



SIEMENS

Documentación didáctica/ para cursos de formación

Siemens Automation Cooperates with Education
(SCE) | A partir de la versión V14 SP1

Módulo TIA Portal 031-200

Principios básicos de la programación
de FB con SIMATIC S7-1200

[siemens.com/sce](https://www.siemens.com/sce)

SIEMENS

Global Industry
Partner of
WorldSkills
International



Paquetes SCE apropiados para esta Documentación didáctica

- **SIMATIC S7-1200 AC/DC/RELÉ paquete de 6 "TIA Portal"**
Ref.: 6ES7214-1BE30-4AB3
- **SIMATIC S7-1200 DC/DC/DC 6er "TIA Portal"**
Ref.: 6ES7214-1AE30-4AB3
- **Upgrade SIMATIC STEP 7 Basic V14 SP1 (para S7-1200) 6er "TIA Portal"**
Ref.: 6ES7822-0AA04-4YE5

Tenga en cuenta que estos paquetes SCE pueden ser sustituidos por paquetes actualizados. Encontrará una relación de los paquetes SCE actualmente disponibles en la página:

www.siemens.com/sce/tp

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 didáctica/para cursos de formación de 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 esta documentación didáctica/para cursos de formación 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 la documentación didáctica/para cursos de formación.

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 esta documentación didáctica/para cursos de formación.

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	Modo automático: motor de cinta.....	16
6.3	Esquema tecnológico	17
6.4	Tabla de asignación.....	18
7	Instrucciones paso a paso estructuradas.....	19
7.1	Desarchivación de un proyecto existente	19
7.2	Creación de una nueva tabla de variables.....	20
7.3	Creación de nuevas variables dentro de una tabla de variables	22
7.4	Importación de la "Tag table_sorting station (Tabla_variables_planta_clasificación)".....	23
7.5	Creación del bloque de función FB1 "MOTOR_AUTO" para el motor de cinta en modo automático	27
7.6	Definición de la interfaz del FB1 "MOTOR_AUTO".....	29
7.7	Programación del FB1: MOTOR_AUTO.....	32
7.8	Programación del bloque de organización OB1: control de la marcha de la cinta hacia delante en modo automático.....	40
7.9	Resultado en el lenguaje de programación KOP (Esquema de contactos)	45
7.10	Guardado y compilación del programa.....	46
7.11	Carga del programa.....	47
7.12	Visualización de los bloques de programa	48
7.13	Archivar proyecto.....	51
7.14	Lista de comprobación.....	52
8	Ejercicio	53
8.1	Tarea planteada: ejercicio.....	53
8.2	Esquema tecnológico	53
8.3	Tabla de asignación.....	54

8.4	Planificación	54
8.5	Lista de comprobación: ejercicio	55
9	Información adicional	56

Principios básicos de la programación de FB

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 programará con él un bloque de función FB1 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 CPU1214C. Sin embargo, también puede trabajar 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:

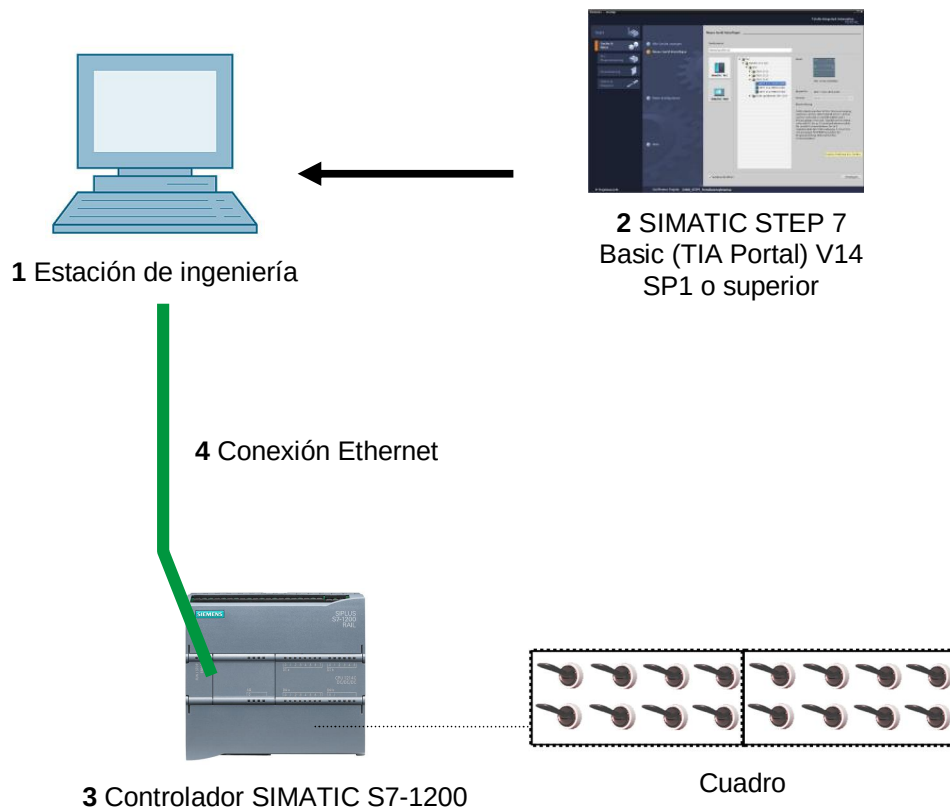
SCE_ES_011-101_Hardware Configuration_CPU1214C.zap14

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 Basic en el TIA Portal – V14 SP1 o superior
- 3 Controlador SIMATIC S7-1200, p. ej., CPU 1214C DC/DC/DC con Signal Board ANALOG OUTPUT SB1232, 1 AO – firmware V4.2.1 o superior

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.:

- Ejecución de un re arranque (en caliente)
- Actualización de las imágenes de proceso de las entradas y de las salidas
- Llamada cíclica del programa de usuario
- Registro de alarmas y llamada de los OB 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 re arranque completo (en caliente) con ayuda de OB 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 con fines de tratamiento de errores.

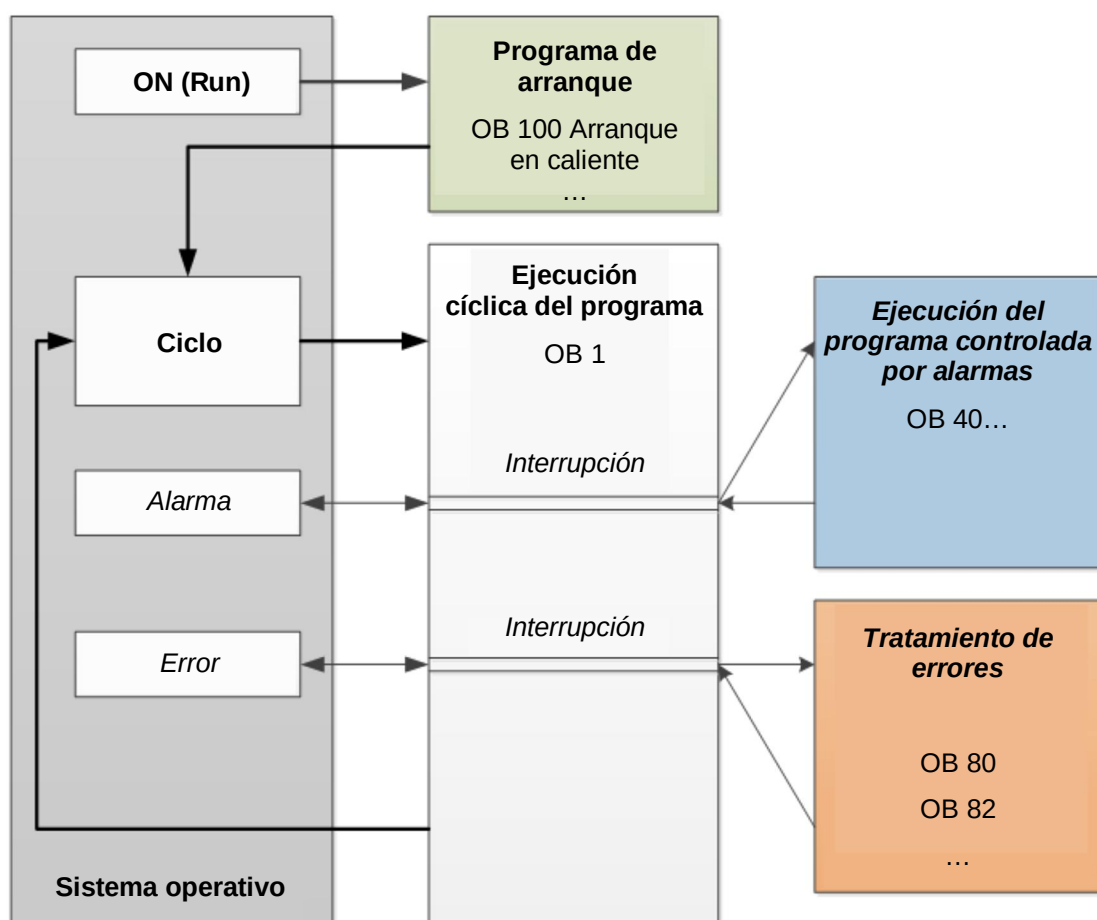


Figura 1: Eventos de arranque en el sistema operativo y llamadas de OB

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 muestra ejemplos de distintos eventos de arranque para un SIMATIC S7-1200. Se muestran también los posibles números de OB y las reacciones predeterminadas del sistema que se producirán si el correspondiente bloque de organización (OB) no está presente en el controlador.

Evento de arranque	Números de OB posibles	Reacción del sistema predeterminada
Arranque	100, ³ 123	Ignorar
Programa cíclico	1, ³ 123	Ignorar
Alarma horaria	10 a 11	-
Alarma de actualización	56	Ignorar
Se ha excedido una vez el tiempo de vigilancia del ciclo	80	Ignorar
Se ha excedido dos veces el tiempo de vigilancia del ciclo	80	STOP
Alarma de diagnóstico	82	Ignorar

Tabla 1: Números de OB para distintos eventos de arranque

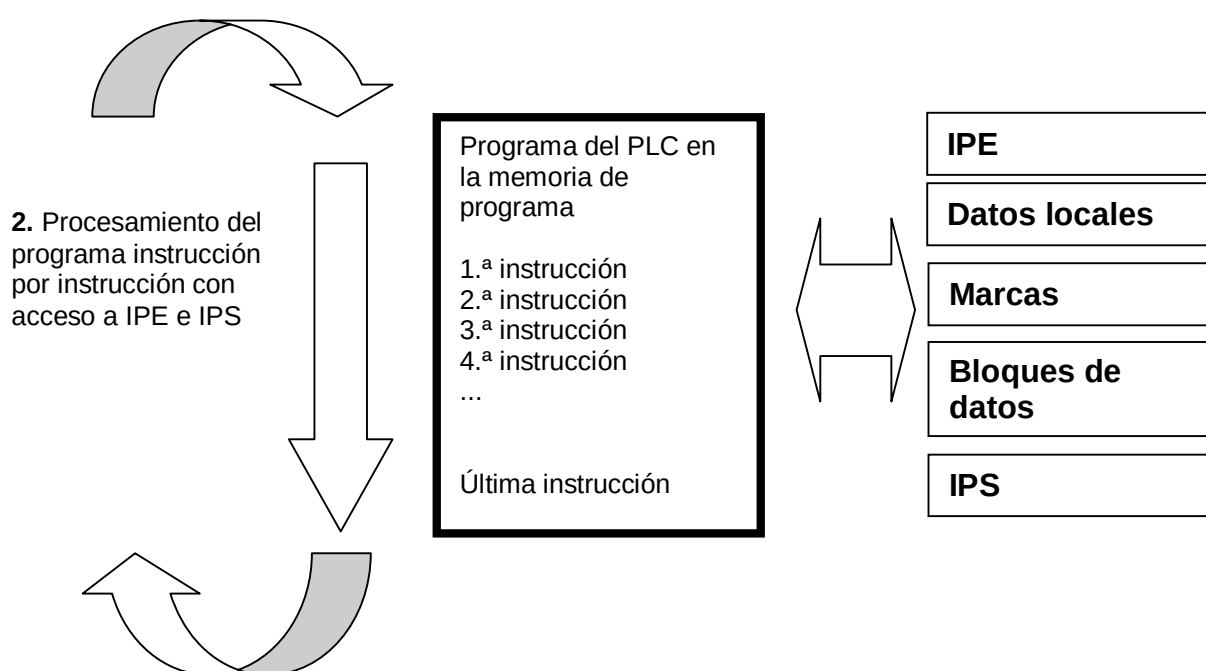
4.3 Imagen de proceso y ejecución cíclica del programa

Si en el programa de usuario cíclico se accede a las entradas (E) y salidas (S), normalmente no se consultan los estados de señal directamente en los módulos de entrada y salida, sino que se accede a un área de la memoria 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. Tras ello, el procesador ejecuta el programa guardado 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 prosigue 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, como controles individuales con operaciones lógicas binarias.

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

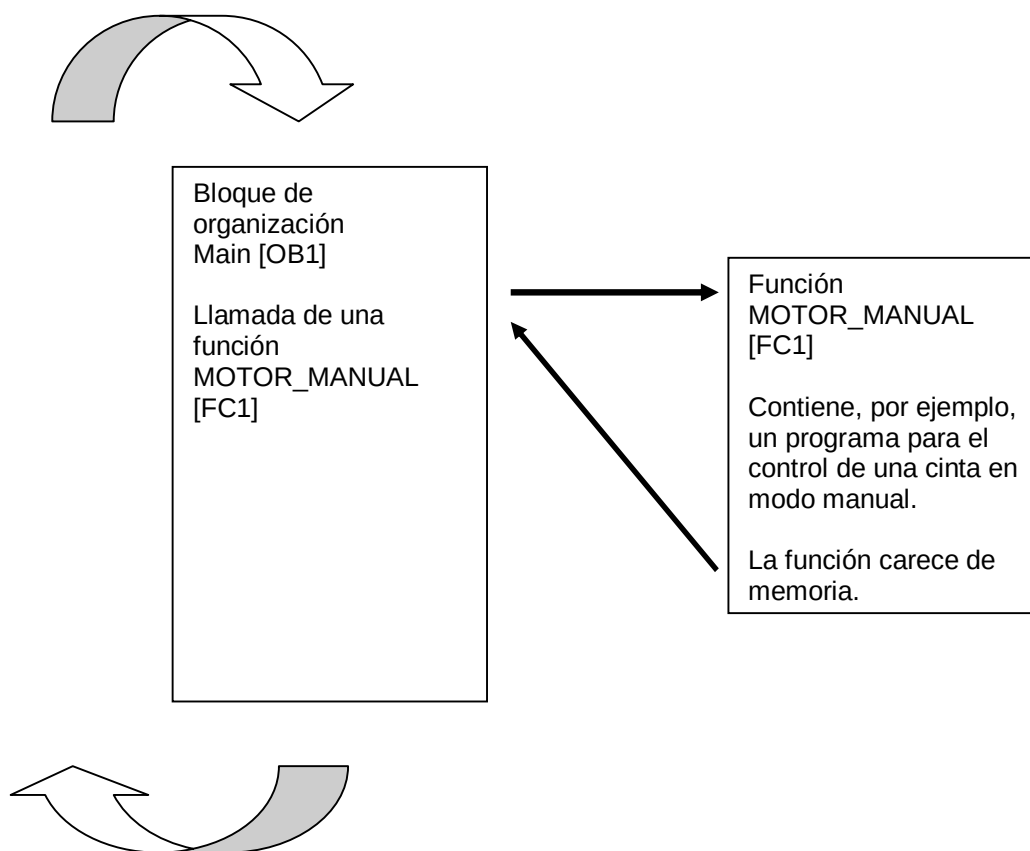


Figura 3: Función con llamada desde el bloque de organización Main[OB1]

4.5 Bloques de función y bloques de datos de instancia

Los bloques de función son bloques lógicos que depositan sus variables de entrada, de 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 de 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.

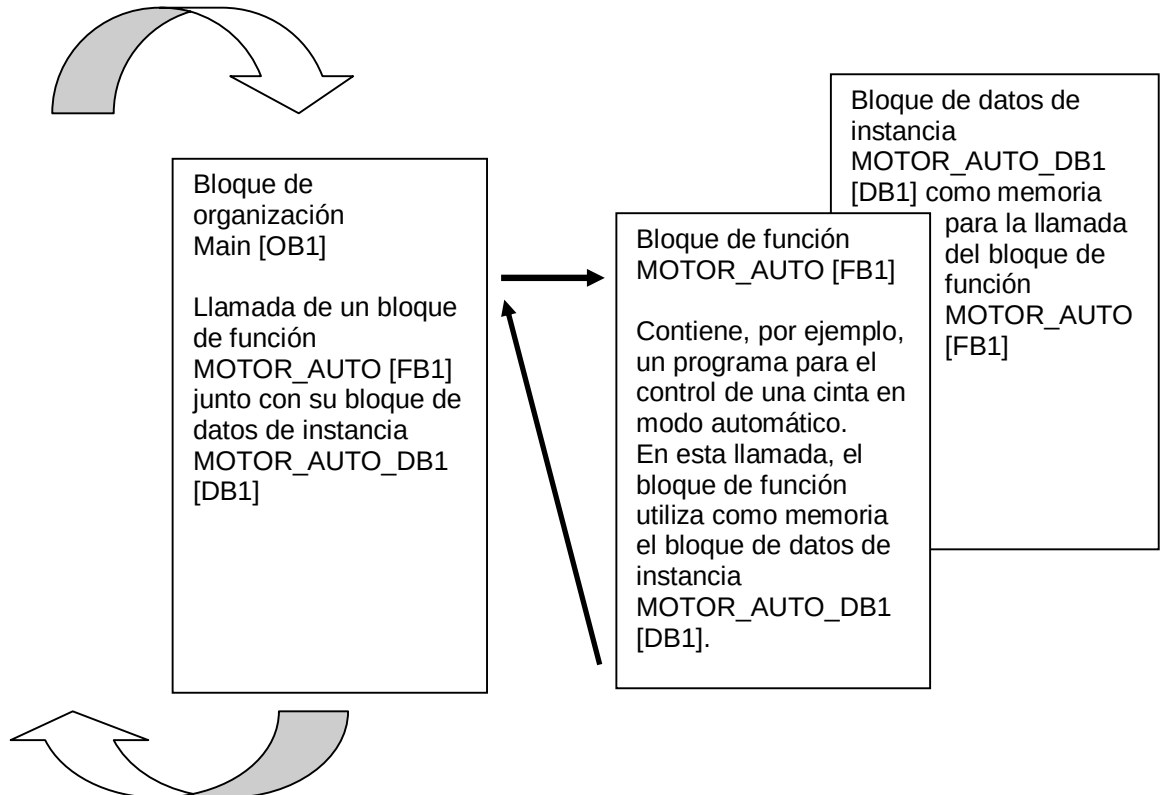


Figura 4: Bloque de función e instancia con llamada desde el bloque de organización Main[OB1]

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.

