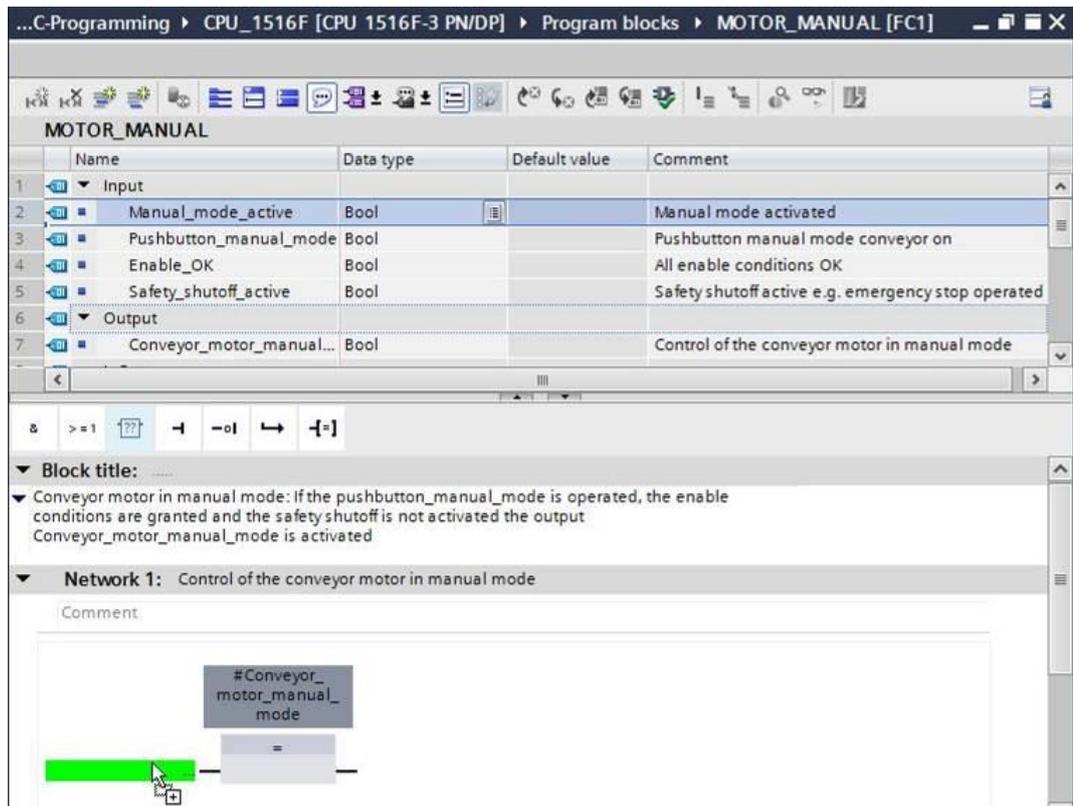
→ Ahora, desplace el parámetro de salida #Conveyor_motor_manual_mode (Motor_cinta_modos_Jog), mediante "arrastrar y soltar", a `<??.?>`, por encima del bloque que acaba de insertar. La mejor manera de seleccionar un parámetro en la descripción de interfaces es agarrarlo por el icono azul .

(→  Conveyor_motor_manual_mode (Motor_cinta_modos_Jog))



→ Con esto se determina que el parámetro #Conveyor_motor_manual_mode (Motor_cinta_modos_Jog) se escriba mediante este bloque. Para que esto sea así, faltan todavía las condiciones de entrada. Desplace el parámetro de entrada #Manual_mode_active (Modo_manual_activo) mediante "arrastrar y soltar" a "..." en el lado izquierdo del bloque de asignación.

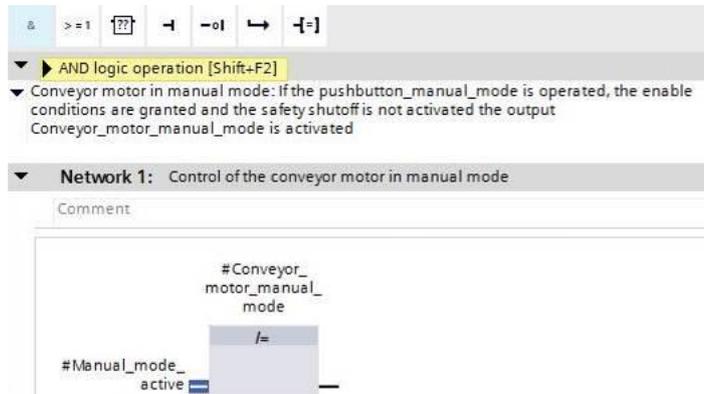
(→  Manual_mode_active (Modo_manual_activo))



→ Además, se combinará la entrada del bloque de asignación con otros parámetros con el operador lógico Y. Para ello, haga clic en primer lugar en la entrada del bloque en el que está interconectado #Manual_mode_active (Modo_manual_activo). La raya de entrada aparecerá sobre fondo azul.

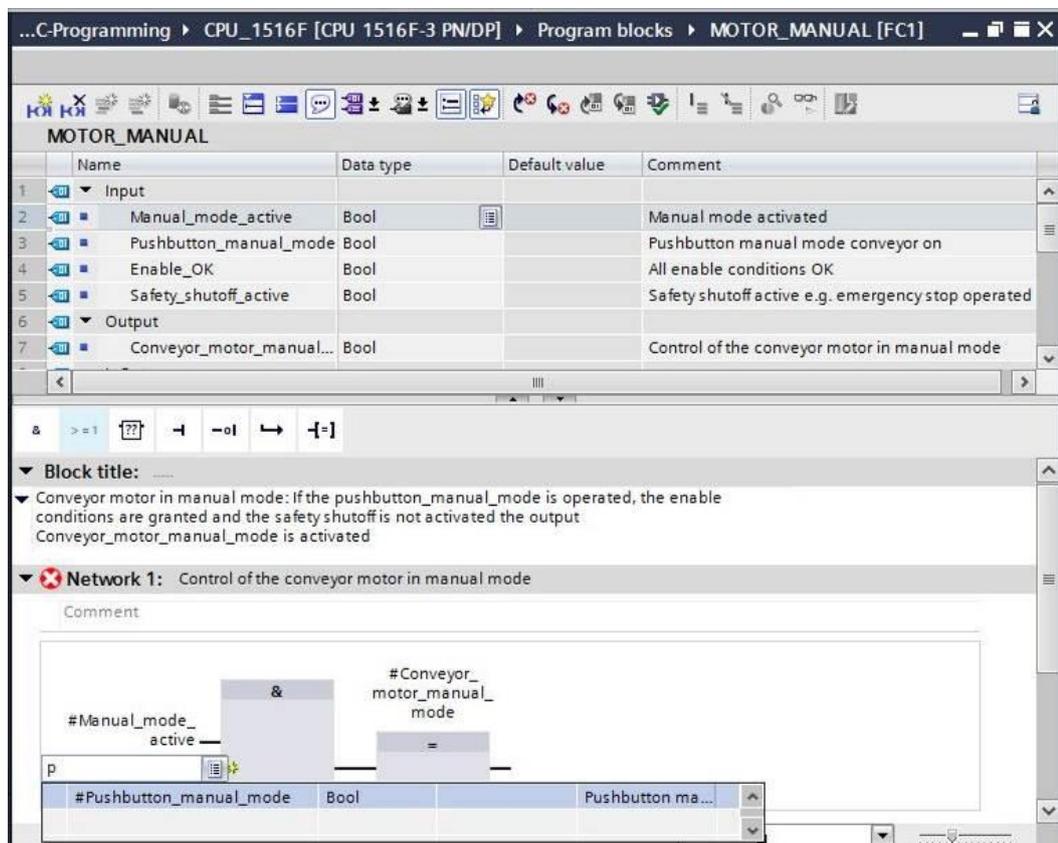


- Haga clic en el icono  de la barra de herramientas lógicas para insertar una combinación mediante operador Y entre la variable #Manual_mode_active (Modo_manual_activo) y el bloque de asignación.



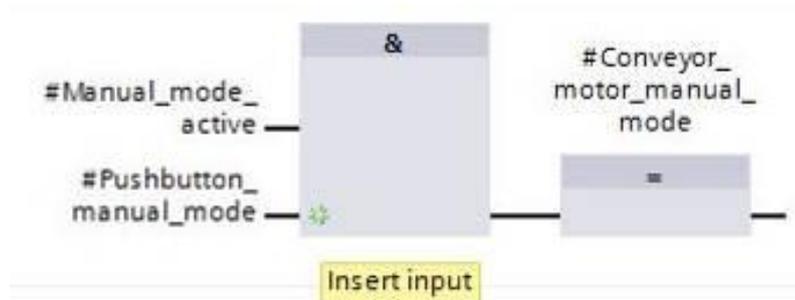
- Haga doble clic en la segunda entrada de la combinación lógica Y  y, en el campo que se abrirá, introduzca la letra "P" para ver una lista de las variables disponibles que empiezan por esa letra. Haga clic en la variable #Pushbutton_manual_mode (Pulsador_modos_Jog) → Intro para aceptar.