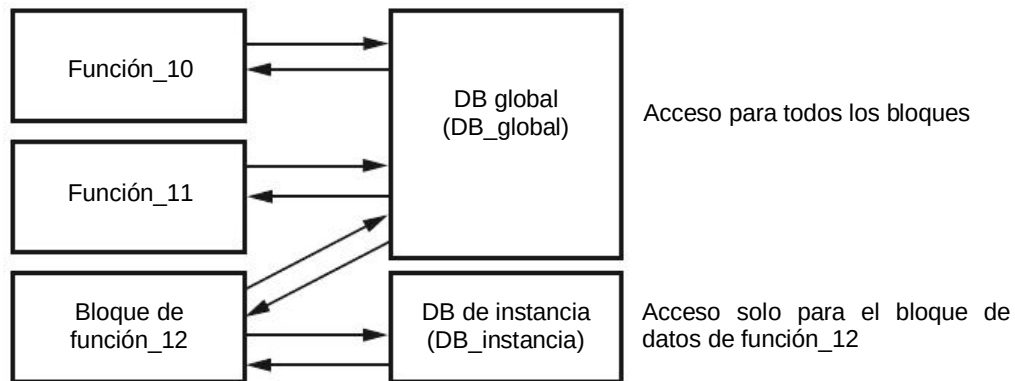


Figura 5: Diferencia entre DB global y DB de instancia.

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.

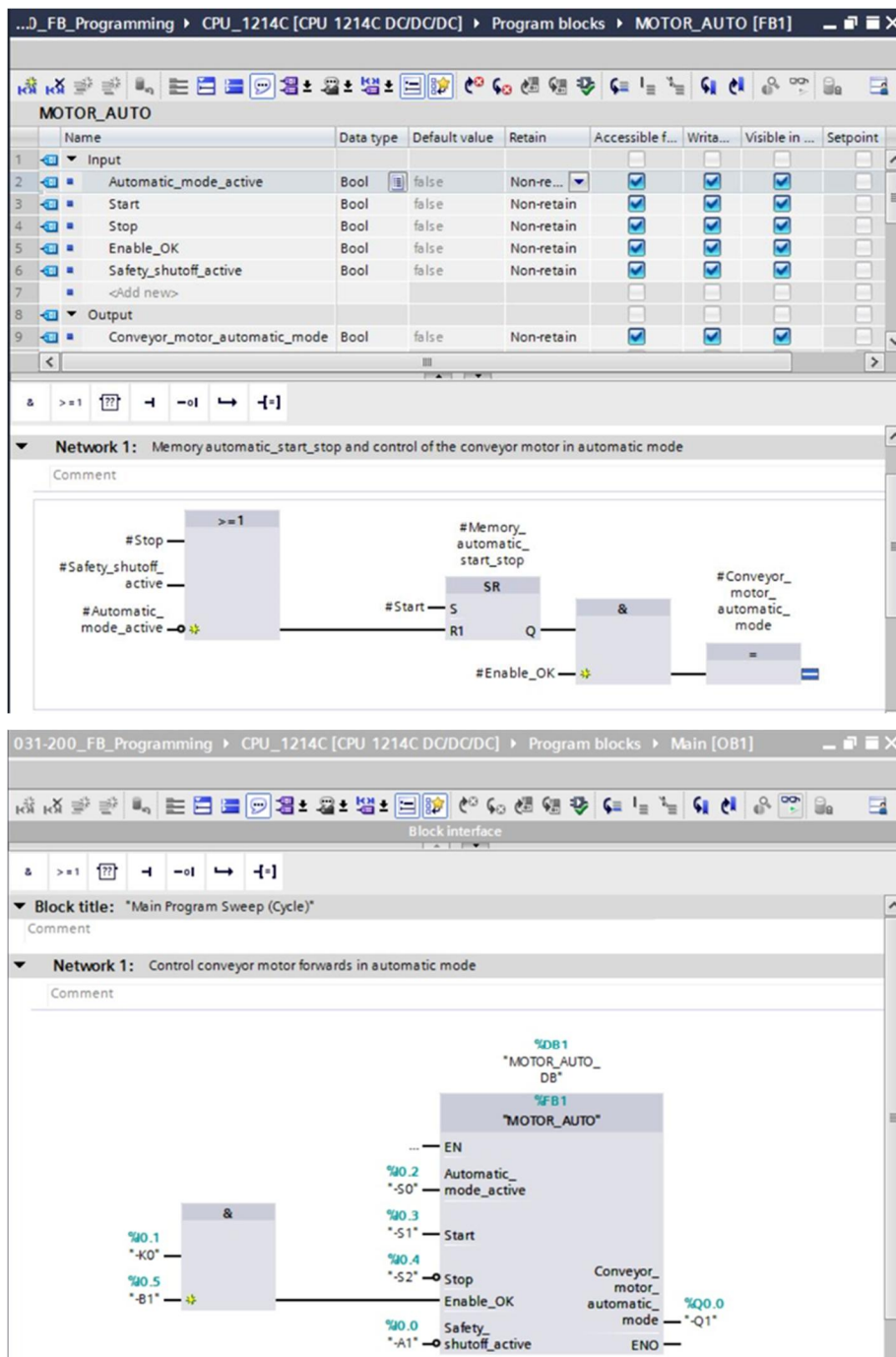


Figura 6: Bloque de función apto para librería con llamada en el OB1

4.8 Lenguajes de programación

Para programar funciones y bloques de función para el controlador SIMATIC S7-1200, se dispone de los lenguajes de programación Diagrama de funciones (FUP), Esquema de contactos (KOP) y Structured Control Language (SCL).

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. Los operandos lógicos "Y", "O" y "O exclusiva" son ejemplos de funciones binarias (ver la Figura 7).

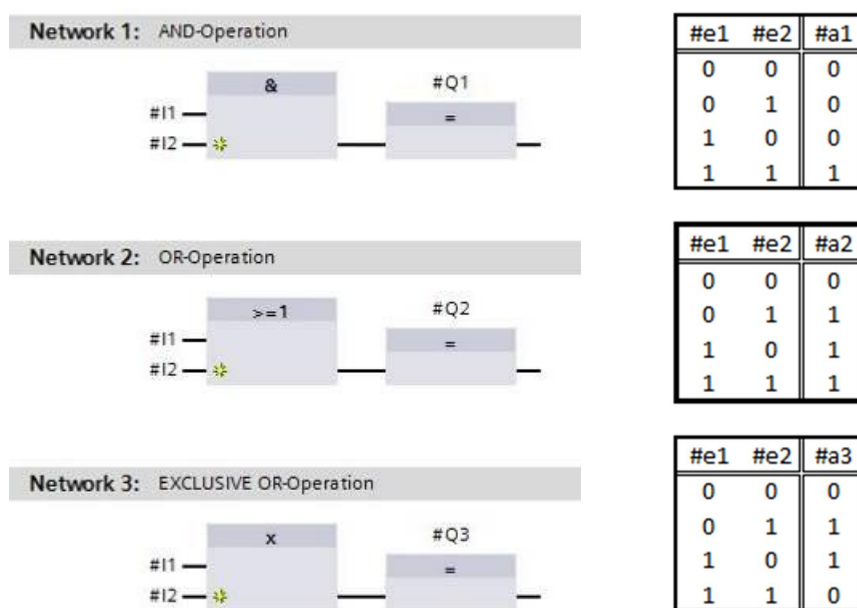


Figura 7: Funciones binarias en FUP y tabla lógica correspondiente

Estas instrucciones simples permiten, por ejemplo, controlar salidas binarias, evaluar flancos o ejecutar funciones de salto dentro de un 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)/de 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". Si se produce 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 del ejemplo de proceso basado en una planta de clasificación.

- Modo automático: motor de cinta

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 al FB 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 Modo automático: motor de cinta

Se desea encapsular el modo automático del motor de cinta en un bloque de función (FB) "MOTOR_AUTO". Con ello, por un lado, se garantiza la claridad en el OB1, y, por el otro, se hace posible la reutilización en caso de ampliarse la planta con una nueva cinta transportadora. Los parámetros planificados se muestran en la Tabla 2.

Input (Entrada)	Tipo de datos	Comentario
Automatic_mode_active (Modo_automático_activo)	BOOL	Modo de operación automático activado
Start (Arranque)	BOOL	Comando de arranque para el modo automático
Stop (Parada)	BOOL	Comando de parada para el modo automático
Enable_OK (Habilitación_OK)	BOOL	Se cumplen todas las condiciones para la habilitación
Safety_shutoff_active (Desconexión de seguridad activa)	BOOL	Desconexión de seguridad activa, p. ej., parada de emergencia accionada
Output (Salida)		
Conveyor_motor_automatic_mode (Motor_cinta_modos_automático)	BOOL	Control del motor de cinta en modo automático
Static (Estática)		
Memory_automatic_start_stop (Memoria_modos_automático_arranque/parada)	BOOL	Memoria para la función de arranque y parada en modo automático

Tabla 2: Parámetros de FB "MOTOR_AUTO"

El parámetro Memory_automatic_start_stop (Memoria_modos_automático_arranque/parada) se conecta con memoria con Start (Arranque), pero solo si no se cumplen las condiciones de desactivación.

El parámetro Memory_automatic_start_stop (Memoria_modo_automático_arranque/parada) se desactiva si está presente Stop (Parada), la desconexión de seguridad está activa o el modo automático no está activado (modo manual).

La salida Conveyor_motor_automatic_mode (Motor_cinta_modo_automático) se controla cuando está activado el parámetro Memory_automatic_start_stop

(Memoria_modo_automático_arranque/parada) y se cumplen las condiciones de habilitación.

6.3 Esquema tecnológico

Este es el esquema tecnológico para la tarea asignada.

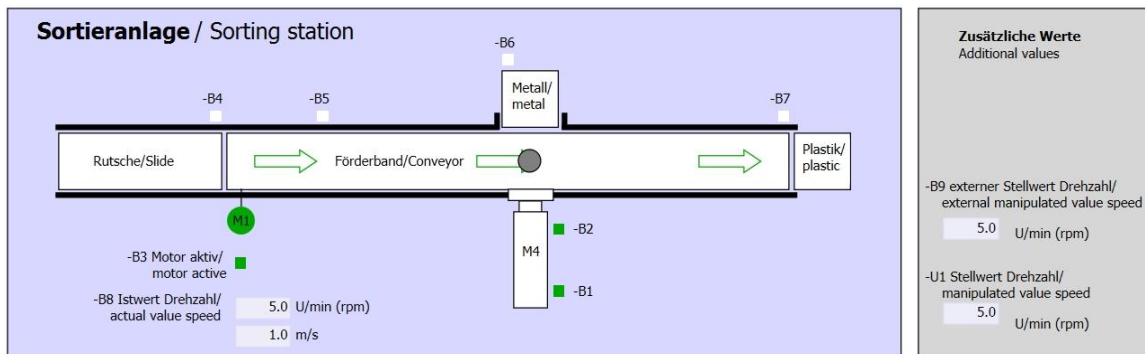


Figura 8: Esquema tecnológico



Figura 9: Pupitre de mando

6.4 Tabla de asignación

Para esta tarea se requieren las siguientes señales como operandos.

DI	Tipo	Identificador	Función	NC/NA
I 0.0	BOOL	-A1	Aviso PARADA DE EMERGENCIA OK	NC
I 0.1	BOOL	-K0	Planta "ON"	NA
I 0.2	BOOL	-S0	Selector modo de operación manual (0)/automático (1)	Manual = 0 Automático = 1
I 0.3	BOOL	-S1	Pulsador de arranque automático	NA
I 0.4	BOOL	-S2	Pulsador de parada automática	NC
I 0.5	BOOL	-B1	Sensor cilindro -M4 introducido	NA

DO	Tipo	Identificador	Función	
Q 0.0	BOOL	-Q1	Motor de cinta -M1 hacia delante, velocidad fija	

Leyenda de la lista de asignación

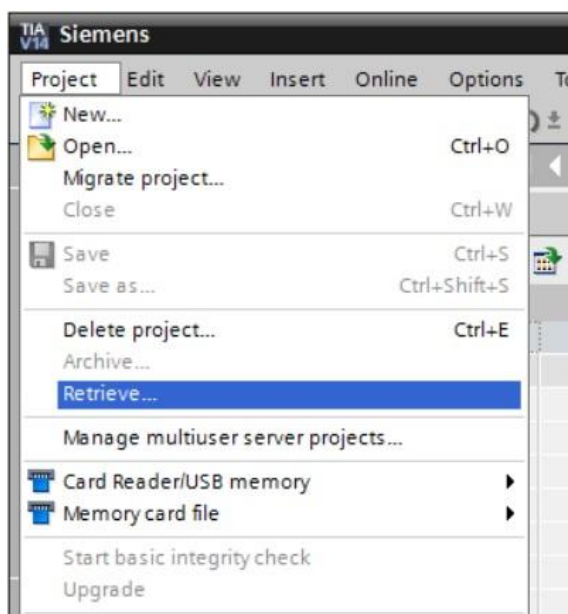
DI	Entrada digital	DQ	Salida digital
AI	Entrada analógica	AQ	Salida analógica
E	Entrada	S	Salida
NC	Normalmente cerrado		
NA	Normalmente abierto		

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

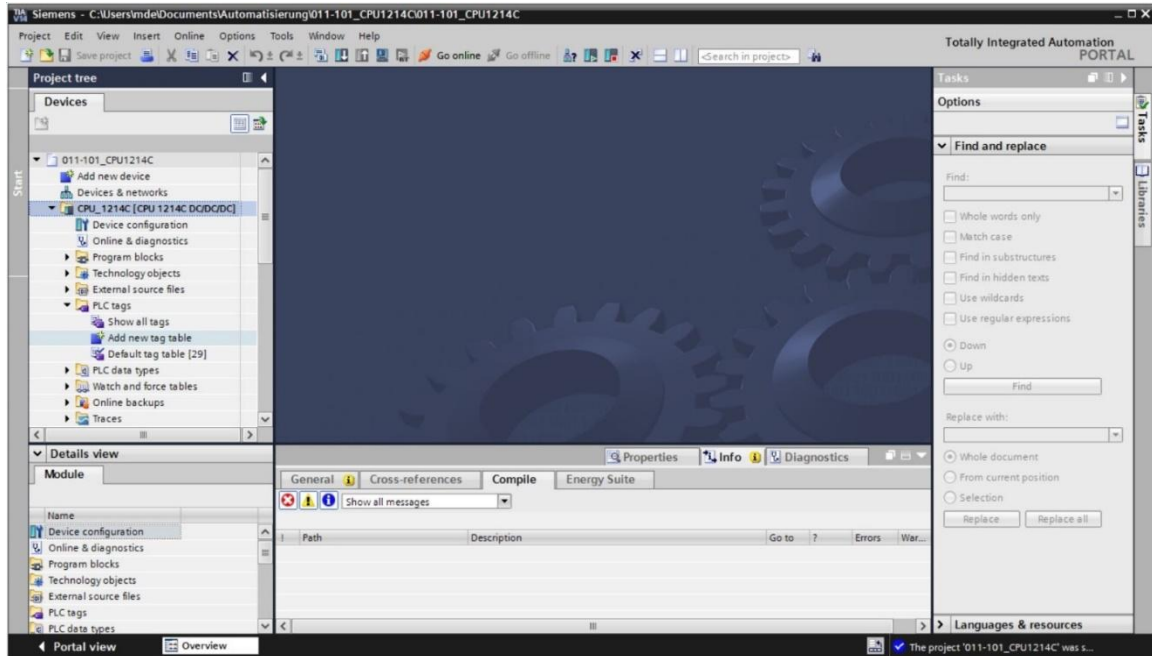
- ® Antes de empezar a programar el bloque de función (FB) "MOTOR_AUTO", se necesita un proyecto con una configuración hardware (p. ej., SCE_ES_011-101_Hardware Configuration_S7-1214C....zap). Para desarchivar un proyecto existente desde la vista del proyecto, escoja el fichero en cuestión en ® Project (Proyecto) ® Retrieve (Desarchivar). A continuación, confirme la selección con Open (Abrir) (® Project (Proyecto) ® Retrieve (Desarchivar) ® seleccionar un fichero .zap ® Open (Abrir)).



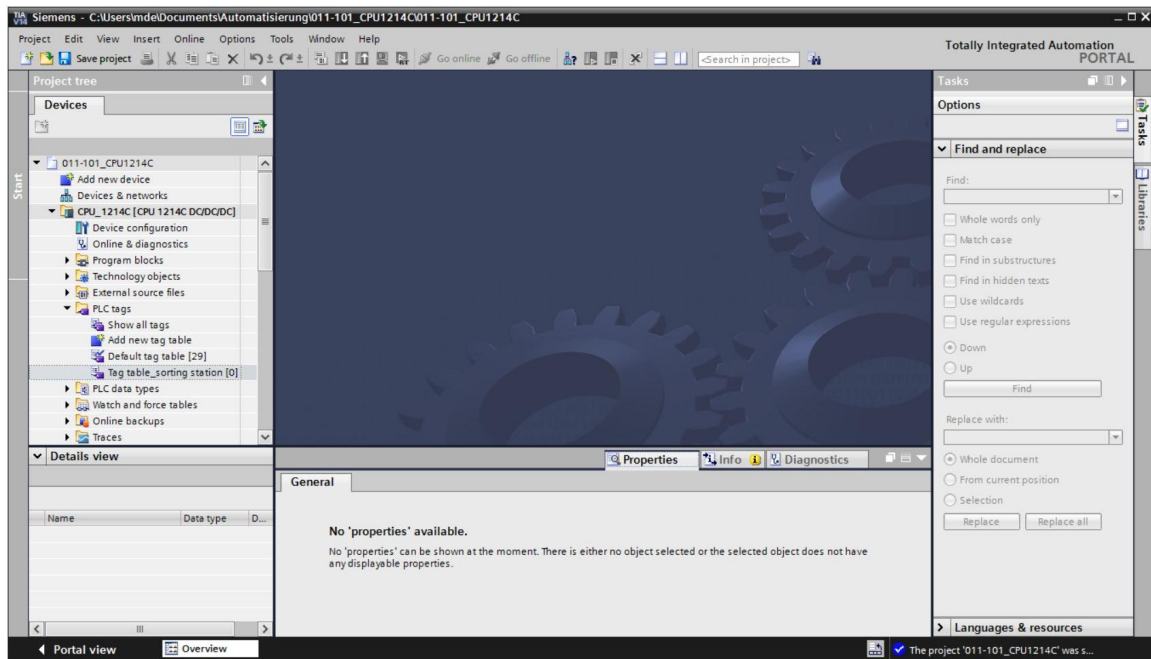
- ® A continuación puede seleccionarse el directorio de destino en el que se guardará el proyecto desarchivado. Confirme la selección con "OK (Aceptar)" (® directorio de destino ® OK (Aceptar)).

7.2 Creación de una nueva tabla de variables

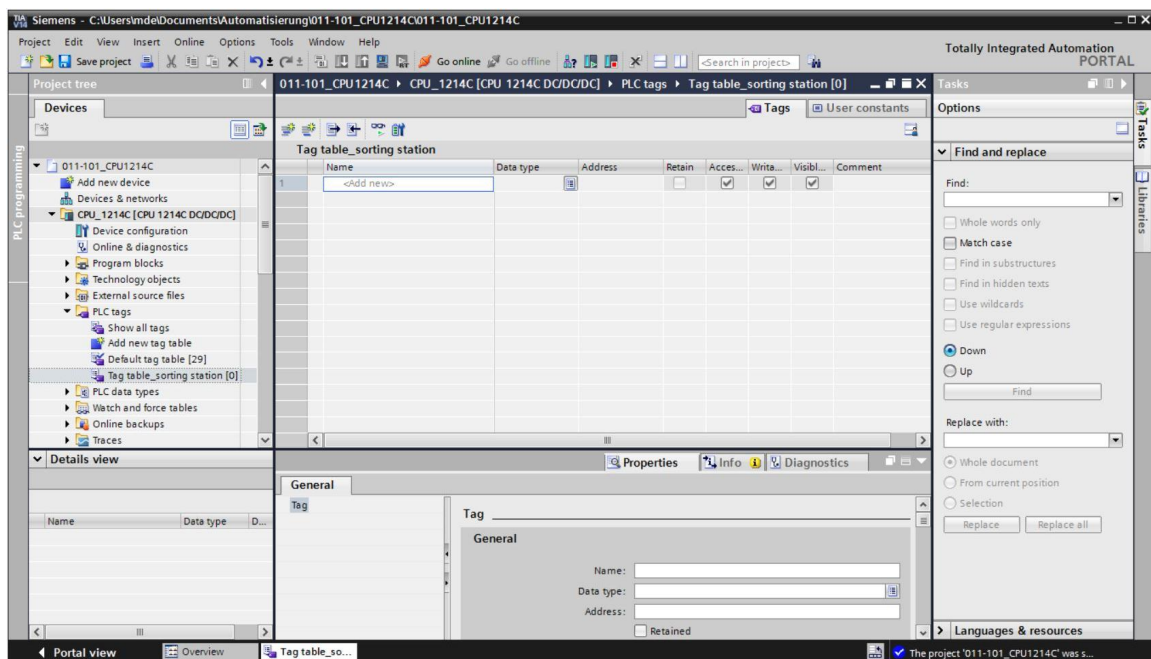
- ® Navegue por el árbol del proyecto hasta las ® variables de PLC de su controlador y cree una nueva tabla de variables haciendo doble clic en ® Add new tag table (Agregar nueva tabla de variables).



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

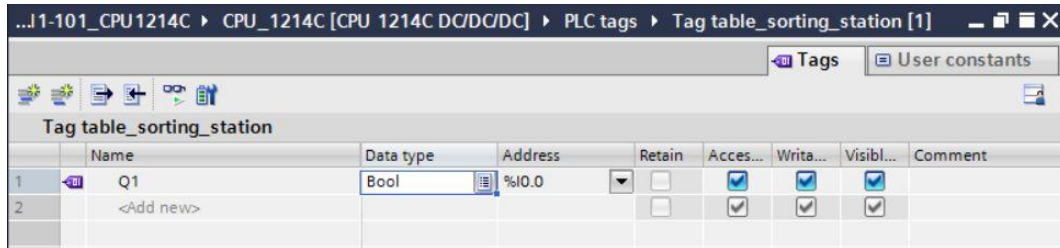


- ® A continuación ábrala haciendo doble clic (® Tag table_sorting station (Tabla_variables_planta_clasificación)).

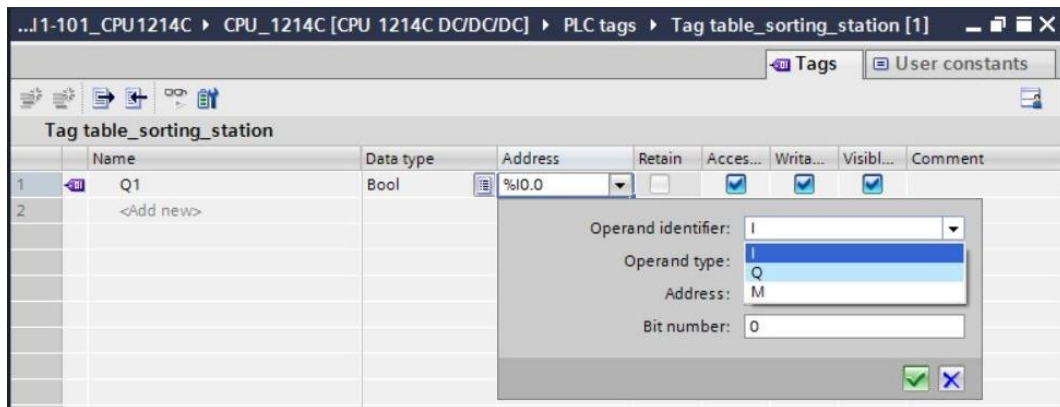


7.3 Creación de nuevas variables dentro de una tabla de variables

- ® 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. 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)".



- ® Agregue en la línea 2 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)".

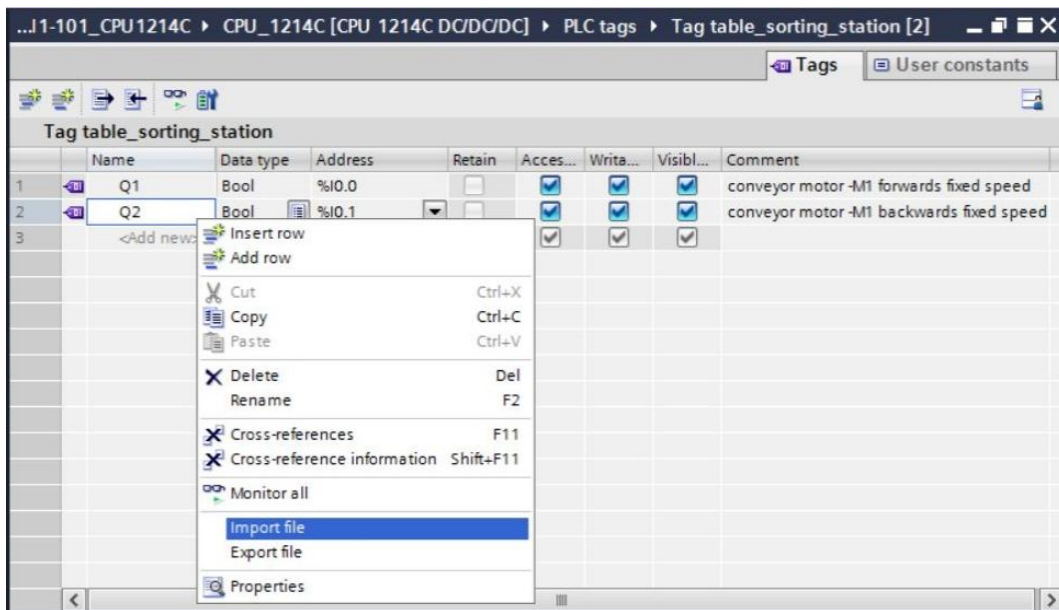
(® <Add new> (Agregar) ® Q2 ® Intro® comentario ® conveyor motor -M1 backwards fixed speed (Motor de cinta M1 hacia atrás, velocidad fija))

	Name	Data type	Address	Retain	Acces...	Writa...	Visible in ...	Comment
1	-Q1	Bool	%Q0.0	☐	☒	☒	☒	conveyor motor -M1 forwards fixed speed
2	-Q2	Bool	%Q0.1	☐	☒	☒	☒	conveyor motor -M1 backwards fixed speed

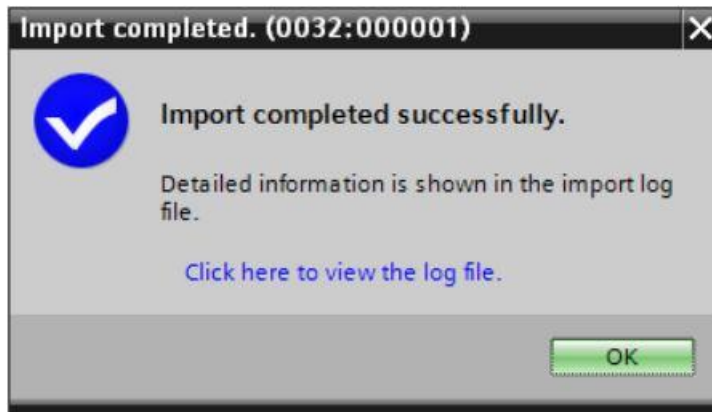
7.4 Importación de la "Tag table_sorting station (Tabla_variables_planta_clasificación)"

- ® 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)" creada. Seleccione en el menú contextual "Import file (Importar archivo)".

(® 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_Process Description 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.

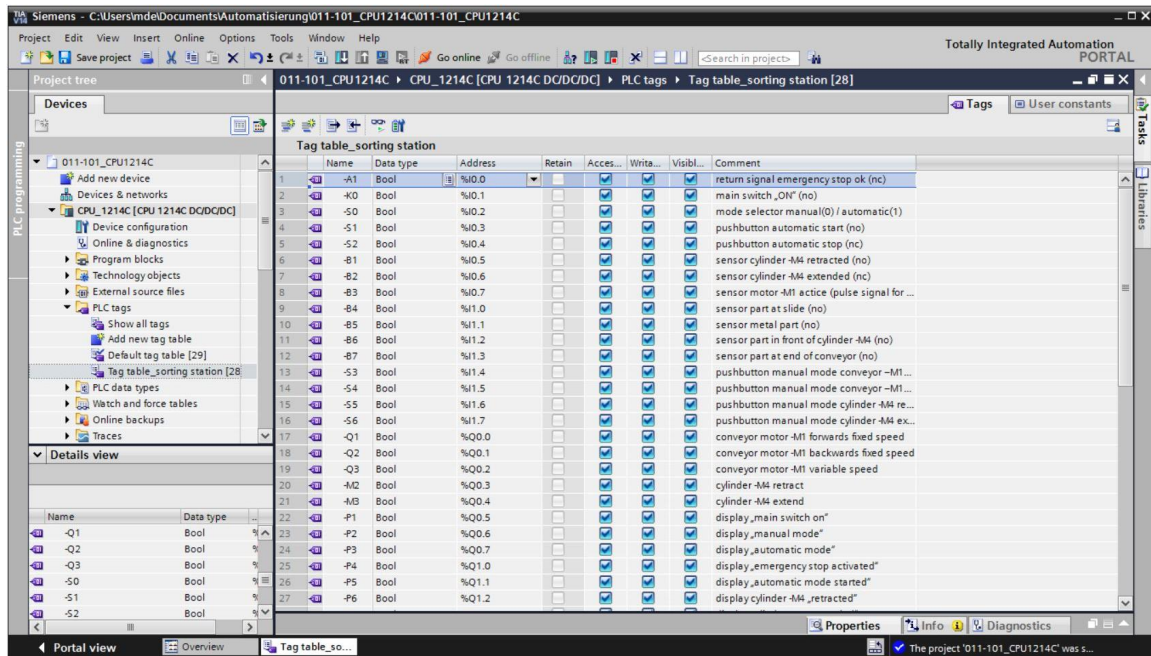
- ® Borre las variables duplicadas seleccionando las líneas correspondientes mediante la tecla Supr de su teclado o la opción "Delete (Eliminar)" del menú contextual.

(® Clic con el botón derecho en las variables seleccionadas ® Delete (Eliminar))

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	Q1	Bool	%Q0.0		☒	☒	☒	conveyor motor -M1 forwards fixed speed
2	Q2	Bool	%Q0.1		☒	☒	☒	conveyor motor -M1 forwards fixed speed
3	-A1	Bool	%I0.0		☒	☒	☒	return signal emergency stop ok (nc)
4	-K0	Bool	%I0.1		☒	☒	☒	main switch „ON“ (no)
5	-S0	Bool	%I0.2		☒	☒	☒	mode selector manual(0) / automatic(1)
6	-S1	Bool	%I0.3		☒	☒	☒	pushbutton automatic start (no)
7	-S2	Bool	%I0.4		☒	☒	☒	pushbutton automatic stop (nc)
8	-B1	Bool	%I0.5		☒	☒	☒	sensor cylinder -M4 retracted (no)
9	-B2	Bool	%I0.6		☒	☒	☒	sensor cylinder -M4 extended (nc)
10	-B3	Bool	%I0.7		☒	☒	☒	sensor motor -M1 actice (pulse signal for ...
11	-B4	Bool	%I1.0		☒	☒	☒	sensor part at slide (no)
12	-B5	Bool	%I1.1		☒	☒	☒	sensor metal part (no)
13	-B6	Bool	%I1.2		☒	☒	☒	sensor part in front of cylinder -M4 (no)
14	-B7	Bool	%I1.3		☒	☒	☒	sensor part at end of conveyor (no)
15	-S3	Bool	%I1.4		☒	☒	☒	pushbutton manual mode conveyor -M1...
16	-S4	Bool	%I1.5		☒	☒	☒	pushbutton manual mode conveyor -M1...
17	-S5	Bool	%I1.6		☒	☒	☒	pushbutton manual mode cylinder -M4 re...
18	-S6	Bool	%I1.7		☒	☒	☒	pushbutton manual mode cylinder -M4 ex...
19	-Q1	Bool	%Q0.0		☒	☒	☒	conveyor motor -M1 forwards fixed speed
20	-Q2	Bool	%Q0.1		☒	☒	☒	conveyor motor -M1 backwards fixed speed
21	-Q3	Bool	%Q0.2		☒	☒	☒	conveyor motor -M1 variable speed
22	-M2	Bool	%Q0.3		☒	☒	☒	cylinder -M4 retract
23	-M3	Bool	%Q0.4		☒	☒	☒	cylinder -M4 extend
24	-P1	Bool	%Q0.5		☒	☒	☒	display „main switch on“
25	-P2	Bool	%Q0.6		☒	☒	☒	display „manual mode“
26	-P3	Bool	%Q0.7		☒	☒	☒	display „automatic mode“

® Ahora tendrá en su pantalla una tabla de símbolos completa de las entradas y salidas digitales. Guarde el proyecto con el nombre 031-200_Programación_FB.

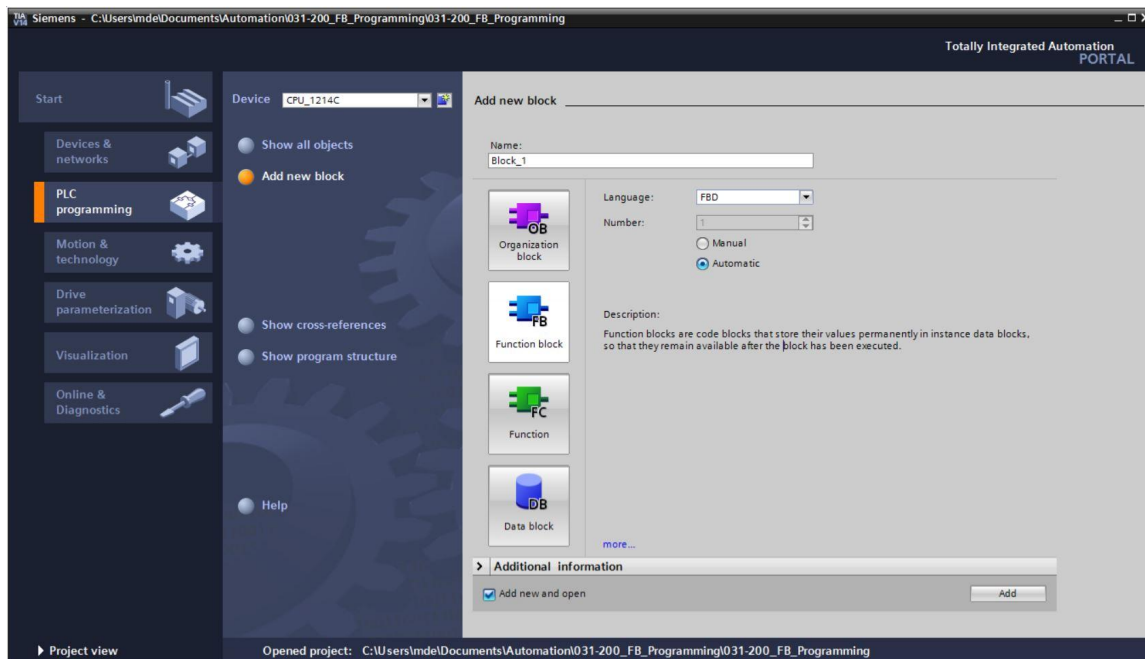
(® Project (Proyecto) ® Save as ... (Guardar como ...) ® 031-200_Programación_FB ® Save (Guardar))



7.5 Creación del bloque de función FB1 "MOTOR_AUTO" para el motor de cinta en modo automático

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

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



- ® Asigne al nuevo bloque el nombre: "MOTOR_AUTO", seleccione el lenguaje FBD (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 en "Add (Agregar)".

(® Name (Nombre): MOTOR_AUTO® Language (Lenguaje): FBD (FUP) ® Number (Número): Automatic (Automático) ® ☒ Add new and open (Agregar y abrir) ® Add (Agregar))

Add new block

Name:

Language:

Number:

☐ Manual
☒ Automatic

Description:
Function blocks are code blocks that store their values permanently in instance data blocks, so that they remain available after the block has been executed.

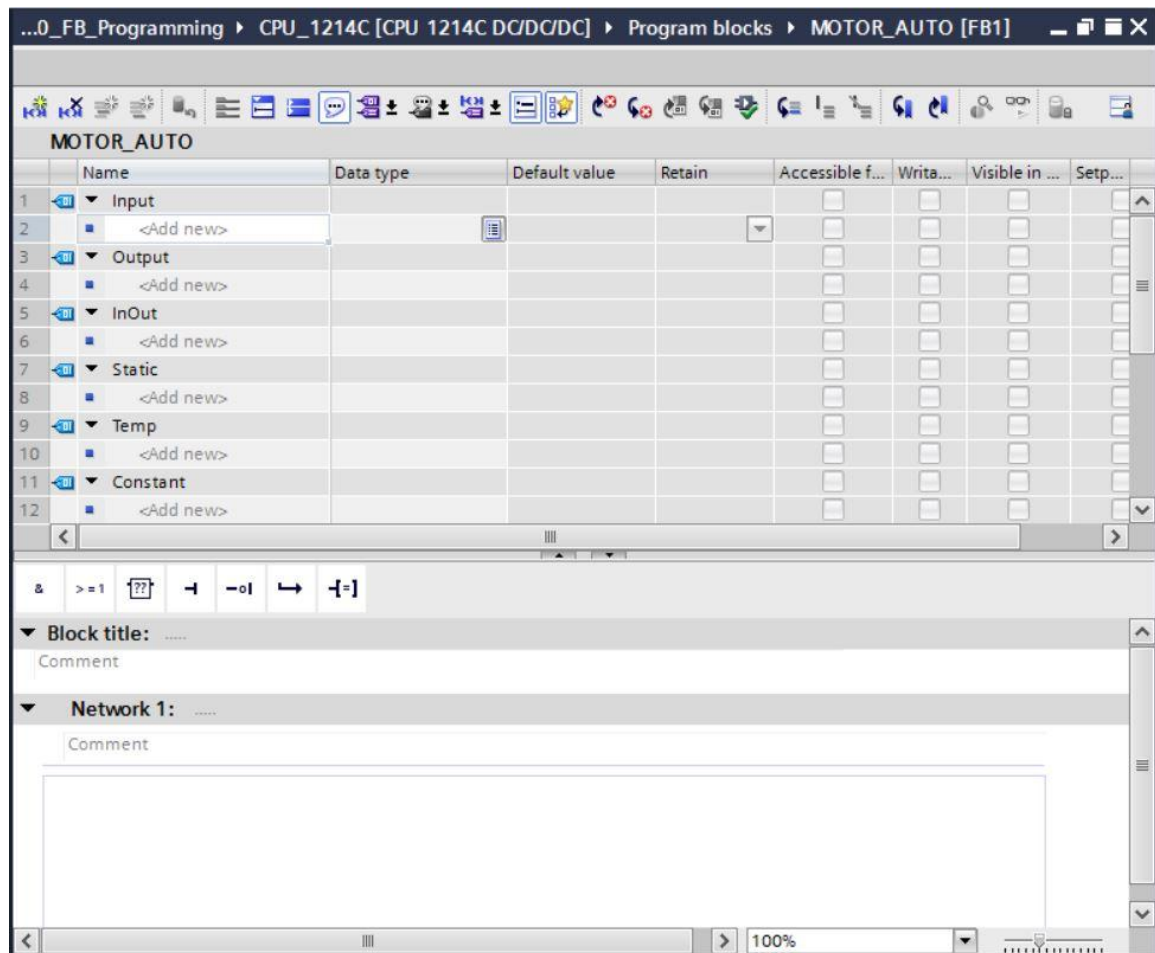
[More...](#)

> **Additional information**

☒ Add new and open

7.6 Definición de la interfaz del FB1 "MOTOR_AUTO"

- ® 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 del bloque de 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_automatic_mode (Motor_cinta_modo_automático), del tipo "Bool". Asigne al parámetro el comentario "Control of the conveyor in automatic mode (Control del motor de cinta en modo automático)".