(→ Bloque & →  → P → #Pushbutton_manual_mode (Pulsador_modos_Jog) → Intro)

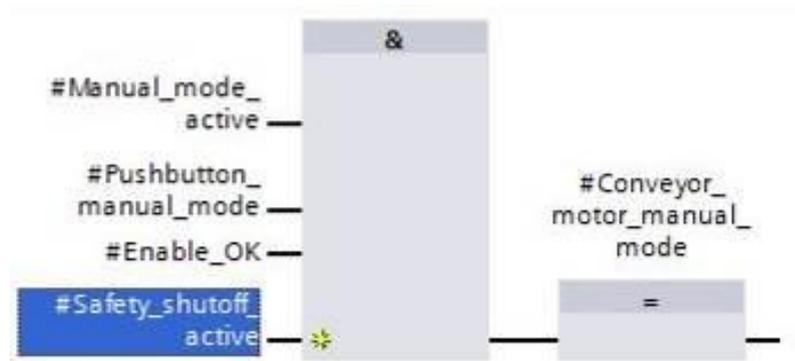


Nota: En esta modalidad de asignación de variables, existe riesgo de confusión con las variables globales de la tabla de variables. Por ello es preferible usar la modalidad con "arrastrar y soltar" desde la descripción de interfaces.

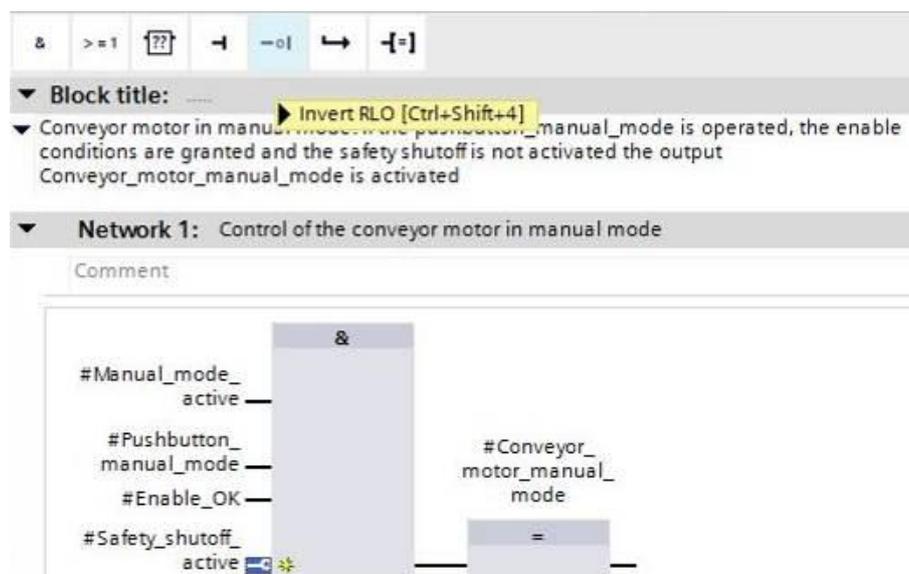
- Para que solo sea posible controlar la salida cuando se haya otorgado la habilitación y no esté activa la desconexión de seguridad, deben combinarse mediante el operador Y las variables de entrada #Enable_OK (Habilitación_OK) y #Safety_shutoff_active (Desconexión de seguridad activa). Para ello, haga doble clic en el asterisco amarillo  del elemento Y a fin de agregar dos entradas más.



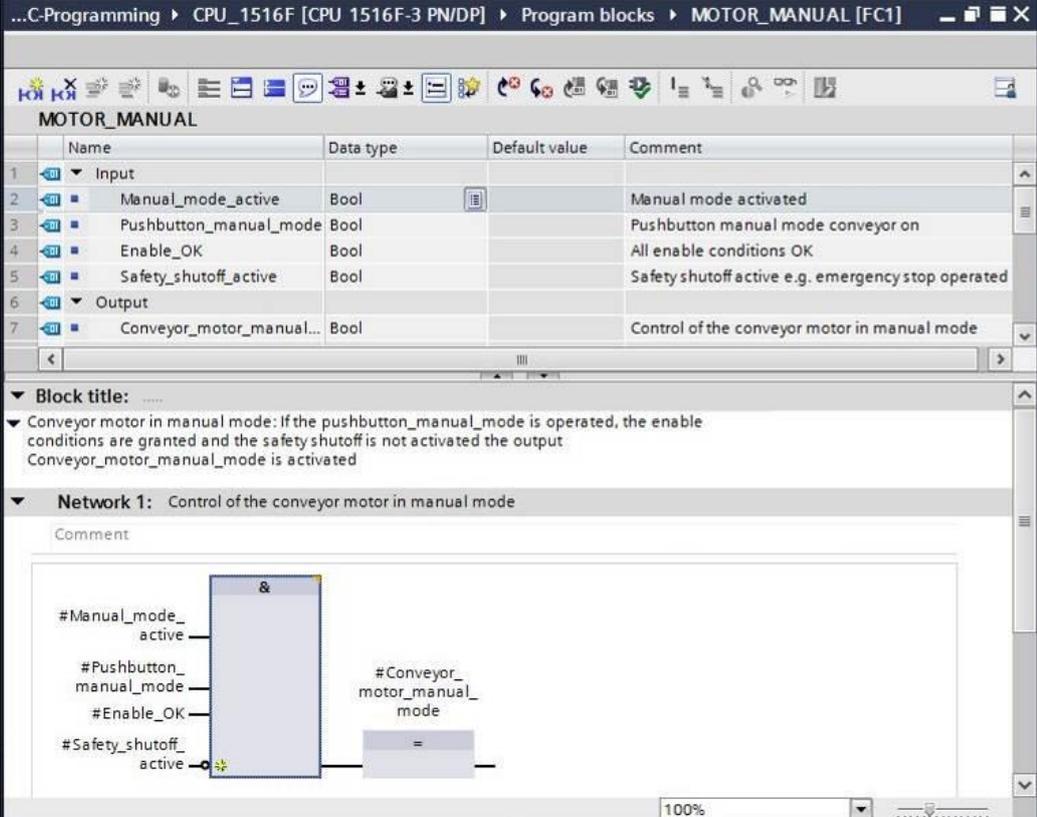
- Inserte las variables de entrada #Enable_OK (Habilitación_OK) y #Safety_shutoff_active (Desconexión de seguridad activa) en las nuevas entradas del elemento Y.



- Niegue la entrada conectada con el parámetro #Safety_shutoff_active (Desconexión de seguridad activa); para ello, selecciónelo y haga clic en .



→ No olvide hacer clic en  Save project. A continuación se muestra la función ya creada "MOTOR_MANUAL" [FC1] de FUP.



The screenshot displays the configuration for the "MOTOR_MANUAL" function block. It features a table for variable declarations, a descriptive block title, a comment, and a network diagram.

	Name	Data type	Default value	Comment
1	Input			
2	Manual_mode_active	Bool		Manual mode activated
3	Pushbutton_manual_mode	Bool		Pushbutton manual mode conveyor on
4	Enable_OK	Bool		All enable conditions OK
5	Safety_shutoff_active	Bool		Safety shutoff active e.g. emergency stop operated
6	Output			
7	Conveyor_motor_manual...	Bool		Control of the conveyor motor in manual mode

Block title:

Conveyor motor in manual mode: If the pushbutton_manual_mode is operated, the enable conditions are granted and the safety shutoff is not activated the output Conveyor_motor_manual_mode is activated

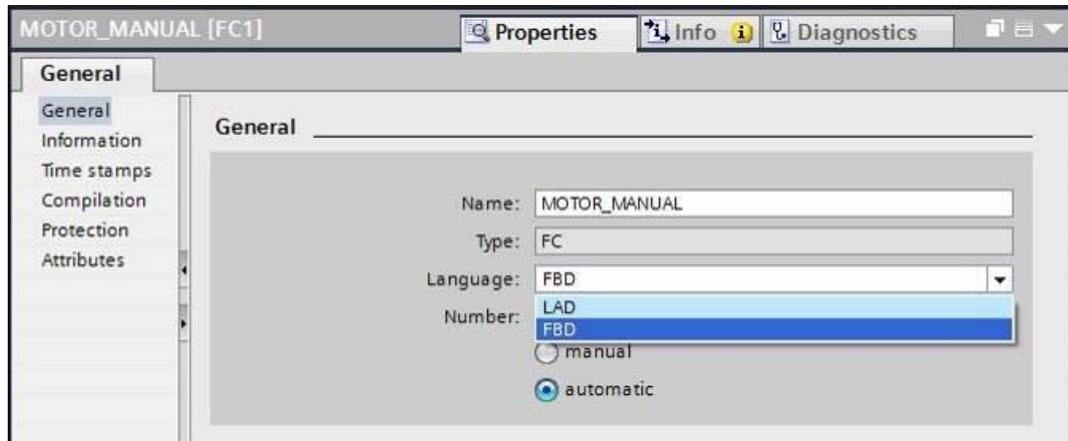
Network 1: Control of the conveyor motor in manual mode

Network Diagram:

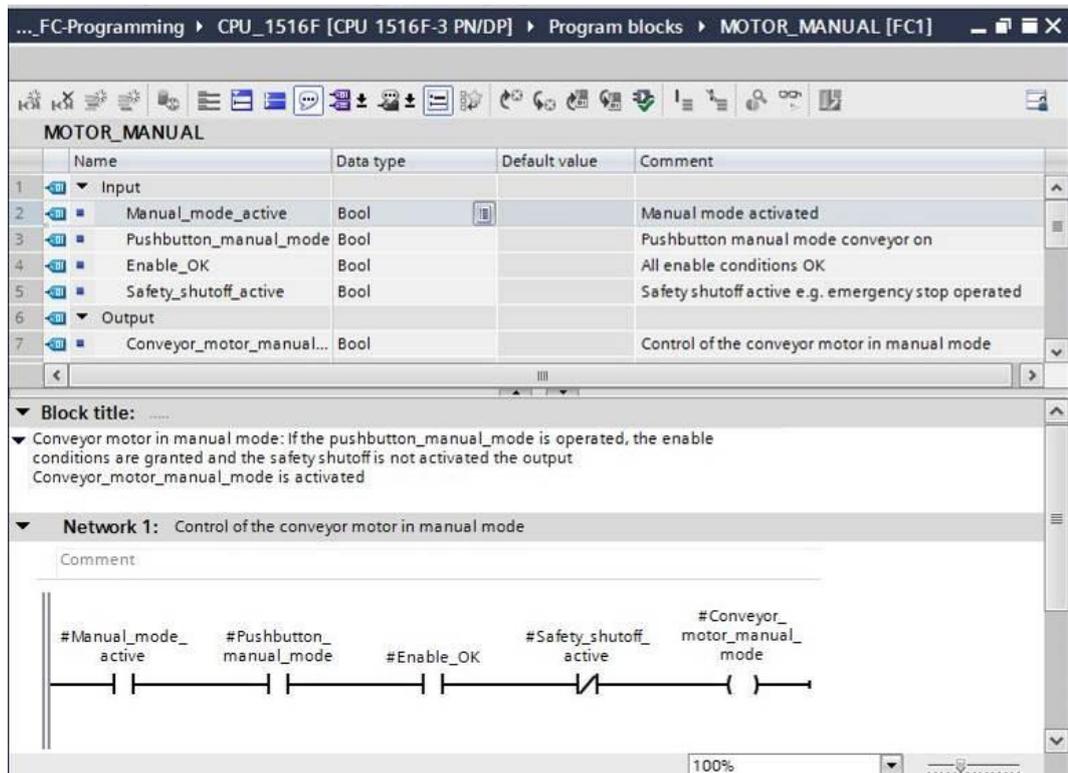
```

    graph LR
      subgraph Inputs
        M1["#Manual_mode_active"]
        M2["#Pushbutton_manual_mode"]
        M3["#Enable_OK"]
        M4["#Safety_shutoff_active"]
      end
      subgraph Logic
        AND("&")
        EQ("=")
      end
      subgraph Output
        O1["#Conveyor_motor_manual_mode"]
      end
      M1 --- AND
      M2 --- AND
      M3 --- AND
      M4 --- AND
      AND --- EQ
      EQ --- O1
  
```

- En las propiedades del bloque, opción "General", puede cambiar el "Language (Lenguaje)" a KOP (Esquema de contactos). (→ Properties (Propiedades) → General → Language (Lenguaje): LAD (KOP))



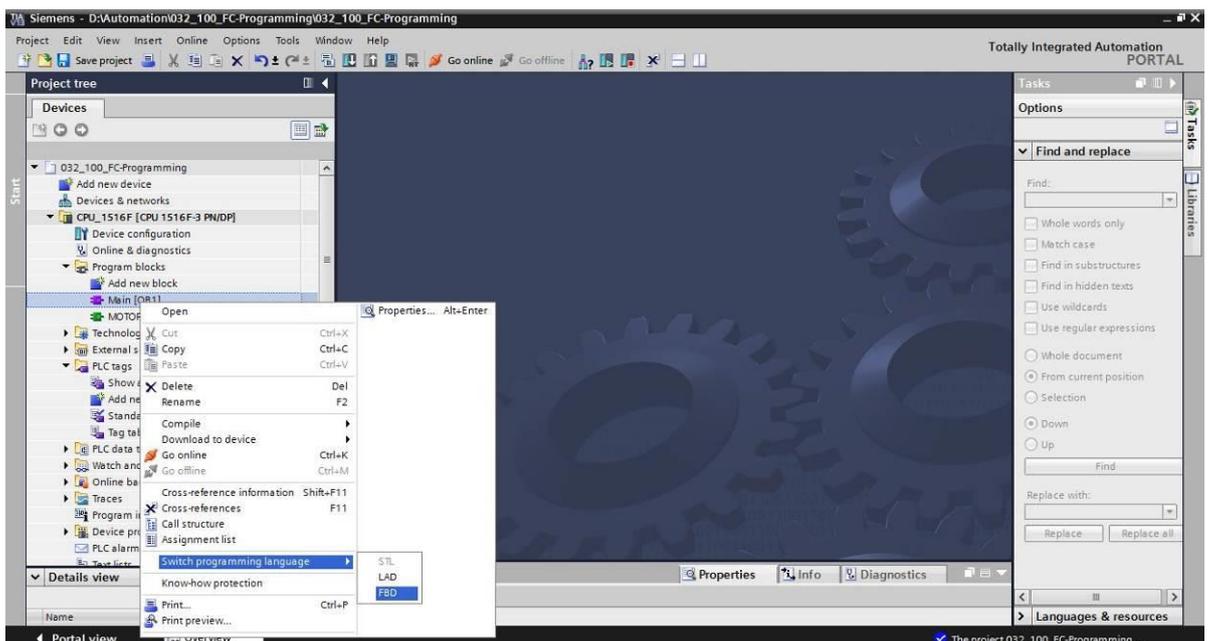
- En KOP, el programa es como se indica a continuación.



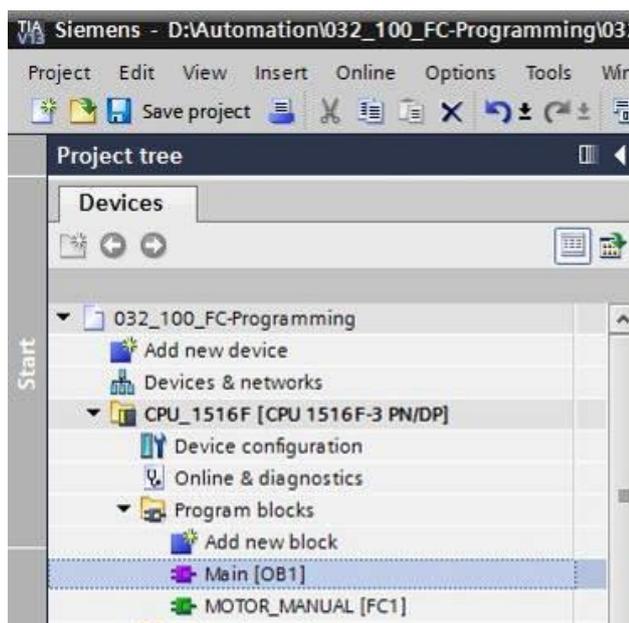
7.8 Programación del bloque de organización OB1: control de la marcha de la cinta hacia delante en modo manual

→ Antes de programar el bloque de organización "Main[OB1]", cambiaremos el lenguaje de programación a FUP (Diagrama de funciones). Para ello, haga clic en primer lugar con el botón izquierdo del ratón en la carpeta "Program blocks (Bloques de programa)", opción "Main[OB1]".

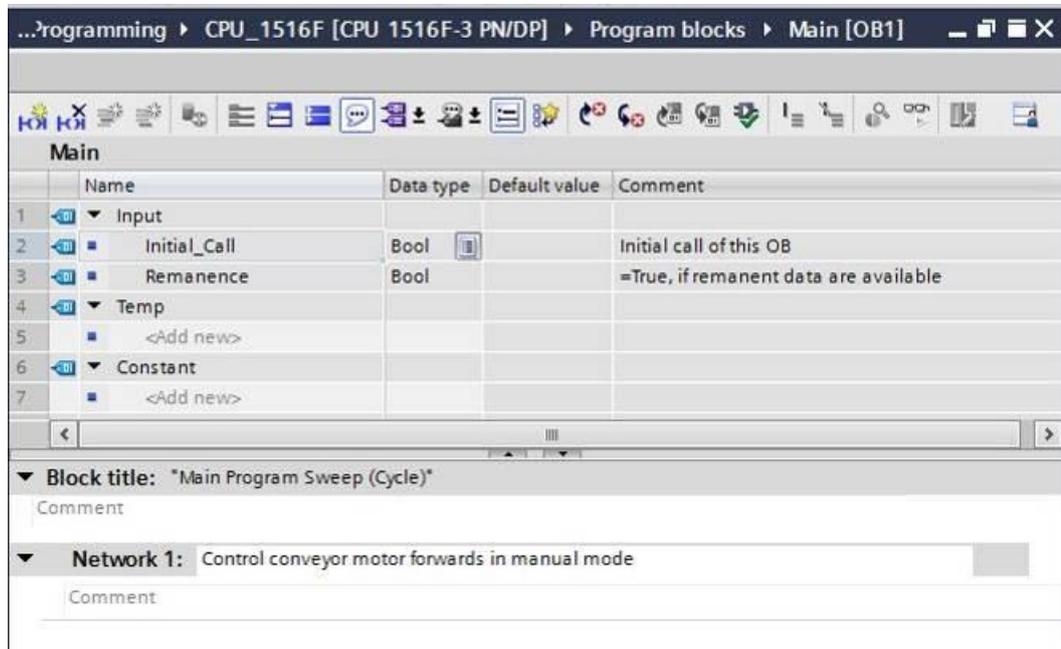
(→ CPU_1516F [CPU 1516F-3 PN/DP → Program blocks (Bloques de programa) → Main [OB1] → Switch programming language (Cambiar lenguaje de programación) → FBD (FUP))