(® Output (Salida)® Conveyor_motor_automatic_mode (Motor_cinta_modo_automático)® Bool ® Control of the conveyor in automatic mode (Control del motor de cinta en modo automático))

Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1 Input								
2 <Add new>								
3 Output								
4 Conveyor_motor_automatic_mode	Bool	false	Non-retain	☒	☒	☒		Control of the conveyor motor in automatic mode
5 <Add new>								
6 InOut								
7 <Add new>								
8 Static								
9 <Add new>								
10 Temp								
11 <Add new>								
12 Constant								

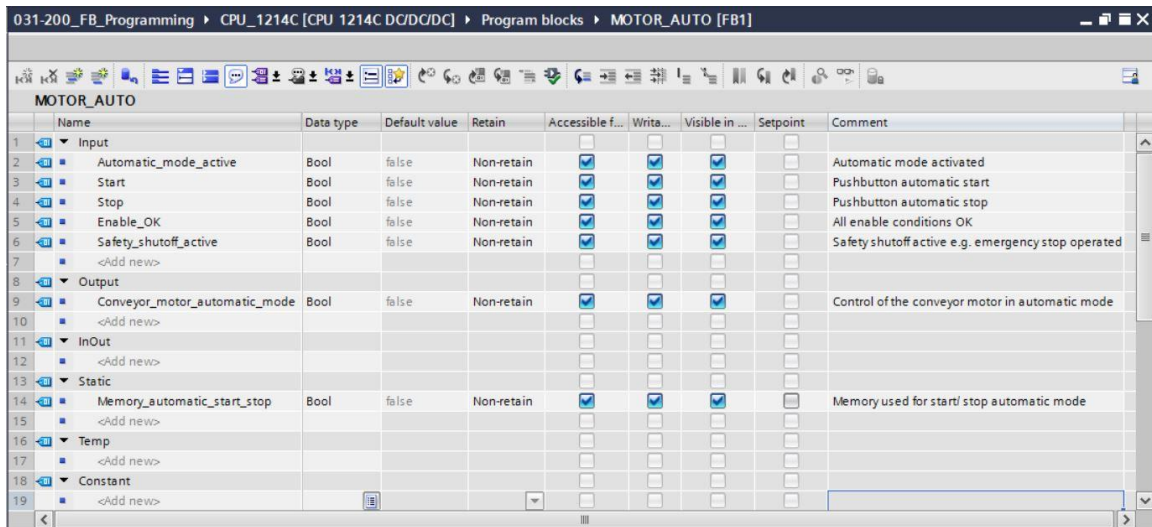
- ® En primer lugar, agregue como interfaz de entrada en Input (Entrada) el parámetro #Automatic_mode_active (Modo_automático_activo) y confirme con la tecla Intro o saliendo del campo de entrada. Se asignará automáticamente el tipo de datos "Bool". El tipo de datos se mantendrá. A continuación introduzca el comentario "Automatic mode activated (Modo de operación automático activado)".

(® Input (Entrada)® Automatic_mode_active (Modo_automático_activo) ® Bool ® Automatic mode activated (Modo de operación automático activado))

- ® A continuación introduzca en Input (Entrada), como parámetros de entrada binarios, #Start (Arranque), #Stop (Parada), #Enable_OK (Habilitación_OK) y #Safety_shutoff_active (Desconexión_seguridad_activa) y compruebe los tipos de datos de dichos parámetros. Añada comentarios descriptivos.

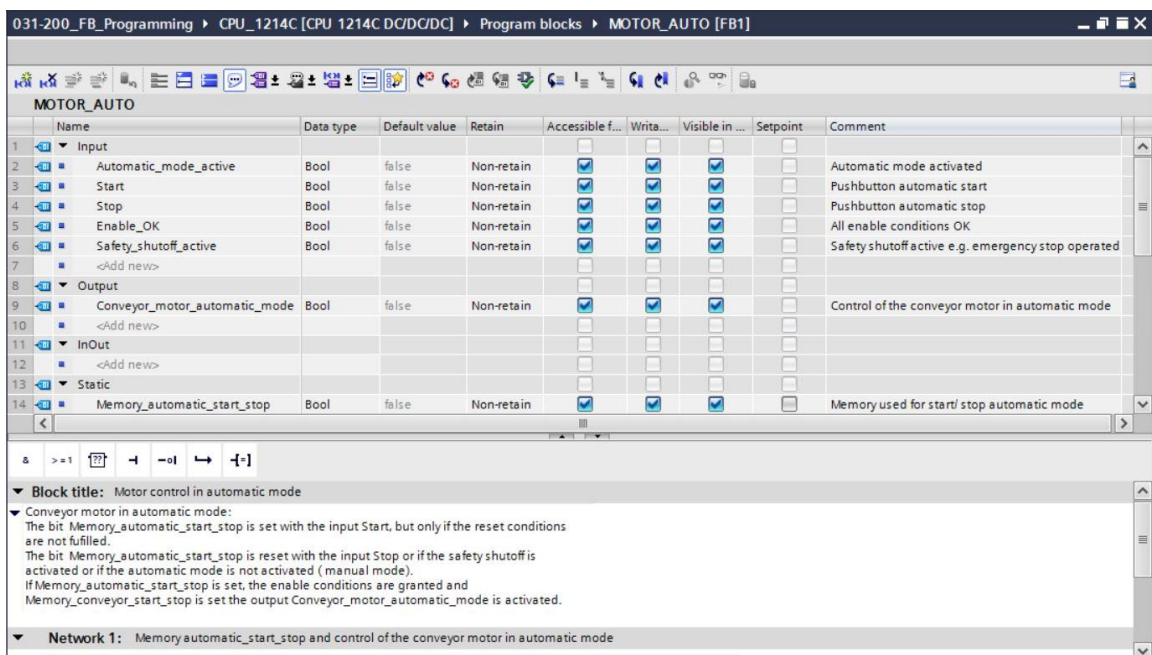
Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1 Input								
2 Automatic_mode_active	Bool	false	Non-retain	☒	☒	☒		Automatic mode activated
3 Start	Bool	false	Non-retain	☒	☒	☒		Pushbutton automatic start
4 Stop	Bool	false	Non-retain	☒	☒	☒		Pushbutton automatic stop
5 Enable_OK	Bool	false	Non-retain	☒	☒	☒		All enable conditions OK
6 Safety_shutoff_active	Bool	false	Non-retain	☒	☒	☒		Safety shutoff active e.g. emergency stop operated
7 <Add new>								
8 Output								
9 Conveyor_motor_automatic_mode	Bool	false	Non-retain	☒	☒	☒		Control of the conveyor motor in automatic mode
10 <Add new>								
11 InOut								
12 <Add new>								
13 Static								
14 <Add new>								
15 Temp								

- ® El arranque y la parada del motor de la cinta se realizan mediante pulsadores. Por ello se necesita una variable "Static" como memoria. Agregue en Static la variable #Memory_automatic_start_stop (Memoria_modo_automático_arranque/parada) 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 "Memory used for start/stop automatic mode (Memoria para la función de arranque y parada en modo automático)" (® Static (Estática)® Memory_automatic_start_stop (Memoria_modo_automático_arranque/parada) ® Bool ® Memory used for start/stop automatic mode (Memoria para la función de arranque y parada en modo automático)).



Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1 Input								
2 Automatic_mode_active	Bool	false	Non-retain	☒	☒	☒	☐	Automatic mode activated
3 Start	Bool	false	Non-retain	☒	☒	☒	☐	Pushbutton automatic start
4 Stop	Bool	false	Non-retain	☒	☒	☒	☐	Pushbutton automatic stop
5 Enable_OK	Bool	false	Non-retain	☒	☒	☒	☐	All enable conditions OK
6 Safety_shutoff_active	Bool	false	Non-retain	☒	☒	☒	☐	Safety shutoff active e.g. emergency stop operated
7 <Add new>								
8 Output								
9 Conveyor_motor_automatic_mode	Bool	false	Non-retain	☒	☒	☒	☐	Control of the conveyor motor in automatic mode
10 <Add new>								
11 InOut								
12 <Add new>								
13 Static								
14 Memory_automatic_start_stop	Bool	false	Non-retain	☒	☒	☒	☐	Memory used for start/ stop automatic mode
15 <Add new>								
16 Temp								
17 <Add new>								
18 Constant								
19 <Add new>								

- ® Para documentar el programa, defina el título del bloque, un comentario de bloque y un título de segmento explicativo para el segmento 1. (® Block title (Título del bloque): Motor control in automatic mode (Control de motor en modo automático) ® Network 1 (Segmento 1): Memory_automatic_start_stop (Memoria_modo_automático_arranque/parada) y Control of the conveyor motor in automatic mode (Control del motor de cinta en modo automático)).



Name	Data type	Default value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1 Input								
2 Automatic_mode_active	Bool	false	Non-retain	☒	☒	☒	☐	Automatic mode activated
3 Start	Bool	false	Non-retain	☒	☒	☒	☐	Pushbutton automatic start
4 Stop	Bool	false	Non-retain	☒	☒	☒	☐	Pushbutton automatic stop
5 Enable_OK	Bool	false	Non-retain	☒	☒	☒	☐	All enable conditions OK
6 Safety_shutoff_active	Bool	false	Non-retain	☒	☒	☒	☐	Safety shutoff active e.g. emergency stop operated
7 <Add new>								
8 Output								
9 Conveyor_motor_automatic_mode	Bool	false	Non-retain	☒	☒	☒	☐	Control of the conveyor motor in automatic mode
10 <Add new>								
11 InOut								
12 <Add new>								
13 Static								
14 Memory_automatic_start_stop	Bool	false	Non-retain	☒	☒	☒	☐	Memory used for start/ stop automatic mode
15 <Add new>								
16 Temp								
17 <Add new>								
18 Constant								
19 <Add new>								

Block title: Motor control in automatic mode

Conveyor motor in automatic mode:
The bit Memory_automatic_start_stop is set with the input Start, but only if the reset conditions are not fulfilled.
The bit Memory_automatic_start_stop is reset with the input Stop or if the safety shutoff is activated or if the automatic mode is not activated (manual mode).
If Memory_automatic_start_stop is set, the enable conditions are granted and Memory_conveyor_start_stop is set the output Conveyor_motor_automatic_mode is activated.

Network 1: Memory automatic_start_stop and control of the conveyor motor in automatic mode

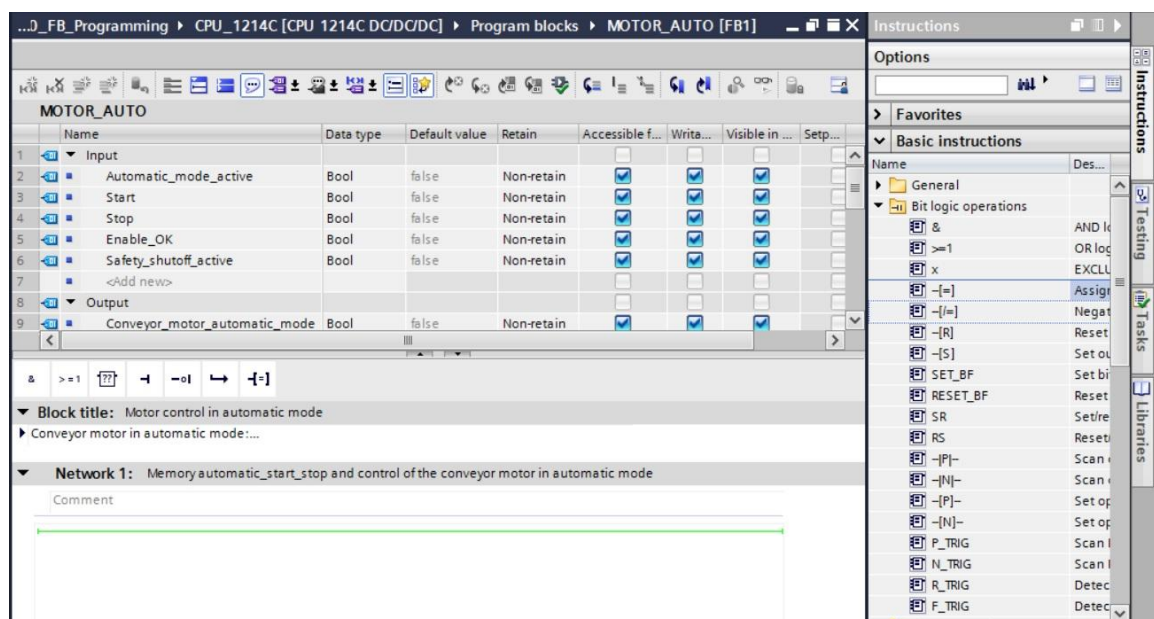
7.7 Programación del FB1: MOTOR_AUTO

- ® 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.



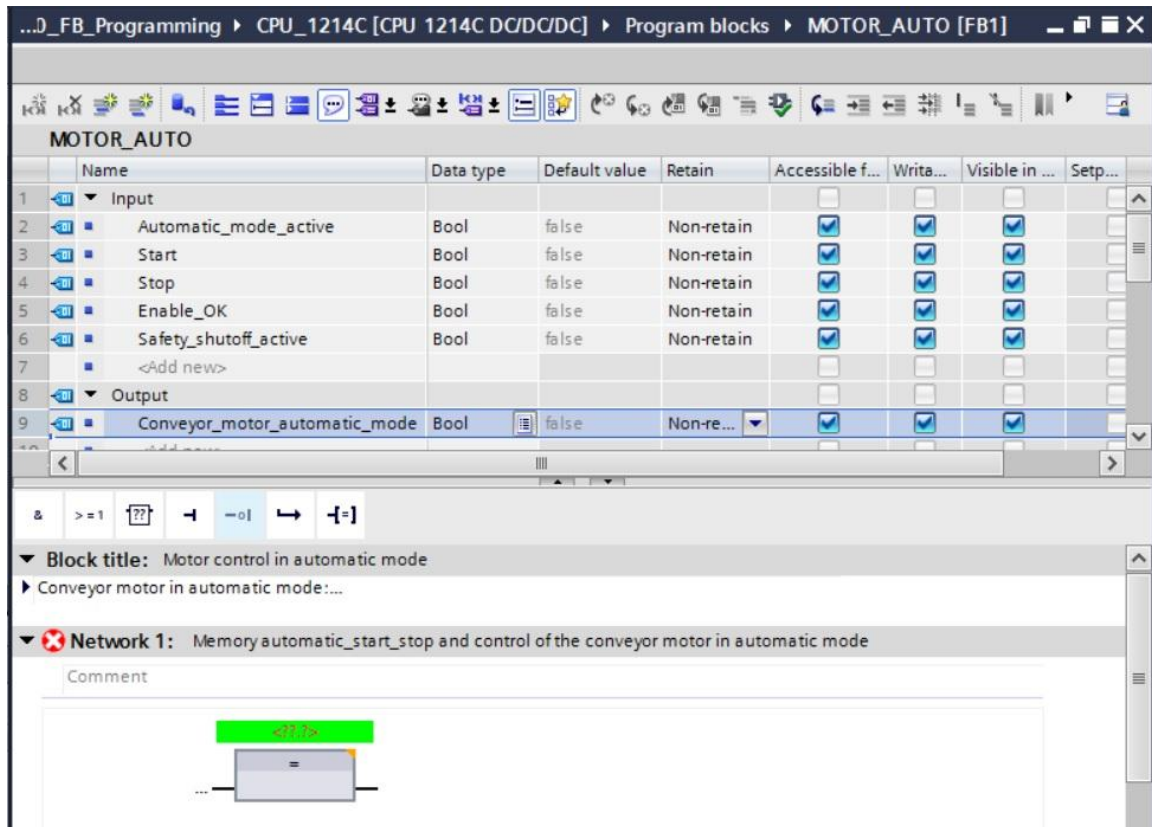
- ® En el lado derecho de la ventana de programación hay una lista de instrucciones que pueden utilizarse en el programa. Busque en ® Basic instructions (Instrucciones básicas) ® Bit logic operations (Operaciones lógicas con bits) la función (asignación) y cópiela mediante "arrastrar y soltar" a su Network 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) ®)

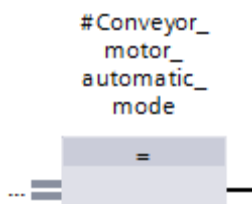


Ahora, desplace el parámetro de salida #Conveyor_motor_automatic_mode (Motor_cinta_modo_automático), mediante "arrastrar y soltar", a $\langle ?? \rangle$, por encima del bloque que acaba de insertar. La mejor manera de seleccionar un parámetro en la descripción de interfaces es agarrarlo por el icono azul .

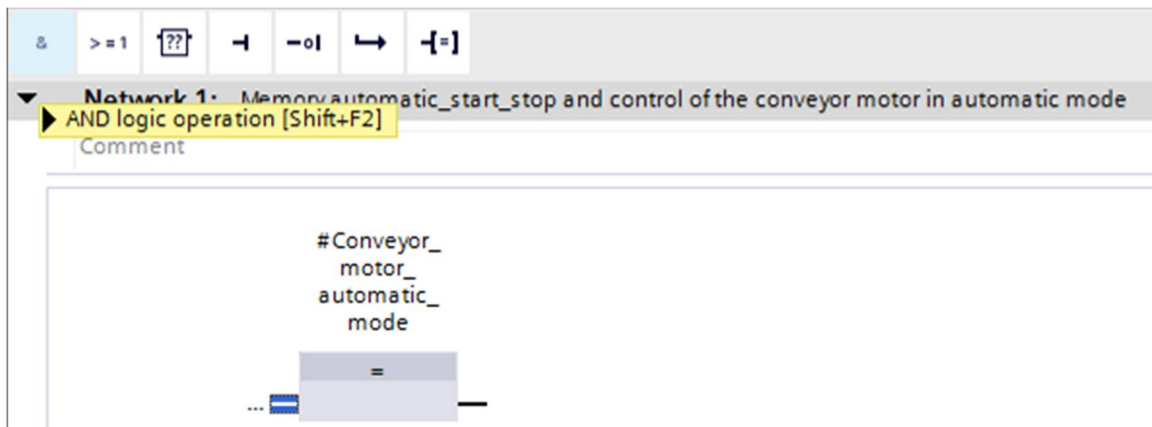
(®  Conveyor_motor_automatic_mode (Motor_cinta_modo_automático))



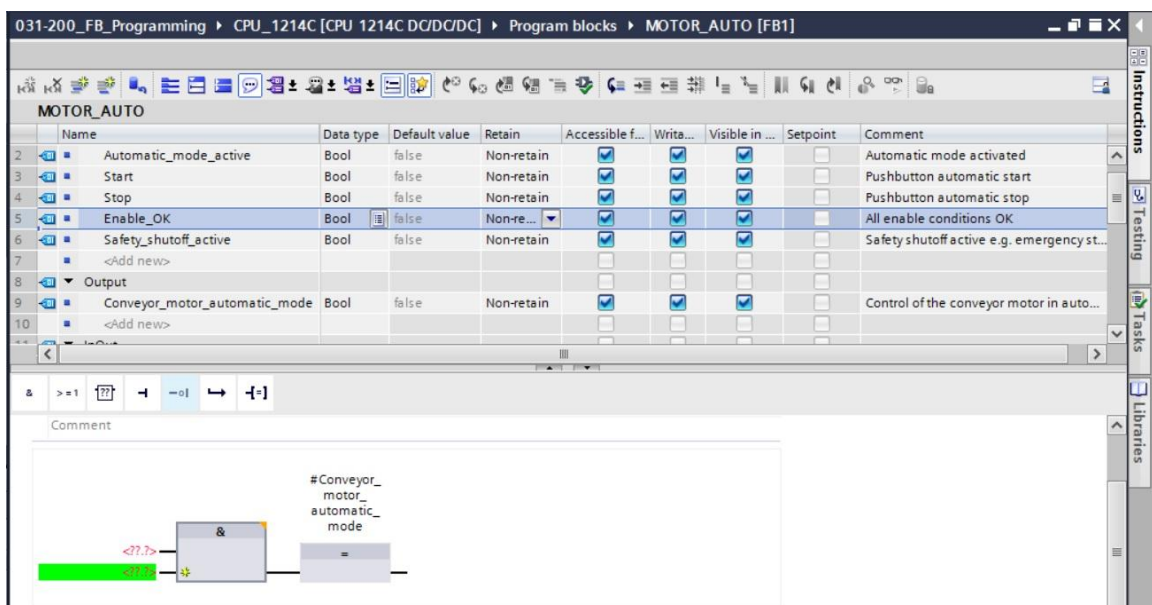
® Con esto se determina que el parámetro #Conveyor_motor_automatic_mode (Motor_cinta_modo_automático) se escriba mediante este bloque. Para que esto sea efectivamente así, faltan todavía las condiciones de entrada. En la entrada del bloque de asignación deben combinarse un biestable SR y el parámetro #Enable_OK (Habilitación_OK) con el operador lógico Y. Para ello, haga clic en primer lugar en la entrada del bloque. La raya de entrada aparecerá sobre fondo azul.



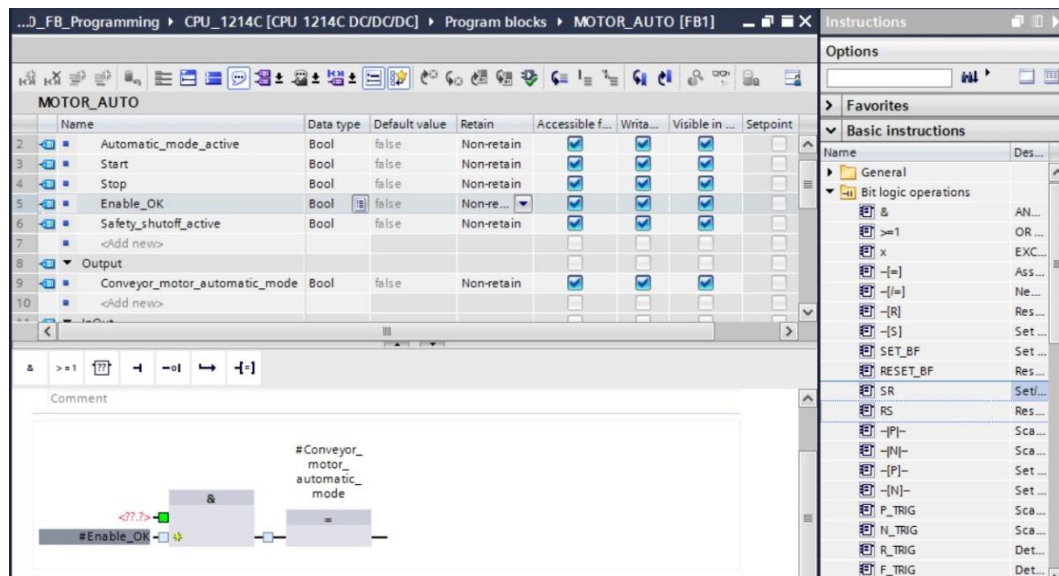
- ® Haga clic en el icono  de la barra de herramientas lógicas para insertar una combinación mediante operador Y delante del bloque de asignación.



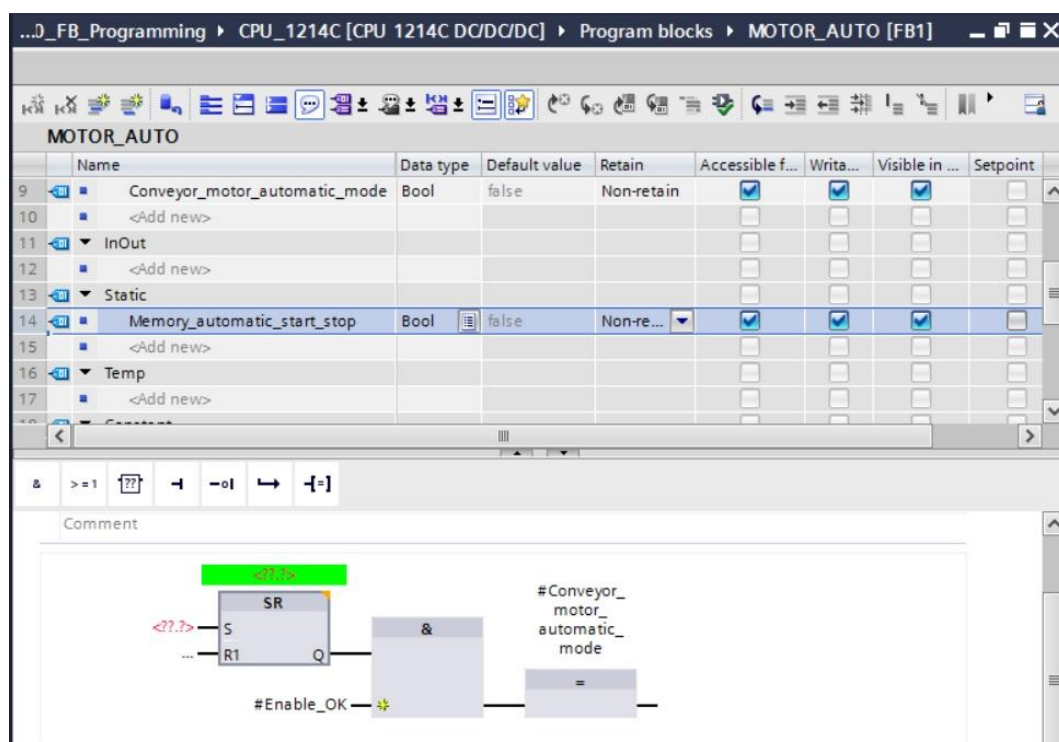
- ® Tras ello, desplace el parámetro de entrada #Enable_OK (Habilitación_OK) mediante "arrastrar y soltar" a la segunda entrada de la combinación Y  (®  Habilitación_OK).



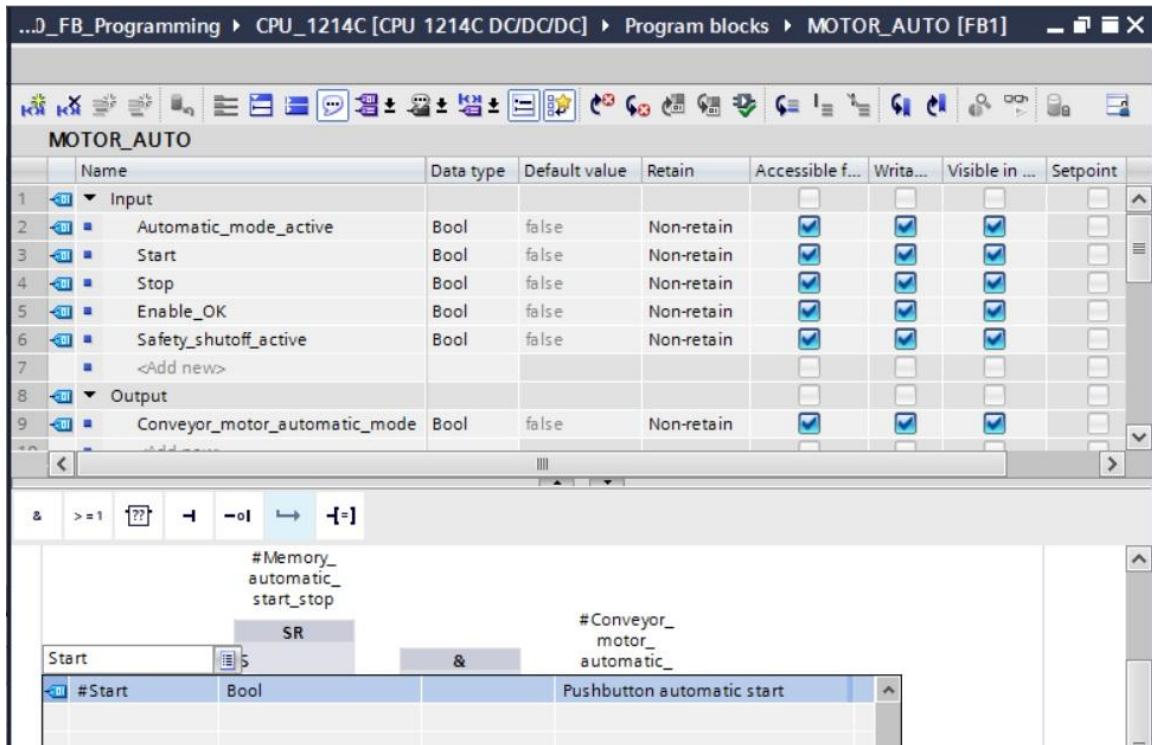
- ® En la lista de instrucciones, ® Basic instructions (Instrucciones básicas) ® Bit logic operations (Operaciones lógicas con bits), copie la función Set/Reset Flipflop (Setear/resetear biestable) a la primera entrada de la combinación Y mediante "arrastrar y colocar". (® Instructions (Instrucciones) ® Basic instructions (Instrucciones básicas) ® Bit logic operations (Operaciones lógicas con bits) ® SR ®)



- ® El biestable SR necesita una variable de memoria. Desplace el parámetro Static #Memory_automatic_start_stop (Memoria_modo_automático_arranque/parada) a <??.?> por encima del biestable SR mediante "arrastrar y soltar". (® Memory_automatic_start_stop (Memoria_modo_automático_arranque/parada))

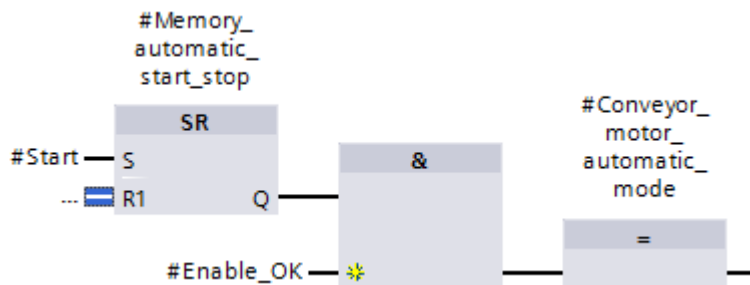


- ® El parámetro #Memory_automatic_start_stop (Memoria_modo_automático_arranque/parada) se seteará con la variable de entrada #Start (Arranque). Para ello, haga doble clic en la entrada S del biestable SR <??.?> y, en el campo que se abrirá, introduzca "Start (Arranque)" para ver una lista de las variables disponibles que empiezan por esa palabra. Haga clic en la variable #Start (Arranque) y pulse ® Intro para aceptar. (® SR Flipflop (Biestable SR) ® <??.?> ® Start (Arranque) ® #Start (Arranque) ® Intro)

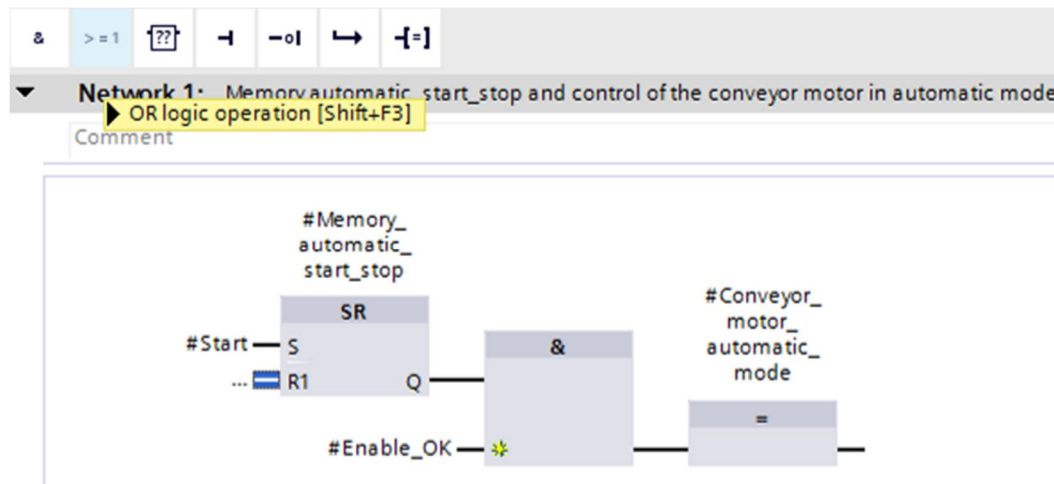


Nota: en esta modalidad de asignación de variables, existe riesgo de confusión con las variables globales de la tabla de variables. Por ello es preferible usar la modalidad con "arrastrar y soltar" desde la descripción de interfaces.

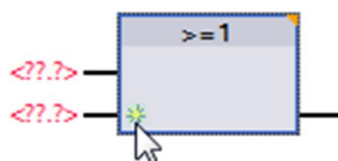
- ® Existirán varias condiciones para la parada de la cinta. Por ello, en la entrada R1 del biestable SR se necesita un bloque O. Haga clic en primer lugar en la entrada R1 del biestable SR. La raya de entrada aparecerá sobre fondo azul.



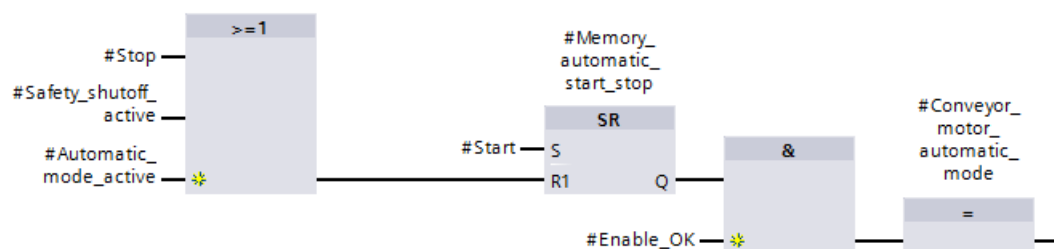
- ® Haga clic en el icono  de la barra de herramientas lógicas para insertar una combinación mediante operador O.



- ® Inicialmente, el bloque O tiene solo 2 entradas. Para poder combinar una variable de entrada adicional, haga clic en el asterisco amarillo  del elemento O.

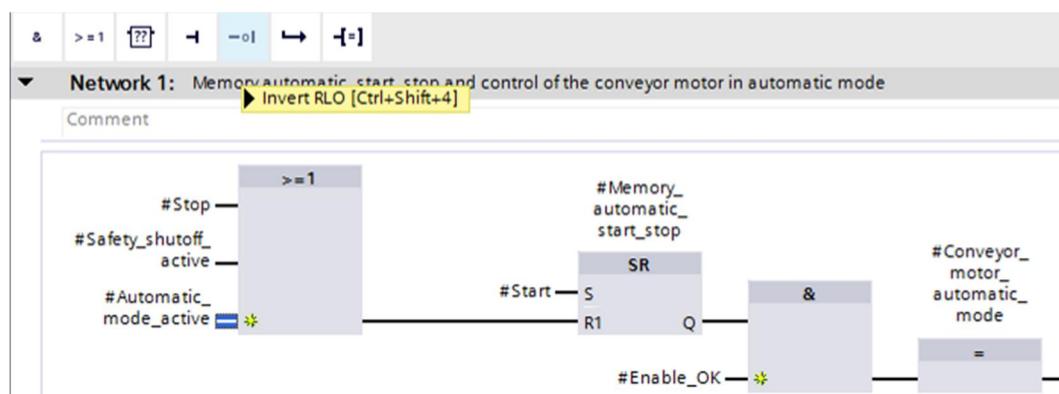


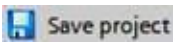
- ® Inserte en las tres entradas del elemento O las variables de entrada #Stop (Parada), #Safety_shutoff_active (Desconexión_seguridad_activa) y #Automatic_mode_active (Modo_automático_activo).

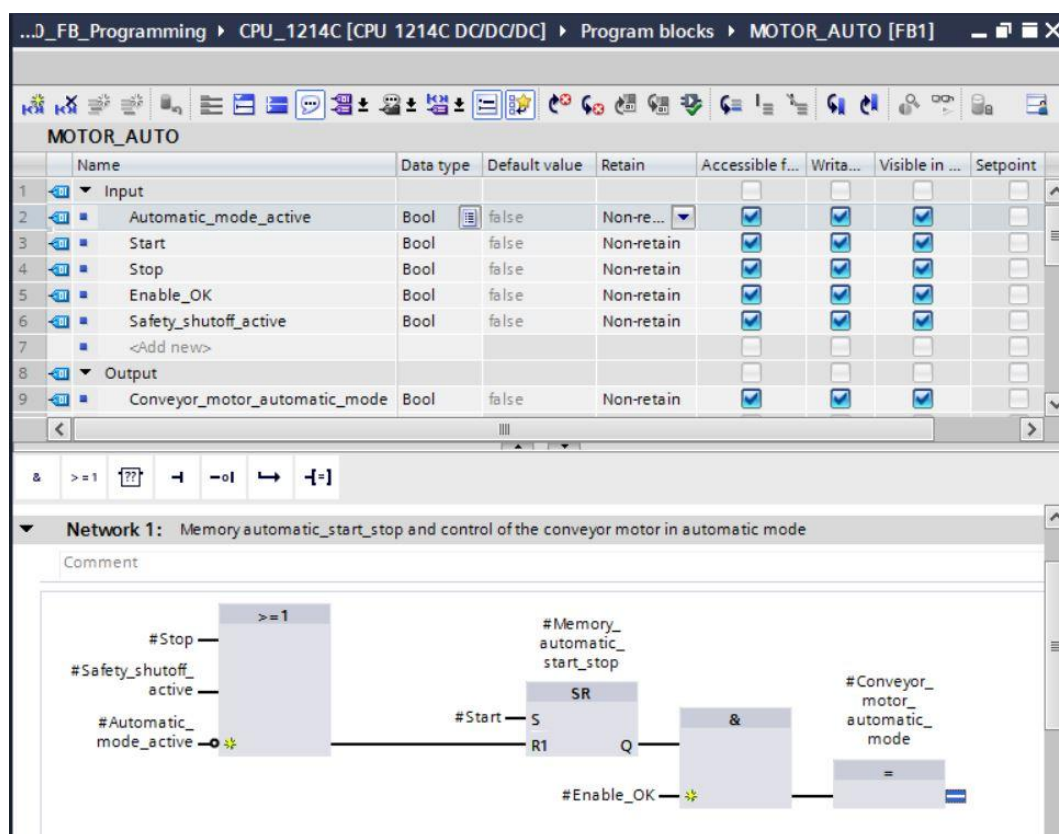


- ® Niegue la entrada conectada con el parámetro #Automatic_mode_active

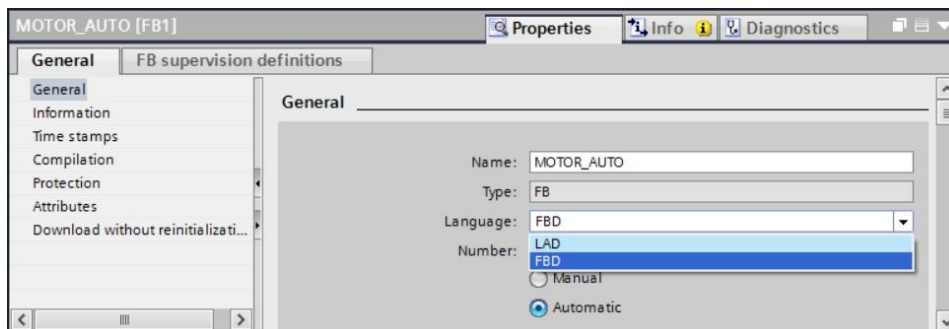
(Modo_automático_activo); para ello, selecciónelo y haga clic en .