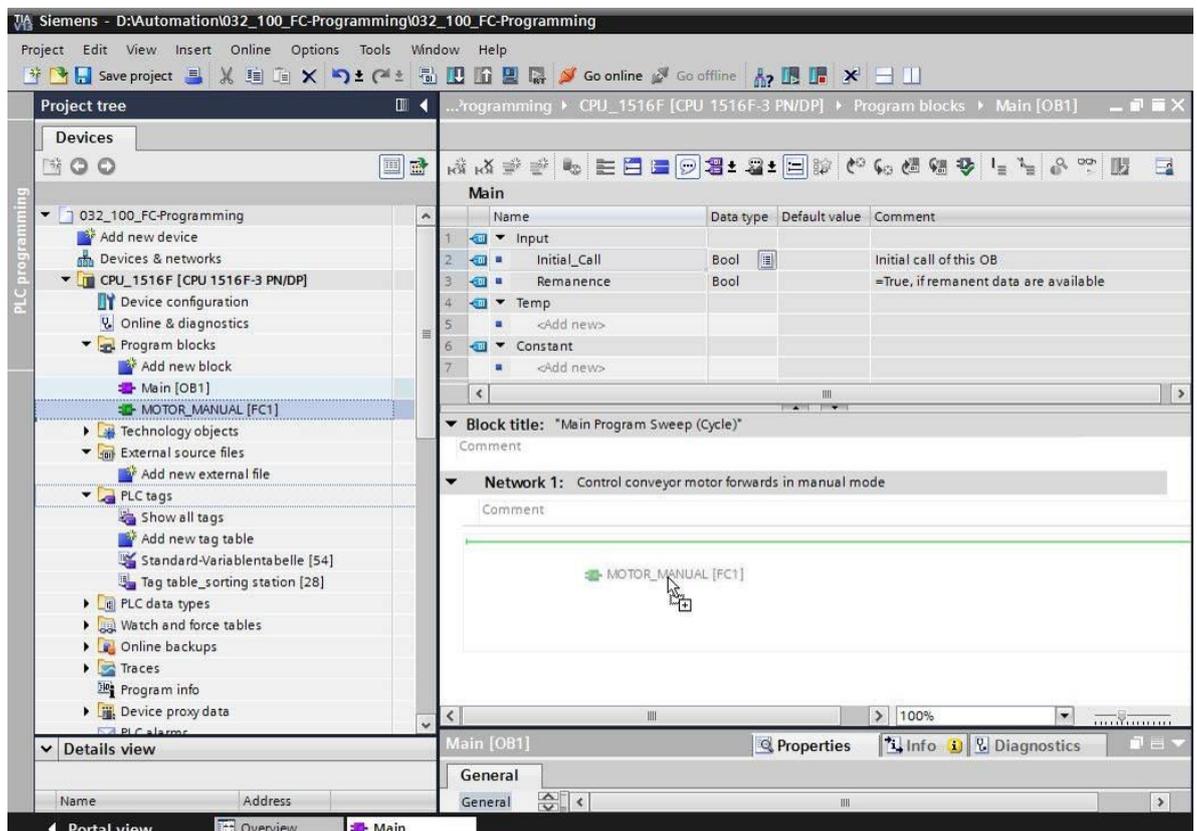
→ Abra el bloque de organización "Main [OB1]" haciendo doble clic.



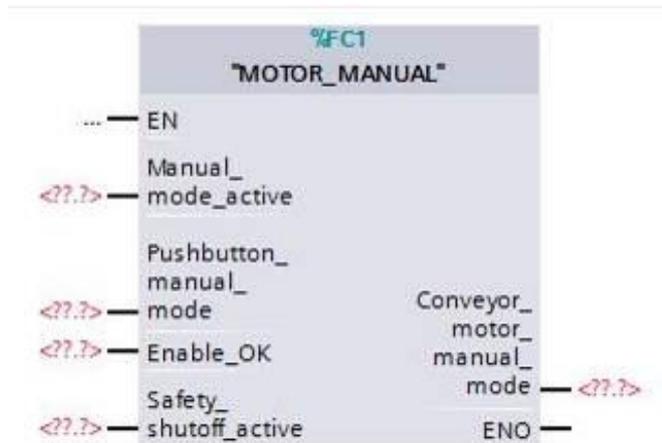
- Asigne al segmento 1 el nombre "Control conveyor motor forwards in manual mode (Control de la marcha de la cinta hacia delante en modo manual o Jog)".
- (→ Network 1 (Segmento 1):... → Control conveyor motor forwards in manual mode (Control de la marcha de la cinta hacia delante en modo manual o Jog))



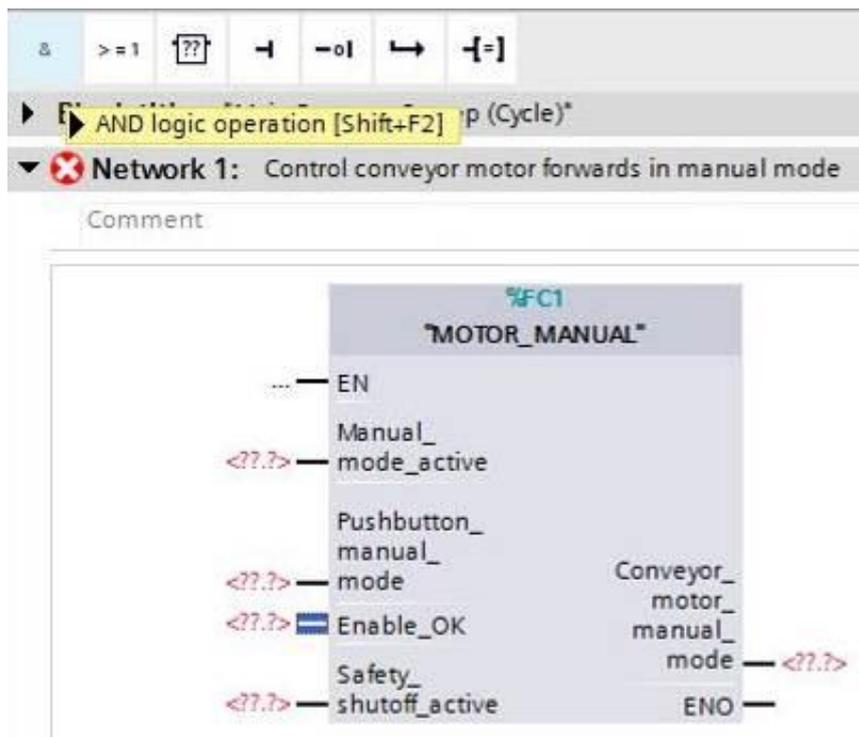
- Ahora desplace la función "MOTOR_MANUAL [FC1]" hasta la línea verde del segmento 1 mediante "arrastrar y soltar".



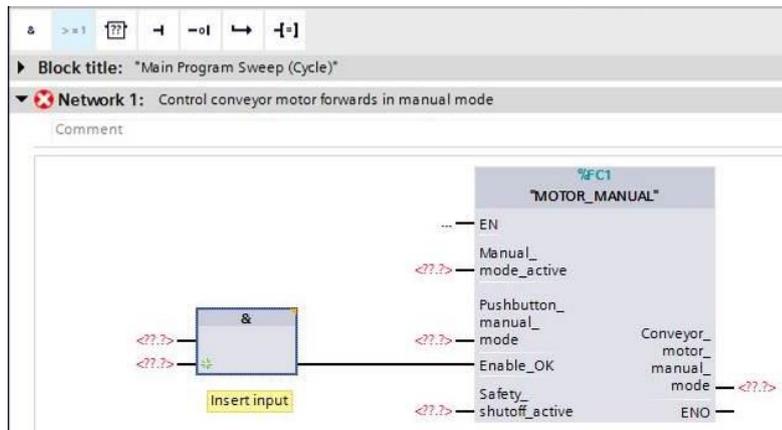
→ Se insertará en el Network 1 (Segmento 1) un bloque con la interfaz que se ha definido y las conexiones EN y ENO.



→ Para insertar un Y antes del parámetro de entrada "Enable_OK (Habilitación_OK)", seleccione dicha entrada e inserte el Y haciendo clic en el icono  de la barra de herramientas lógicas. (→ )

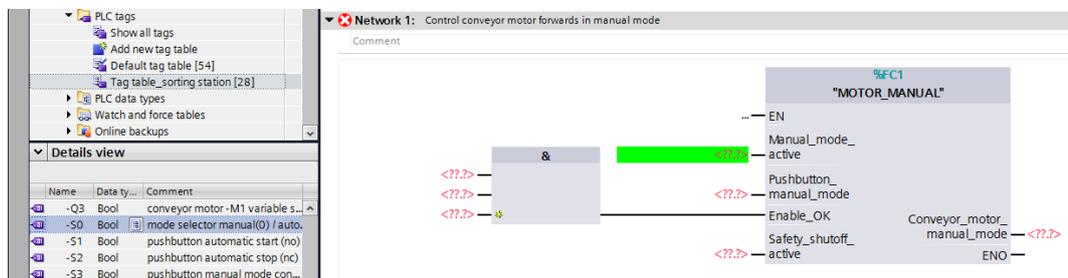


→ Para ello, haga clic en el asterisco amarillo  del elemento Y a fin de agregar una entrada más. (→ )