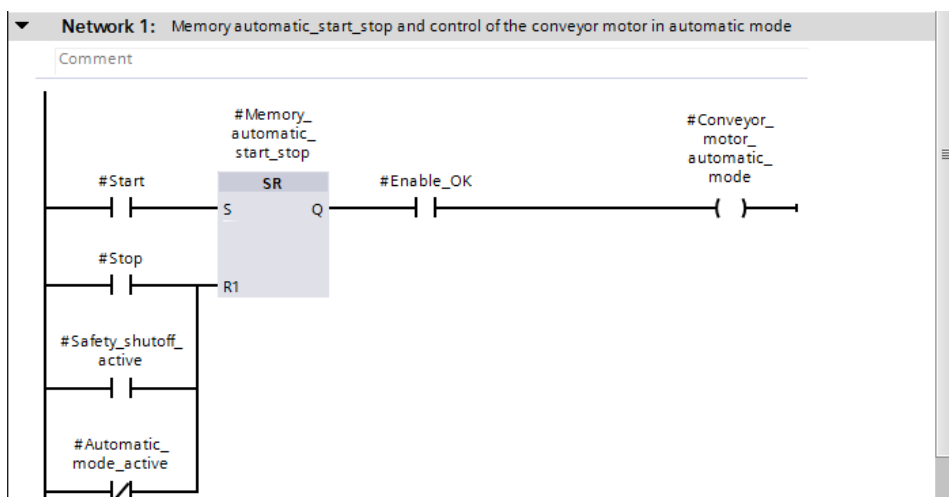
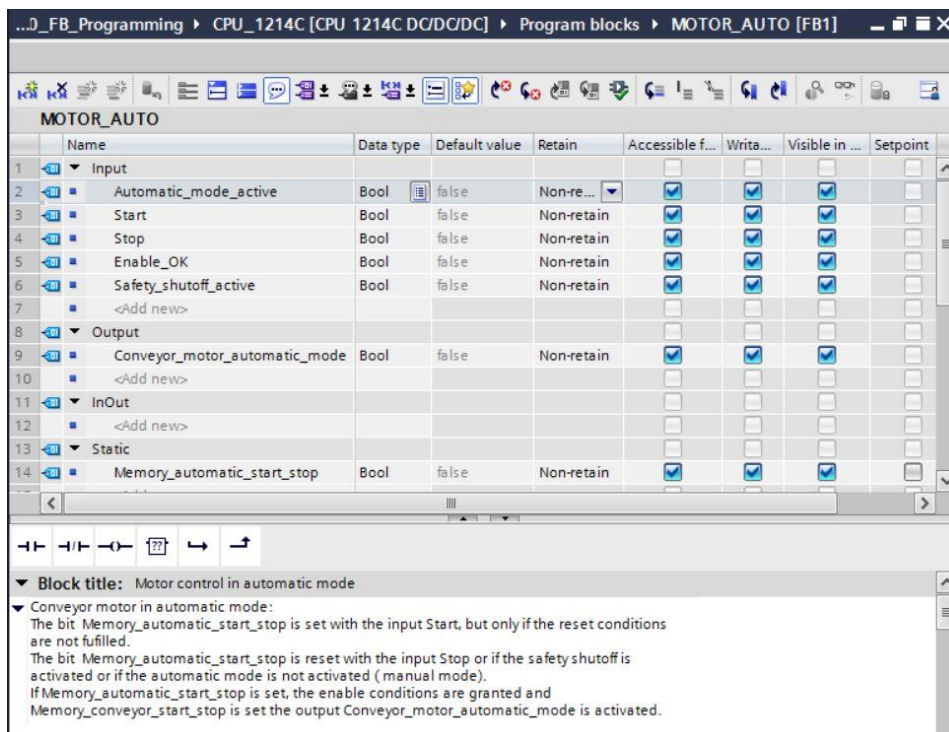
- ® No olvide hacer clic regularmente en . A continuación se muestra el bloque de función ya creado "MOTOR_AUTO" [FB1] en FUP.



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

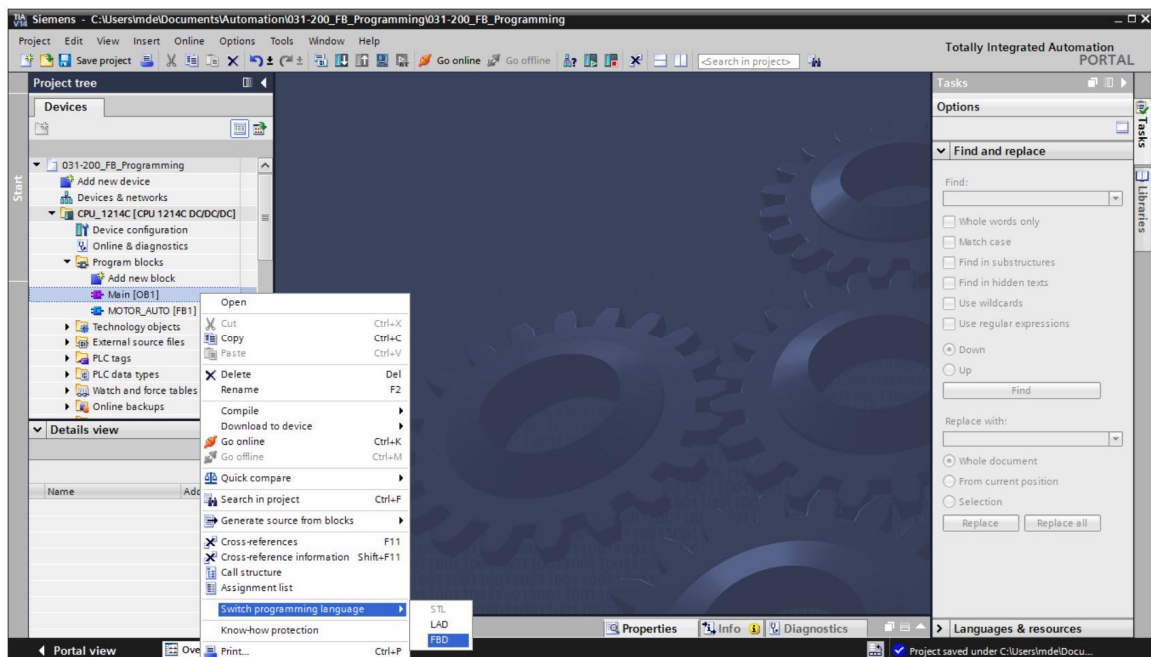


- ® En KOP, el programa es como se indica a continuación.

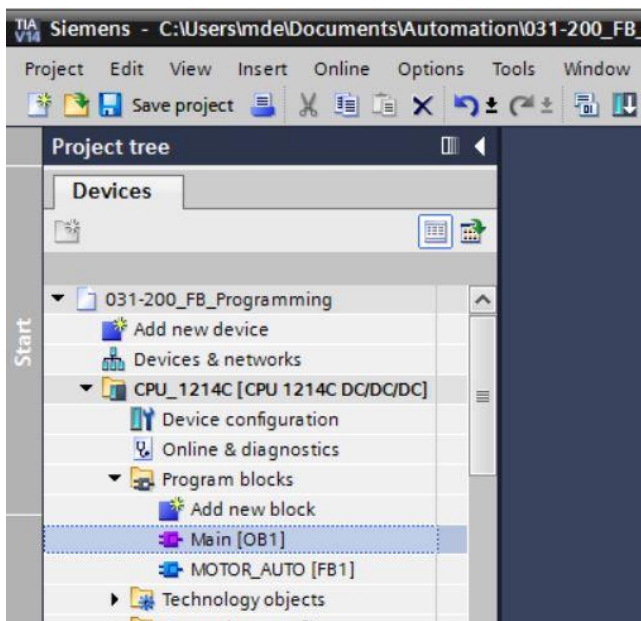


7.8 Programación del bloque de organización OB1: control de la marcha de la cinta hacia delante en modo automático

- ® Antes de programar el bloque de organización "Main[OB1]", cambiaremos el lenguaje de programación a FUP (Esquema 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_1214C [CPU 1214C DC/DC/DC] ® Program blocks (Bloques de programa) ® Main [OB1] ® Switch programming language (Cambiar lenguaje de programación) ® FBD (FUP))

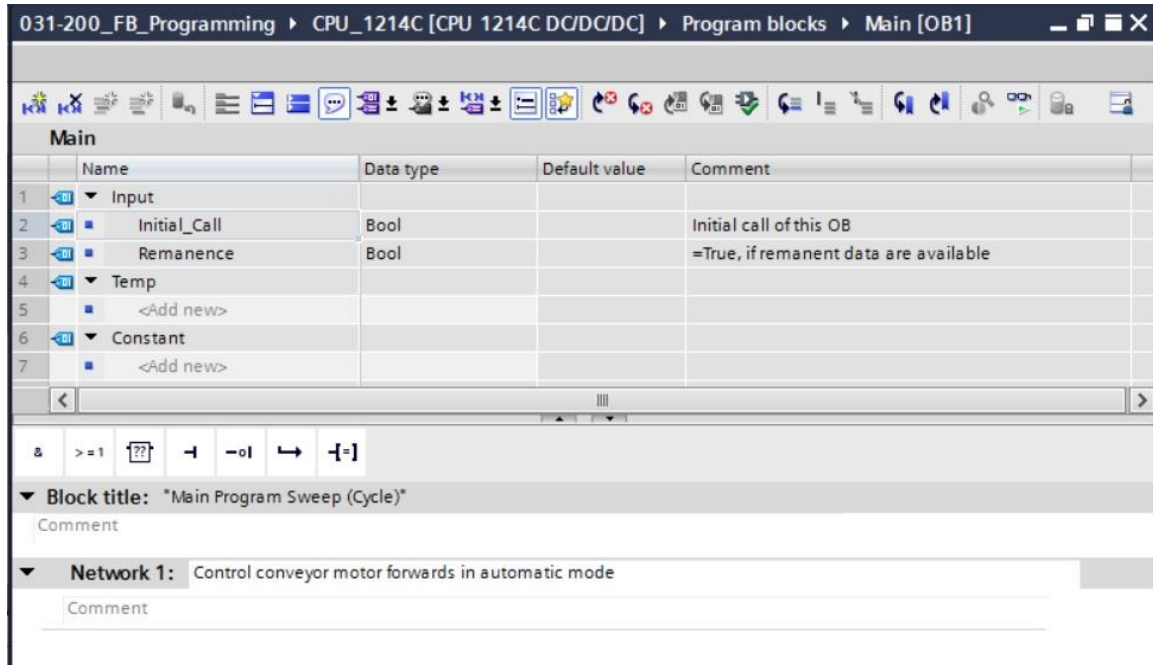


- ® Abra el bloque de organización "Main [OB1]" haciendo doble clic.

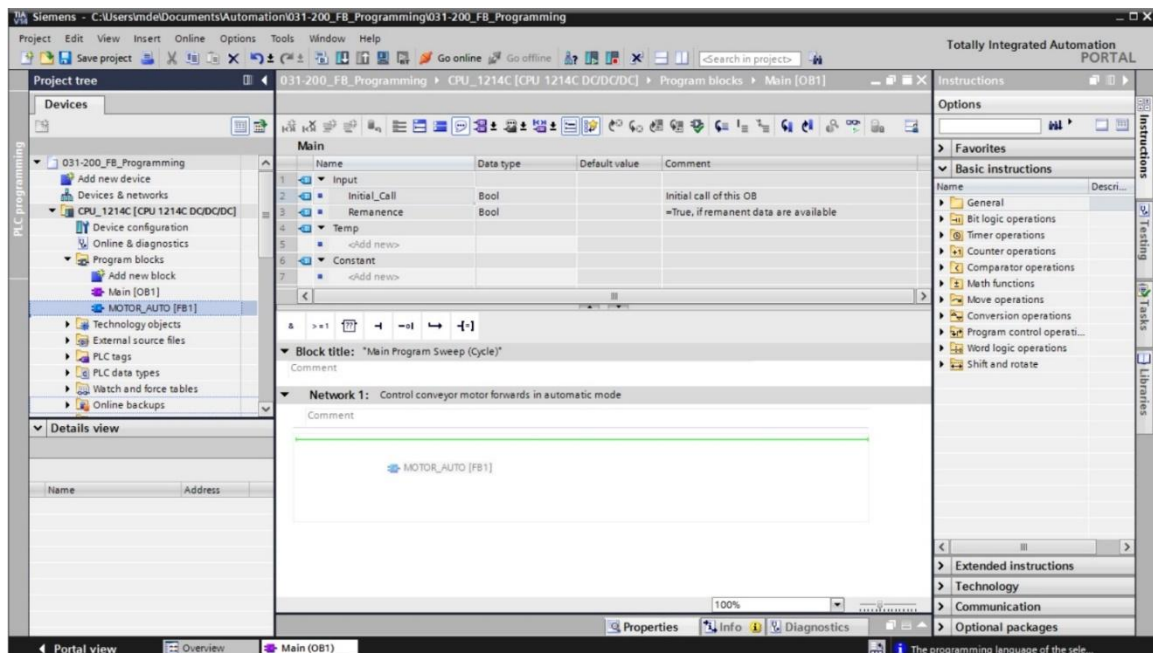


® Asigne al Network 1 (Segmento 1) el nombre "Control conveyor motor forwards in automatic mode (Control de la marcha de la cinta hacia delante en modo automático)".

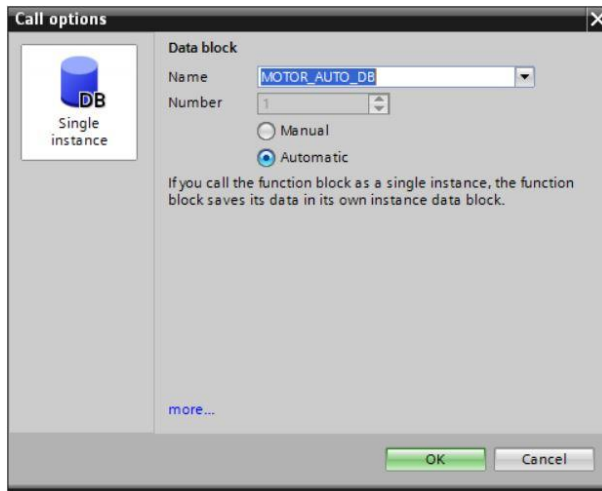
(® Network 1 (Segmento 1):... ® Control conveyor motor forwards in automatic mode (Control de la marcha de la cinta hacia delante en modo automático))



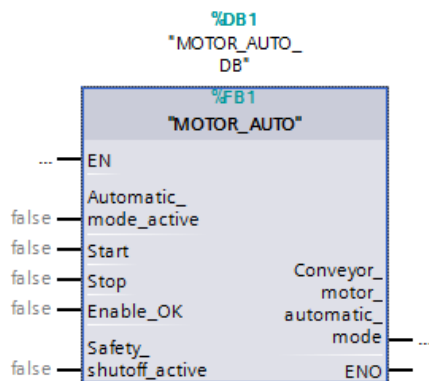
® Ahora desplace el bloque de función "MOTOR_AUTO [FB1]" hasta la línea verde del Network 1 (Segmento 1) mediante "arrastrar y soltar".



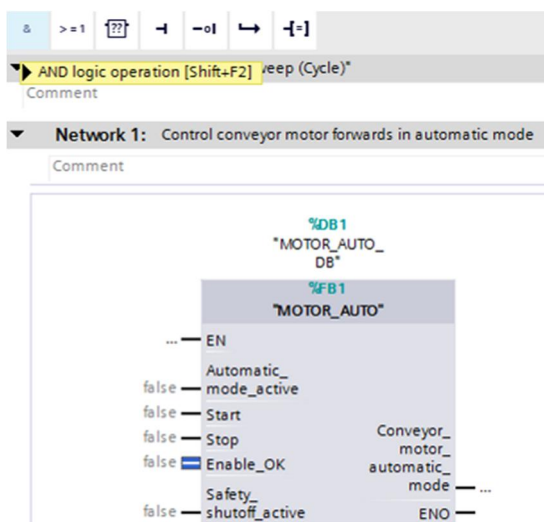
- ® El bloque de datos de instancia para esta llamada del FB1 se crea automáticamente. Asigne un nombre y acéptelo con OK (Aceptar) (® MOTOR_AUTO_DB1 ® OK (Aceptar)).



- ® Se insertará en el Network 1 (Segmento 1) el bloque de datos de instancia con la interfaz que se ha definido y las conexiones EN y ENO.