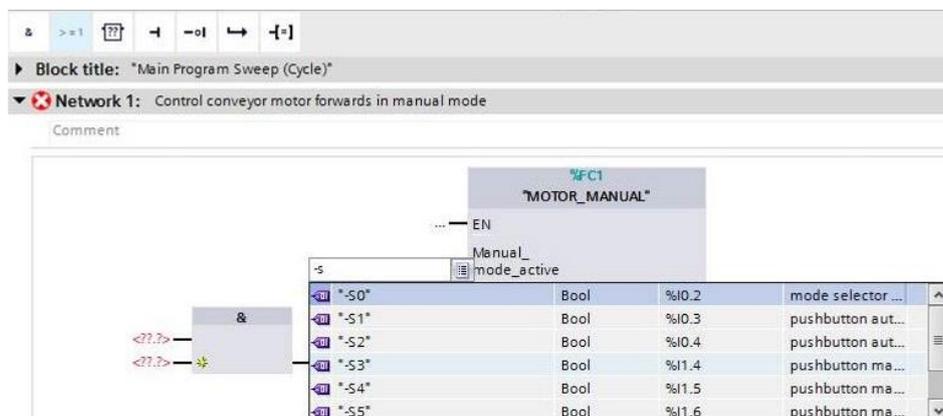


→ Para interconectar el bloque con las variables globales de la "Tag table_sorting station (Tabla_variables_planta_clasificación)", disponemos de dos posibilidades:

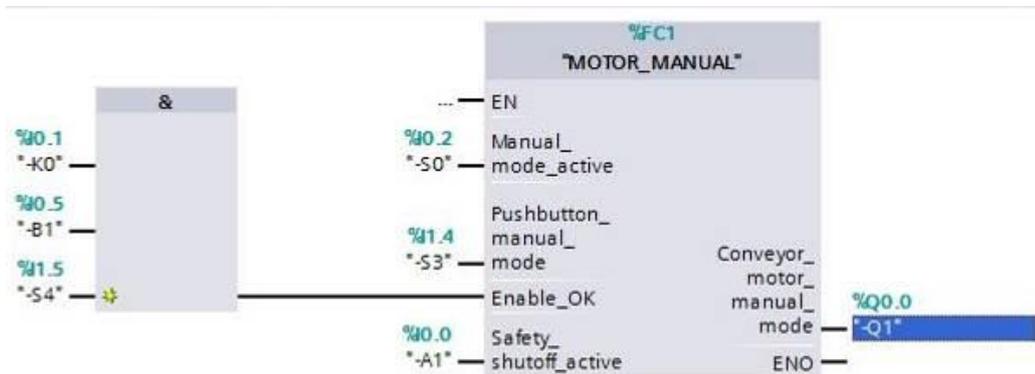
→ La primera consiste en seleccionar la "Tag table_sorting station (Tabla_variables_planta_clasificación)" en el árbol del proyecto y copiar la variable global deseada desde la vista detallada a la interfaz de FC1 mediante "arrastrar y soltar" (→ Tag table_sorting station (Tabla_variables_planta_clasificación) → Details view (Vista detallada) → -S0 → Manual_mode_active (Modo_manual_activo))



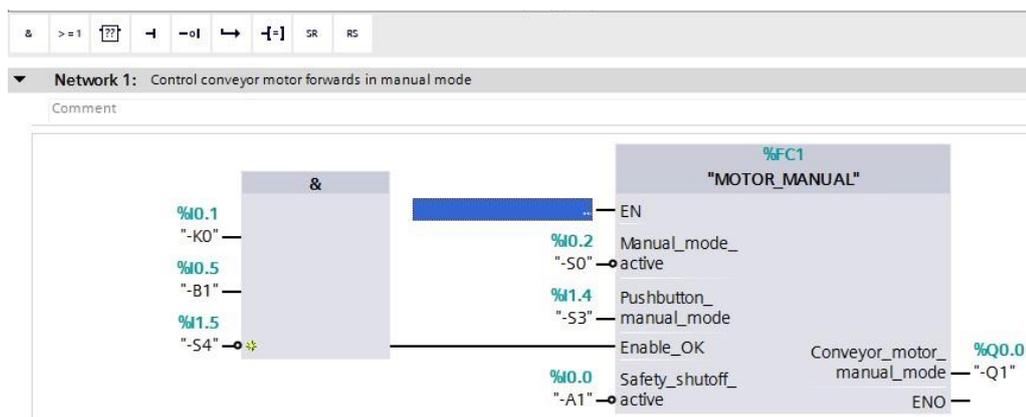
→ La segunda, en introducir en $\langle ???.? \rangle$ las letras iniciales de la variable global deseada (p. ej., "-S") y, en la lista que aparecerá, seleccionar la variable global de entrada "-S0" (%I0.2). (→ Manual_mode_active (Modo_manual_activo) → -S → -S0)



- Inserte las restantes variables de entrada "-S3", "-K0", "-B1", "-S4" y "-A1" y, a continuación, en la salida "Conveyor_motor_manual_mode (Motor_cinta_modos_Jog)", la variable de salida "-Q1" (%Q0.0).

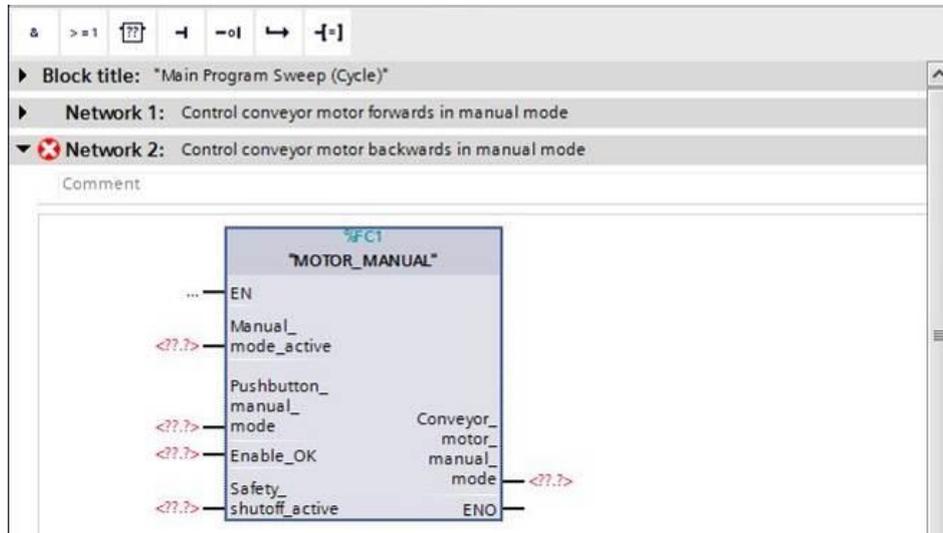


- Niegue las consultas de las variables de entrada "-S0", "-S4" y "-A1" seleccionándolas y haciendo clic a continuación en . (→ -S0 → → -S4 → → -A1 →)