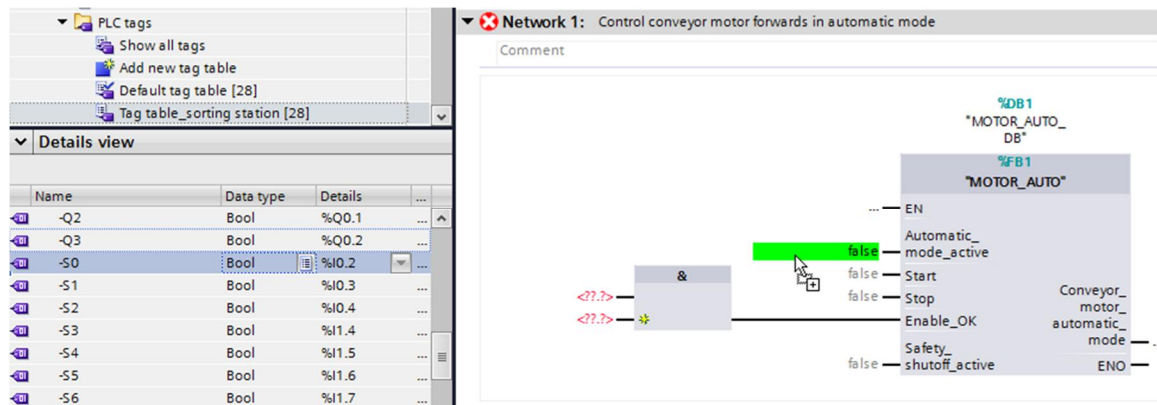


- ® Para insertar un Y antes del parámetro de entrada "Enable_OK (Habilitación_OK)", seleccione dicha entrada e inserte el Y haciendo clic en el icono  de la barra de herramientas lógicas. (® )

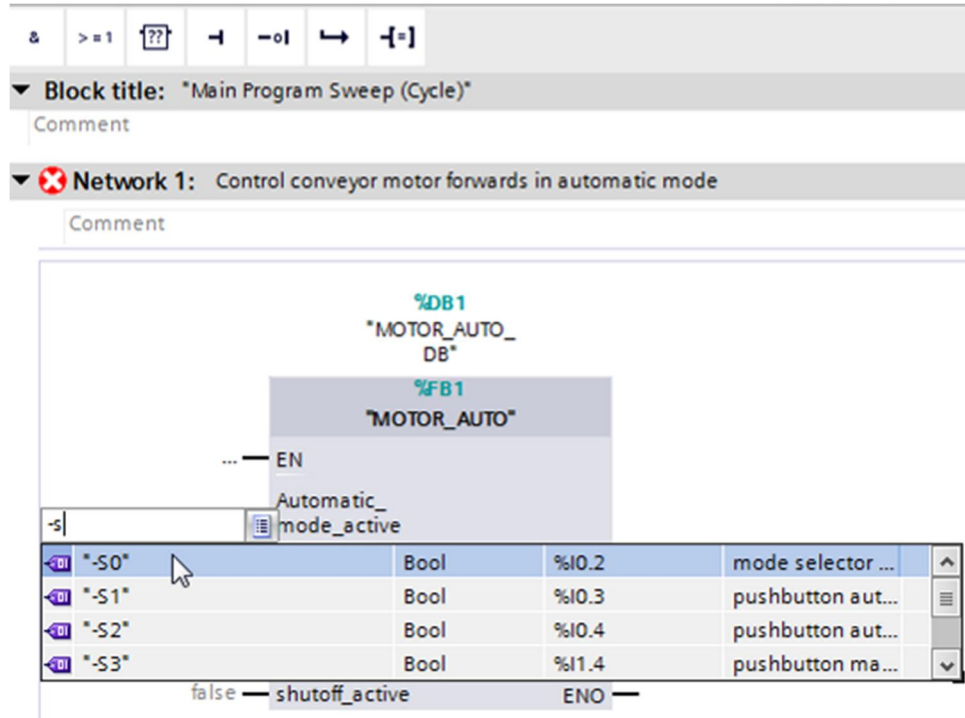


- ® Para interconectar el bloque con las variables globales de la "Tag table_sorting station (Tabla_variables_planta_clasificación)", disponemos de dos posibilidades:

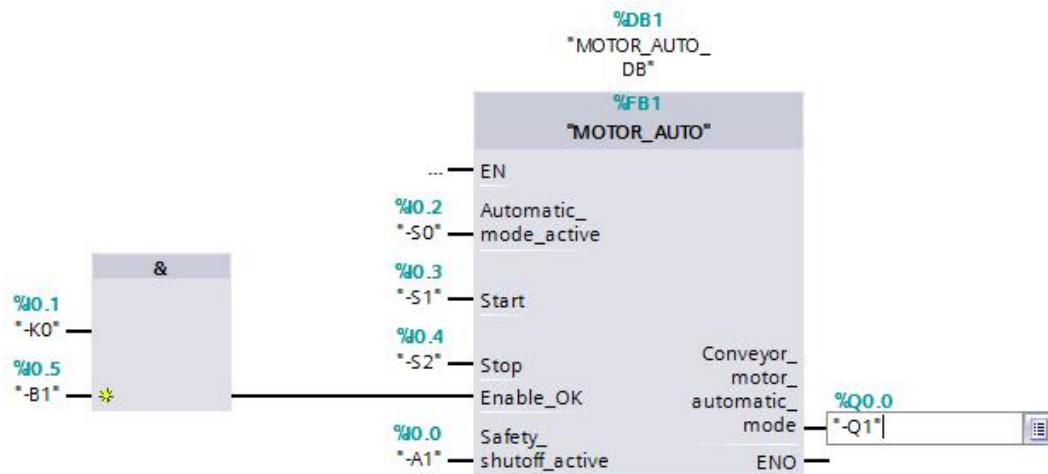
La primera consiste en seleccionar la "Tag table_sorting station (Tabla_variables_planta_clasificación)" en el árbol del proyecto y copiar la variable global deseada desde la vista detallada a la interfaz de FC1 mediante "arrastrar y soltar" (® Tag table_sorting station (Tabla_variables_planta_clasificación)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



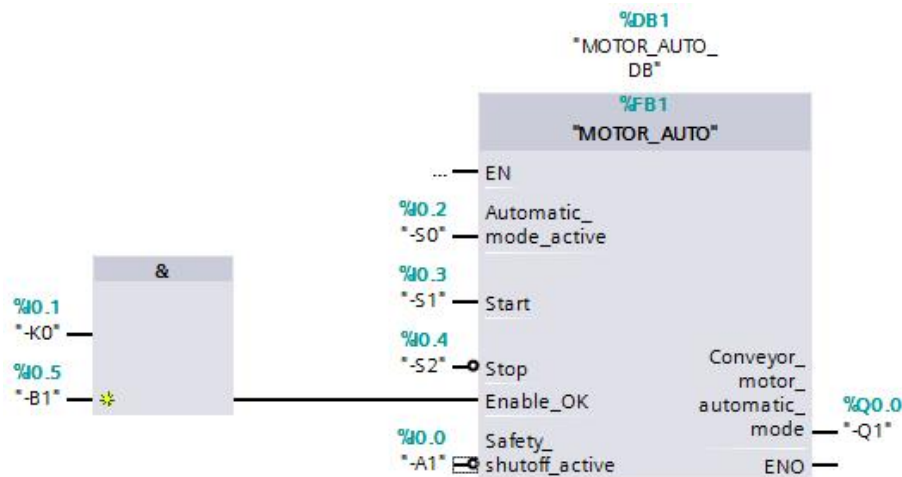
- ® La segunda, en introducir en <??.?> las letras iniciales de la variable global deseada (p. ej., "-S") y, en la lista que aparecerá, seleccionar la variable global de entrada "-S0" (%I0.2) (® Automatic_mode_active (Modo_automático_activo) ® -S ® -S0).



- ® Inserte las restantes variables de entrada "-S1", "-S2", "-K0", "-B1" y "-A1" y, a continuación, en la salida "Conveyor_motor_automatic_mode (Motor_cinta_modos_automático)", la variable de salida "-Q1" (%Q0.0).

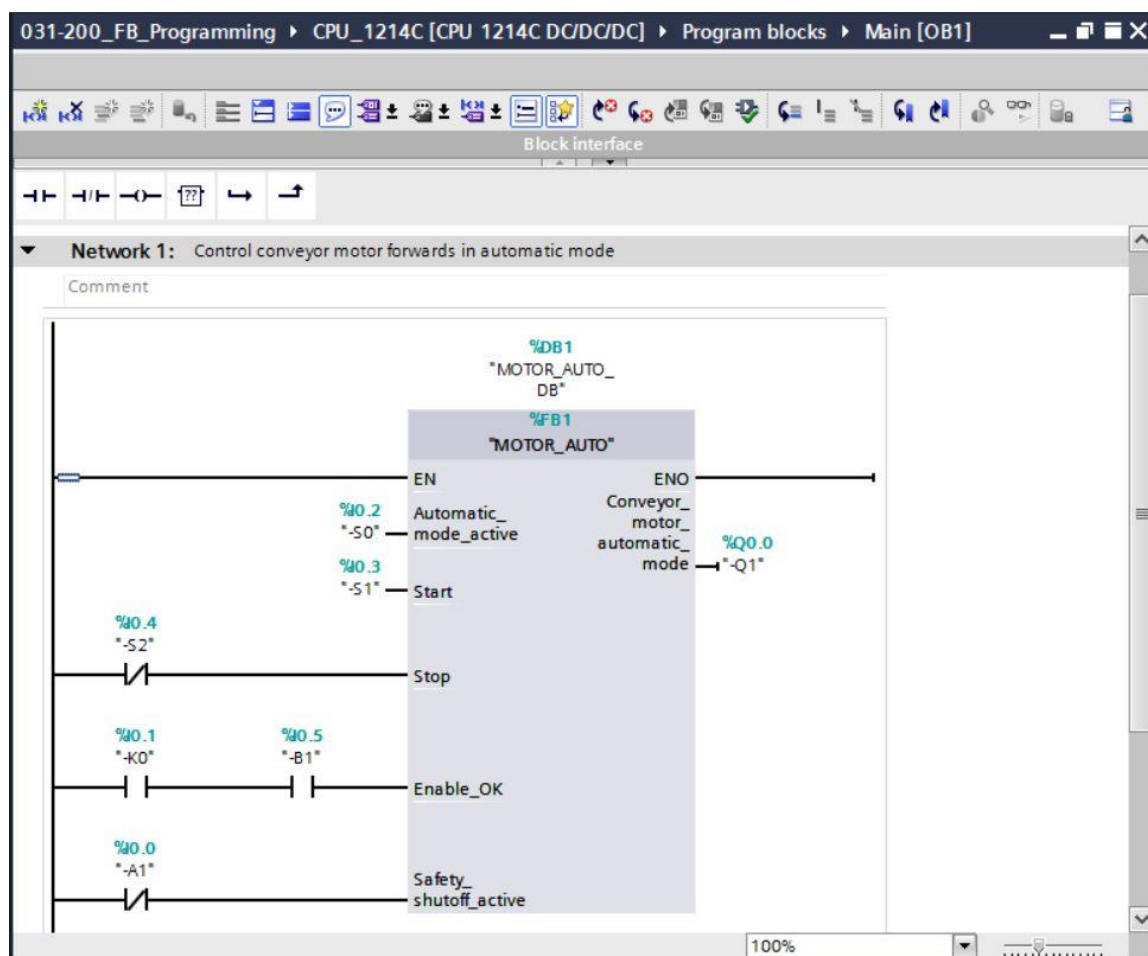


- ® Niegue las consultas de las variables de entrada "-S2" y "-A1" seleccionándolas y haciendo clic a continuación en  (® -S2 ®  ® -A1 ® ).



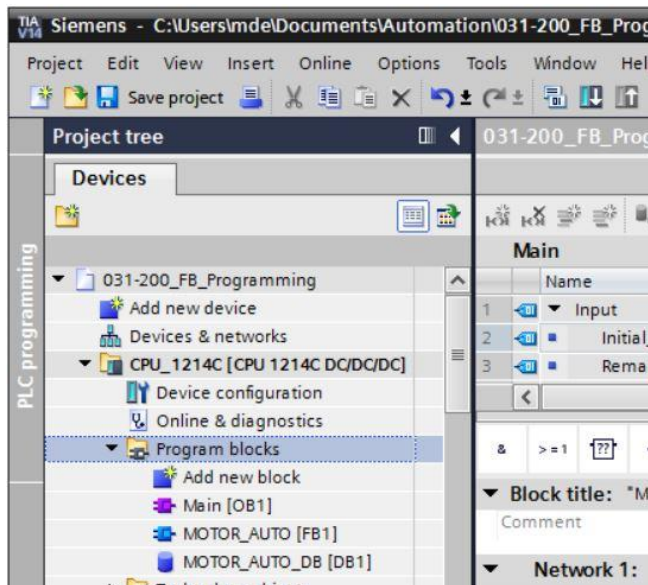
7.9 Resultado en el lenguaje de programación KOP (Esquema de contactos)

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

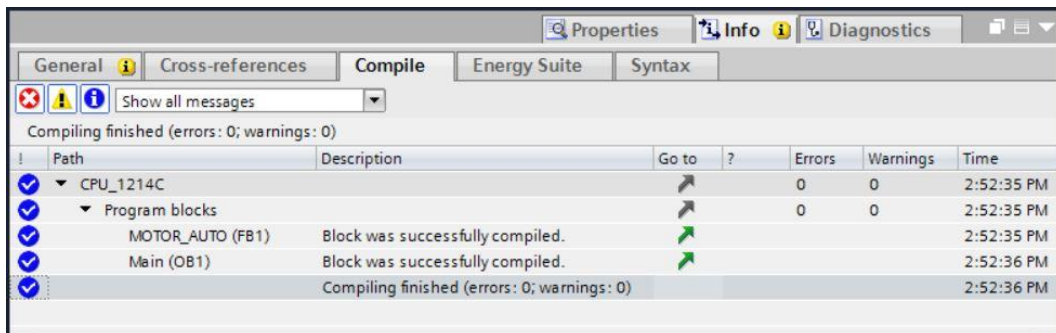


7.10 Guardado y compilación del programa

- ® 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 a continuación seleccione en el menú el icono de compilación  (®  Save project ® Program blocks (Bloques de programa) ® ).

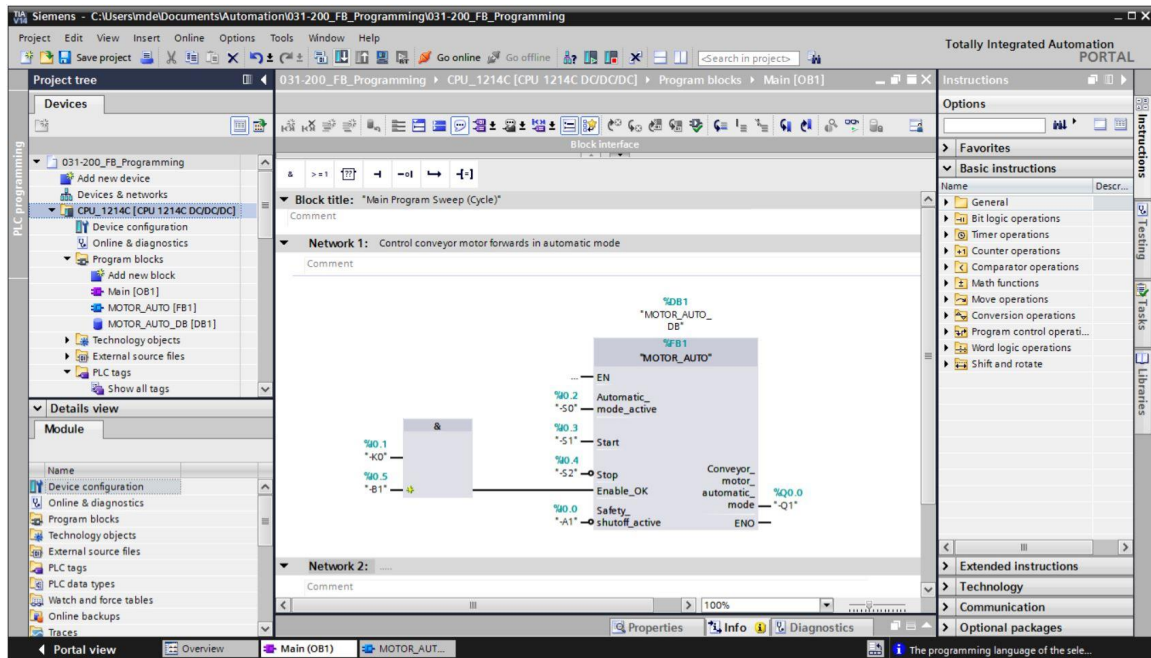


- ® A continuación se mostrarán en la pestaña "Info (Información)" "Compile (Compilar)" los bloques que se han podido compilar correctamente.



7.11 Carga del programa

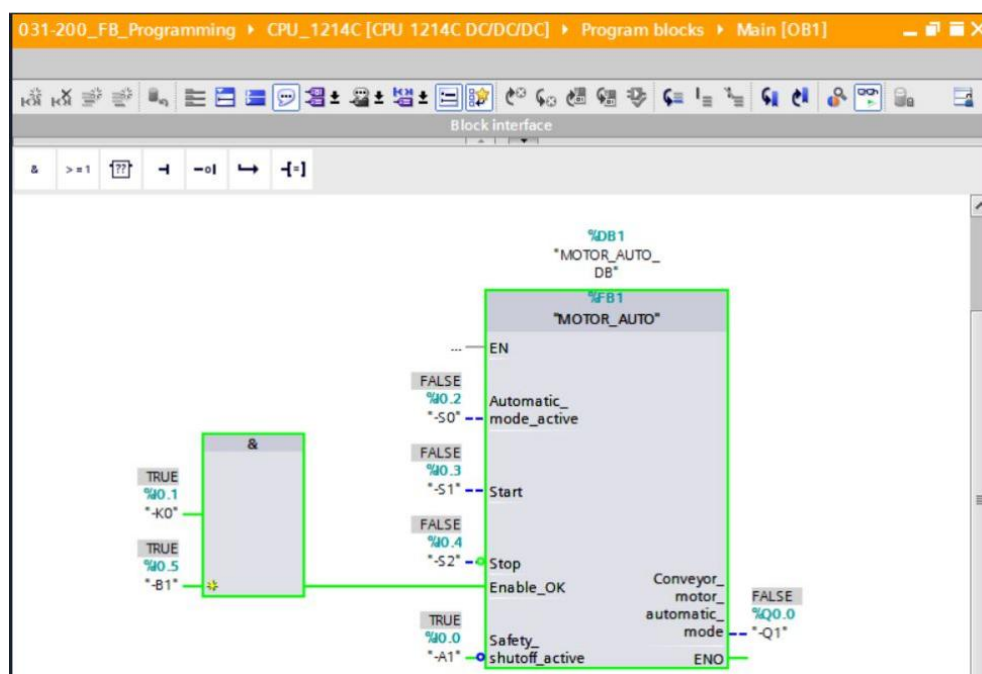
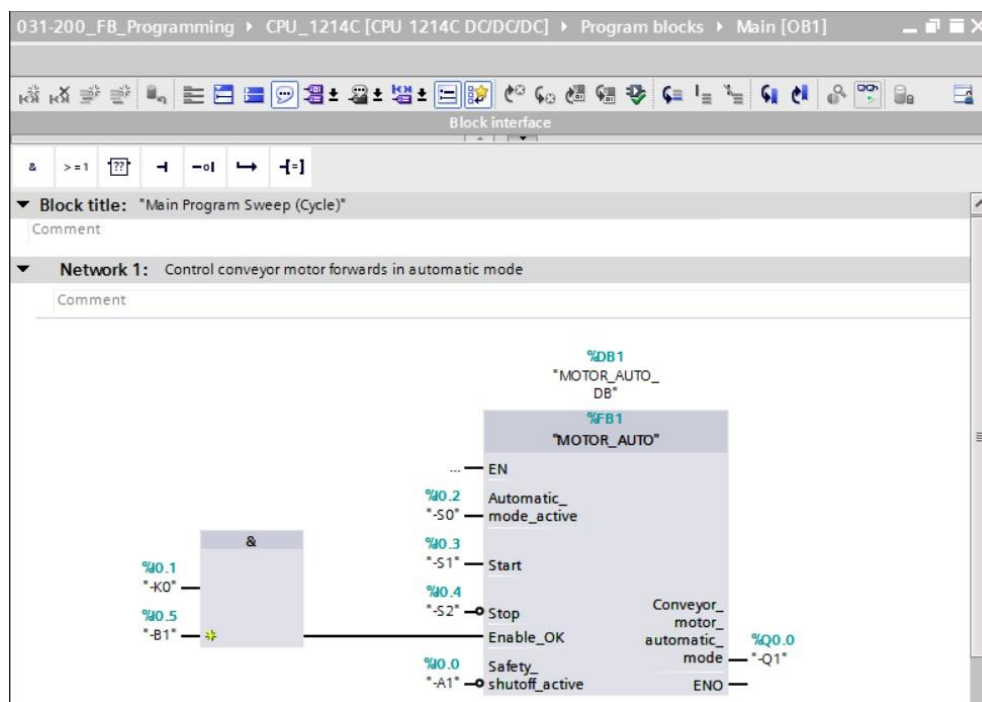
Una vez realizada la compilación correctamente, puede cargar el controlador completo, con el programa que ha creado, del modo descrito en los módulos dedicados a la configuración hardware (® ).