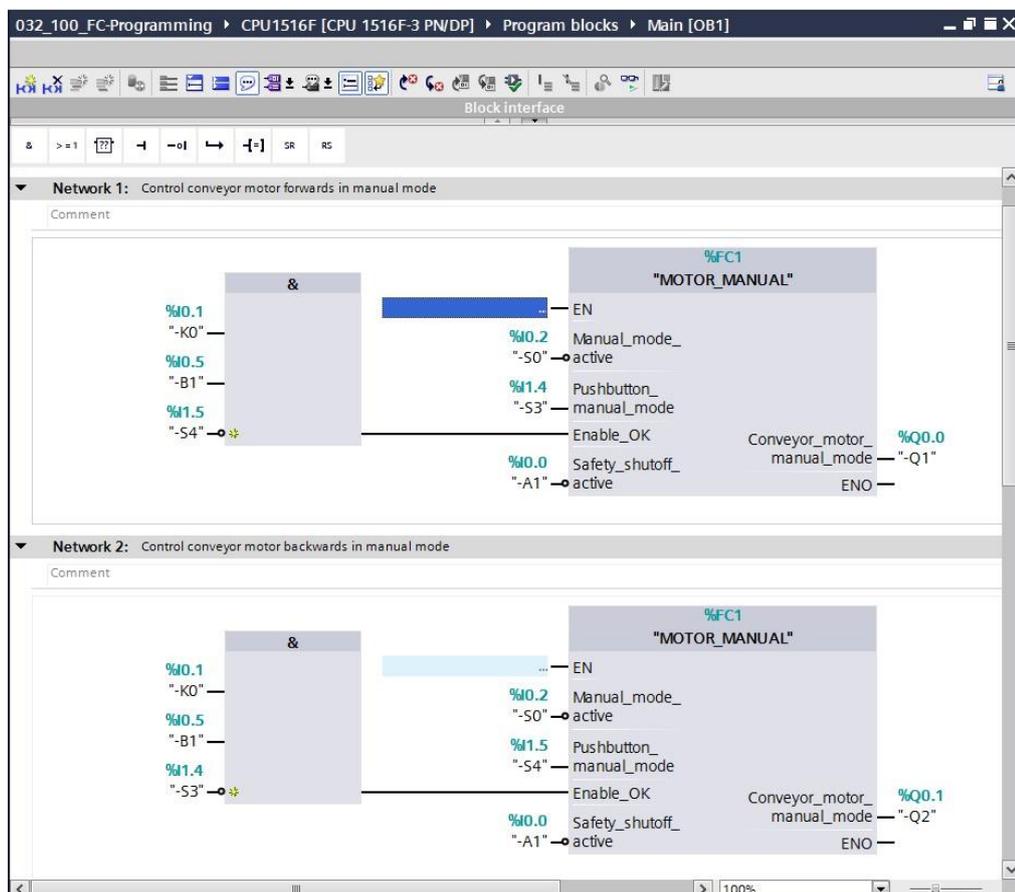


7.9 Programación del bloque de organización OB1: control de la marcha de la cinta hacia atrás en modo manual

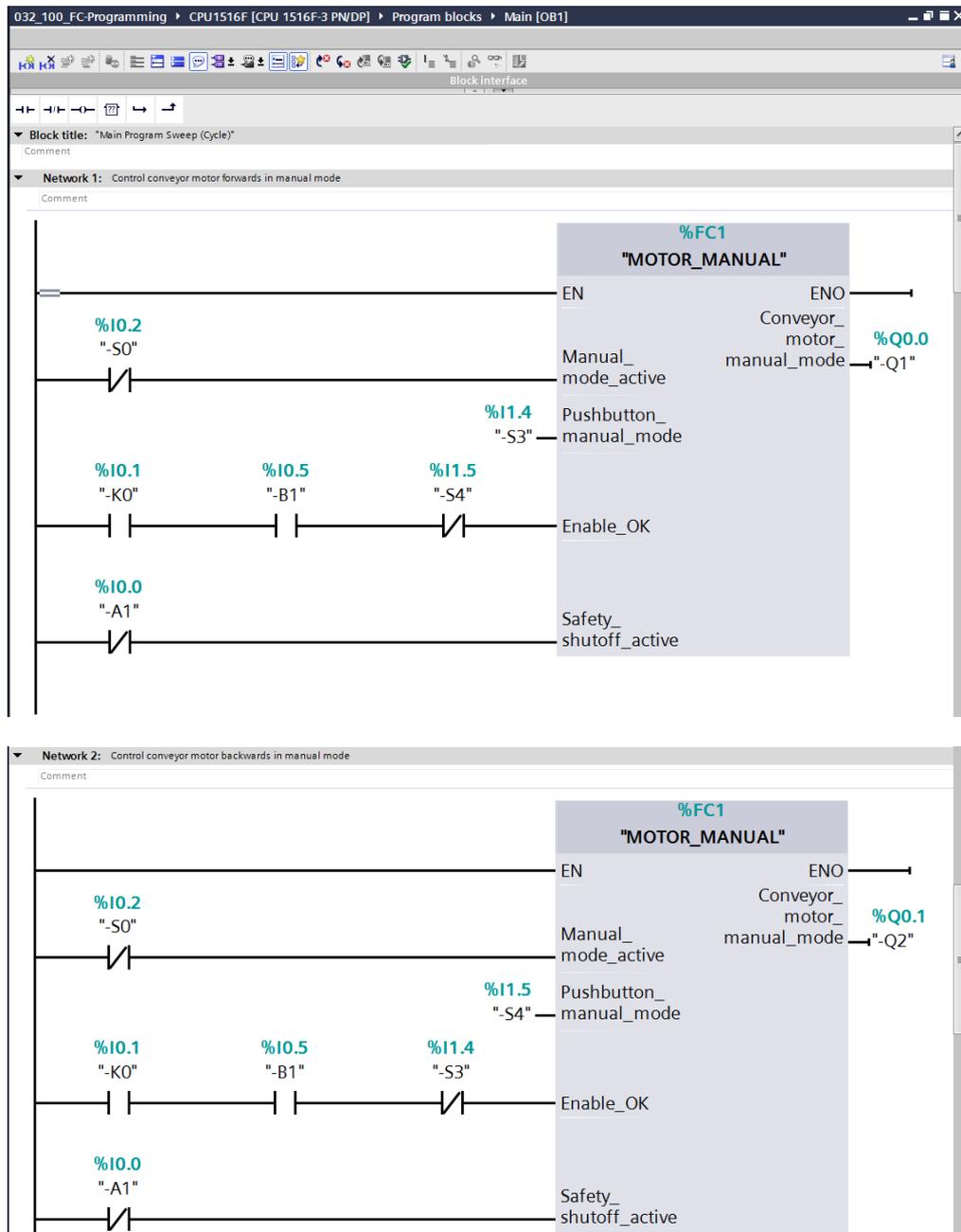
- Asigne al segmento 2 el nombre "Control conveyor motor forwards in manual mode" (Control de la marcha de la cinta hacia delante en modo manual o Jog) e inserte la función "MOTOR_MANUAL [FC1]" mediante "arrastrar y soltar" como ya hizo en el segmento 1.



- Interconecte la función como se muestra aquí. En el lenguaje de programación FUP (Diagrama de funciones) obtendrá el siguiente resultado.

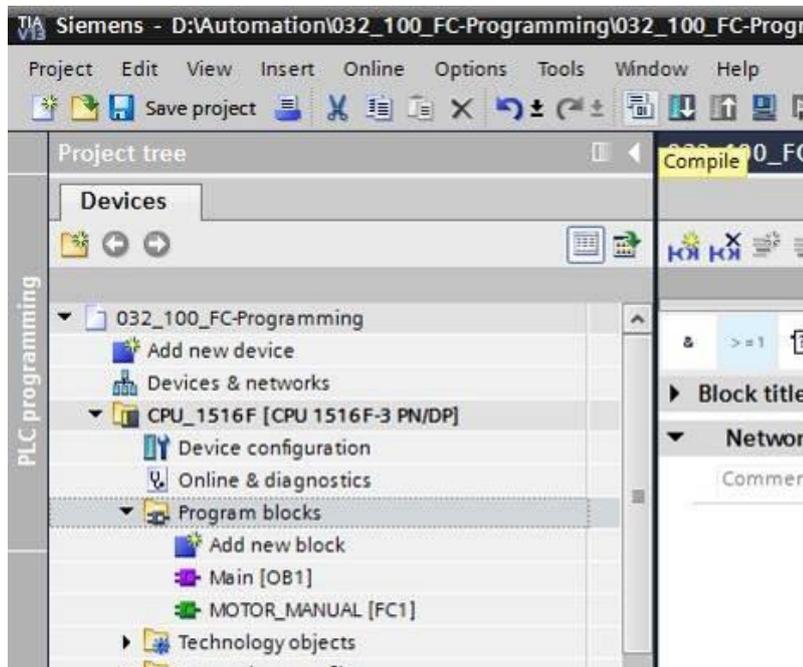


→ En el lenguaje de programación KOP (Esquema de contactos), el resultado es como sigue.



7.10 Almacenamiento y compilación del programa

- Para guardar el proyecto, seleccione en el menú el botón  Save project. Para compilar todos los bloques, haga clic en la carpeta "Program blocks (Bloques de programa)" y seleccione en el menú el icono  de compilación. (→  Save project → Program blocks (Bloques de programa) → )

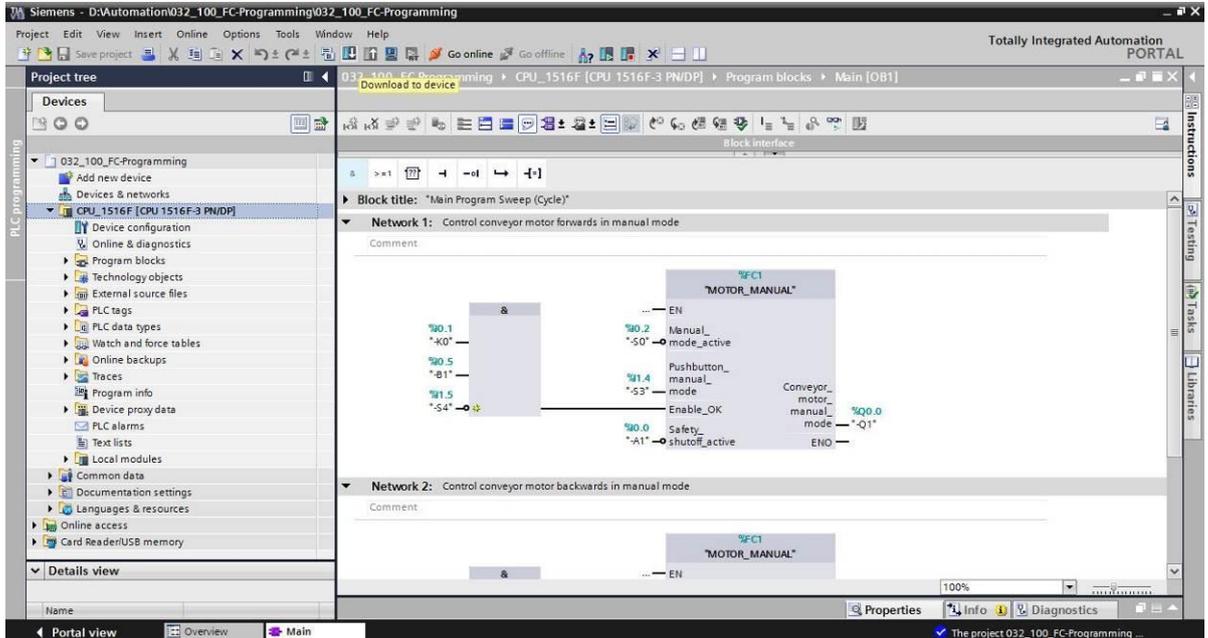


- A continuación se mostrarán, en la pestaña "Info (Información)" "Compile (Compilar)", los bloques que se han podido compilar correctamente.

Path	Description	Go to ?	Errors	Warnings	Time
▼ CPU_1516F		↗	0	0	12:16:44 PM
▼ Program blocks		↗	0	0	12:16:44 PM
MOTOR_MANUAL (FC1)	Block was successfully compiled.	↗			12:16:44 PM
Main (OB1)	Block was successfully compiled.	↗			12:16:44 PM
	Compiling completed (errors: 0; warnings: 0)				12:16:44 PM

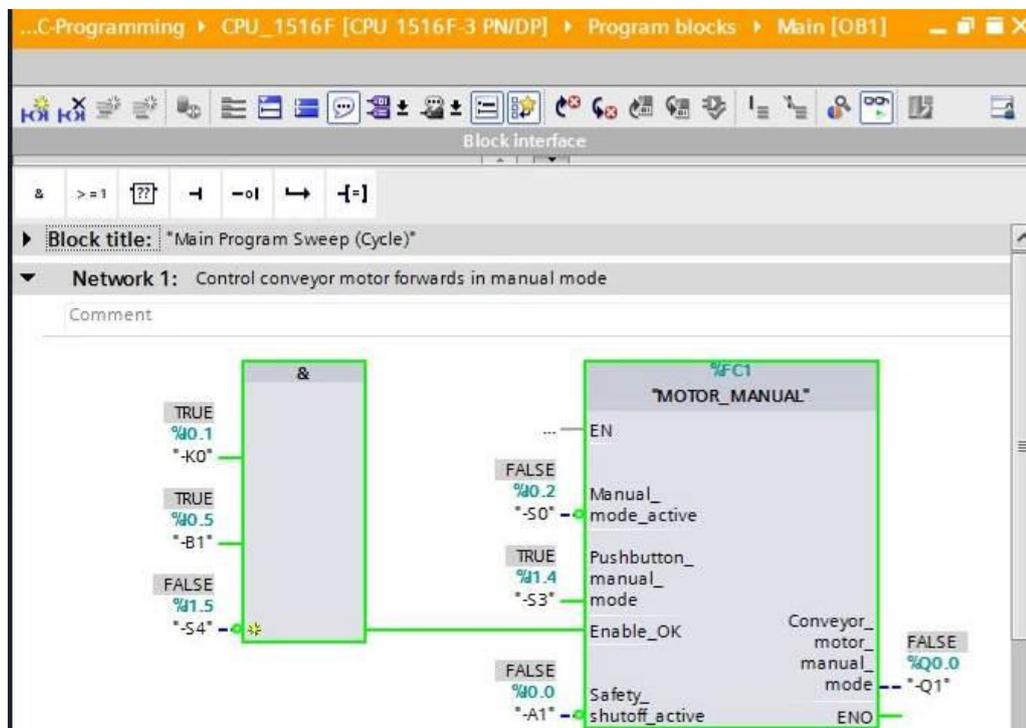
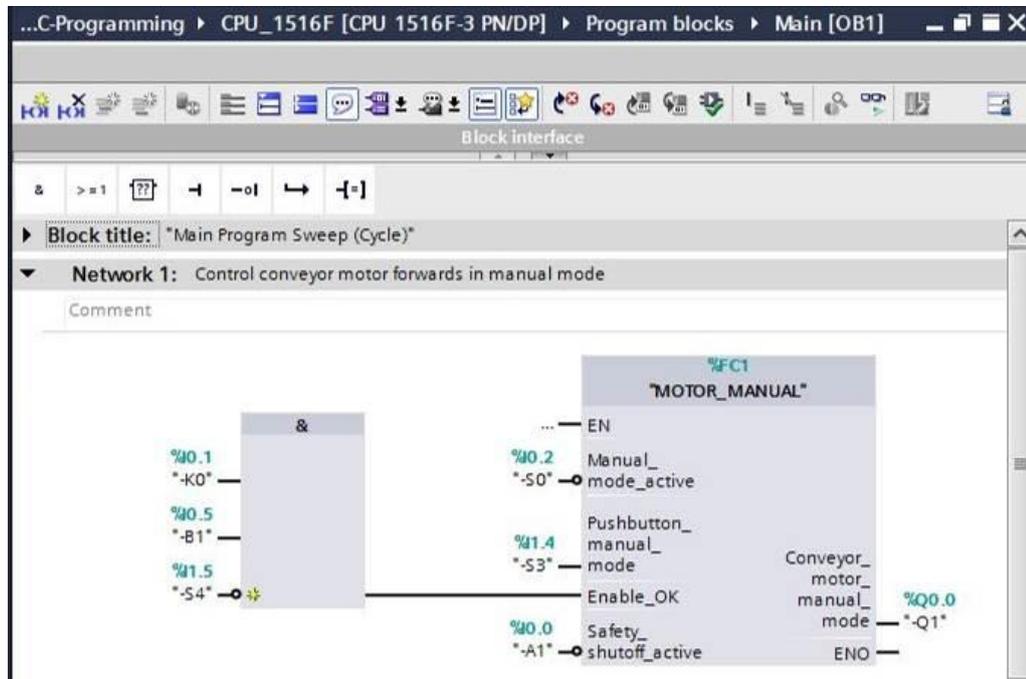
7.11 Carga del programa

- Una vez realizada la compilación correctamente, puede cargar el controlador completo con el programa que ha creado, del modo descrito en los módulos dedicados a la configuración hardware. (→ )



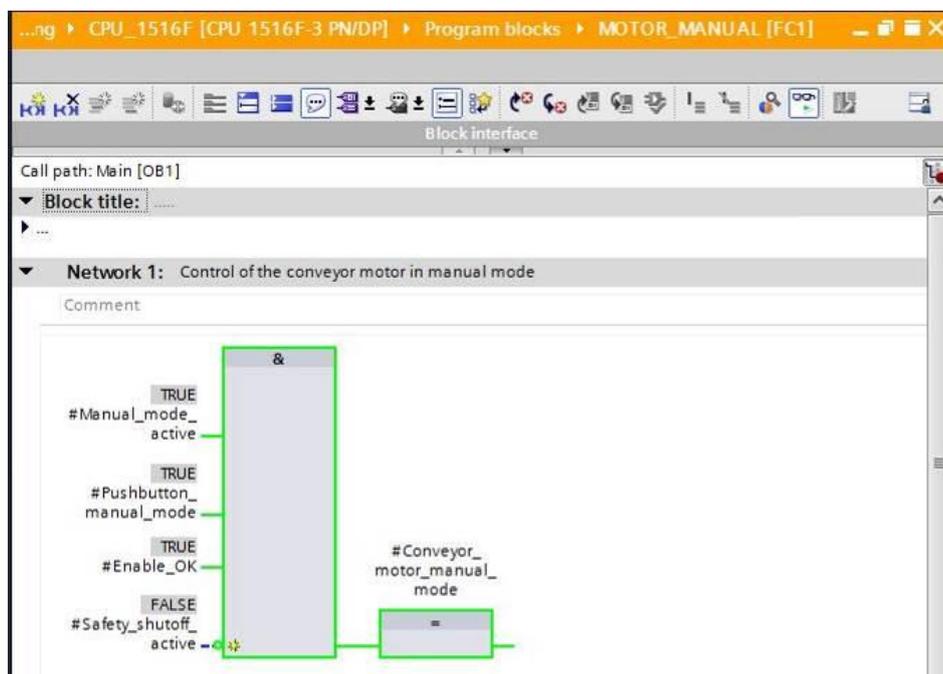
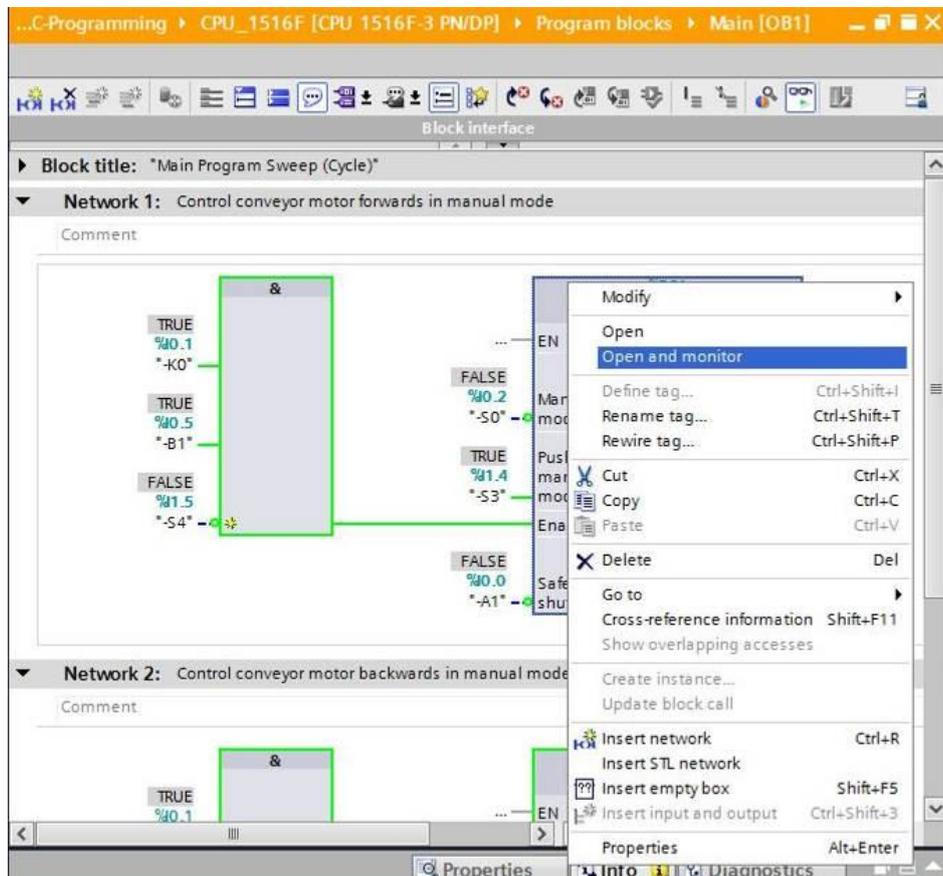
7.12 Visualización de los bloques de programa

→ Para observar el programa cargado, debe estar abierto el bloque deseado. Haciendo clic en el icono  se activa/desactiva la visualización. (→ Main [OB1] → )