7.12 Visualización de los bloques de programa

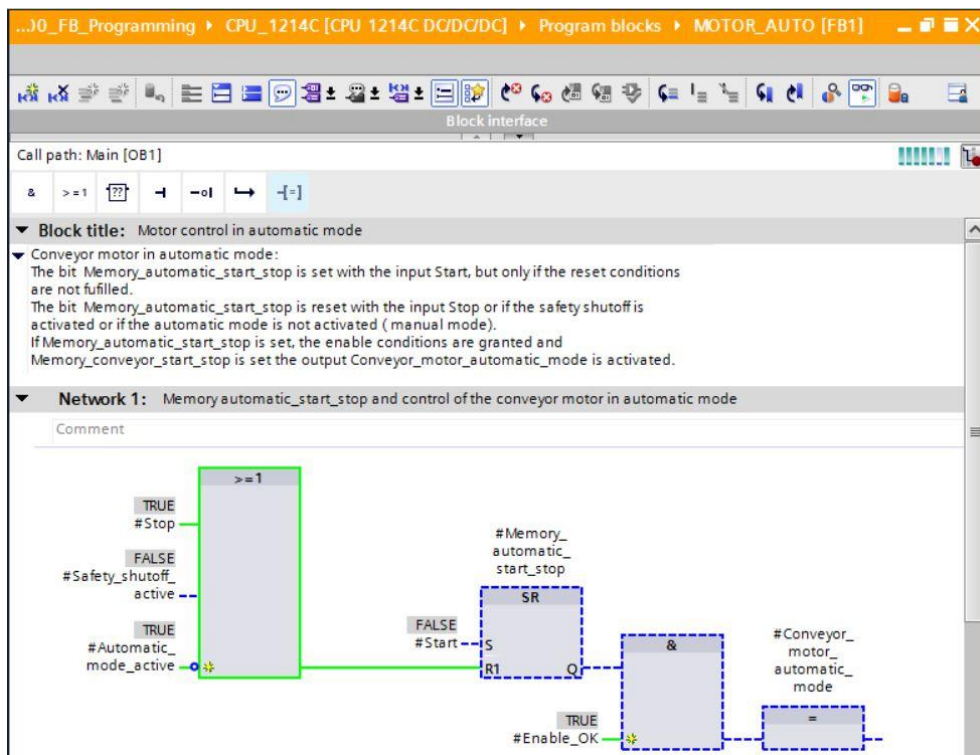
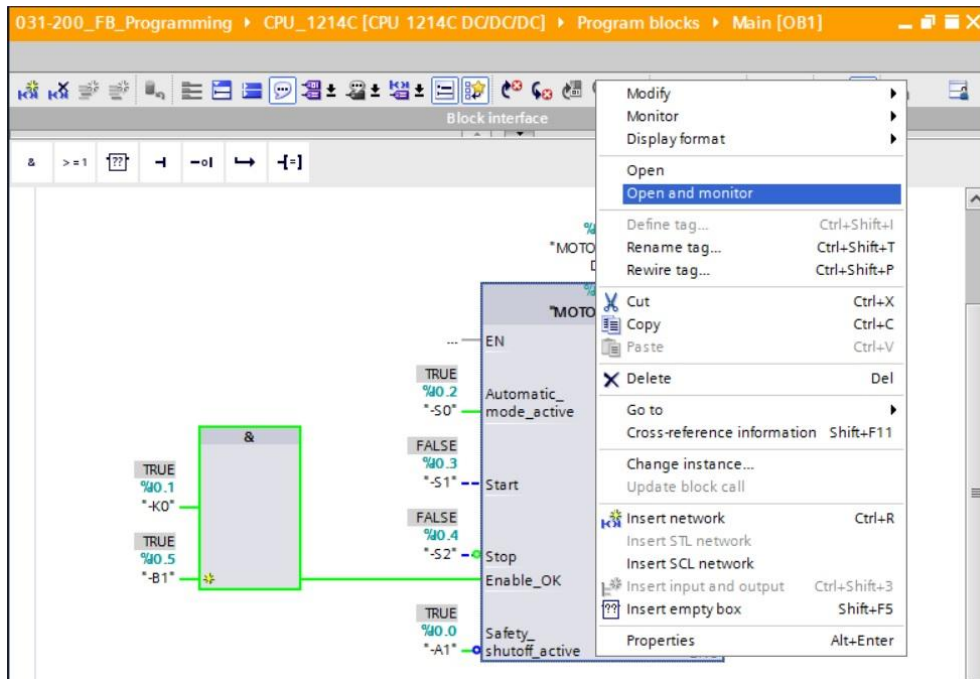
® Para visualizar el programa cargado, debe estar abierto el bloque deseado. Entonces puede activarse o desactivarse la visualización haciendo clic en el icono .

(® Main [OB1] ® ).



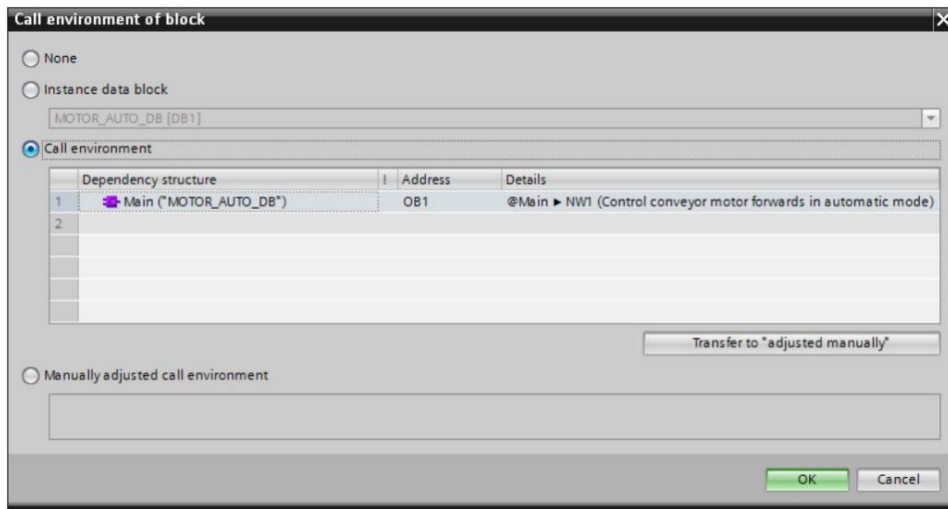
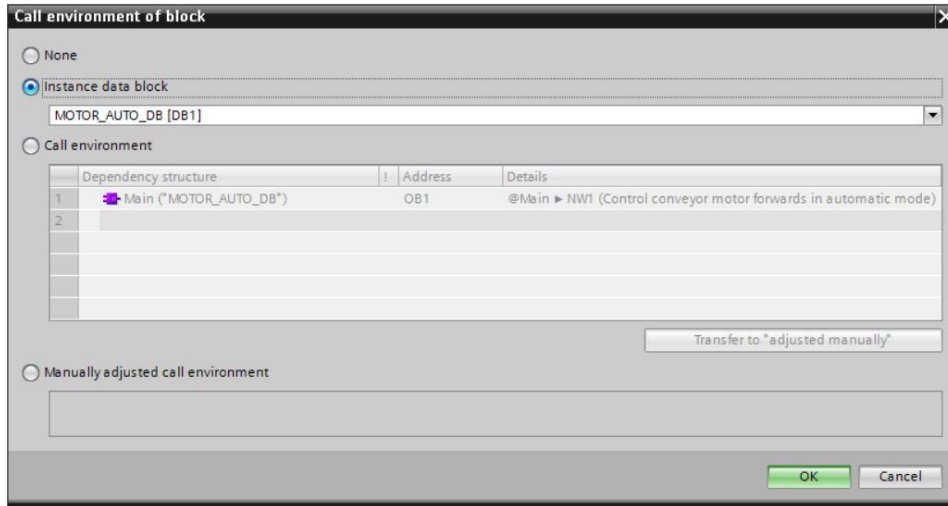
Nota: la visualización se realiza con referencia a la señal y dependiendo del controlador. Los estados lógicos de los bornes se indican mediante TRUE y FALSE.

- ® Para abrir y visualizar el bloque de función "MOTOR_AUTO" [FB1] llamado en el bloque de organización "Main [OB1]", selecciónelo directamente tras hacer clic en él con el botón derecho del ratón (® "MOTOR_AUTO" [FB1] ® Open and monitor (Abrir y visualizar)).



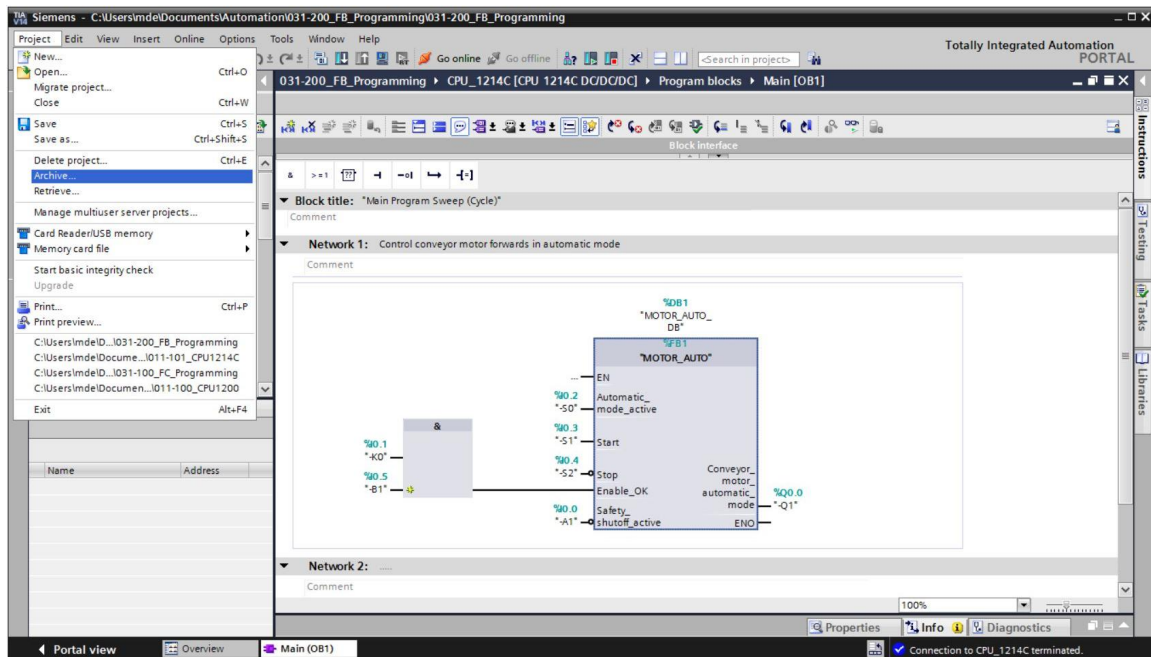
Nota: la visualización se realiza con referencia a la función y dependiendo del controlador. La activación de los sensores y el estado de la instalación se indican mediante TRUE y FALSE.

- ® Si se desea visualizar una ubicación concreta de un bloque de función "MOTOR_AUTO" [FB1] que se llama varias veces, puede usarse el icono . Existen dos posibilidades: usar el entorno de llamada o definir el entorno de llamada mediante el bloque de datos de instancia (®  ® Instance data block (Bloque de datos de instancia) ® MOTOR_AUTO_DB1 [DB1] ® Call environment (Entorno de llamada) ® Address (Dirección): OB1 ® Details (Detalles): Main NW1 ® OK (Aceptar)).



7.13 Archivar 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) ® 031-200_Programación_FB.... ® Save (Guardar)).



7.14 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 AUTOMÁTICO (-S0 = 1) Pulsador de parada automática no accionado (-S2 = 1) Accionar brevemente el pulsador de arranque automático (-S1 = 1) Tras ello, el motor de cinta se pone en marcha hacia delante a velocidad fija (-Q1 = 1) y permanece encendido.	
4	Accionar brevemente el pulsador de parada automática (-S2 = 0) ® -Q1 = 0	
5	Activar PARADA DE EMERGENCIA (-A1 = 0) ® -Q1 = 0	
6	Modo de operación manual (-S0 = 0) ® -Q1 = 0	
7	Desconectar la instalación (-K0 = 0) ® -Q1 = 0	
8	Cilindro no introducido (-B1 = 0) ® -Q1 = 0	
9	Proyecto archivado correctamente	

8 Ejercicio

8.1 Tarea planteada: ejercicio

En este ejercicio se añadirá una función de ahorro energético al bloque de función MOTOR_AUTO [FB1]. El bloque de función ampliado se planificará, programará y probará.

Por motivos de ahorro de energía, la cinta solo debe funcionar cuando realmente haya una pieza sobre ella.

Por ello, la salida Conveyor_motor_automatic_mode (Motor_cinta_modo_automático) solo se activará cuando esté seteado el parámetro Memory_automatic_start_stop (Memoria_modo_automático_arranque/parada), se cumplan las condiciones de habilitación y esté seteado el parámetro Memory_conveyor_start_stop (Memoria_cinta_arranque/parada).

El parámetro Memory_conveyor_start_stop (Memoria_cinta_arranque/parada) se activa cuando Sensor_slide (Sensor_deslizador) indica la presencia de una pieza y se desactiva cuando Sensor_end_of_conveyor (Sensor_fin_cinta) genera un flanco negativo, la desconexión de seguridad está activa o el modo automático no está activado (operación manual).

8.2 Esquema tecnológico

Este es el esquema tecnológico para la tarea asignada.

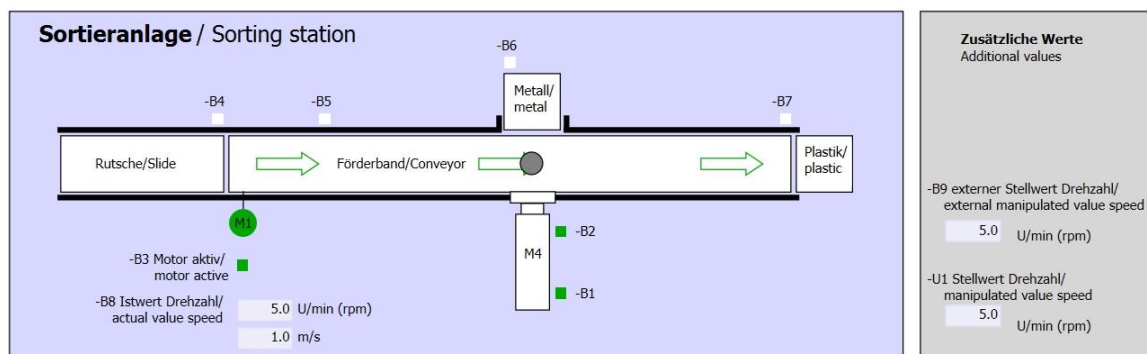


Figura 10: Esquema tecnológico



Figura 11: Pupitre de mando

8.3 Tabla de asignación

Para esta tarea se requieren las siguientes señales como operandos globales.

DI	Tipo	Identificador	Función	NC/NA
I 0.0	BOOL	-A1	Aviso PARADA DE EMERGENCIA OK	NC
I 0.1	BOOL	-K0	Planta "ON"	NA
I 0.2	BOOL	-S0	Selector modo de operación manual (0)/automático (1)	Manual = 0 Automático = 1
I 0.3	BOOL	-S1	Pulsador de arranque automático	NA
I 0.4	BOOL	-S2	Pulsador de parada automática	NC
I 0.5	BOOL	-B1	Sensor cilindro -M4 introducido	NA
I 1.0	BOOL	-B4	Sensor deslizador ocupado	NA
I 1.3	BOOL	-B7	Sensor de pieza al final de la cinta	NA

DO	Tipo	Identificador	Función	
Q 0.0	BOOL	-Q1	Motor de cinta -M1 hacia delante, velocidad fija	

Leyenda de la lista de asignación

DI	Entrada digital	DQ	Salida digital
AI	Entrada analógica	AQ	Salida analógica
E	Entrada	S	Salida
NC	Normalmente cerrado		
NA	Normalmente abierto		

8.4 Planificación

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

Nota: encontrará información acerca del uso del flanco negativo en SIMATIC S7-1200 en la Ayuda en pantalla.

8.5 Lista de comprobación: ejercicio

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 AUTOMÁTICO (-S0 = 1) Pulsador de parada automática no accionado (-S2 = 1) Accionar brevemente el pulsador de arranque automático (-S1 = 1) Sensor deslizador ocupado, activado (-B4 = 1) Tras ello, el motor de cinta se pone en marcha hacia delante a velocidad fija (-Q1 = 1) y permanece encendido.	
4	Sensor de fin de cinta, activado (-B7 = 1) ® -Q1 = 0	
5	Accionar brevemente el pulsador de parada automática (-S2 = 0) ® -Q1 = 0	
6	Activar PARADA DE EMERGENCIA (-A1 = 0) ® -Q1 = 0	
7	Modo de operación manual (-S0 = 0) ® -Q1 = 0	
8	Desconectar la instalación (-K0 = 0) ® -Q1 = 0	
9	Cilindro no introducido (-B1 = 0) ® -Q1 = 0	
10	Proyecto archivado correctamente	

9 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/s7-1200

Vista previa "Información adicional"

Getting Started, Videos, Tutorials, Apps, Manuals, Trial-SW/Firmware

-  TIA Portal Videos
-  TIA Portal Tutorial Center
-  Getting Started
-  Programming Guideline
-  Easy Entry in SIMATIC S7-1200
-  Download Trial Software/Firmware
-  Technical Documentation SIMATIC Controller
-  Industry Online Support App
-  TIA Portal, SIMATIC S7-1200/1500 Overview
-  TIA Portal Website
-  SIMATIC S7-1200 Website
-  SIMATIC S7-1500 Website

Más información

Siemens Automation Cooperates with Education
[siemens.com/sce](https://www.siemens.com/sce)

Documentación didáctica/para cursos de formación de SCE
[siemens.com/sce/documents](https://www.siemens.com/sce/documents)

Paquetes para instructores de SCE
[siemens.com/sce/tp](https://www.siemens.com/sce/tp)

Personas de contacto de SCE
[siemens.com/sce/contact](https://www.siemens.com/sce/contact)

Digital Enterprise
[siemens.com/digital-enterprise](https://www.siemens.com/digital-enterprise)

Industria 4.0
[siemens.com/future-of-manufacturing](https://www.siemens.com/future-of-manufacturing)

Totally Integrated Automation (TIA)
[siemens.com/tia](https://www.siemens.com/tia)

TIA Portal
[siemens.com/tia-portal](https://www.siemens.com/tia-portal)

Controladores SIMATIC
[siemens.com/controller](https://www.siemens.com/controller)

Documentación técnica de SIMATIC
[siemens.com/simatic-docu](https://www.siemens.com/simatic-docu)

Industry Online Support
support.industry.siemens.com

Catálogo de productos y sistema de pedidos online Industry Mall
mall.industry.siemens.com

Siemens AG
Digital Factory
P.O. Box 4848
90026 Nuremberg
Alemania

Sujeto a cambios sin previo aviso; no nos responsabilizamos de posibles errores.
© Siemens AG 2018

[siemens.com/sce](https://www.siemens.com/sce)