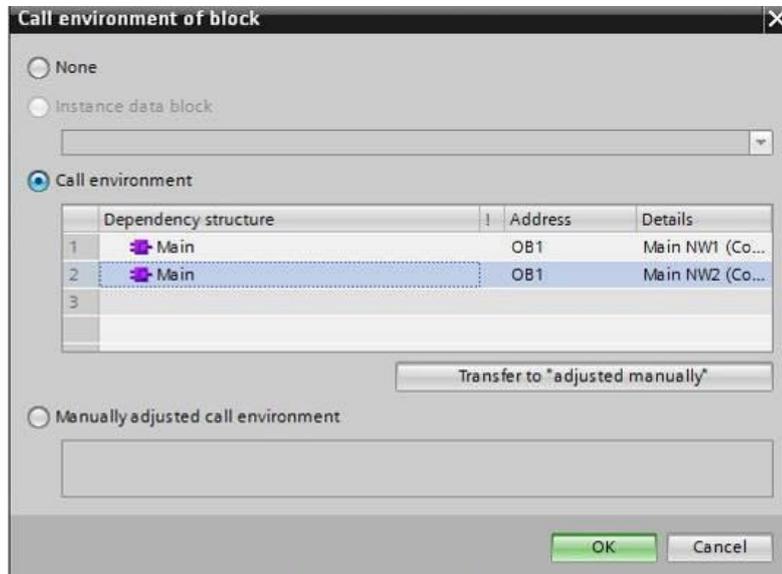
Nota: La observación se realiza con referencia a la señal y en función del controlador. Los estados lógicos de los bornes se indican mediante TRUE y FALSE.

- Para abrir y observar la función "Main [OB1]" llamada en el bloque de organización "MOTOR_MANUAL" [FC1], selecciónela directamente tras hacer clic en ella con el botón derecho del ratón. (→ "MOTOR_MANUAL" [FC1] → Open and monitor (Abrir y observar))



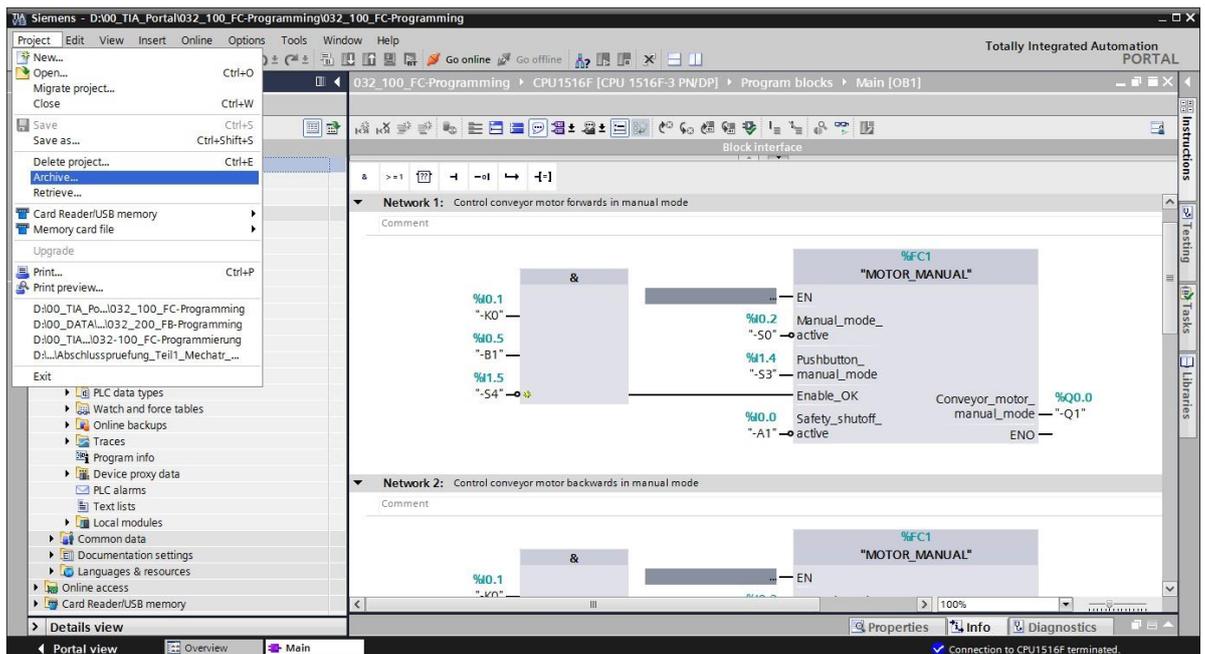
Nota: La observación se realiza con referencia a la función y dependiendo del controlador. La activación de los sensores y el estado de la instalación se indican mediante TRUE y FALSE.

- Si se desea observar una ubicación concreta de la función "MOTOR_MANUAL" [FC1], debe seleccionarse el entorno de llamada mediante el icono . (→  → Call environment (Entorno de llamada) → OK (Aceptar))



7.13 Archivación del proyecto

- Para finalizar vamos a archivar el proyecto completo. Seleccione en el menú → "Project (Proyecto)" la opción → "Archive ... (Archivar...)". Seleccione la carpeta en la que desee archivar el proyecto y guárdelo con el tipo de archivo "TIA Portal project archives (Archivos de proyecto del TIA Portal)". (→ Project (Proyecto) → Archive (Archivar) → TIA Portal project archives (Archivos de proyecto del TIA Portal) → 032-100 FC Programming... → Save (Guardar))



8 Lista de comprobación

N.º	Descripción	comprobado
1	Compilación correcta y sin avisos de error	
2	Carga correcta y sin avisos de error	
3	Conectar la instalación (-K0 = 1) Cilindro introducido/respuesta activada (-B1 = 1) Parada de emergencia (-A1 = 1) no activada Modo de operación MANUAL (-S0 = 0) Activar modo Jog cinta hacia delante (-S3 = 1), luego motor cinta hacia delante, velocidad fija (-Q1 = 1)	
4	Como 3, pero activando parada de emergencia (-A1 = 0) → -Q1 = 0	
5	Como 3, pero en modo de operación AUTO (-S0 = 1) → -Q1 = 0	
6	Como 3, pero desconectando la instalación (-K0 = 0) → -Q1 = 0	
7	Como 3, pero con el cilindro no introducido (-B1 = 0) → -Q1 = 0	
8	Conectar la instalación (-K0 = 1) Cilindro introducido/respuesta activada (-B1 = 1) Parada de emergencia (-A1 = 1) no activada Modo de operación MANUAL (-S0 = 0) Activar modo Jog cinta hacia atrás (-S4 = 1), luego motor cinta hacia atrás, velocidad fija (-Q2 = 1)	
9	Como 8, pero activando PARADA DE EMERGENCIA (-A1 = 0) → -Q2 = 0	
10	Como 8, pero en modo de operación AUTO (-S0 = 1) → -Q2 = 0	
11	Como 8, pero desconectando la instalación (-K0 = 0) → -Q2 = 0	
12	Como 8, pero con el cilindro no introducido (-B1 = 0) → -Q2 = 0	
13	Como 8, pero activando también Modo Jog cinta hacia delante (-S3 = 1) → -Q1 = 0 y también -Q2 = 0	
14	Proyecto archivado correctamente	

9 Ejercicio

9.1 Tarea planteada: ejercicio

En este ejercicio se planificarán, programarán y probarán las siguientes funciones del ejemplo de proceso basado en una planta de clasificación.

- Modo manual: extraer cilindro
- Modo manual: introducir cilindro

Nota: Tenga en cuenta la posibilidad de reutilización o encapsulamiento de las funciones.

9.2 Planificación

Ahora, planifique de forma autónoma la implementación de la tarea.

9.3 Lista de comprobación: ejercicio

N.º	Descripción	comproba
1	Función (FC): se ha creado CILINDRO_MANUAL	
2	Se han definido las interfaces	
3	Se ha programado la función	
4	Se ha insertado la función FC2 en el segmento 3 del OB1	
5	Se han interconectado las variables de entrada para introducir cilindro	
6	Se han interconectado las variables de salida para introducir cilindro	
7	Compilación correcta y sin avisos de error	
8	Se ha insertado la función FC2 en el segmento 4 del OB1	
9	Se han interconectado las variables de entrada para extraer cilindro	
10	Se han interconectado las variables de salida para extraer cilindro	
11	Compilación correcta y sin avisos de error	
12	Carga correcta y sin avisos de error	
13	Conectar la instalación (-K0 = 1) Cilindro introducido/respuesta activada (-B1 = 1) Parada de emergencia (-A1 = 1) no activada Modo de operación MANUAL (-S0 = 0) No activar introducir cilindro (-S5 = 0) Activar extraer cilindro (-S6 = 1), luego extraer cilindro (-M3 = 1), OK	
14	Conectar la instalación (-K0 = 1) Cilindro extraído/respuesta activada (-B2 = 0) Parada de emergencia (-A1 = 1) no activada Modo de operación MANUAL (-S0 = 0) No activar extraer cilindro (-S6 = 0) Activar introducir cilindro (-S5 = 1), luego introducir cilindro (-M2 = 1), OK	
15	Introducir y extraer cilindro no activables simultáneamente	
16	Proyecto archivado correctamente	

10 Información adicional

Con fines orientativos, se ofrece también información adicional para la puesta en práctica y la profundización, como, p. ej.: Getting Started (primeros pasos), vídeos, tutoriales, aplicaciones, manuales, guías de programación y versiones de prueba del software y el firmware, todo ello en el siguiente enlace:

www.siemens.com/sce/s7-1500